Predicting Solar Flares Using a Novel Deep Convolutional Neural Network

Xuebao Li1,3, Yanfang Zheng1,3,4, Xinshuo Wang1, and Lulu Wang2

1 College of Electrical and Information Engineering, Jiangsu University of Science and Technology, Zhangjiagang, People’s Republic of China; zyf062856@163.com
2 College of Software Engineering, East China Normal University, Shanghai, People’s Republic of China

Received 2019 November 5; revised 2019 December 23; accepted 2020 January 15; published 2020 February 27

Abstract

Space weather forecasting is very important, and the prediction of space weather, especially for solar flares, has increasingly attracted research interests with the numerous recent breakthroughs in machine learning. In this study, we propose a novel convolutional neural network (CNN) model to make binary class prediction for both \( \geq C \)-class and \( \geq M \)-class flares within 24 hr. We collect magnetogram samples of solar active regions (ARs) provided by the Space-weather Helioseismic and Magnetic Imager Active Region Patches (SHARP) data from 2010 May to 2018 September. These samples are used to construct 10 separate data sets. Then, after training, validating, and testing our model, we compare the results of our model with previous studies in several metrics, with a focus on the true skill statistic (TSS). The major results are summarized as follows. (1) We propose a method of shuffle and split cross-validation (CV) based on AR segregation, which is the first attempt to verify the validity and stability of the model in flare prediction. (2) The proposed CNN model achieves a relatively high score of TSS = 0.749 \( \pm \) 0.079 for \( \geq M \)-class prediction, and TSS = 0.679 \( \pm \) 0.045 for \( \geq C \)-class prediction, which is greatly improved compared with previous studies. (3) The model trained on 10 CV data sets is considerably robust and stable in making flare prediction for both \( \geq C \) class and \( \geq M \) class. Our experimental results indicate that our proposed CNN model is a highly effective method for flare forecasting, with quite excellent prediction performance.

Unified Astronomy Thesaurus concepts: Convolutional neural networks (1938); Magnetic fields (994); Solar activity (1475); Solar flares (1496); Astronomical techniques (1684)

1. Introduction

A solar flare is the result of the energy release of a magnetic field mainly in the form of electromagnetic radiation and high-energy particles. Therefore, a strong flare occasionally causes rapid and significant changes in the near-Earth space environment, which is referred to as space weather (e.g., Handzo et al. 2014; Hayes et al. 2017). There have been several reports that space weather events associated with major flares affect space assets, harm human health, and degrade critical infrastructure, such as the malfunction of satellites, radiation hazards for astronauts, and the interruption of radio communication and electric power (e.g., Schrijver et al. 2014; Ponomarchuk et al. 2015). Hence, it is very important to establish a reliable system for flare prediction to effectively prevent various types of damage caused by powerful flare events.

The triggering mechanism of solar flares is not well understood, thus many flare prediction models have been developed with different methods in recent years. Some of them are based on statistical methods (Song et al. 2009; Mason & Hoeksema 2010; Bloomfield et al. 2012; Barnes et al. 2016), while most of them are based on machine-learning methods. Traditional machine-learning methods commonly used for flare prediction cover artificial neural network (Qahwaji & Colak 2007; Ahmed et al. 2013; Li & Zhu 2013; Nishizuka et al. 2018), k-nearest neighbors (Li et al. 2008; Huang et al. 2013), support vector machine (Yuan et al. 2010; Bobra & Couvidat 2015; Nishizuka et al. 2017; Sadykov & Kosovichev 2017), random forests (Liu et al. 2017; Florios et al. 2018), and ensemble learning (Colak & Qahwaji 2009; Huang et al. 2010; Guerra et al. 2015). Because flares originate from the magnetic field around sunspots in an active region (AR), many of the above studies utilize photospheric magnetic observational data of AR for binary class prediction. Both statistical methods and traditional machine-learning algorithms learn from features that are manually selected and extracted from the observational data. The predictive performance of these methods mainly relies on the quality of the features used.

Deep learning neural networks (DNNs), which are as a new branch of machine learning, have become a highly reliable technique for solving large-scale learning problems in astronomy and other branches of science (Abraham et al. 2018). A famous method of DNNs, convolutional neural networks (CNNs; LeCun et al. 2015), have become very popular in the area of image processing and computer vision. It could learn directly from the raw data instead of features. CNNs are usually composed of several convolutional layers, and each convolutional layer is trained to automatically select and extract increasingly complex features from input data. There are some literatures reporting that CNNs have been successfully applied to solar flare prediction. Park et al. (2018) applied a CNN model to predict a binary class flare within 24 hr, which is based on a combination of GoogLeNet (Szegedy et al. 2014) and DenseNet (Huang et al. 2016). After trained by full-disk solar line of sight (LOS) magnetograms, their model could only make binary class prediction for a \( \geq C \)-class flare, which lacked research on the prediction for a \( \geq M \)-class major flare. Huang et al. (2018) presented a CNN model to predict a binary class flare, which is based on the classic CNN model architecture including two convolutional layers with 64 \( 11 \times 11 \) filters. After trained by many patches of solar
ARs from LOS magnetograms located within ±30° of the solar disk center, their model could predict a ≥C/M/X-class flare, but their predictive performance needs to be further improved.

