Draft version October 12, 2021
Typeset using LaTeX default style in AASTeX631

Predicting Solar Flares using CNN and LSTM on Two Solar Cycles of Active Region Data

Zeyu Sun [1], Monica G. Bobra [2], Xiantong Wang [3], Yu Wang [4], Hu Sun [4], Tamas Gombosi [3], Yang Chen [4], and Alfred Hero [1, 4]

[1] Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, USA
[2] W.W. Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA 94305, USA
[3] Department of Climate and Space Sciences and Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[4] Department of Statistics, University of Michigan Ann Arbor, MI 48109, USA

ABSTRACT

One major challenge of solar flare prediction with machine learning methods is the scarcity of large flares. This issue of low positive sample size is even more severe for data observed in the relatively weak Solar Cycle 24, for example, the SHARPs data product. This partly hampers the successful application of deep learning methods, especially those dealing with high-dimensional spatial and/or temporal data. By joining SHARPs with Space-Weather MDI Active Region Patches (SMARPs), a new data product derived from observations in Solar Cycle 23, we are able to obtain a fused dataset with nearly tripled positive samples. We evaluated two deep learning methods, LSTM and CNN, using the selected parameter sequences and image snapshots in the fused dataset. Experiment results show that the two models trained on the fused dataset achieve better or equivalent test set performance than those trained on a single solar cycle. In addition, we demonstrate the improvement of the performance of the stacking ensemble that combines LSTM and CNN. We provided interpretation to CNN using modern visual attribution methods in computer vision. The results show that CNN is able to identify flare-related signatures in magnetograms.

1. INTRODUCTION

Solar flares are abrupt electromagnetic explosions in magnetically active regions on the solar surface. Intense solar flares can be followed by coronal mass ejections and eruptions of solar energetic particles, which may disturb or disable satellites, terrestrial communication systems, and power grids. Predicting such strong flares from solar observations is therefore of particular significance and has been one of the primary tasks in space weather research.

Solar flares, like many other solar eruptive events, are known to be magnetically driven. Over the past decade, a great amount of flare prediction studies have benefited from an active region data product called Space-Weather HMI Active Region Patches (SHARPs, Bobra et al. 2014). The SHARP database is derived from full-disk observations of the Helioseismic and Magnetic Imager (HMI, Schou et al. 2012) aboard the *Solar Dynamics Observatory* (SDO), containing maps and summary parameters of automatically tracked active regions from May 2010 to the present day, covering much of Solar Cycle 24. However, Solar Cycle 24 is the weakest solar cycle in a century and, consequently, the SHARP database only contains a limited number of strong events, which is not favorable for data-driven learning methods.

Recently, a new data product, Space-Weather MDI Active Region Patches (SMARPs, Bobra et al. 2021), was developed as an effort to extend backward the SHARP database to include active regions observations in Solar Cycle 23, a much stronger solar cycle with significantly more flaring events. In fact, Solar Cycle 23 is the longest solar cycle (147 months) in the past 150 years [1]. The SMARP database was derived from the Michelson Doppler Imager (MDI, Scherrer et al. 1995) aboard the *Solar and Heliospheric Observatory* (SoHO), which observed the sun from 1996 to 2010. Compared to its successor HMI, MDI's measurement of the solar surface magnetic field is only restricted to the line-of-sight component, with lower spatial resolution, lower signal-to-noise ratio, and shorter cadence. As such,

Corresponding author: Zeyu Sun
zeyusun@umich.edu

[1] Source: https://ntrs.nasa.gov/api/citations/20130013068/downloads/20130013068.pdf

SMARP is not of as high quality as SHARP. Nonetheless, SMARPs' coverage of a stronger solar cycle and its partial compatibility with SHARPs make it a valuable dataset to use with SHARPs when larger sample size is desired.

Flare prediction is posed as a classification problem, asking for, most commonly, a binary decision whether an active region will flare in a future time window given its observation and/or flaring history. In this paper, we consider a "strong-vs-quiet" flare prediction problem, distinguishing active regions that will produce an M- or X- class flare in the future 24 hours from those that stay flare quiescent.

Many machine learning methods for flare prediction have been proposed. They roughly fall into three categories in terms of how flare pertinent features are extracted from data. The first category uses *explicit* parameterization of observational data that are considered relevant to flare production, e.g., SHARP parameters that characterize the photospheric magnetic field. Much of the effort in data-driven flare forecasting has been made in this category, exploring a wide range of machine learning algorithms including discriminant analysis (Leka & Barnes 2003), regularized linear regression (Jonas et al. 2018), support vector machine (Yuan et al. 2010; Bobra & Couvidat 2015; Nishizuka et al. 2017; Florios et al. 2018), k-nearest neighbors (Nishizuka et al. 2017), extremely random trees (Nishizuka et al. 2017), random forests (Liu et al. 2017; Florios et al. 2018), multi-layer perceptrons (MLP) (Florios et al. 2018), residual networks (Nishizuka et al. 2018, 2020), long short-term memory (LSTM) networks (Chen et al. 2019; Liu et al. 2019), etc. The second category learns features from images using fixed transformations, e.g., random filters (Jonas et al. 2018), Gabor filters (Jonas et al. 2018), wavelet transforms (Hada-Muranushi et al. 2016). The third category, only popularized more recently, *implicitly* learns flare indicative signatures directly from active region magnetic field maps. This category features mainly convolutional neural networks (CNNs) (Huang et al. 2018; Li et al. 2020). Note that the three categories are not mutually exclusive. For example, methods in the second category typically also depend on explicitly constructed features (e.g. Jonas et al. 2018) as the information within transformation coefficients is often limited.

Another taxonomy to categorize machine learning methods for flare prediction is how the method uses the temporal evolution of active regions. Most of the literature uses static data, be it images or parameters. Some studies take advantage of temporal evolution, for example, sunspot classification evolutions (McCloskey et al. 2018), or moments of the time series of magnetogram summary statistics (Ahmadzadeh et al. 2021). Only limited exploration has been made in directly modeling time series and extracting dynamic information, most notably the LSTM (Chen et al. 2019; Liu et al. 2019).

In this study, two representative deep learning methods, LSTM and CNN, are considered. LSTM uses times series of keyword parameters derived from line-of-sight magnetograms, whereas CNN uses static point-in-time magnetograms. We explore the possibilities of combining CNN and LSTM for a better performance by considering a meta-learning scheme called stacking. We also provide a visual explanation of how CNN makes a decision using visual attribution methods. In particular, we examine the features picked up by the CNN using one attribution method, Integrated Gradients.

The contribution of this paper to the solar flares prediction research lies in the following five folds:

1. We demonstrated the utility of SMARP on flare prediction when combined with SHARP.

2. We first compared the flare prediction performance of LSTM and CNN on an equal footing in terms of using the same dataset.

3. We first applied the stacking method that combines LSTM and CNN in flare prediction and demonstrate improvement in certain settings. We called attention to the convexity of stacking criteria in solar flare prediction. We also evaluated and compared some convex objectives with conventional metrical objectives.

