



Solar Flare Forecasting Using Hybrid Neural Networks

Dan Xu, Pengchao Sun, Song Feng , Bo Liang , and Wei Dai

Faculty of Information Engineering and Automation, Kunming University of Science and Technology and Yunnan Key Laboratory of Computer Technologies Application, Kunming 650500, People's Republic of China; feng.song@kust.edu.cn

Received 2024 June 24; revised 2024 December 10; accepted 2024 December 20; published 2025 February 5

Abstract

Solar flares are one of the most intense solar activities, the result of a sudden large-scale release of magnetic energy in the form of electromagnetic radiation and energetic particles. Intense solar flares can severely threaten communication and navigation systems, oil pipelines, and power grids on Earth. Therefore, it is crucial to establish highly accurate solar flare prediction models to enable humans to anticipate solar flare eruptions in advance, thereby reducing human and economic losses. In this paper, we utilized the solar active region (AR) magnetogram provided by the Solar Dynamics Observatory's Helioseismic and Magnetic Imager and the associated feature parameters of the magnetic field; specifically, the feature vectors of the magnetic field's spatial structure characteristics and the magnetic field feature parameters are fused to predict solar flares. We built two solar flare prediction models based on a combination of convolutional neural networks (CNN) and a temporal convolutional network (TCN), called CNN-TCN, and predicted whether a $\geq C$ - or $\geq M$ -class flare event would erupt in ARs in the next 24 hr, respectively. Then, after training and testing our model, we focused on the true skill statistic (TSS). Through the model superiority discussion, the model obtained high average TSS values, with the $\geq C$ and $\geq M$ models achieving TSS scores of 0.798 ± 0.032 and 0.850 ± 0.074 , respectively, suggesting that our models have good forecasting performance. We speculate that some key features automatically extracted by our model may not have been previously identified, and these features could provide important clues for studying the mechanisms of flares.

Unified Astronomy Thesaurus concepts: [Solar flares \(1496\)](#)

1. Introduction

A solar flare is an intense solar activity, mainly in the form of electromagnetic radiation and high-energy particles, which is an intense burst of radiation occurring in the solar atmosphere that releases large amounts of energy into space. The electromagnetic radiation and high-energy particles produced by intense solar flares can severely threaten the space environment and the safety of human activities and infrastructure. To reduce or prevent the hazards of solar flares, establishing accurate and robust solar flare prediction models is of great significance for space weather forecasting.

So far, there is still a lack of clear physical theories to explain the flare eruption mechanisms in active regions (ARs). Previously, many traditional machine learning algorithms have played an important role in solar flare prediction. Common traditional machine learning methods used for flare prediction include support vector machines (e.g., Y. Yuan et al. 2010; N. Nishizuka et al. 2017; V. M. Sadykov & A. G. Kosovichev 2017), the artificial neural network (e.g., R. Qahwaji & T. Colak 2007; O. W. Ahmed et al. 2013; R. Li & J. Zhu 2013; N. Nishizuka et al. 2018), the K-nearest neighbors algorithm (e.g., R. Li et al. 2008; X. Huang et al. 2013), the random forest algorithm (e.g., C. Liu et al. 2017; K. Florios et al. 2018), and the ensemble learning algorithm (e.g., T. Colak & R. Qahwaji 2009; X. Huang et al. 2010; J. A. Guerra et al. 2015). These traditional machine learning algorithms have achieved high levels of performance and improvement in solar flare prediction. All of these techniques learn from manually

selected and extracted features from observational data, but artificially extracting effective physical parameters from solar ARs has become a bottleneck affecting the performance of solar flare prediction models.

Recently, deep learning has emerged as a reliable technique to advance the solar flare forecasting learning problem. The most representative of these, the 2D convolutional neural networks (2D CNNs), have been used multiple times for solar flare prediction (X. Huang et al. 2018; E. Park et al. 2018; X. Li et al. 2020). 2D CNNs have good performance in the field of solar flare forecasting. Since flare eruptions are closely related to the evolution of the magnetic field in ARs (H. Wang et al. 2017), these 2D CNN-based solar flare prediction models primarily use magnetogram samples of ARs as training data. However, 2D CNNs only utilize the spatial structure characteristics of the magnetic field in ARs, and their drawback is that they cannot account for the temporal evolution features of the magnetic field in these regions. Additionally, in order to consider the temporal evolution features of the magnetic field, some other studies mainly use as model inputs the AR parameters provided from the Space-weather HMI Active Region Patches (SHARPs; M. G. Bobra et al. 2014) to develop LSTM-based solar flare forecasting models (Y. Chen et al. 2019; R. Tang et al. 2020; X. Wang et al. 2020; K. Yi et al. 2020). The successful application of the SHARP parameters in LSTM-based solar flare forecasting shows the importance of these parameters for triggering solar eruptions. Many studies have proposed that larger sunspots with a large number of more complex magnetic flux structures tend to produce strong solar flares (D. S. Bloomfield et al. 2012), and thus the spatial structure of the magnetic field is also characterized as an important factor to be considered for solar flare forecasting. However, the SHARP parameter we use is a scalar value that



Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

summarizes information about the entire AR and does not contain information about the 2D spatial structure of the magnetic field in the AR (H. Sun et al. 2021). P. Sun et al. (2022) proposed a 3D CNN-based solar flare prediction model. 3D CNN extends the 2D convolution kernel of 2D CNN to a 3D structure by adding a temporal dimension and simultaneously extracts the spatial distribution and the temporal evolution of the AR magnetic field for solar flare prediction with a forecast period of 24 hr. This method significantly enhances the prediction of flare class, suggesting that it is more advantageous to consider the spatial distribution and temporal evolution of the AR magnetic field together. Indeed, this method performs well on classification problems, but still has limitations in the work on AR magnetic field temporal evolution feature capture, with limited ability to capture long-time dependencies.

To address the shortcomings of existing solar flare forecasting methods mentioned above, we propose a solar flare forecasting model based on the combination of a CNN and a temporal convolutional network (TCN). The aim is to extract both the spatial distribution and the temporal evolution features of AR magnetic fields simultaneously and to better capture the complex temporal evolution information of the AR magnetogram. The magnetograms of the ARs are processed by the CNN, which extracts the spatial structure characteristics of the magnetogram through comparative representation learning. These features are then concatenated with the magnetic field feature parameters to form a feature set that includes both the spatial distribution characteristics and the temporal evolution features of the AR magnetic fields. Multiple feature sets are organized into temporal sequences in chronological order. The temporal sequences are then processed by the TCN to extract the magnetic field evolution features, allowing for the prediction of $\geq C$ - and $\geq M$ -class flares produced by an AR within the next 24 hr simultaneously.

The structure of this paper is organized as follows: Section 2 describes the data set in detail. Section 3 provides an overview of our proposed method and corresponding model. In Section 4, we evaluate the performance of the proposed solar flare prediction model. Section 5 discusses feature visualization analysis. Finally, Section 6 presents our conclusions.