All the above studies are based on statistical methods or machine-learning algorithms, and simulate the prediction process of future flare activity given current solar data by segregating the data into the training and testing data set. The method of constructing data set including training and testing data set also plays a very important role in the model evaluation and flare prediction. According to previous studies, there are mainly two data segregation methods for flare prediction. One method is selecting randomly shuffled data sets with some similarities between the training and testing data set due to closeness in time (e.g., Bobra & Couvidat 2015; Nishizuka et al. 2017; Florios et al. 2018), which most probably leads to the improvement of the prediction performance of the model. The other method is splitting of the data into one single training and testing data set in chronological order (e.g., Bloomfield et al. 2012; Huang et al. 2018; Nishizuka et al. 2018; Park et al. 2018), where the model was trained and evaluated by one single data set to provide the best value of evaluation metric. To show the validity and stability of the prediction model, the method of cross-validation (CV) are usually utilized, which is the standard approach in the field of machine learning. Many of the above studies used the CV method to divide the data into training and testing data by random selection (e.g., Liu et al. 2017; Huang et al. 2018; Park et al. 2018). This method cannot ensure that the data samples in the testing data set are not similar to those in the training data set, which is likely to result in an increase in the validity and stability of the prediction model.

In this paper, we attempt to propose a new CNN model that is different from previous models to make binary class prediction for both ≥C-class and ≥M-class flares that would occur in an AR within 24 hr. Moreover, we propose a CV method based on AR segregation, which is also different from previous studies and more suited for model evaluation and flare prediction. The raw input data for the model in our work is from LOS magnetograms of ARs provided by the Helioseismic and Magnetic Imager (HMI; Schou et al. 2012) on board the Solar Dynamics Observatory (SDO; Pesnell et al. 2012). The output result for the model is compared with Geostationary Operational Environment Satellite (GOES) observations of the daily flare occurrence.

The rest of this paper is organized as follows. The data is described in Section 2, and the method is introduced in Section 3. Results are given in Section 4, and finally, conclusions and discussions are provided by Section 5.

2. Data

In this study, we adopt the data product named Space-weather HMI Active Region Patches (SHARP; Bobra et al. 2014), which is provided by the SDO/HMI team. These data was released at the end of 2012, and is now publicly available at the Joint Science Operations Center. The SHARP data contains automatically identified and tracked ARs in map patches, which is convenient for flare prediction. We survey flare events that occurred from 2010 May 1 to 2018 September 13 covering the main peak of solar cycle 24, using the GOES X-ray flare catalogs provided by the National Centers for Environmental Information (NCEI, https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-features/solar-flares/x-rays/goes/xrs/), and select flares with identified ARs in the NCEI. It is worth noting that many records of the flare events are short of locations and National Oceanic and Atmospheric Administration (NOAA) AR numbers. Thus we supplement the missing location information and AR numbers for the associated flares referring to the information from Solar Geophysical Data Solar event reports (http://www.solarmonitor.org/index.php). Finally, the LOS magnetograms of ARs as the raw input data of the model can be obtained from the SHARP data. The cadence of the data is 12 minutes. To avoid the projection effects (Ahmed et al. 2013; Bobra et al. 2014), we choose the LOS magnetograms located within ±45° of the central meridian.

We need to build data sets to train and evaluate the model. Then the data sets are prepared in the following way: (1) if the AR does not flare within 24 hr after the observation time, the No-flare (weaker than C1.0) label is assigned to the magnetogram sample in the same AR. (2) If the C/M/X-level flare occurs within 24 hr after the observation time, the corresponding flare label (i.e., C, M, or X) is assigned to the magnetogram sample. Note that there are a number of ARs producing recurring flares with different flare levels within 24 hr. For the first flare of one AR, the corresponding flare label is assigned to the magnetogram sample within 24 hr after the observation time. Then for the following flares of the same AR, the corresponding labels are assigned to the magnetogram in the period from the end of its prior flare to the end of this flare. (3) We adopt a four-level AR classification scheme based on the maximum GOES-level flare an AR ever yields (Song et al. 2009; Yuan et al. 2010; Liu et al. 2017). In other words, ARs are further categorized into four levels (i.e., No-flare, C, M, and X) if they yield at least one flare with such GOES-level criterion but no flares with higher GOES-level criterion: “Level = X” indicates that an AR yields at least one X-level flare; “Level = M” indicates that an AR yields at least one M-level flare but no X-level flares; “Level = C” indicates that an AR yields at least one C-level flare but no M/X-level flares; “Level = No-flare” indicates that an AR only yields microflares (weaker than C1.0 flares). Finally, we gather 870 ARs and 136134 magnetogram samples in total, including 443 X-level, 6534 M-level, 72412 C-level, and 56745 No-flare level samples. Note that the magnetogram samples with multiple ARs (Bobra et al. 2014) are not included in our work. For the ≥M class, magnetogram samples of M/X-level flare in an AR are defined as positive class, while all the others are defined as negative class. For the ≥C class, magnetogram samples of C/M/X-level flare in an AR are defined as positive class, while all the others are defined as negative class.