4. We provided visual explanations of CNN using visual attribution methods including Deconvolution, Guided Backpropagation, Integrated Gradients, DeepLIFT, and Grad-CAM. We demonstrate the potential of these methods in identifying flare indicative signatures, interpreting CNN's decisions, revealing model limitations, and suggesting methodical modifications.

The rest of the paper is organized as follows. Section 2 introduces in detail the data sources and how they are processed into machine learning ready datasets. Section 3 describes the flare prediction methods, stacking ensemble, and visual attribution methods. Section 4 presents and compares the flare prediction performance on the datasets. Section 5 concludes the paper by presenting the lessons learned from the experiments.

Table 1: Active region summary parameters. Note that `MEANGBL` has unit Gauss/pixel, and that the pixel size, denoted as another keyword `CDELT1`, is different in SHARP and SMARP.

Keyword	Description	Pixels	Formula	Unit
`USFLUXL`	Total line-of-sight unsigned flux	Pixels in the TARP/HARP region	$\sum \lvert B_{\rm LoS}\rvert dA$	Maxwell
`MEANGBL`	Mean gradient of the line-of-sight field	Pixels in the TARP/HARP region	$\sqrt{\left(\frac{\partial B_{\rm LoS}}{\partial x}\right)^2 + \left(\frac{\partial B_{\rm LoS}}{\partial y}\right)^2}$	Gauss/pixel
`R_VALUE`	R, or a measure of the unsigned flux near polarity inversion lines (Schrijver 2007)	Pixels near polarity inversion lines	$\log\left(\sum \lvert B_{\rm LoS}\rvert dA\right)$	Maxwell
`AREA`	De-projected area of patch on sphere in micro-hemisphere	Pixels in the TARP/HARP region	$\sum dA$	mH

2. DATA

2.1. *Data sources*

SHARP contains automatically-detected active regions, referred to as HMI Active Region Patches, or HARPs, from May 2010 to the present day. SMARP contains Track Active Region Patches, or TARPs, from 1996 to 2010. We download SHARP and SMARP records in Cylindrical Equal-Area (CEA) coordinates from Joint Science Operations Center[2]. Only good quality SMARPs and SHARPs within $\pm 70^\circ$ of the central meridian matching at least one NOAA active region are considered. We query SMARP records from 1996 April 23 to 2010 October 28 and SHARP records from 2010 May 1 to 2020 December 1, both at a cadence of 96 minutes. For keyword parameters, we use four common keyword parameters in SMARP and SHARP, i.e., `USFLUXL`, `MEANGBL`, `R_VALUE`, and `AREA`. Definitions and calculations of those keywords are listed in Table 1. For images, we use photospheric line-of-sight magnetic field maps, or magnetograms, from the two data products.

The SHARP data product overlaps with the SMARP data product from May 1 to October 28 in 2010. This overlap period provides an opportunity to calibrate the two data products. In this study, we use the overlap period to derive transformations that map SHARP magnetograms and keyword parameters to "SMARP proxy data". In the following, we describe and analyze the data fusion transformations for magnetograms and keyword parameters.

The difference between SHARPs and SMARPs magnetograms poses a challenge to use them jointly. SHARP contains active region magnetograms at the HMI resolution of about $0.5''$ per pixel, whereas SMARP magnetograms inherit the MDI resolution of about $2''$ per pixel. To compare HMI and MDI magnetograms, Liu et al. (2012) reduced HMI spatial resolution to match MDI's by convolving a two-dimensional Gaussian function with an FWHM of 4.7 HMI pixels and truncated at 15 HMI pixels. Then, the HMI pixels enclosed in each MDI pixel are averaged to generate an MDI proxy pixel. After that, a pixel value transformation $\mathrm{MDI} = -0.18 + 1.40 \times \mathrm{HMI}$ is applied. This conversion was also used by Huang et al. (2018) in their flare prediction work. In this work, we took a simplified approach by subsampling SHARP magnetograms 4 times in both dimensions to match the resolution of SMARP magnetograms. In addition, we do not perform pixel value transformation because we found by histogram equating (Riley et al. 2014) that the data distribution in the overlap period of MDI and HMI are similar, with the correlation coefficient very close to 1 (Figure 1). We use histogram equating because highly precise alignment of CEA-projected active region patches between SHARP and SMARP is not yet available (Bobra et al. 2021). Our multiplicative conversion factor (1.099) agrees well with that in Riley et al. (2014, Table 2) (0.99 ± 0.13). The discrepancy between our result and Liu et al. (2012) (1.099 vs. 1.40) may be because they considered full-disk magnetograms whereas we focus on active regions. In addition, they considered only 12 pairs in June – August 2010, whereas we considered every possible matching in May – October 2010. Furthermore, they performed pixel-to-pixel match of full-disk magnetograms, whereas we use histogram-based methods. Pixel selection rules may also contribute to the difference.

The keyword parameters in the two databases also need to be calibrated. Designed to represent the same physical quantity, keywords with identical names in SHARP and SMARP are calculated from two pipelines with different

[2] See http://jsoc.stanford.edu.