2. Data

Solar flares are typically classified based on their radiation intensity, particularly the soft X-ray radiation intensity. Internationally, the most commonly used classification standard is the soft X-ray radiation intensity observed by Geostationary Operational Environmental Satellites (GOES) in the 1 Å to 8 Å range. According to the different radiation intensities, solar flares are divided into five classes, A, B, C, M, and X, representing increasing levels of radiation intensity. Based on flare eruption records provided by the National Oceanic and Atmospheric Administration (NOAA)¹, this paper collects data on the eruption time, classification, and AR number of flares from 2010 May to 2019 December. Since NOAA only records information on flare eruptions and many active regions do not experience any flare eruptions, we collected data on AR without any flare eruptions from solar geophysical data solar event reports.² This ensures that the trained model is more

Table 1

SHARP Parameter Maps; the Formulas Are Applied to Every Pixel of the HMI Magnetogram

Parameter	Description	Formula
USFLUX	Total unsigned flux	$\Phi = \sum B_z dA$
MEANGAM	Mean angle of field from radial	$\bar{\tau} = \frac{1}{N} \sum \arctan\left(\frac{B_h}{B_z}\right)$
MEANGBT	Horizontal gradient of total field	$ \nabla B_{\text{tot}} = \frac{1}{N} \sum \sqrt{\left(\frac{\partial B}{\partial x}\right)^2 + \left(\frac{\partial B}{\partial y}\right)^2}$
MEANGBZ	Horizontal gradient of vertical field	$ \nabla B_z = \frac{1}{N} \sum \sqrt{\left(\frac{\partial B_z}{\partial x}\right)^2 + \left(\frac{\partial B_z}{\partial y}\right)^2}$
MEANGBH	Horizontal gradient of horizontal field	$ \nabla B_h = \frac{1}{N} \sum \sqrt{\left(\frac{\partial B_h}{\partial x}\right)^2 + \left(\frac{\partial B_h}{\partial y}\right)^2}$
MEANJZD	Vertical current density	$\bar{J}_z \propto \frac{1}{N} \sum \left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y} \right)$
TOTUSJZ	Total unsigned vertical current	$J_{z\text{total}} = \sum J_z dA$
MEANALP	Characteristic twist parameter, α	$\alpha_{\text{total}} \propto \frac{\sum J_z B_z}{\sum B_z^2}$
MEANJZH	Current helicity (B_z contribution)	$\bar{H}_c \propto \frac{1}{N} \sum B_z J_z$
TOTUSJH	Total unsigned current helicity	$H_{c\text{total}} \propto \sum B_z J_z $
ABSJZH	Absolute value of the net current helicity	$H_{c\text{abs}} \propto \sum B_z J_z $
SAVNCPP	Sum of the modulus of the net current per polarity	$J_{z\text{sum}} \propto \sum B_z^+ J_z dA + \sum B_z^- J_z dA $
MEANPOT	Proxy for mean photospheric excess magnetic energy density	$\bar{\rho} \propto \frac{1}{N} \sum (B^{\text{Obs}} - B^{\text{Pot}})^2$
TOTPOT	Proxy for total photospheric magnetic free energy density	$\rho_{\text{tot}} \propto \sum (B^{\text{Obs}} - B^{\text{Pot}})^2 dA$
MEANSHR	Shear angle	$\bar{\Gamma} = \frac{1}{N} \sum \arccos\left(\frac{B^{\text{Obs}} \cdot B^{\text{Pot}}}{ B^{\text{Obs}} B^{\text{Pot}} }\right)$
SHRGT45	Fractional area with shear $>45^\circ$	area with shear $>45^\circ$ / HARP area

generalizable. Based on the collected data, we downloaded a total of 46,800 magnetograms from the 24 hr prior to flare eruptions, along with the corresponding 16 magnetic field feature parameters from the Joint Science Operations Center³ to build our data set. These data are taken continuously with a cadence of 12 minutes. The 16 magnetic field feature parameters we used are listed in Table 1. These parameters are computed from the vector magnetic field in AR patches (M. G. Bobra et al. 2014). These magnetic field feature parameters have been proven to aid in solar flare prediction (X. Wang et al. 2020). Additionally, to minimize or avoid the influence of projection effects, only ARs located within $\pm 45^\circ$ of the central meridian are considered for this work.

In our study, we only predict whether $\geq C$ - and $\geq M$ -class flares will erupt (Y. Zheng et al. 2019; X. Wang et al. 2020; P. Sun et al. 2022). For the $\geq C$ -class prediction model, C-, M-, and X-class magnetograms and magnetic field feature parameters are considered positive samples, while the rest are negative samples. For the $\geq M$ -class prediction model, positive samples are defined as M- and X-class magnetograms and magnetic field feature parameters, with the rest being negative samples. CNN requires that all input images must have a fixed size, and all magnetograms are resized to 224×224 pixels.

¹ <https://www.ngdc.noaa.gov>

² <http://www.solarmonitor.org/>

³ <http://jsoc.stanford.edu>

Table 2
Data Set Description and Distribution Details

Flare Class	Data Set	Type	Positive			Negative		
			ARs	N_f	Sequences	ARs	N_q	Sequences
$\geq C$	Magnetogram	2D image	188	22560	1504	202	24240	1616
	Magnetic feature	Sixteen 1D values	188	22560	1504	202	24240	1616
$\geq M$	Magnetogram	2D image	55	6600	440	335	40200	2680
	Magnetic feature	Sixteen 1D values	55	6600	440	335	40200	2680

Note. N_f and N_q represent the sample numbers with flares and no flares, respectively.

Then, we used the method provided by P. Sun et al. (2022). The 120-frame magnetograms from the 24 hr before the flare eruptions were composed into eight magnetogram sequences, each sequence containing 15 frames of magnetograms. This resulted in a total of 3120 magnetogram sequences, with the temporal resolution of our data set downsampled to 96 minutes. The magnetic field feature parameter sequences are composed in the same way as the magnetogram sequences. Then, the spatial structure magnetogram sequences are combined with the normalized magnetic field feature parameters to form 24 hr temporal sequences through feature fusion. The data set used for this study is shown in Table 2, including the number of solar ARs used for the $\geq C$ and $\geq M$ models, the number of magnetograms and magnetic field feature parameters, and the number of positive and negative samples in the time series. N_f and N_q represent the number of positive samples with flares and negative samples without flares, respectively. There are a total of 390 ARs, with a total of 46,800 magnetograms and 3120 temporal sequences.

In our study, we used fivefold cross validation to divide the data set, which is the standard method in this field. First, all the data are randomly shuffled, and then the entire data set is divided into five equal parts. For each experiment, one part is used to test the model, while the remaining parts are used to train the model. Therefore, the ratio of the testing set to the training set is 1:4. Then, this process is repeated five times, each time selecting a different testing set and training set, resulting in a data set with five folds. We used cross validation to ensure that all data in the data set have been used both as a testing set and a training set. In fivefold cross validation, each fold independently completes the training and testing of the model, and the average of the five evaluation results is taken as the final evaluation of the model. Note that it is difficult to correctly divide the entire data set into a training set and a testing set, so we chose to segregate the training set and the testing set using the NOAA activity zone numbers in the same way as in X. Li et al. (2020) and Z. Deng et al. (2021). This isolation of ARs ensures that the samples in the testing set have never been seen by the training set, and it avoids performance gains brought by similar samples from the same AR.