The CV methods are usually used to show the validity and stability of prediction model. There are several kinds of CV methods, such as K-fold CV, leave-one-out CV, etc. (Nishizuka et al. 2017). As the number of ARs in four levels is strongly imbalanced, we propose a method of shuffle and split CV based on AR segregation in our study, which is used in flare prediction for the first time. Figure 1 shows the main flowchart of 10 separate CV data sets constructed by the method of shuffle and split CV based on AR segregation. As those of the magnetograms in the same AR have some similarity, we perform this segregation by NOAA AR number. First, we randomly shuffle the AR numbers in different levels of No-flare/C/M/X, and then split the AR numbers at a ratio of around 20%-80%. Second, the magnetogram samples
Figure 1. Flowchart of 10 separate CV data sets constructed by the method of shuffle and split CV based on AR segregation.
Figure 2. Number of solar magnetogram samples and ARs for 10 separate data sets. In every level of each data set, “Original” represents the number of magnetogram samples before data preprocessing, “Preprocessed” represents the number of magnetogram samples after data preprocessing used for our study, and “AR numbers” represents the number of ARs after data preprocessing used for our study. Data preprocessing indicates the sample images are processed through excluding the samples with multiple ARs, image augmentation, and undersampling.
belonging to 20% ARs are assigned to the testing data set, and those belonging to 80% ARs are assigned to the training data set. It demands that not only the magnetogram samples but also ARs in the testing data set do not appear in the training data set. It is pretty obvious that the number of M/X-level magnetogram samples is far less than that of No-flare/C-level magnetogram samples, which consists with the fact most ARs do not yield major flares in the period of any given 24 hr. This would result in serious class-imbalance issue, which is a major problem in the field of machine learning. Third, we employ undersampling and image augmentation schemes to alleviate this issue. On the one hand, we undersample the data set by randomly choosing No-flare/C-level samples with around 2 samples per 10 samples. One the other hand, we adopt image augmentation schemes to artificially increase the number of M/X-level samples by rotating and reflecting images, which makes the model invariant to the rotation and reflection in pixel values. As the CNNs require a fixed size for all input images, these images from our data sets are resampled to 128 × 128 pixels following a similar method of Huang et al. (2018). Finally, we complete the construction of one single data set containing training and testing data set. Repeating the above process of building one single data set 10 times in total, we obtain 10 separate CV data sets consisting of training and testing data sets. Figure 2 shows the number of solar magnetogram samples and ARs for 10 CV data sets. The advantage of this method is that in each of the 10 data sets, not only the samples in the testing data set do not overlap with those in the training data set, but also the ARs in the testing data set are disjoint from those in the training data set. We train and evaluate our model on these 10 separate training and testing data sets.

3. Method

CNNs generally contain convolutional layers, pooling layers, and fully connected layers, which are some of most important components of CNN models. The input to a convolutional layer is usually images, and the output channels of each convolutional layer are called feature maps. To yield feature maps, the convolutional layer performs a convolution operation between the input of the layer and a set of weights called kernels or filters. The output $k$th feature map can be mathematically represented as

$$y^k = \sigma \left( \sum_m W^k_m \otimes x_m + b^k \right),$$

where $\otimes$ is the convolution operator, $b^k$ is the bias, and $W^k_m$ represents the filters. $\sigma$ is the activation function (or nonlinearity), which is usually rectified linear unit (ReLU; Nair & Hinton 2010). The pooling layer performs downsampling of the input feature maps by taking average (average-pooling) or taking maximum (max-pooling) value within a square neighborhood. Thus it reduces the dimensionality of the feature maps and makes the model insensitive to small shifts and variations (Boureau et al. 2010). The fully connected layer is used to carry out classification within the network. Each neuron in this layer is connected to all activated outputs in the previous layer. Neurons in this layer are randomly turned off by putting to zero a random selection of connection weights with certain probability during training. This procedure is called dropout and is a mechanism to avoid overfitting (Hinton et al. 2012). Furthermore, batch normalization (BN; Ioffe & Szegedy 2015) has been developed to regularize the network and stabilize the training process, which is usually added behind the convolutional layer or fully connected layer in the model. In general, a common design of CNNs consists of a mixture of convolutional and pooling layers, followed by fully connected layers.