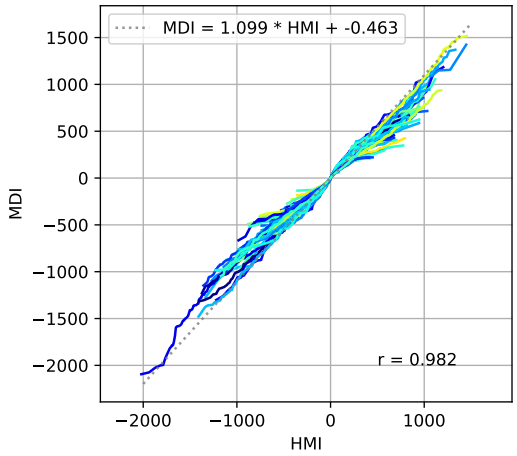


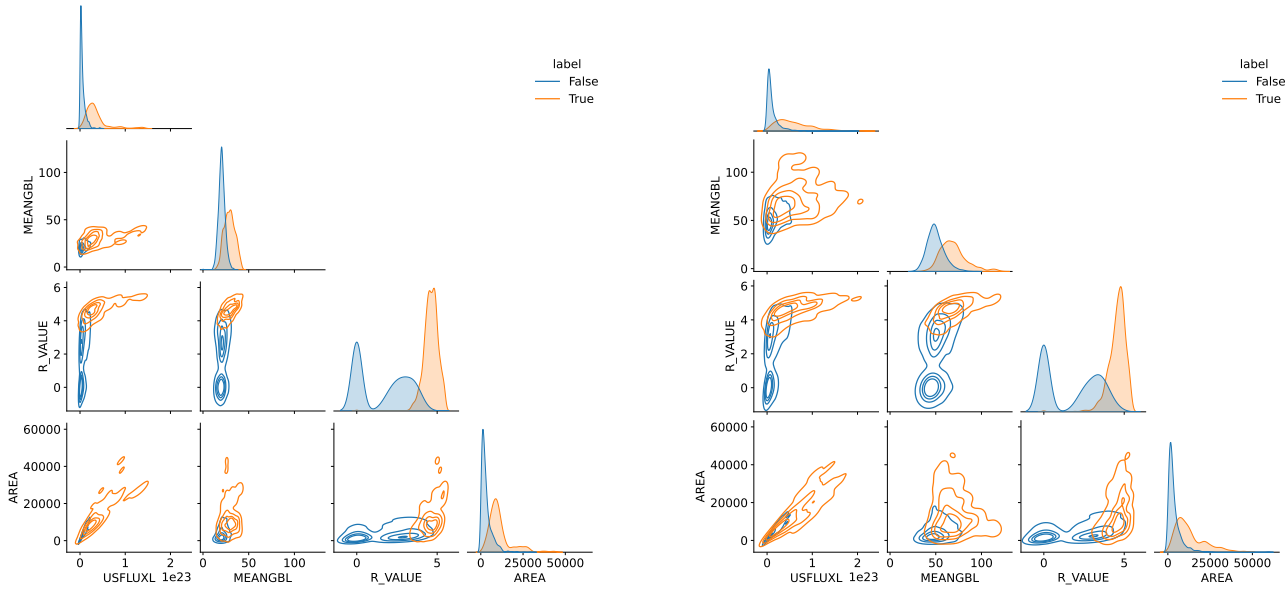
Figure 1: Q-Q (quantile-quantile) plot of 50 matched pairs of HARP and TARP from 2010-05-01 to 2010-10-28. Active regions with pixels outside of ±70° from the central meridian are not used. For each pair, the co-temporal magnetograms are sampled at a rate of every 8 hours. The pixels within the intersection of the bounding boxes of active region pairs are used. Lighter color indicates higher latitude.

source data, and the difference between them cannot be neglected. Bobra et al. (2021) investigated such difference by comparing the marginal and the joint distribution of co-temporal SMARP and SHARP keywords for 51 NOAA active regions in the overlap period of MDI and HMI (Bobra et al. 2021, Figure 3). We extend this investigation by looking at long-term distributions of the keywords in SMARP and SHARP, respectively. In addition, we perform univariate linear regression of SMARP parameters on their counterparts in the SMARP database. The results are shown in Figure 2 and 3. In Figure 3 shows that `USFLUXL` is the most correlated parameter between SHARP and SMARP, with r=0.970, whereas MEANGBL is the least correlated parameter, with $r = 0.796$. We do not see a significant improvement when regressing SMARP parameters on multiple co-temporal SHARP parameters. Therefore, the linear transformation seems to be a reasonable choice to convert SHARP data.

To label data samples, we take advantage of the GOES solar flare events. Based on the peak magnitude of 1–8 Å soft X-ray flux measured by *Geostationary Operational Environmental Satellites* (GOES), solar flare events are classified into five increasingly intense classes: A, B, C, M, and X, often appended with a number that indicates the finer scale. M- and X- classes are referred to as strong flares throughout the paper. Each solar flare event is associated with an NOAA active region, which is used to cross-reference the `NOAA_ARS` keyword in SHARP (or SMARP) databases to associate the flare with a HARP (or TARP). The GOES event records are queried using the Sunpy package (The SunPy Community et al. 2020) from the beginning of 1996 to the end of 2020, covering the period of the SMARP and SHARP observations used in this paper. There are 61 event records with unknown GOES event class, most of them in the year 1996, that are excluded in this study. Of note, although the GOES catalog is widely considered as the "go-to" record database in solar flare forecasting, it is not error-free. There are cases in which flares, even the major ones, are not assigned to any active region (Leka et al. 2019). Moreover, small-sized flares could be buried under the background radiation, which is frequently observed for A-class flares in the majority of a solar cycle and B-class flares after a major flare occurs. As such, works that only consider C-class flares and above are not uncommon (e.g. McCloskey et al. 2018).

2.2. *Sample extraction and labeling*

To build a dataset for the 24-hour "strong-vs-quiet" flare prediction task, record sequences of active regions in SMARP and SHARP need to be organized into observation samples and labeled according to flare activity. Throughout this paper, we define a *sample* as a 24-hour long observation of an active region, in the form of a time sequence of magnetograms or keyword parameters. The 24-hour time window of observation is called the *observation period*, and the following 24 hour time window immediately after the observation is called the *prediction period*. A sample is



(a) SHARP (a) SMARP

Figure 2: Pairplots of keywords `USFLUXL`, `MEANGBL`, `R_VALUE`, and `AREA` in the selected and labeled dataset of SHARP (a) and SMARP (b), respectively. Shown are kernel density estimations of the marginal and the joint distribution of the keywords. The axes of the two plots at the same position in (a) and (b) are scaled equally for comparison.

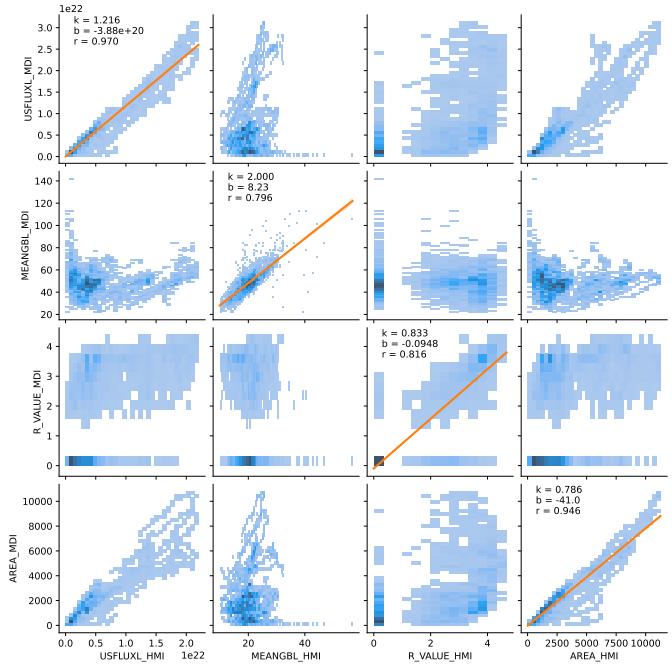


Figure 3: 2D histograms of keywords `USFLUXL`, `MEANGBL`, `R_VALUE`, and `AREA` between SHARP and SMARP. SHARP keywords are suffixed with `_HMI` and SMARP with `_MDI`. The orange lines in the diagonal blocks are the least square fit of SMARP keywords on the corresponding SHARP keywords, with coefficient k, intercept b, and correlation coefficient r displayed in the corner.