3. Method

3.1. Model Architecture

The $\geq C$ - and $\geq M$ -class solar flare prediction models share the same structure. Our solar flare prediction model is performed in two stages. The first stage is the CNN stage, which aims to extract spatial structure features from AR magnetograms and represent them as feature vectors. These spatial structure feature vectors are then fused with magnetic field feature parameters, the composition of the feature data set

containing both the spatial structural characteristics and the feature parameters of the magnetic field in the AR. The feature data set is composed of temporal sequences as described in Section 2. The second stage is the TCN stage, which focuses on solar flare classification prediction. The fused feature data set from the first stage is inputted into TCN. The fused feature data set includes the temporal sequences containing the spatial structural characteristics and the magnetic field feature parameters. We extract the magnetic field evolution features of the temporal sequences and complete the solar flare classification forecast by using the TCN.

The model architecture is illustrated in Figure 1, the 15 magnetograms from the 24 hr are formed into a sequence of magnetograms in chronological order and then input into a CNN model for spatial feature extraction. The CNN acts as a pretrained encoder trained through contrastive learning, converting each 2D AR magnetogram into a 128-dimensional feature vector. In addition, 15 sets of magnetic field features of the AR over 24 hr are each formed as a temporal sequence, with each set containing the 16 magnetic field feature parameters. These magnetic field feature parameters are normalized and then fused with the obtained the spatial feature vectors in the time dimension. This forms a 15×144 dimensional feature vector, which contains the spatial structure and the magnetic field feature parameters of the ARs over 24 hr. Each of these feature vectors serves as a sample input to the TCN for further extraction of magnetic field evolution features. After the TCN completes the temporal sequence modeling, the final features are passed to a fully connected layer for classification prediction.

3.2. CNN Model

CNNs are one of the deep learning methods that have recently become very popular in the fields of image processing and computer vision. Previously, CNNs have been used for solar flare forecasting many times. Based on the basic layer of a CNN, we built a CNN for magnetogram spatial structure characteristic extraction, but the standard convolution usually has a large number of covariates and computational complexity when dealing with the magnetogram data set of the AR. To solve this problem, we propose a lightweight CNN model based on depth separable convolution (J. Shi et al. 2020) by using depth separable convolution instead of standard convolution and introducing a standard combination of batch normalization (BN; S. Ioffe & C. Szegedy 2015) and rectified linear unit (ReLU; X. Glorot et al. 2011) activation functions. The model aims to reduce the number of parameters and computational complexity of the CNN. Depth separable convolution divides the conventional convolution process into two steps: depthwise convolution and pointwise convolution.

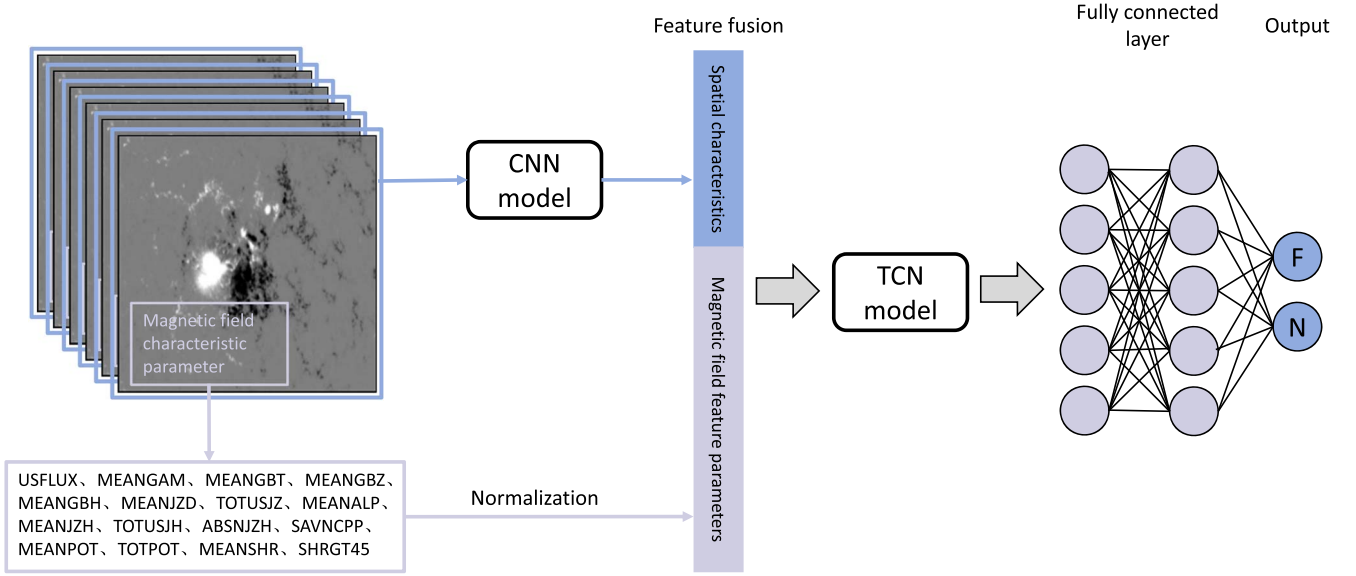


Figure 1. Solar flare prediction model framework. The blue box represents the AR magnetograms, and the purple box represents the 16 magnetic field feature parameters.

The first stage of the work is to extract the spatial structural characteristic of the magnetogram and then to perform feature fusion with the feature parameters of the magnetic field in the AR. Therefore, the 2D magnetograms need to be transformed into 1D representation vectors. The CNN model in our study is used as a feature extractor, and the CNN is trained to be a more effective magnetogram encoder through contrastive learning. Contrastive learning is a type of self-supervised learning that does not make use of the sample labels but instead uses information about the data itself as a supervisory signal (A. Jaiswal et al. 2020). Contrastive learning is often used as an upstream task (J.-B. Grill et al. 2020; K. He et al. 2020; A. Jaiswal et al. 2020), and in our study it is also used as an upstream task to obtain a high-quality representation of the raw magnetogram data, which is then used in the downstream task of classifying solar flare forecasts. Our research used the SimCLR framework (T. Chen et al. 2020), a self-supervised learning method widely used as a contrastive learning framework for learning visual representations. There are three main processes involved in the computation of SimCLR. The first step is to construct positive and negative sample pairs. Neural networks train models by inputting data in batches. For example, a batch consists of N magnetograms, and SimCLR performs two different data enhancement operations on each tile to generate $2N$ expanded magnetograms. For each magnetogram, the original magnetogram forms a positive sample pair with its corresponding two expanded magnetograms and a negative sample pair with $2N - 2$ magnetograms other than the original magnetogram. For example, magnetograms x_i and x_j form positive sample pairs with each other, while x_i forms negative sample pairs with all magnetograms except x_j . The second step is to encode the magnetogram using an encoder, and since it is a manipulation of an image, we use a CNN as the encoder. The magnetograms x_i and x_j are processed by a weight-sharing encoder $f(\cdot)$ to obtain the representation vectors h_i and h_j . These representations are then used for downstream tasks. After the encoder, in order to improve the quality of these features, SimCLR introduces a projection head unit consisting of a multilayer perceptron (MLP) after the representation. This projection head maps the features of the