In this paper, we consider a novel CNN model to make binary class prediction for both $\geq C$-class and $\geq M$-class flares. The model is trained and tested on the Keras framework using the TensorFlow (Abadi et al. 2016) backend in Python programming language. The output of the last fully connected layer in our model is fed to a softmax activation function, which converts the score of each class to probability $p(y)$. Here, for the $\geq M$ class ($\geq C$ class, respectively), $(y_1, y_2) = (0, 1)$ is the class of $\geq M$-level ($\geq C$-level, respectively) flare events, and $(y_1, y_2) = (1, 0)$ is the class of $< M$-level ($< C$-level, respectively) flare events. The softmax function computes two probabilities for $\geq M$-level ($\geq C$-level, respectively) flare events and for $< M$-level ($< C$-level, respectively) flare events, and then the final prediction result is the highest probability class.

We design the proposed CNN model in this study, which is inspired by the VGG network (Simonyan & Zisserman 2015) and the Alexnet network (Krizhevsky et al. 2012). We attempt to construct different structures of the model with different numbers of convolutional layers and different sizes of filters, and finally we only describe the structure of the best performance network in this paper. Figure 3 shows the architecture of the model for both $\geq C$-class and $\geq M$-class flare prediction. As shown in Figure 3, the architecture of the model for $\geq C$ class is the same as that for $\geq M$ class. The model has 28 different functions and layers including the different convolutional layers, pooling layers, fully connected layers, BN layers, dropout functions, and ReLU and softmax activation functions. In the first two convolutional layers of the model, we select the filter of size $11 \times 11$ to learn basic features, and $11 \times 11$ is the optimum size obtained after many attempts, which is learned from the Alexnet network. However, in the last three convolutional layers of the model, we select the filter of size $3 \times 3$ instead of $11 \times 11$ to learn complicated high-level features, which is learned from the VGG network. The filter of large size (e.g., $11 \times 11$) will increase the number of model parameters to be computed, which is not beneficial to the increase of the model depth. Different from the Alexnet network and the VGG network, we add a BN layer behind each convolutional layer and fully connected layer in our model to regularize the network.

The mini-batch strategy is used to achieve faster convergence of training error (Goodfellow et al. 2016). Here, the mini-batch size is equal to the number of training samples in one forward and backward pass. To maximize the prediction accuracy, the model is trained to minimize a loss function, which is usually computed from the cross entropy loss (Hinton & Salakhutdinov 2006). As the number of ARs in our data sets is imbalanced, we adopt the loss function calculated from the

---

5 We provide the source codes for downloading four different levels of samples, the codes for constructing 10 CV data sets by the method of shuffle and split CV, and the codes for the proposed CNN model at https://github.com/FlarePrediction/Repository.
Figure 3. Architecture of the model for both ≥C-class and ≥M-class flare prediction.
weighted cross entropy loss as follows:

$$\text{loss} = \sum_{n=1}^{N} \sum_{k=1}^{K} \omega_{k} y_{nk} \log_{e}(\tilde{y}_{nk}),$$  \hspace{0.5cm} (2)

$$\omega_{k} = f(A_{k}) f(S_{k}) \beta_{k},$$  \hspace{0.5cm} (3)

where $N$ is the number of training samples in each mini-batch size, $K$ is the number of classes, $\tilde{y}_{nk}$ and $y_{nk}$ denote the predicted output and the expected output of the $k$th class during a forward pass, respectively. $\omega_{k}$ indicates the weight of the $k$th class, which is used for weighting the loss function during training. $A_{k}$ represents the ARs of the $k$th class, and $f(A_{k})$ refers to the number of statistics for ARs of the $k$th class. $S_{k}$ represents the expected output of the $k$th class during a forward pass, respectively. $\omega_{k}$ indicates the weight of the $k$th class, which is used for weighting the loss function during training. $A_{k}$ represents the ARs of the $k$th class, and $f(A_{k})$ refers to the number of statistics for ARs of the $k$th class. $S_{k}$ represents the

$\beta_{k}(k=0, 1)$ is the optimized parameter used to adjust $\omega_{k}$, which is obtained through experiment. It is related to the product of $f(A_{k})$ and $f(S_{k})$, and the relationship between $\beta_{k}$ and the product of $f(A_{k})$ and $f(S_{k})$ presents a trend of inverse proportion. $\beta_{0}$ and $\beta_{1}$ comes in pairs for the negative class and positive class. In our study, $(\beta_{0}, \beta_{1})$ is $(1, 0.8)$ for the $\geq$C-class and $(1, 30)$ for the $\geq$M class, respectively. The choice of training parameters such as learning rate, momentum, class weight, etc. is also significant for achieving better prediction performance. The training parameters used in our CNN model are summarized in Table 1. As shown in Table 1, our model is trained over 80