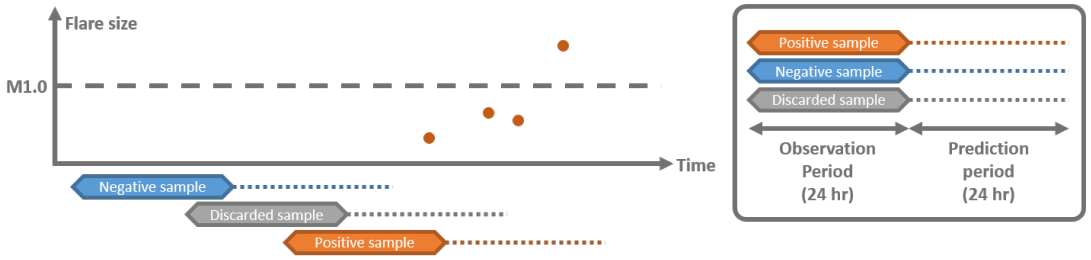


Figure 4: Demonstration of the sample extraction and labeling procedure of an active region. The dark orange dots represent flares that occurred in an active region, with the last flare exceeding the M1.0 threshold. The blue sample is labeled as negative because no flare of any class occurs in the observation and the prediction period. The gray sample is discarded because all flares in the prediction period are weaker than M1.0. The orange sample is labeled as positive because the prediction period contains a flare of size exceeding M1.0. Note that the lag time between samples (96 minutes) is not depicted proportionally.

assigned to the positive class if the active region has at least one flare of size exceeding M1.0 occurring in the prediction period, and to the negative class if the active region has no flare of any class in both the observation period and the prediction period. The steps to extract and label samples are detailed in the following pipeline and illustrated in Figure 4.

1. *Discard records subject to severe projection effects.* For the record sequence of each HARP region or TARP region in the aforementioned time periods with an associated NOAA active region number, we only keep the records with the entire active region bounding box inside $\pm 70^\circ$ of the central meridian.

2. *Extract subsequences from each active region record sequence.* We segment the record sequence of the active region into 24-hour long (or 16 time steps), partially overlapping subsequences that are 96 minutes apart. Hence, the observation period of each sample is 24 hours.

3. *Label subsequences.* A subsequence belongs to the positive class (or event class) if there is an M- or X-class flare within the 24-hour prediction period, i.e., 24 hours after the subsequence ends. A subsequence belongs to the negative class (or quiet class) if there is no flare of any class within the observation period and the prediction period. Any subsequence that cannot be categorized into the above two classes is discarded.

4. *Discard subsequences with too many missing data.* We define a "bad image" as one with Not-a-Number (NaN) pixels or with either dimension (height or width) deviating more than 2 pixels from the median dimension of the subsequence. For each subsequence, if one of the following conditions are satisfied, the subsequence is considered beyond imputation and thus discarded: (1) there are more than 2 "bad images", (2) the last image is a "bad image", (3) there are more than two missing values in any keyword subsequence, or (4) the last record has missing keywords.

The motivation to extract sequences in Step 2 is to provide a common collection of samples to evaluate and compare methods working with time series and those working with static point-in-time observations. Both magnetograms and keyword parameters are considered in this pipeline so that we can also compare methods working with images and those working with parameters. This pipeline enables sample-level inspection. It also eliminates the randomness from sample selection, a long-standing problem for methodical comparison (e.g. Barnes et al. 2016). In our case, a reasonably fair comparison can be made between LSTM that takes the parameter sequences and CNN that takes the last magnetogram of each sequence.

We also note that, in step 3, discarded samples have flaring patterns belonging to one of the following two cases: (1) samples with only weak flares in the prediction period, and (2) samples that flare in the observation period but not in the prediction period. Samples with pattern (1) are discarded because we want better contrast between the two classes. This not only makes the learning easier, but also avoids the concern about the granularity of labels (for instance, an M1.0 class flare and a C9.9 class flare relieve a similar amount of energy but are categorized differently). Samples with pattern (2) are in the decline phase of activity. Predicting those samples equates to answering the question of whether

Table 2: Sample sequences extracted from SMARP and SHARP

	Positive (M1.0+)	Negative (Quiet)	Event Rate
SMARP	4601	130695	0.0340
SHARP	2849	66349	0.0412

a flaring active region will return quiet in the near future. This problem could be intrinsically harder, but also less interesting from an operational forecasting point of view. Therefore, samples with this pattern are also discarded.

After the above sample selection pipeline, the number of positive and negative samples extracted from SMARP and SHARP is shown in Table 2. The count of negative samples is observed to dominate in both SMARP and SHARP. To address the issue of significant class imbalance, we randomly undersample the negative samples, which will be detailed in Section 2.4.

2.3. *Train/validation/test split*

Machine learning algorithms typically require data samples to be partitioned into disjoint subsets, also referred to as splits. A common practice is to divide the dataset into three splits: a training set on which the model is fitted, a validation set on which hyperparameters are selected, and a test set on which the model is evaluated for generalization performance. Each split serves a different goal which could be interfered and compromised by the inter-splits correlation. On the other hand, the success of generalization hinges on the distributional similarity among splits. Therefore, it is important that splits are sufficiently similar in distribution while being statistically independent.

Due to the temporal coherence of an active region in its lifetime, a random split of data samples will have samples coming from one active region categorized into different splits. Such correlation constitutes an undesirable information leakage among splits. For instance, information leaking from the training set into the test set will likely result in an overly optimistic estimate of the generalization performance. Much of the flare prediction literature deals with this issue by taking a chronological split, e.g., a year-based split (e.g. Bobra & Couvidat 2015; Chen et al. 2019). Unfortunately, it is observed that the splits may not share the same distribution due to solar cycle dependency (Wang et al. 2020). Some other works take an active-region-based split, where data samples from the same active region must belong to the same split (e.g. Guerra et al. 2015; Campi et al. 2019; Zheng et al. 2019; Li et al. 2020). Compared to splitting by years, this approach has the advantage that active regions in each split are randomly dispersed in different phases of a solar cycle, removing the bias introduced by artificially specifying splits. This distributional consistency between splits comes at the price of an additional source of information leakage due to sympathetic flaring in co-temporal active regions. Yet, such phenomenon is observed to be weak (?) and is ignored in our analysis.

2.4. *Random undersampling*

Both SMARP and SHARP exhibit class imbalance as shown in Table 2. However, a balanced dataset is typically easier for machine learning models to learn from. Class imbalance can be dealt with at two levels, the data level and the model level. At the data level, one could undersample the majority class and/or oversample the minority class. A significant side effect of resampling strategy is that it changes the class distribution. This has to be considered critically. At the model level, one could adjust the penalty of misclassification of different classes in the loss function. This approach is widely applied in solar flare forecasting (e.g. Bobra & Couvidat 2015; Nishizuka et al. 2018; Liu et al. 2019). A recent work by Ahmadzadeh et al. (2021) provided a fairly thorough review of the class imbalance in solar flare forecasting as well as empirical evaluation of different approaches to tackle this issue.