magnetograms h_i and h_j into the z_i and z_j spaces. T. Chen et al. (2020) indicates that incorporating a projection head makes contrastive learning more effective. Finally, there is the computation of the loss, and inputting N samples results in $2N$ feature vectors. Taking z_i as an example, z_i and z_j form a positive sample pair and the other feature vectors form a negative sample pair, so there are $2N - 1$ feature vector pairs. SimCLR uses the cosine distance to calculate the similarity between pairs of feature vectors. The formula is as follows (T. Chen et al. 2020):

$$\sin(z_i, z_j) = \frac{z_i^\top z_j}{\|z_i\| \|z_j\|}. \quad (1)$$

The normalized temperature-scaled cross-entropy loss (CELoss; Z. Wu et al. 2018) is then used as the final optimization objective, and the final loss value is calculated using the following formula (T. Chen et al. 2020):

$$\ell_{ij} = -\log \frac{\exp\left(\frac{\sin(z_i, z_j)}{\tau}\right)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(\frac{\sin(z_i, z_k)}{\tau}\right)}, \quad (2)$$

where $1_{[k \neq i]}$ is the indicator function and τ is the temperature parameter, and the numerator is the distance between positive sample pairs of magnetograms and the denominator is the distance between negative sample pairs of magnetograms. To ensure optimal performance, we conducted parameter tuning experiments, trying different temperature parameter values and observing their effects on the loss of contrast on the validation set and on the performance of downstream tasks. Through experiments, we found that when the temperature parameter is set to $\tau = 0.07$, it helps the model to perform better on the hard-to-distinguish negative sample pairs.

Table 3 shows all the units and specific parameters of the CNN stage in the solar flare prediction model. Our CNN uses five convolutional blocks, including two convolutional block 1 and three convolutional block 2. Convolutional block 1 consists of a depth separable convolutional layer. Each depth

Table 3
Specific Parameter Settings for CNN

Layer	Convolution Block	Operation	Regularization	Kernel Size	Output Size
Conv1	Block1	DepthwiseSeparableConv	BN	5×5	$64 \times 111 \times 111$
Conv2	Block1	DepthwiseSeparableConv	BN	5×5	$64 \times 55 \times 55$
Conv3	Block2	DepthwiseSeparableConv	BN	3×3	$128 \times 55 \times 55$
Pool3	Block2	MaxPool2d	...	2×2	$128 \times 27 \times 27$
Conv4	Block2	DepthwiseSeparableConv	BN	3×3	$128 \times 27 \times 27$
Pool4	Block2	MaxPool2d	...	2×2	$128 \times 13 \times 13$
Conv5	Block2	DepthwiseSeparableConv	BN	3×3	$128 \times 13 \times 13$
Pool5	Block2	MaxPool2d	...	2×2	$128 \times 6 \times 6$
AAP	...	Adaptive average pooling	$128 \times 1 \times 1$
FC1	...	Fully connected	128×1
FC2	...	Fully connected	128×1

separable convolutional layer consists of a depthwise convolutional module and a pointwise convolutional module. Both the depthwise convolution module and the pointwise convolution module contain 2D convolutional layers (Conv2d) with a convolutional kernel size of 5×5 . The purpose of using a larger convolutional kernel is to obtain a larger receptive field; it also includes 2D batch normalization (BatchNorm2d) and a ReLU activation function. BatchNorm2d accelerates training and improves model stability by normalizing the activation values for each batch of data, and ReLU activation function accelerates training convergence by introducing nonlinearities, which enables the model to learn and represent more complex features. Convolutional block 2 consists of a depth separable convolutional layer with the same structure as convolutional block 1 and a 2D maximum-pooling layer (MaxPool2d) with kernel size 2×2 . However, the size of the convolution kernel for Conv2d in the depth separable convolutional layer of convolutional block 2 is reduced to 3×3 . Pooling is a form of downsampling. The maximum-pooling operation preserves the critical feature information by reducing the spatial dimension and computational effort of the feature map while retaining the key feature information. The final feature vector with a dimension of $128 \times 1 \times 1$ obtained by the adaptive average pool (AAP) layer and the spreading operation is the feature used as a downstream task in SimCLR. We used the MLP recommended by SimCLR as the model's projection head, which consists of two fully connected layers and a ReLU activation function. After the projection head completes the feature mapping, the final contrastive loss function is used to compute the loss.

3.3. TCN Model

The second stage of the TCN model involves passing temporal sequences with the spatial structural characteristics and the magnetic field feature parameter information to a TCN to extract magnetic field evolution features and simultaneously classify and forecast solar flares, and the structure of the TCN is illustrated in Figure 2. When looking at an image, we usually pay more attention to the parts we are interested in. Similarly, the attention mechanism helps the neural network to better utilize the input information. In the first residual block, the convolution kernel size is set to 3 and the dilation factor is set to 1. In the second residual block, the convolution kernel size is still set to 3, and the dilation factor is set to 2. The purpose of increasing the dilation factor is to obtain a larger receptive field, so as to capture the longer history information in the temporal

sequences. Here the random deactivation ratio of the Dropout mechanism is 0.3.

Within the residual block of TCN, we added the channel attention mechanism (CA); the position of CA is shown in Figure 2(b) as "ChannelAttention." The process of CA is shown in Figure 2(c). The CA module can adaptively assign weights to each channel, thus enhancing the representation of important features in the solar flare fusion feature data. Therefore, we integrated this module into the residual block of the TCN to improve the overall model performance.

After the temporal sequences have passed through two layers of TCN residual blocks, the obtained features are then fed into a classifier consisting of fully connected layers (FC1) to classify the flare forecasts. The first fully connected layer of the classifier, which is used to take over the TCN output features and perform feature mapping, is followed by a BN layer that can normalize the network. In addition, in the TCN model classifier, we use the sigmoid-weighted linear unit (SiLU) for the nonlinear mapping of the features and add a Dropout mechanism (N. Srivastava et al. 2014) to mitigate the overfitting of the model, which randomly deactivates a group of neurons during the training process, with the deactivation ratio q set at 0.5. The second fully connected layer (FC2) is the output layer of the network, which has only two neurons representing the two prediction categories of the flare class (F) and the no-flare class (N) two prediction classes. Finally, the final prediction is determined by the Softmax function.

As shown in Table 2, especially for the $\geq M$ model, the positive and negative sample time series are 440 and 2680, respectively, with a ratio of 1:6, and the number of positive and negative samples in our data set is severely imbalanced. For our TCN model, we use the label-distribution-aware margin loss (LDAMLoss; K. Cao et al. 2019) to deal with the data imbalance problem. The loss function is defined as

$$\mathcal{L}_{LDAM}((x, y); f) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}}, \quad (3)$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1, \dots, k\}. \quad (4)$$

In the formula, j represents a specific class, n_j is the number of samples in that class, and C is a constant. In our experiments, we conducted several experiments on the validation set, trying different values of C and observing the performance of the model. Eventually, we found that the constant $C = 30$ was able to achieve the best performance, so we chose 30 as our constant