### Table 1

| Model         | Optimizer | Learning Rate | Momentum | $\beta_{0}$ | $\beta_{1}$ | Batch Size | Epoch |
|---------------|-----------|---------------|----------|-------------|-------------|------------|-------|
| Model ($\geq$C class) | SGD       | $1.0 \times 10^{-3}$ | 0.5      | (1, 0.8)    | 16          | 80         |
| Model ($\geq$M class) | SGD       | $1.0 \times 10^{-3}$ | 0.5      | (1, 30)     | 16          | 80         |

Figure 4. Learning curves showing the result of training and validating loss per epoch for our CNN model. The 10 different colored curves represent the changes of training and validating loss with epochs for the model trained and validated by 10 separate data sets. (a) and (b), (c) and (d), respectively, show the results of training and validating loss per epoch for the model used to predict $\geq$C-class ($\geq$M-class, respectively) flares.
epochs using stochastic gradient descent (SGD; LeCun et al. 1998) as an optimizer with a mini-batch size of 16, where one epoch means that a whole training data set is passed through the model. In the training phase, the gradients of the loss function with respect to the weights of the network are estimated with mini-batch samples at each iteration, and then the error backpropagation adjusts the weights of the network by the gradient descent to minimize the loss function.

### 4. Results

We treat flare level prediction as a binary classification task. The $\geq$M-class ($\geq$C-class, respectively) samples are considered as positive class, and the $<$M-class ($<$C-class, respectively) samples are considered as negative class. Therefore, there are four possible outcomes. The samples correctly classified as positive are defined as true positive (TP), while the samples correctly classified as negative are defined as true negative (TN). On the other hand, the samples wrongly predicted as positive are defined as false positive (FP), and the samples wrongly predicted as negative are defined as false negative (FN). These four quantities construct a confusion matrix, also called a contingency table, which can characterize the prediction results of the model. In our study, the prediction performance is measured by the metrics as following: recall = $TP / (TP + FN)$, precision = $TP / (TP + FP)$, accuracy = $[TP + TN] / [TP + FN + TN + FN]$, false alarm ratio (FAR) = $FP / (TP + FP)$, Heidke skill score (HSS; Heidke 1926) [45], and the true skill statistics (TSS; Hanssen & Kuipers 1965) [46]. HSS and TSS are defined as follows, respectively,

$$HSS = \frac{2[(TP \times TN) - (FN \times FP)]}{(TP + FN)(FN + TN) + (TP + FN)(FP + TN)}.$$  

(4)

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN}.$$  

(5)

The recall, precision, and accuracy have a score range of 0 to 1, with 1 representing perfect prediction. The range of FAR is from $-\infty$ to 1, with 1 representing perfect prediction and less than 0 representing no skill. TSS ranges from $-1$ (no correct predictions) to 1 (all predictions correct), with a value of 0 meaning that the model performs no better than one using random chance. Among the above six metrics, only the TSS score is not sensitive to the class-imbalance ratio (Bloomfield et al. 2012; Bobra & Couvidat 2015). Therefore, we also follow the recommendation of Bloomfield et al. (2012) adopting the TSS score as primary metric and others as secondary ones.

The proposed CNN model is trained by the training data set over 80 epochs with a mini batch size of 16. The validating of the model is implemented during every epoch to track the learning performance. We use the testing data set as the validating data set to validate the model. By monitoring the value of the validating loss, we select and save the best trained model among many models produced at every epoch, which is similar to Park et al. (2018) and Huang et al. (2018). As 10 CV

| Dataset | Model ($\geq$M class) | Model ($\geq$C class) |
|---------|-----------------------|-----------------------|
|         | PP        | PN       | PP        | PN       |
| No. 1   | AP 2201   | 445      | 3893      | 509      |
|         | AN 245    | 2842     | 233       | 1098     |
| No. 2   | AP 600    | 336      | 2121      | 404      |
|         | AN 205    | 3117     | 355       | 1378     |
| No. 3   | AP 1762   | 500      | 3224      | 417      |
|         | AN 84     | 2677     | 361       | 1021     |
| No. 4   | AP 2097   | 147      | 3591      | 288      |
|         | AN 280    | 2920     | 352       | 1213     |
| No. 5   | AP 1289   | 529      | 3132      | 372      |
|         | AN 210    | 3140     | 441       | 1223     |
| No. 6   | AP 2524   | 476      | 4308      | 333      |
|         | AN 169    | 2972     | 378       | 1122     |
| No. 7   | AP 2760   | 408      | 3814      | 379      |
|         | AN 298    | 2306     | 216       | 1363     |
| No. 8   | AP 921    | 141      | 1717      | 270      |
|         | AN 59     | 2325     | 219       | 1240     |
| No. 9   | AP 2205   | 273      | 3585      | 366      |
|         | AN 244    | 2642     | 369       | 1044     |
| No. 10  | AP 1316   | 328      | 2466      | 444      |
|         | AN 203    | 2580     | 246       | 1271     |