In our work, we perform random undersampling on the negative samples to arrive at a balanced dataset with equal numbers of positive and negative samples. The random undersamling is applied to all splits separately to ensure each split is balanced. We note that, for an operational forecast that needs to report generalization performance on the new data with unaltered climatological rate, resampling can only be applied to the training set; applying it to the test set leads to systematic bias to the results. However, the distributional difference among splits is undesirable for model training: a model generalizes the best on the same data distribution as what it is trained on. In our case, we choose to value distributional consistency. Thus, the test set performance is not to be interpreted in an operational setting, nor should it be compared to other forecasting methods that sample data differently.

Both train/validation/test split and random undersampling are random. Repeating these two steps with different seeds enables uncertainty quantification to the evaluation results in Section 4. It is worth noting that, to date,

uncertainty quantification in forecasting metrics can only provide guidance (Leka et al. 2019). Commonly used schemes that estimate the variance of skill scores such as random splitting (Bobra & Couvidat 2015) and cross validation (Jonas et al. 2018) are usually biased (Efron & Tibshirani 1997; Bengio & Grandvalet 2004). Even bootstrap estimate of the uncertainty incurs bias due to non-distinct observations in the bootstrap samples (Efron & Tibshirani 1997).

2.5. *Image resizing*

The CNN requires all input images to be of the same size, but the active region cutouts are of different sizes and aspect ratios. Resizing (via interpolation), zero padding, and cropping are among mostly used methods to convert different-sized images into a uniform size. In this work, we decide to resize all active region magnetograms to 128×128 pixels using bilinear interpolation, similar to Huang et al. (2018) and Li et al. (2020).

We note that resizing is more of an empirical decision than a scientifically justified one. Resizing obviously does not preserve the area and aspect ratio of the active region, which may well be correlated or informative of flaring activities. However, when compared to other options like zero padding and cropping, resizing gives better convergence and test set performance and hence is adopted.

2.6. *Standardization*

Magnetogram pixel values and keyword metadata are different physical quantities in different units and ranges. Unlike physical modeling, many machine learning algorithms are invariant to the input scaling; they only care about the relative position of a quantity in the feature distribution. Moreover, drastically different ranges of features may hurt the convergence and stability of many algorithms. Therefore, the data of different scales are typically transformed into the same range via a process called standardization, also known as normalization. In particular, Z-score standardization transforms the input data by removing the mean and then dividing by the standard deviation. In this work, we apply the Z-score standardization to image data using the mean and standard deviation of images in SHARP. This is because we consider the pixel values between SMARP and SHARP are similar. We apply the Z-score standardization to SMARP and SHARP keywords separately. That is, the mean and standard deviation are calculated for SHARP and SMARP separately, and data in one dataset is standardized using the mean and the standard deviation in that dataset. The transformation is "global" (Ahmadzadeh et al. 2021) in that it is calculated regardless of the splits. Empirical evaluation in Ahmadzadeh et al. (2021) showed a global normalization is better than the local normalization, i.e., the mean and standard deviation are calculated only for the training split. We note that, with this normalization, the linear transformation converting SHARP keywords to SMARP proxy data is not needed anymore; any coefficients and bias will have no effect after standardization.

3. METHODOLOGY

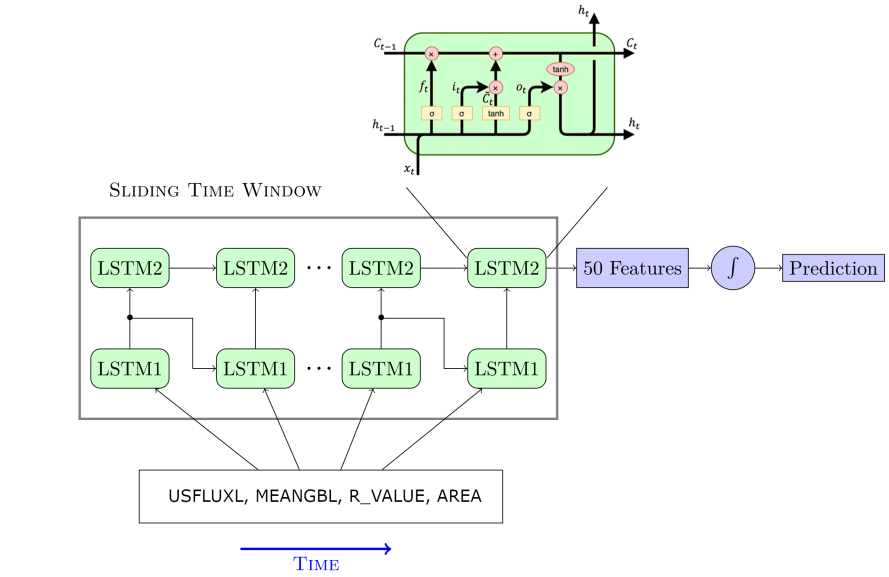
In this section, we first introduce two deep learning models, LSTM and CNN, used for flaring active region prediction. Then we describe the stacking ensemble. After that, we describe forecast verification methods including metrics and graphical tools. Following that, we introduce the paired t-test used in making statistically significant claims. Lastly, we introduce the visual attribution methods used to interpret CNNs.

3.1. *Deep learning models*

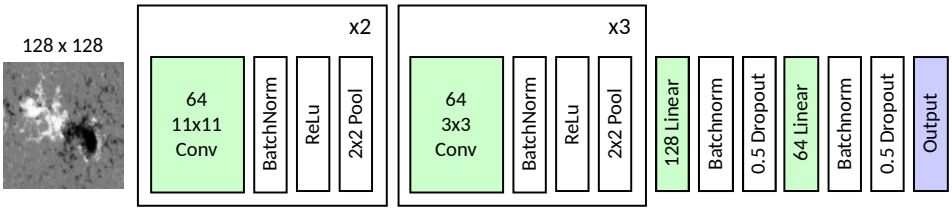
We use two deep neural network models, CNN and LSTM, to predict strong flares from active region observation. CNN takes an active region magnetogram as input, whereas LSTM takes a time sequence of keyword parameters. Both networks output the probability that the sample belongs to the positive class, i.e., the probability that the active region will produce a strong flare the next day, rather than continue to be flare-quiescent.

Long short-term memory (LSTM) network (Hochreiter & Schmidhuber 1997) is a type of recurrent neural network that learns from sequential data such as text and speech. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. In solar flare prediction, LSTM has been applied to SHARP parameter series (Chen et al. 2019; Liu et al. 2019). The architecture of the LSTM used in this paper is adapted from (Chen et al. 2019), shown in Figure 5(a). Two LSTM layers, each with 64 hidden states, are stacked. The last output of the second LSTM layer, as a 64-dimensional vector, is sent to a linear layer with 2 outputs. The softmax is applied to this 2-dimensional output to get the predicted probabilities of the positive and the negative class.

Convolutional neural network (CNN) is a neural network architecture that learns from images. CNN has been applied to solar flare forecasting by Huang et al. (2018) and Li et al. (2020). We take the architecture used in Li et al.



(a) LSTM architecture



(b) CNN architecture

Figure 5: Neural network architectures. (a) shows the LSTM architecture. (b) shows the CNN architecture.