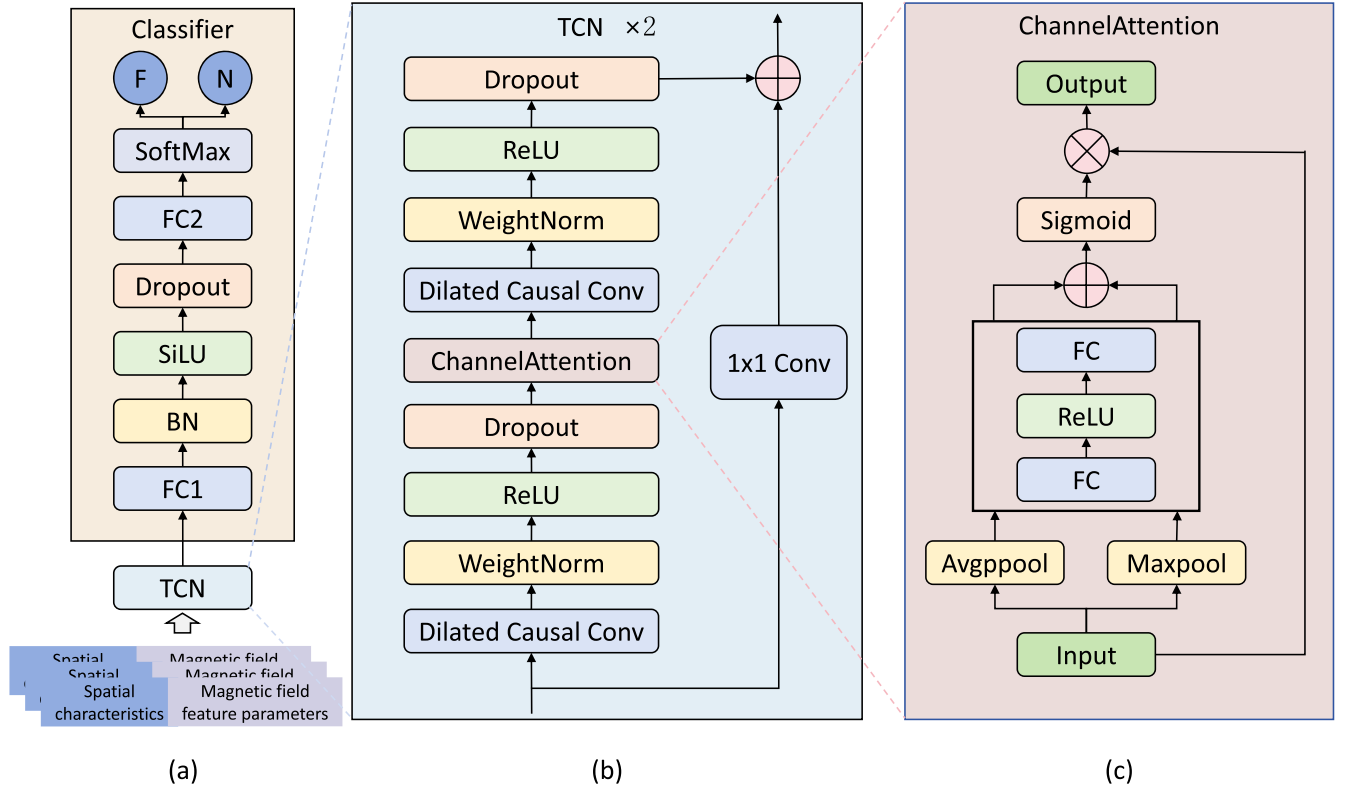


Figure 2. Specific structure of the TCN network. Panel (a) represents the flow of the TCN model. Panel (b) represents the TCN residual block. When the residual inputs and outputs have different dimensions, a 1×1 convolution is added. Panel (c) represents the flow of the channel attention mechanism. Channel attention pools the feature maps in the spatial dimension and preserves the feature information of the channel. The input feature maps are first pooled with global maximum pooling and global average pooling, respectively, and then the spatial dimension is compressed to 1 to preserve the channel information. The two pooled features are fed into a shared multilayer perceptron (MLP) to extract features. Finally, the pooled features after the MLP are summed up and sigmoid activation is performed to obtain the final channel attention weights.

C . y is the label corresponding to sample x . Let an example be (x, y) and a model f . Then $z_y = f(x)y$ denotes the model output for that sample. If there is no Δ_y , then it is the traditional CELoss, the softmax value of the correct category class is found first and then the negative logarithm is taken. The LDAMLoss is changed from Δ_y to $z_y - \Delta_y$, in the case that the output of other classes remains unchanged, when the output of the model is $z_y + \Delta_y$, the loss of LDMA is the same as the loss of CE when the output of the model is z_y , that is to say, the LDMA algorithm hopes that the output of the correct class can be Δ_y higher than the original one, so as to achieve the optimal classification effect.

3.4. Feature Visualization

Feature visualization (M. D. Zeiler & R. Fergus 2014) is a model interpretation technique that analyzes what information is extracted from the input data of a deep learning model by visualizing the results of the computation of the intermediate layers of the model. This method is often used in CNNs, where the output feature maps from a layer of the model are mapped to the range of 0–255 and then displayed as an image. After feature visualization, the visualized feature map is compared and analyzed with the original input map to explain the model. Also, the feature map visualization is generally used in the model after the training has been completed. Only the information extracted and how decisions were made are of interest to the final obtained model focus.

Table 4
Contingency Table for Binary Solar Flare Forecasting

	Forecasting Flaring	Forecasting Nonflaring
Observed Flaring	True Positive (TP)	False Negative (FN)
Observed Nonflaring	False Positive (FP)	True Negative (TN)

4. Forecast Results

4.1. Performance Metrics

Our solar flare prediction is a binary classification problem with two output types, representing the classes of flare (F) and no flare (N). To quantitatively evaluate the performance of this binary prediction model, the definition of the list of columns is shown in Table 4. The binary prediction model has four possible outcomes. Flare samples are considered to be in the positive category and no-flare samples are considered to be in the negative category. The number of samples correctly predicted as positive is true (TP), while the number of samples correctly predicted as negative is true negative (TN). The number of samples incorrectly predicted as positive is false positive (FP), while the number of samples incorrectly predicted as negative is false negative (FN).

Using the values of TP, TN, FP, and FN, many commonly used model evaluation metrics can be calculated, such as recall, precision, false alarm ratio (FAR), accuracy (ACC), Heidke Skill Score (HSS; P. Heidke 1926), and true skill statistics (TSS; A. Hanssen 1965). In our study, we measure the

Table 5
Description of the Six Metrics

Metric	Formula	Range
Recall	$TP/[TP + FN]$	0–1
Precision	$TP/[TP + FP]$	0–1
Accuracy	$[TP + TN]/[TP + FP + TN + FN]$	0–1
FAR	$FP/[FP + TP]$	0–1
HSS	$2[(TP \times TN) - (FP \times FN)] / [(TP + FN) \times (FN + TN) + (TP + FP) \times (FP + TN)]$	$-\infty$ –1
TSS	$TP/[TP + FN] - FP/[FP + TN]$	–1–1

performance of our models by the six metrics. Table 5 shows the formulas and ranges for the six metrics. We know ranges for recall, precision, and accuracy, with unity representing perfect results. We know the range of FAR, with zero representing a perfect result. We know the range of HSS, with unity indicating perfect result and less than zero indicating no skill. We know the range of TSS, with unity indicating that all predictions are correct, and zero indicating no skill. Among these metrics, only the TSS score is insensitive to the class imbalance ratio (D. S. Bloomfield et al. 2012). Recent studies, including our own, adhere to the recommendation by D. S. Bloomfield et al. (2012) to use the TSS score as the primary evaluation metric, while considering other metrics as supplementary indicators.