Figure 5. Confusion matrices for the testing of the model for both $\geq$C class and $\geq$M class on each of 10 data sets. PP represents predicted positive class, PN represents predicted negative class, AP represents actual positive class, and AN represents actual negative class.
data sets are used to train and validate our model (see Section 2), we obtain the results of training and validating loss per epoch. Figure 4 gives the learning curves showing the result of training and validating loss per epoch for our model. The 10 different colored curves represent the changes of training and validating loss with epochs for the model trained and validated by 10 separate data sets. As shown in Figures 4(a)–(d), the result of training and validating loss tends to converge after 20 epochs for the model, respectively. However, the validating loss curves seem to fluctuate after 20 epochs, most probably because the number of ARs is small and unbalanced in the training and validating data set. Although there are fluctuations in the validating loss curves in Figure 4, the testing or prediction results of our trained model perform better in the testing phase below. In general, Figure 4 shows that our model does not suffer from severe overfitting for both $\geq$C-class and $\geq$M-class flare prediction.

The confusion matrices for the testing of the model for both $\geq$C class and $\geq$M class on each of 10 data sets are shown in Figure 5. The other verification metrics can be calculated from these matrices. The data of the prediction result is more centralized on the main diagonal of the confusion matrice, indicating that the metrics such as TSS in Table 2 are better. The prediction results of our CNN model are shown and compared with previous studies in Table 2. We select previous models in Table 2 to be compared with our model, because they use a chronological data set to ensure that not only the samples in the testing data set and those in the training data set are disjoint, but also the ARs in the testing data set and those in the training data set do not intersect. In addition, the previous models of Huang et al. (2018) and Park et al. (2018) are selected because they both apply the CNN model without features manually selected and extracted from the observational data. The model by Nishizuka et al. (2018) is based on deep residual neural network, and that of Bloomfield et al. (2012) is based on statistics. It can be seen from Table 2 that, our model is trained and tested on 10 CV data sets to provide the mean and standard deviation of each metric, while other models are trained and tested on one single data set to provide the best value. Park et al. (2018) and Huang et al. (2018) also adopt tenfold CV to provide the values of the mean and standard deviation. However, they are not included in Table 2 in our study, because their data sets used for CV are randomly shuffled and segregated, which cannot assure that the ARs in the testing data set do not appear in the training data set.

Our proposed CNN model has better mean values of the TSS and HSS for $\geq$M class than those for $\geq$C class. For $\geq$M-class flare forecasting, our model achieves TSS = $0.749 \pm 0.079$, which is much better than that of Huang et al. (2018) and Bloomfield et al. (2012), and comparable to that of Nishizuka et al. (2018). For $\geq$C-class flare forecasting, the model achieves TSS = $0.679 \pm 0.045$, which is much better than that of Huang et al. (2018) and Bloomfield et al. (2012), and comparable to that of Park et al. (2018) and Nishizuka et al. (2018). It is worth noting that Nishizuka et al. (2018) achieved the highest TSS score in previous studies in Table 2, and they only used the best TSS score where the model was trained and evaluated by one single data set. However, we provide the mean of TSS with standard deviation, and the best TSS score of the model is $0.847$ for $\geq$M class and $0.773$ for $\geq$C class, which is calculated from the confusion matrice of No. 4 and No. 7, respectively, in Figure 5, and better than that of Nishizuka et al. (2018). The HSS value of the model is $0.759 \pm 0.071$ for $\geq$M class, which is much better than that of previous models, while the HSS value of the model is $0.671 \pm 0.040$ for $\geq$C class, which is much better than that of Huang et al. (2018), Nishizuka et al. (2018), and Bloomfield et al. (2012), and comparable to that of Park et al. (2018). The scores of FAR and precision of the model are much better than those of previous models for both $\geq$C class and $\geq$M class. In particular, the FAR and precision of the model are about eight times better than those of previous models for $\geq$M class. In addition, we achieve quite good scores in terms of recall and accuracy at the same time. In summary, experiment results reveal that the overall performance of our model is greatly improved compared with previous studies for both $\geq$C-class and $\geq$M-class flare prediction.