(2020), illustrated in Figure 5(b), which is in turn inspired by the VGG network (Simonyan & Zisserman 2014) and the Alexnet network (Krizhevsky et al. 2012). The first two convolutional layers have kernels of size 11×11, designed to learn low-level and concrete features. The three following convolutional layers have kernels of size 3×3, designed to learn more high-level, abstract concepts. Batch normalization is used after all convolutional and linear layers to speed convergence. ReLU nonlinearity is applied to only convolutional layers. The batch normalization outputs of the two linear layers are randomly dropped out with probability 0.5 in training to reduce overfitting. The 2-dimensional output is passed to softmax to generate a probability assignment between the positive and the negative class. More details of this architecture can be found in Li et al. (2020).

The training procedures of the LSTM and the CNN are similar. For both models, the Adam optimizer (Kingma & Ba 2014) is used to minimize the cross-entropy loss with learning rate 10^{-3} and batch size 64. Both models are evaluated on the validation set after each epoch of training. To prevent overfitting, the training is early-stopped if no improvement on the validation True Skill Statistic (or TSS, explained later in Section 3.3) is observed for a certain number of epochs called *patience*. The LSTM is trained for at most 20 epochs with a patience of 5 epochs, whereas the CNN is trained at most 20 epochs with a patience of 10 epochs. After training, the LSTM or the CNN with the best validation TSS among the checkpoints of all epochs is selected and evaluated on the test set to estimate its generalization performance.

3.2. *Stacking ensemble*

In a sense, physical parameters represent the known aspects of flare-related signatures in active regions, whereas magnetograms contain much richer information, some of which has not yet been characterized or even understood by humans. One might expect that LSTM and CNN fitted on these two types of data could provide somewhat different perspectives to the same physical process. It is then natural to ask whether it is possible to combine the

two perspectives for better performance. This idea, in fact, belongs to ensemble learning, a learning paradigm that capitalizes on different models to achieve a better performance than any of the models alone. Famous examples of ensemble learning include bagging (Breiman 1996), boosting (Freund & Schapire 1997; Friedman 2001), and stacking (Wolpert 1992). Bagging (bootstrap aggregating) produces an aggregated output (typically average for continuous response and voting for discrete response) from many classifiers trained on bootstrapped samples to reduce variance. Boosting harvests the boosted performance from a multitude of weak learners. Stacking usually involves only a few strong learners. In our case, it is most appropriate to take the stacking approach to ensemble LSTM and CNN.

First introduced by Wolpert (1992), stacking has been studied extensively in a wealth of literature. The earliest effort that applied stacking in solar flare prediction can be traced back at least to a seminal machine learning work by Džeroski & Ženko (2004), in which a general stacking method using multi-response model trees was proposed. The authors showed their stacking method performed better than the best classifier based on experiments on the *UCI Repository of machine learning databases* (Dua & Graff 2017), including a dataset with 1389 flare instances, each characterized by 10 categorical attributes. Guerra et al. (2015) first attempted to use stacking over operational forecasts in flare prediction. They combined the full-disk probabilistic forecasts from four operational forecasting methods using 13 active regions selected from 2012 to 2014. Combination weights are chosen to maximize HSS under the constraint that the weights sum to 1. Guerra et al. (2020) continued in this direction with a larger ensemble of forecasting methods and they also considered an unconstrained linear combination with a climatological frequency term. They found most ensembles perform better than a bagging model that essentially averages the members' predictions. However, they overlooked the nonconvexity of the objective in training the meta-learner. We will discuss this issue and provide solutions later in this section.

In its most basic form, stacking uses a linear combination of the outputs of a collection of models as the output of the ensemble. The collection of models are called *base learners*, and the linear combination of base learners is called the *meta-learner*. Stacking is typically performed in two stages. In the first stage, the base learners are fitted on the training set. In the second stage, the predicted probabilities by all base learners on the validation set, as well as their labels, are collected into the so-called "level-one" data, on which the meta-learner is fitted to figure out the optimal combination weights of the base learners. Cross-validation is frequently used in place of a simple train-validation split so that the validation sets in different folds can be combined into "level-one" dataset of the same size as the training set. Either way, it is important that the "level-one" data are out-of-sample data for base learners to prevent overfitting.

In our case with two base learners, we formulate the stacking ensemble as follows. Let p_i, q_i denote the predicted probabilities of instance x_i by the independently trained LSTM and CNN, respectively. The stacking ensemble constructs a probability prediction as a weighted average $r_i = \alpha p_i + (1-\alpha) q_i$, $0 \leq \alpha \leq 1$. The mete-learner is parameterized by a single scalar α. To prevent overfitting, stacking requires the meta-learner parameters to be fitted on a dataset different from the datasets on which the base learners were fitted. Therefore, we use the validation set to find the best α.

There are multiple ways to formulate the optimization objective to estimate α. One natural way is to directly optimize the metric of interest. However, the loss function constructed by metrics may not be convex or even differentiable. For instance, categorical metrics such as ACC, TSS, and HSS are closely related to 0-1 loss which is neither convex nor differentiable. Intuitively speaking, smoothness of the loss function makes it possible to deduce the loss function's behavior at the neighborhood given its behavior at one point, whereas convexity ensures the uniqueness (and sometimes the existence with stronger conditions) of the minimizer; both are desired properties of optimization problems, making them easier to solve in theory and practice (Nocedal & Wright 2006). In Guerra et al. (2020), nonconvex objectives are optimized using sequential quadratic programming. However, due to the aforementioned issues, the algorithm is not guaranteed to converge to the global minimum of the objective, which may contribute to the instability of optimized weights observed in Guerra et al. (2020) for certain metrics. To resolve this issue, the authors repeatedly ran the algorithm with random initialization and take the mean as the final weights. In our case with only two base learners, the feasible region is constrained to a one-dimensional space. A grid search can be applied to locate the global solution. In general, however, nonconvex objectives are difficult to deal with, which motivates the use of convex objectives.

Convex loss functions are surrogate objectives in cases where the verification metric is not the loss function itself. Nonetheless, a loss function can have its own motivation; the optimizer is optimal in that sense. One example is maximum likelihood estimation (MLE). Within the meta-learning framework we formulated above, MLE minimizes

		Predicted		
		Negative	Positive	Total
True	Negative	TN	FP	N
	Positive	FN	TP	P
	Total	N′	P′	N + P

Table 3: A contingency table consisting of TP (true positive), FP (false positive), FN (true negative), and TN (true negative).

the negative log-likelihood loss function

$$L(\alpha) = -\log \prod_{i=1}^{n} r_i^{y_i} (1 - r_i)^{1-y_i} \tag{1}$$

$$= \sum_{i=1}^{n} \underbrace{(-y_i \log r_i - (1 - y_i) \log(1 - r_i))}_{L_i} . \tag{2}$$

The MLE objective can also be interpreted as the binary cross-entropy loss, a divergence measure between the distributions of ground truth labels and predicted probabilities. This loss function can be decomposed into the summation of instance-wise loss L_i, with the gradient and the Hessian

$$L_i'(\alpha) = \left(-\frac{y_i}{r_i} + \frac{1 - y_i}{1 - r_i} \right) (p_i - q_i) , \tag{3}$$

$$L_i''(\alpha) = \left(\frac{y_i}{r_i^2} + \frac{1 - y_i}{(1 - r_i)^2} \right) (p_i - q_i)^2 \geq 0. \tag{4}$$

Since the Hessian is nonnegative, minimizing L on $\alpha \in [0, 1]$ is a convex problem and the grid search will recover the unique optimizer. When the number of dimensions scales up, as is the case with multiple base learners, the grid search is no longer feasible. However, thanks to the convexity and differentiability of the loss function, iterative procedures can be performed to efficiently recover the minimizer with guaranteed algorithmic convergence. Examples of such algorithms include projected gradient descent and Newton's method.