4.2. Evaluations

We calculated the values in the confusion matrix (TP, TN, FP, FN) after fivefold cross validation. The validation metrics can be calculated from these matrices. We trained and tested our model on five training/testing split data sets and used the means and the standard deviations of each metric on the five testing sets as our evaluation results. This evaluation approach is similar to those of X. Li et al. (2020), Z. Deng et al. (2021), and P. Sun et al. (2022). The metric scores of the different models (N. Nishizuka et al. 2018; H. Liu et al. 2019; Y. Zheng et al. 2019; X. Wang et al. 2020; and P. Sun et al. 2022), along with their means and standard deviations, are compared with previously published works, as listed in Table 6.

For our $\geq C$ model, the recall and precision are 0.952 ± 0.028 and 0.967 ± 0.033 , respectively, which are slightly higher than in the previous studies. In the FAR metric, both our $\geq C$ and $\geq M$ models perform better than the other models. In the ACC metric, both our $\geq C$ model and $\geq M$ model perform relatively well, but ACC is sensitive to imbalanced data sets, so we do not give much consideration to the ACC metric. For the HSS metric, our two models achieve relatively high scores of 0.922 ± 0.056 and 0.752 ± 0.103 , respectively, and for TSS, which is the main metric, the $\geq C$ model achieves a TSS score of 0.798 ± 0.032 and the $\geq M$ model achieves a TSS score of 0.850 ± 0.074 , which are higher than the results of other previous studies.

Our evaluation shows that both proposed models performed well on most metrics, particularly the TSS metric. While both our study and the compared studies used similar types of data sets, the differences such as the time range and data preprocessing may lead to variations in how the models perform on different data sets. These differences limit the validity of direct comparisons. Nevertheless, the comparison results provide useful initial insights into the relative

Table 6
Performance Evaluation between This Work and Previous Studies

Metric	Method	$\geq C$	$\geq M$
Recall	N. Nishizuka et al. (2018)
	H. Liu et al. (2019)	0.773	0.885
	Y. Zheng et al. (2019)	0.671	0.617
	X. Wang et al. (2020)	0.621	0.730
	P. Sun et al. (2022)	0.862	0.925
	This work	0.952 ± 0.028	0.741 ± 0.128
Precision	N. Nishizuka et al. (2018)
	H. Liu et al. (2019)	0.541	0.222
	Y. Zheng et al. (2019)	0.670	0.699
	X. Wang et al. (2020)	0.635	0.282
	P. Sun et al. (2022)	0.878	0.595
	This work	0.967 ± 0.033	0.871 ± 0.141
FAR	N. Nishizuka et al. (2018)	0.470	0.820
	H. Liu et al. (2019)
	Y. Zheng et al. (2019)	0.330	0.301
	X. Wang et al. (2020)	0.068	0.049
	P. Sun et al. (2022)	0.107	0.099
	This work	0.031 ± 0.031	0.024 ± 0.028
ACC	N. Nishizuka et al. (2018)	0.820	0.860
	H. Liu et al. (2019)	0.826	0.907
	Y. Zheng et al. (2019)	0.812	0.849
	X. Wang et al. (2020)	0.883	0.945
	P. Sun et al. (2022)	0.879	0.904
	This work	0.961 ± 0.028	0.943 ± 0.024
HSS	N. Nishizuka et al. (2018)	0.530	0.260
	H. Liu et al. (2019)	0.526	0.323
	Y. Zheng et al. (2019)	0.535	0.551
	X. Wang et al. (2020)	0.557	0.382
	P. Sun et al. (2022)	0.756	0.667
	This work	0.922 ± 0.056	0.752 ± 0.103
TSS	N. Nishizuka et al. (2018)	0.630	0.800
	H. Liu et al. (2019)	0.612	0.792
	Y. Zheng et al. (2019)	0.538	0.534
	X. Wang et al. (2020)	0.553	0.681
	P. Sun et al. (2022)	0.756	0.826
	This work	0.798 ± 0.032	0.850 ± 0.074

performance of the different methods in solving similar problems.

4.3. Discussion of Model Superiority

To further discuss the superiority of the proposed model, solar flare forecasting was performed using the CNN model and the TCN model individually, and the evaluation results were compared with the CNN-TCN model. We discuss below the superiority of integrating both the spatial and temporal features over only a single factor for solar flare forecasting.

A single CNN model only considers the forecasting effect of the spatial structure characteristics of the magnetic field in AR. The CNN model architecture used here is modified from the CNN in the CNN-TCN model. In the original CNN model, the fully connected layer at the end was used as the projection head in contrastive learning, performing feature mapping rather than flare classification prediction. The original projected head was modified to a classifier, and the rest of the model was kept the same as the original. The hyperparameters used in the training of the CNN model also remain the same as the original. The

Table 7
Evaluation Results of Different Models

Metric	Method	$\geq C$	$\geq M$
Recall	CNN	0.987 ± 0.013	0.999 ± 0.001
	TCN	0.926 ± 0.027	0.637 ± 0.021
	CNN-TCN	0.952 ± 0.028	0.741 ± 0.128
Precision	CNN	0.640 ± 0.014	0.638 ± 0.297
	TCN	0.938 ± 0.014	0.944 ± 0.029
	CNN-TCN	0.967 ± 0.033	0.871 ± 0.141
FAR	CNN	0.237 ± 0.015	0.164 ± 0.162
	TCN	0.058 ± 0.015	0.012 ± 0.006
	CNN-TCN	0.031 ± 0.031	0.024 ± 0.028
ACC	CNN	0.830 ± 0.009	0.860 ± 0.139
	TCN	0.934 ± 0.010	0.908 ± 0.006
	CNN-TCN	0.961 ± 0.028	0.943 ± 0.024
HSS	CNN	0.649 ± 0.015	0.671 ± 0.285
	TCN	0.868 ± 0.020	0.706 ± 0.020
	CNN-TCN	0.922 ± 0.056	0.752 ± 0.103
TSS	CNN	0.719 ± 0.095	0.726 ± 0.092
	TCN	0.805 ± 0.028	0.783 ± 0.223
	CNN-TCN	0.798 ± 0.032	0.850 ± 0.074

single TCN model only considers the forecasting effect of the magnetic field evolution characteristics of AR, and it uses only the 16 magnetic field characteristic parameter data of AR. The TCN model is consistent with the TCN model in the CNN-TCN model. In our study, forecasting whether or not to erupt $\geq C$ and $\geq M$ level flares, the single CNN and the single TCN solar flare forecasting models each have the same structure when forecasting $\geq C$ and $\geq M$ level flares, respectively.