Figure 6 shows the skill scores of six different metrics for the evaluating of the trained model on each of the 10 CV data sets. The skill scores for the model used to predict $\geq$C-class and $\geq$M-class flare are given in Figures 6(a) and (b), respectively. As shown in Figure 6(a), both the TSS curve and the curves of
other five metrics appear to fluctuate within a very small range. However, as shown in Figure 6(b), the curves of six metrics seem to fluctuate moderately. This indicates that the model is more robust and stable in predicting $\geq$C-class flare than in predicting $\geq$M-class flare, which happens most probably because the number of ARs and samples between positive class and negative class for $\geq$C class is more balanced than that for $\geq$M class. Overall, along with the small standard deviation of each metric (see Table 2), Figure 6 demonstrates that the proposed CNN model trained by 10 CV data sets is relatively robust and stable in making flare prediction for both $\geq$C class and $\geq$M class.

5. Conclusions and Discussions

In this paper, we propose a new CNN model to make binary class prediction for both $\geq$C-class and $\geq$M-class flares within 24 hr. We collect 870 ARs and 136134 LOS magnetogram samples provided by the SHARP in the period between 2010 May and 2018 September. Based on these AR samples, we propose a method of shuffle and split CV to construct 10 independent data sets. In each of 10 data sets, the data set is split into the training and testing data set according to NOAA AR number, which is more suited to flare prediction. Note that it is not easy to carry out this segregation, because it requires that not only the samples in the testing data set do not appear in the training data set, but also the ARs in the testing data set do not overlap with those in the training data set. However, previous studies have focused on chronologically splitting the data into training and testing data set, or randomly choosing shuffled data set with certain similarities between training and testing data set. As a matter of fact, the difference in constructing data sets may influence the performance comparisons, but it is not easy to construct common data sets. To alleviate the class imbalance, we also adopt the method of undersampling and image augmentation, which is usually used for data preprocessing in the field of machine learning. Ultimately, the resulting data sets are used to train, validate, and evaluate our model, and our model is not subject to excessive overfitting.

The main results are summarized as follows. (1) We propose a novel CNN model, which can make binary class prediction for both $\geq$C class and $\geq$M class, and is different from the architecture of previous CNN models. (2) We propose a method of shuffle and split CV based on AR segregation to construct data sets, which is the first attempt to verify the validity and stability of the model in flare prediction and different from previous studies. (3) Our model achieves a relatively high score of TSS = 0.749 ± 0.079 (0.679 ± 0.045, respectively) for $\geq$M-class ($\geq$C-class, respectively) prediction, which is much better than or comparable to previous studies. The best TSS score of our model among the 10 CV results is 0.847 (0.773, respectively) for $\geq$M-class ($\geq$C-class, respectively) prediction, which is better than the highest TSS score in previous studies in Table 2 (Nishizuka et al. 2018). In addition, the model obtains fairly good scores in the other five metrics for both $\geq$C-class and $\geq$M-class flare prediction. (4) The standard deviations of all metrics are significantly small, indicating our model is considerably robust and stable in predicting flares for both $\geq$C class and $\geq$M class. The experimental results show that our proposed model has achieved unprecedented predictive performance of flare forecasting. Therefore, we deduce that there could be some undiscovered features automatically extracted by the convolution filters in our model, which could help reveal the physical mechanism of flare eruption. However, the learned convolution filters automatically identify features in image data that act more like black boxes, and these features contain high-level information that is difficult to understand. Our current study is not yet able to determine the specific features that would result in such positive results. In the near future, we will attempt to open the black boxes to study these high-level features related to solar flares, to further improve the prediction performance of our model.

Based on all our experiments, we conclude that our proposed CNN model is a very effective method for both $\geq$C-class and $\geq$M-class flare forecasting with fairly excellent prediction performance. At present, our model could do binary class prediction for both $\geq$C-class and $\geq$M-class flares. However, the model does not carry out the prediction for $\geq$X-class flares, because the number of X-level ARs and samples is seriously
insufficient, which leads to a serious class-imbalance issue and makes our model not well learned for \( \geq X \)-class. With the accumulation of observation data at high temporal and spatial resolution, we will gather more ARs and magnetogram samples for X level to solve the above issue. In the near future, to further improve the predictive performance of our model, we plan to incorporate different types of images of ARs in our data sets such as vector magnetic field and extreme ultraviolet images. On the other hand, we also attempt to apply our model to solve other predictive problems (e.g., filament eruptions and CMEs) in solar physics.

We acknowledge the referee for the great effort for anonymously reviewing our paper and for giving us very valuable and useful comments. This work is supported by the National Natural Science Foundation of China (grants No. 11703009, No. 11803010), and the Natural Science Foundation of Jiangsu Province, China (grant No. BK20170566). The data used here is courtesy of NASA/SDO and the HMI science team, as well as the GOES team.