It is worth noting that stacking is made possible in this work thanks to the sample selection scheme. Magnetograms are associated with summary statistic sequences, providing two different modes of the same instance. Each instance can then have two predicted probabilities provided by the CNN and the LSTM respectively, which is the prerequisite for applying the stacking method.

3.3. *Evaluation tools*

Both CNN and LSTM produce probabilistic predictions. With proper discriminating thresholds, those predictions can be made binary decisions, which fall into a contingency table (or confusion matrix) shown in Table 3. The contingency table contains the most complete information for categorical predictions. However, it is often the case that a single numerical metric is needed to summarize the table. For instance, such a metric may be desired when deciding which model is to be deployed in operation. Accuracy and the skill scores adopted in space weather forecasting are examples of such contingency table based metrics.

We start our discussion on metrics with accuracy (ACC), also known as rate correct, the simplest metric that is widely used in all sorts of domains. In terms of the contingency table, accuracy is defined as

$$A = \frac{\mathrm{TN} + \mathrm{TP}}{\mathrm{N} + \mathrm{P}}. \tag{5}$$

For a highly imbalanced classification problem like solar flare prediction, accuracy is generally not considered a useful metric, since a no-skill classifier that assigns the majority label to all samples will be correct most of the time. Therefore, a plethora of skill scores are devised to overcome this issue.

A skill score provides a normalized measure of the improvement against a specific reference method. In its most general form, a skill score can be expressed as

$$\text{Skill} = \frac{A_{\text{forecast}} - A_{\text{reference}}}{A_{\text{perfect}} - A_{\text{reference}}}\,. \tag{6}$$

A higher skill score indicates better performance, with the maximum value 1 corresponding to the perfect performance, 0 corresponding to no improvement over the reference, and negative values corresponding to performance worse than the reference. Below, we introduced some of the mostly used skills scores in flare forecasting. For a more complete discussion, we refer readers to Woodcock (1976) and Wilks (2011).

The Heidke Skill Score (HSS), also known as Cohen's kappa coefficient due to Cohen (1960), uses a random forecast independent from the flare occurrences as a reference. The expected number of correct forecasts made by the random predictor, denoted by E, can be calculated using the law of total expectation as

$$\mathrm{E} = \frac{\mathrm{P}}{\mathrm{N}+\mathrm{P}} \times \mathrm{P}' + \frac{\mathrm{N}}{\mathrm{N}+\mathrm{P}} \times \mathrm{N}'. \tag{7}$$

The accuracy of the random predictor can then be expressed as

$$A_{\text{reference}} = \frac{\mathrm{E}}{\mathrm{N}+\mathrm{P}}. \tag{8}$$

Defined using this reference accuracy, HSS has the form

$$\mathrm{HSS} = \frac{\mathrm{TP}+\mathrm{TN}-\mathrm{E}}{\mathrm{N}+\mathrm{P}-\mathrm{E}} \tag{9}$$

$$= \frac{2[(\mathrm{TP}\times\mathrm{TN}) - (\mathrm{FN}\times\mathrm{FP})]}{(\mathrm{TP}+\mathrm{FN})(\mathrm{FN}+\mathrm{TN}) + (\mathrm{TP}+\mathrm{FP})(\mathrm{FP}+\mathrm{TN})}. \tag{10}$$

HSS quantifies the forecast improvements over a random prediction. Since the random reference forecast is dependent on the event rate (climatology) $\mathrm{P}/(\mathrm{N}+\mathrm{P})$, HSS has to be used with discretion in comparing methods when the event rate varies.

The True Skill Score (TSS), also known as Hanssen & Kuiper's Skill Score (H&KSS), Peirce Skill Score. It has the form

$$\mathrm{TSS} = \underbrace{\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}}_{\substack{\text{probability}\\ \text{of detection}}} - \underbrace{\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}}_{\substack{\text{false alarm}\\ \text{rate}}}. \tag{11}$$

TSS falls into the general skill score definition with a reference accuracy (Barnes et al. 2016)

$$A_{\text{reference}} = \frac{\mathrm{FN}(\mathrm{TN}-\mathrm{FP})^2 + \mathrm{FP}(\mathrm{TP}+\mathrm{FN})^2}{(\mathrm{N}+\mathrm{P})[\mathrm{FN}(\mathrm{TN}-\mathrm{FP}) + \mathrm{FP}(\mathrm{TP}+\mathrm{FN})]}\,, \tag{12}$$

constructed such that both the random and unskilled predictors score 0. A nice property of TSS is its invariance to the class imbalance ratio, and hence is suggested by Bloomfield et al. (2012) to be the standard measure for comparing flare forecasts.

We note that, on a balanced dataset for which the event rate is 0.5, it can be shown that $\mathrm{TSS} = \mathrm{HSS} = 1-2(1-\mathrm{ACC})$. The trend and the paired t-test results for TSS apply to ACC and HSS due to perfect correlation. Therefore, we mainly focus on the discussion on TSS, list ACC as a complement metric, and omit HSS as it is equal to TSS in our setting.

For probabilistic forecasts, the aforementioned metrics (ACC, HSS, and TSS) depend upon the threshold applied to the predicted probability. A common practice is to apply a threshold of 0.5, which is considered to be "random" by many researchers. In contrast, the following two metrics, BSS and AUC, are irrelevant to the threshold, and they need information (i.e., predicted probabilities) beyond the mere contingency table.

The Brier Skill Score (BSS) is a skill score evaluating the quality of a probability forecast. It is of a nature different from those of HSS and TSS, in that it directly uses probabilistic predictions without thresholding them. The BSS also

admits the general skill score formulation, with the accuracy replaced by the Brier Score (BS), defined as the mean squared error between the probability predictions f_i's and binary outcomes o_i's:

$$\mathrm{BS} = \frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2 \,. \tag{13}$$

With a reference forecast that consistently predicts the average event frequency $\bar{o}$ (also known as climatology), the BSS is given by

$$\mathrm{BSS} = \frac{\mathrm{BS}_{\mathrm{forecast}} - \mathrm{BS}_{\mathrm{reference}}}{0 - \mathrm{BS}_{\mathrm{reference}}} = 1 - \frac{\mathrm{BS}_{\mathrm{forecast}}}{\mathrm{BS}_{\mathrm{reference}}} \,. \tag{14}$$

It is sometimes of interest to decompose BS into three components of reliability, resolution, and uncertainty (Murphy 1973; McCloskey et al. 2018). BSS is frequently accompanied by the reliability diagram for more complete information, which will be discussed later.