The CNN model and the TCN model both use fivefold cross validation for performance evaluation. The evaluation criteria are the same as the six performance metrics mentioned earlier, with the means and the standard deviations calculated over the fivefold cross validation. The metric scores for the individual models and the CNN-TCN model are shown in Table 7. For ease of reading, a set of data from Table 6 is also presented in Table 7. As shown in Table 7, we also provide the means and standard deviations of the metric scores. In terms of specific performance indicators, for forecasting $\geq C$ -class flares, the CNN model has the highest recall, indicating its strong ability to predict flare samples. However, at the same time, it also has the highest FAR, which indicates that the CNN model has a tendency to over-forecast, resulting in the generation of too many false alarms. The TCN model outperforms our model only in the TSS metric, but the FAR is also higher than the CNN-TCN model, suggesting that the TCN model has a tendency to over-forecast. For forecasting $\geq M$ -class flares, the CNN model has the highest recall, indicating its strong ability to predict flare samples. Moreover, we note that the overall trend for $\geq C$ -class flare prediction shows higher standard deviation values for the CNN-TCN method. We speculate that this stems from two main reasons. First, the combined CNN-TCN model's higher complexity compared to the single CNN model or the single TCN model. The CNN-TCN model can capture more features, but its sensitivity to data also increases accordingly; this also makes the model more volatile in different fold cross validation. This phenomenon is common in complex architectures of multilayer neural networks. Second,

the use of fivefold cross validation further amplifies this volatility. The prediction results of complex models are prone to large variations in the different training and testing sets, which is an important reason for triggering high standard deviation values (Y. Bengio & Y. Grandvalet 2004). Overall, the CNN-TCN model outperforms both the single CNN and the single TCN models in several key performance metrics. Since the CNN model only uses the magnetogram as training data, it cannot take into account the temporal evolution information of the magnetic field in the AR. The TCN model only uses the parameters of the magnetic field characteristics of the AR as training data, and these parameters do not contain the spatial structure characteristics of the magnetic field in the AR. This indicates that the improved performance of our model is indeed due to the fusion of the spatial structure features of the magnetic field with the magnetic field parameter data in AR. Thus, this suggests that a comprehensive consideration of the temporal and the spatial characteristics of the magnetic field in AR is indeed more conducive to flare forecasting. In summary, the CNN-TCN model has higher flare prediction performance than models using only the single CNN or the single TCN. The experimental results demonstrated the validity and superiority of the methodology.

5. Visualization Analysis and Discussion

Using the CNN model described in Section 3.2, we now visualize the feature activations on the magnetogram data set. Figures 3(b), (d), and (f) show the feature visualization of our CNN model after training. We chose three magnetograms of the AR taken at different times for feature visualization. Figure 3(a) shows the magnetogram of AR 12715 taken on 2018 June 20, at 01:12 UT; Figure 3(c) shows the magnetogram of AR 12381 taken on 2015 July 5, at 21:00 UT; and Figure 3(e) shows the magnetogram of AR 11158 taken on 2011 February 14, at 01:48 UT. The feature visualization is used to show the information extracted by the CNN model at this point in time when processing this magnetogram. Figures 3(a), (c), and (e) show the original AR magnetogram that will be visualized for feature map visualization, and the feature map shown in Figures 3(b), (d), and (f) shows that the corresponding feature maps are the outputs of the third DepthwiseSeparableConv layer of the CNN model in this paper. The output of this convolutional layer contains 64 channels and the feature maps of all the channels are visualized. These maps reflect specific features extracted by the convolutional filters, such as edges, textures, or other spatial patterns (K. Simonyan & A. Zisserman 2014). Unlike the original input images, these feature maps are not intended to be directly interpretable by the human eye. Instead, they capture abstract representations that are useful for the model's decision-making process. The apparent blurriness or lack of clarity in these visualizations is a result of this abstraction (M. D. Zeiler & R. Fergus 2014). As the convolutional layers focus on identifying and amplifying particular features, they naturally omit other details that are less relevant to the task at hand (R. R. Selvaraju et al. 2017). This selective feature extraction leads to images that may appear less distinct, but these images contain the crucial information needed by the model (K. Simonyan & A. Zisserman 2014). The main information retained by the model is boxed using red rectangles in Figures 3(b), (d), and (f), corresponding to the information boxed by red rectangles in the original input magnetograms of

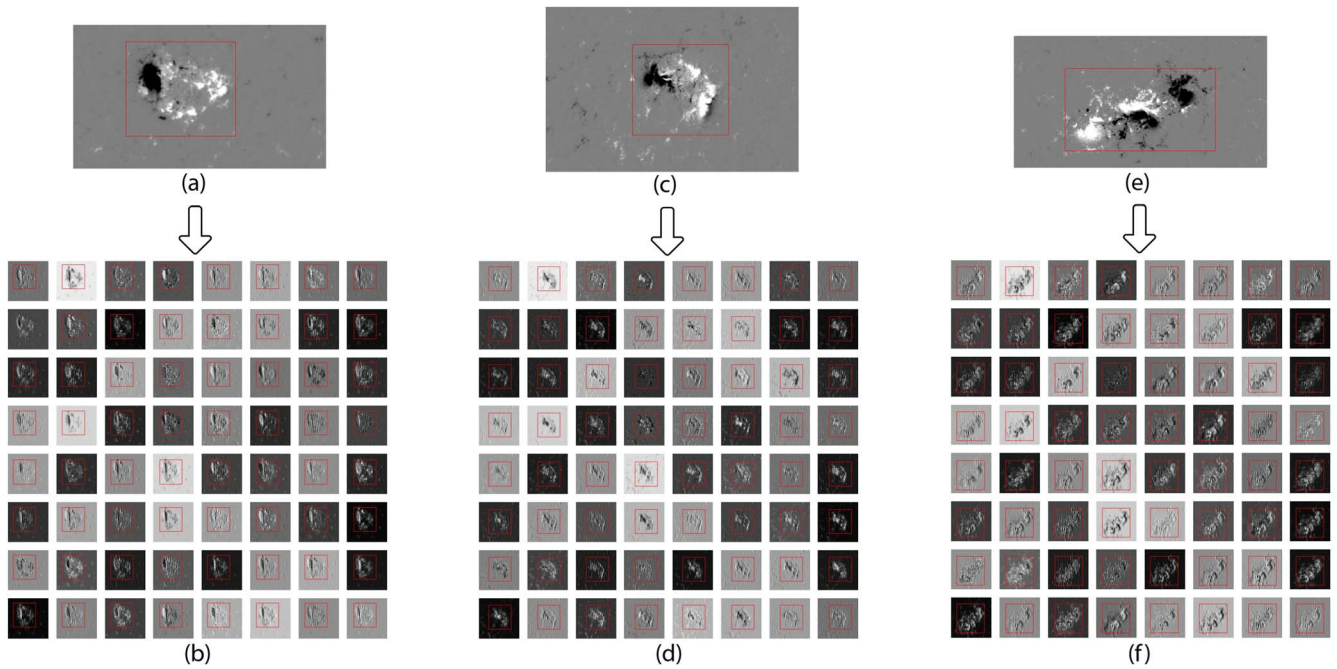


Figure 3. Panels (a), (c) and (e) show the original magnetograms of the AR for feature map visualization; panels (b), (d), and (f) show the corresponding feature visualization results.

Figures 3(a), (c), and (e), respectively. From the visualization results of these three magnetograms, we can intuitively observe that after the convolution operation of the model, the model retains the main morphological structure of the magnetic field in the magnetogram, while eliminating the smaller information and the background part of the magnetogram. The CNN model has preserved the high-frequency part of the magnetogram of AR, which is the main part of the magnetic field of the positive and negative poles in the magnetogram. By presenting magnetograms at different times, we verify the consistency and reliability of the model in dealing with diverse data. These results provide preliminary visual evidence that the model helps to focus on key features when processing magnetograms. However, in order to further validate the generalizability and robustness of this observation, quantitative analyses and assessments of larger samples are needed.