ORCID iDs

Xuebao Li \( \text{https://orcid.org/0000-0003-0397-4372} \)
Yanfang Zheng \( \text{https://orcid.org/0000-0003-0229-3989} \)

References

Abadi, M., Agarwal, A., Barham, P., et al. 2016, arXiv:1603.04467
Abraham, S., Aniyan, A. K., Kembhavi, A. K., Philip, N. S., & Vaghmare, K. 2018, \textit{MNRAS}, 477, 894
Ahmed, O. W., Qahwaji, R., Colak, T., et al. 2013, \textit{SoPh}, 283, 157
Barnes, G., Leka, K. D., Schrijver, C. J., et al. 2016, \textit{ApJ}, 829, 89
Bloomfield, D. S., Higgins, P. A., James McAteer, R. T., & Gallagher, P. T. 2012, \textit{ApJL}, 747, L41
Bobra, M. G., & Couvidat, S. 2015, \textit{ApJ}, 798, 135
Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, \textit{SoPh}, 289, 3549
Boureau, Y. L., Ponce, J., & LeCun, Y. 2010, in Proc. 27th Int. Conf. On Machine Learning (ICML-10) (Madison, WI: Omnipress), 111
Colak, T., & Qahwaji, R. 2009, \textit{SpWea}, 7, S06001

Florios, K., Kontogiannis, I., Park, S. H., et al. 2018, \textit{SoPh}, 293, 28
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA: MIT Press), \url{http://www.deeplearningbook.org}
Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, \textit{SpWea}, 13, 626
Handzo, R., Forbes, J. M., & Reinisch. B. 2014, \textit{SpWea}, 12, 205
Hanssen, A. W., & Kuipers, W. J. A. 1965, Meded. Verb, 81, 2
Hayes, L. A., Gallagher, P. T., McCauley, J., et al. 2017, \textit{IGRA}, 122, 9841
Heidke, P. 1926, \textit{Geogr. Ann.}, 8, 301
Hinton, G., Deng, L., Yu, D., et al. 2012, ISPM, 29, 82
Hinton, G. E., & Salakhutdinov, R. R. 2006, \textit{Sci}, 313, 504
Huang, G., Liu, Z., Maaten, L., & Weinberger, K. Q. 2016, arXiv:1608.06993
Huang, X., Wang, H., Xu, L., et al. 2018, \textit{ApJ}, 856, 7
Huang, X., Yu, D. R., Hu, Q. H., Wang, H. N., & Cui, Y. M. 2010, \textit{SoPh}, 263, 175
Huang, X., Zhang, L., Wang, H., & Li, L. 2013, A&A, 549, A127
Ioffie, S., & Szegedy, C. 2015, arXiv:1502.03167
Krizhevsky, A., Sutskever, I., & Hinton, G. 2012, in Proc. of the Neural Information Processing Systems Conf., ed. F. Pereira et al. (La Jolla, CA: Neural Information Processing Systems Foundation), 1097
LeCun, Y., Bengio, Y., & Hinton, G. 2015, \textit{Natur}, 521, 436
LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. 1998, in Neural Networks: Tricks of the Trade, ed. G. Montavon, G. Orr, & K.-R. Müller (Berlin: Springer), 9
Li, R., Cui, Y., He, H., & Wang, H. 2008, \textit{AdSpR}, 42, 1469
Li, R., & Zhu, J. 2013, \textit{RAA}, 13, 1118
Liu, C., Deng, N., Wang, J. T. L., & Wang, H. M. 2017, \textit{ApJ}, 843, 104
Mason, J. P., & Hoeksema, J. T. 2010, \textit{ApJ}, 723, 634
Nair, V., & Hinton, G. E. 2010, in Proc. 27th Int. Conf. on Machine Learning (ICML-10) (Madison, WI: Omnipress), 507
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, \textit{ApJ}, 835, 156
Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2018, \textit{ApJ}, 858, 113
Park, E., Moon, Y. J., Shin, S., et al. 2018, \textit{ApJ}, 869, 91
Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, \textit{SoPh}, 275, 3
Ponomarchuk, S. N., Kurkin, V. I., Polekhi, N. M., et al. 2015, \textit{Proc. SPIE}, 9680, 96805G
Qahwaji, R., & Colak, T. 2007, \textit{SoPh}, 241, 195
Sadovyk, V. M., & Kosovichev, A. G. 2017, \textit{ApJ}, 849, 148
Schou, J., Scherrer, P. H., Bush, R. L., et al. 2012, \textit{SoPh}, 275, 229
Schrijver, C. J., Dobbins, R., Martagh, W., & Petrinec, S. M. 2014, \textit{SpWea}, 12, 487
Simonyan, K., & Zisserman, A. 2015, arXiv:1409.1556
Song, H., Tan, C., Jing, J., et al. 2009, \textit{SoPh}, 254, 101
Szegedy, C., Liu, W., Jia, Y., et al. 2014, arXiv:1409.4842
Yuan, Y., Shih, F. Y., Jing, J., & Wang, H. M. 2010, \textit{RAA}, 10, 785