6. Conclusions

We investigate a solar flare forecasting model based on a combination of CNN and TCN. The forecasting objective is whether flares of $\geq C$ - and $\geq M$ -class will erupt in the AR in the next 24 hr. To sufficiently utilize both the magnetic field spatial distribution and evolution characteristics, we composed the 24 hr AR magnetograms and related magnetic field feature parameters provided by the Solar Dynamics Observatory's Helioseismic and Magnetic Imager (SDO/HMI) into temporal sequences as our training data. We used the fivefold cross validation method and the six performance metrics for our model evaluation. The results of the discussion based on the model superiority show that models combining the two significantly improve in prediction performance compared to using either the spatial distribution or evolution characteristics of the magnetic field alone. The CNN-TCN model can effectively forecast flares at $\geq C$ - and $\geq M$ -class in the next 24 hr and simultaneously extract the spatial distribution and temporal evolution features of the magnetic field in the AR.

This gives full play to the learning ability of TCN on temporal sequences data to better capture the complex temporal evolution information of the magnetogram in the AR.

Furthermore, by visualizing and analyzing the feature maps of the CNN model, we find that the CNN combined with the contrastive learning method can effectively extract the morphological structure of the positive and negative magnetic fields in the AR and eliminate the interference information in the background, obtaining more useful features of the spatial structure of the magnetic field. This finding provides new ideas and methods to further improve the accuracy of solar flare forecasting. Using the spatial distribution and evolution characteristics of AR magnetic fields for flare prediction holds significant practical application value.

Acknowledgments

We are very grateful to the reviewers for their valuable comments. We are thankful for the use of the SDO/HMI image obtained courtesy of NASA/SDO and the GOES team. The work is supported by the Open Fund of Yunnan Key Laboratory of Computer Technologies Application.

ORCID iDs

Song Feng <https://orcid.org/0000-0003-4709-7818>
Bo Liang <https://orcid.org/0000-0002-0604-0949>

References

- Ahmed, O. W., Qahwaji, R., Colak, T., et al. 2013, *SoPh*, **283**, 157
- Bengio, Y., & Grandvalet, Y. 2004, *JMLR*, **5**, 1089
- Bloomfield, D. S., Higgins, P. A., McAteer, R. J., & Gallagher, P. T. 2012, *ApJL*, **747**, L41
- Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, *SoPh*, **289**, 3549
- Cao, K., Wei, C., Gaidon, A., Arechiga, N., & Ma, T. 2019, *Advances in Neural Information Processing Systems 32*, ed. H. Wallach et al. (NeurIPS), https://papers.nips.cc/paper_files/paper/2019/hash/621461af90cadfdaf0e8d4cc25129f91-Abstract.html

- Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. 2020, in Proc. Machine Learning Research 119, Proc. of the 37th Int. Conf. on Machine Learning, ed. H. Daumé, III & A. Singh (PMLR), 1597, <https://proceedings.mlr.press/v119/chen20j.html>
- Chen, Y., Manchester, W. B., Hero, A. O., et al. 2019, *SpWea*, **17**, 1404
- Colak, T., & Qahwaji, R. 2009, *SpWea*, **7**, S06001
- Deng, Z., Wang, F., Deng, H., et al. 2021, *ApJ*, **922**, 232
- Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, *SoPh*, **293**, 28
- Glorot, X., Bordes, A., & Bengio, Y. 2011, in Proc. Machine Learning Research 14, Proc. 14th Int. Conf. on Artificial Intelligence and Statistics, ed. G. Gordon, D. Dunson, & M. Dudík (PMLR), 315, <http://proceedings.mlr.press/v15/glorot11a.html>
- Grill, J.-B., Strub, F., Altché, F., et al. 2020, Advances in Neural Information Processing Systems 33, ed. H. LaRochelle et al. (NeurIPS), <https://proceedings.neurips.cc/paper/2020/hash/f3ada80d5c4ee70142b17b8192b2958e-Abstract.html>
- Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, *SpWea*, **13**, 626
- Hanssen, A. 1965, *JApMC*, **4**, 172
- He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. 2020, in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 9729
- Heidke, P. 1926, *Geografiska Annaler*, **8**, 301
- Huang, X., Wang, H., Xu, L., et al. 2018, *ApJ*, **856**, 7
- Huang, X., Yu, D., Hu, Q., Wang, H., & Cui, Y. 2010, *SoPh*, **263**, 175
- Huang, X., Zhang, L., Wang, H., & Li, L. 2013, *A&A*, **549**, A127
- Ioffe, S., & Szegedy, C. 2015, arXiv:1502.03167
- Jaiswal, A., Babu, A. R., Zadeh, M. Z., Banerjee, D., & Makedon, F. 2020, *Technologies*, **9**, 2
- Li, R., Cui, Y., He, H., & Wang, H. 2008, *AdSpR*, **42**, 1469
- Li, R., & Zhu, J. 2013, *RAA*, **13**, 1118
- Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, *ApJ*, **891**, 10
- Liu, C., Deng, N., Wang, J. T., & Wang, H. 2017, *ApJ*, **843**, 104
- Liu, H., Liu, C., Wang, J. T., & Wang, H. 2019, *ApJ*, **877**, 121
- Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2018, *ApJ*, **858**, 113
- Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, *ApJ*, **835**, 156
- Park, E., Moon, Y.-J., Shin, S., et al. 2018, *ApJ*, **869**, 91
- Qahwaji, R., & Colak, T. 2007, *SoPh*, **241**, 195
- Sadykov, V. M., & Kosovichev, A. G. 2017, *ApJ*, **849**, 148
- Selvaraju, R. R., Cogswell, M., Das, A., et al. 2017, in Proc. IEEE International Conf. on Computer Vision, ICCV 2017 (Piscataway, NJ: IEEE), 618
- Shi, J., Liu, Y., Liu, Q., Zhang, Q., & Fei, Y. 2020, *JPhCS*, **1693**, 012056
- Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *JMLR*, **15**, 1929
- Sun, H., Manchester IV, W., & Chen, Y. 2021, *SpWea*, **19**, e2021SW002837
- Sun, P., Dai, W., Ding, W., et al. 2022, *ApJ*, **941**, 1
- Tang, R., Zeng, F., Chen, Z., et al. 2020, *Atmos*, **11**, 316
- Wang, H., Liu, C., Ahn, K., et al. 2017, *NatAs*, **1**, 0085
- Wang, X., Chen, Y., Toth, G., et al. 2020, *ApJ*, **895**, 3
- Wu, Z., Xiong, Y., Yu, S. X., & Lin, D. 2018, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 3733
- Yi, K., Moon, Y.-J., Shin, G., & Lim, D. 2020, *ApJL*, **890**, L5
- Yuan, Y., Shih, F. Y., Jing, J., & Wang, H.-M. 2010, *RAA*, **10**, 785
- Zeiler, M. D., & Fergus, R. 2014, in Computer Vision – ECCV 2014, ed. D. Fleet et al. (Cham: Springer), 818
- Zheng, Y., Li, X., & Wang, X. 2019, *ApJ*, **885**, 73