The last metric we introduce is Area Under Curve (AUC), defined as the area under the receiver operating characteristic (ROC) curve. The ROC curve depicts how the probability of detection changes with the false alarm rate by varying the classification threshold. A higher AUC generally implies a higher probability of detection for the same false alarm rate. Although rarely mentioned, AUC also falls into the general formulation of a skill score in a trivial way, with the reference forecast being one that has its prediction separated but in the wrong direction, that is, all negative samples have predicted probability higher than any of the positive samples. This reference forecast gives a zero AUC. Unlike TSS and HSS, AUC is irrelevant to the threshold selected to convert probabilistic forecasts into binary decisions. It is a "fair" metric in that sense. One downside of AUC is that it dismisses some metrics regarded as informative by the community (Leka et al. 2019). Another problem is related to the nature of AUC as being the integrated probability of detection against a uniform measure on the false alarm rate. The reason why this is problematic is that models are rarely operated outside a narrow range of low false alarm rates. Indeed, we observe in experiments that there are a number of cases where AUC follows a different trend, sometimes opposite, to other dichotomous metrics. Due to this reason, we do not use AUC to select models in validation. It is only reported for completeness.

The above numeric values provide one way to directly compare flare prediction models. In addition to metrics, flare forecasts usually also present some graphical tools to provide detailed information for diagnostics and comparison. Common graphical tools used in flare prediction include receiver operating characteristic (ROC) curves, reliability diagrams (RD), and skill score profiles (SSP). All three of them are only meant for forecasts that predict probabilities or continuous scores (e.g., logits) that can be converted to probabilities.

An ROC curve visualizes the trade-off between probability of detection (POD) and false alarm rate (FAR) by altering the dichotomous decision threshold of the predicted probability. A higher ROC curve for up to a particular FAR indicates a more powerful detector within a certain size, or in other words, one that makes fewer Type II errors (misses) under the constraint of Type I error (false alarm) rate not exceeding a certain level. The area under the ROC curve (AUROC, or simply AUC), is a statistic frequently used as a surrogate to measure such quality of a detector. The ROC curve reflects how resolved are the predicted distributions of the positive and the negative class (Leka et al. 2019). It is also worth noting that, if we vary the dichotomous threshold, the highest TSS is achieved by the point on the ROC that is farthest to the diagonal. The TSS can be visually identified as the vertical distance of this point to the diagonal line.

The reliability diagram, also known as the calibration curve, measures how a probabilistic forecast agrees with the observation. The predicted probabilities are binned into groups and the observed event rate within each group is plotted. If the predicted probability agrees well with the observed rate, the points will be close to the diagonal of the plot (the line of perfect reliability). Such a forecast is known as reliable. Any forecast that produces predictions independent of flare activity has all its points close to the horizontal line at climatology. Such a forecast is referred to as one with no resolution. BSS provides a metric considering both reliability and resolution. Figure 6 shows an example of the plane on which the reliability diagram is drawn. The climatology rate is set to be 0.1. The overall BSS can be seen as a histogram weighted average of the contributions of the points on the reliability diagram. The contours are equal contribution lines. The points in the shaded area contribute positively to BSS. The dashed line with slope 1/2 is called the "no skill" line, the points on which have zero contribution to the overall BSS.

A skill score profile plot shows how skill scores change as a function of the probability threshold. A high and flat profile is usually desired, as the method achieves high skill scores and the performance is robust to the changes of the threshold.

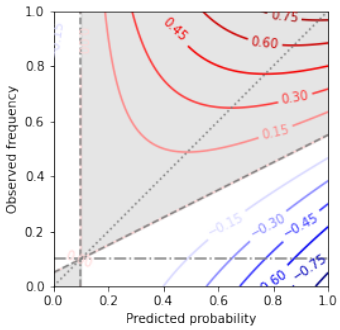


Figure 6: An illustration of the relation between the reliability diagram and BSS.

3.4. *Paired t-tests*

To make sure that our claims based on comparisons are made with statistical significance, we perform hypothesis testing. In particular, we perform a one-sided paired t-test to test if there is an improvement induced by a treatment. Specifically, two competing hypotheses are formulated: the alternative hypothesis H_1 claiming that the measurement with the treatment is higher than that without the treatment, against the null hypothesis H_0 that states otherwise. Measurement pairs on n subjects are collected and formed into two vectors, $\boldsymbol{x}$ and $\boldsymbol{y}$, with x_i and y_i denoting the measurements on the i-th subject with and without treatment, respectively. The t-statistic is calculated as follows:

$$\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x} \qquad \text{(difference)}$$

$$\bar{d} = \frac{\sum_{i=1}^{n} d_i}{n} \qquad \text{(sample mean)}$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{d})^2}{n-1}} \qquad \text{(sample standard deviation)}$$

$$t = \frac{\bar{d}}{s/\sqrt{n}} \qquad (t\text{-statistic})$$

Under H_0, it can be shown that t follows a Student-t distribution with degrees of freedom $n-1$. Consequently, if the resulting t is associated with a right-tail p-value less than a threshold, called significance level and denoted as α, we can say that sufficient evidence has been observed to reject H_0, in the sense that the probability of falsely claiming significant improvement when there is none will be no larger than α. Usually, α is set as a small probability such as 0.05.

In our case, the paired t-test is made possible by enforcing the same test set between the experiments to be compared. For example, in Section 4.1, to test if the treatment of adding SMARP data in the training set of SHARP will improve the predictor's performance on the test set, we treat different test sets resulting from random sampling as subjects. For a given metric, pairs of measurements could be taken on the test set for models trained with and without SMARP data in the training set. The result of the paired t-test will tell us with statistical confidence if adding SMARP in the training set will be of any help in flare prediction.

3.5. *Interpretation of CNN*

Deep learning methods are the essential state-of-the-art in numerous tasks across various domains such as computer vision, natural language processing, speech processing, robotics, and games (see, e.g. He et al. 2016; Silver et al. 2016; Devlin et al. 2018). As of today, deep learning methodology remains to be a black box that lacks a general theory, raising concerns in transparency, accountability, and reliability. However, it is of particular significance to be able to provide interpretation when deep learning methodology is applied to make a scientific discovery. Over the years, many interpretation tools of neural networks have been proposed, revealing aspects of their underlying decision process.

One way to interpret a black-box model, often referred to as "attribution", is to see how different parts of the input contribute to the model's output. An attribution method generates a vector of the same size as the input, with each element indicating how much the corresponding element in the input contributes to the model decision for that input. In the context of CNN, the attribution vector is a heatmap of the same size as the input image.

A multitude of attribution methods have been proposed for CNN in the task of image classification. One type of approach is perturbation-based methods, among which occlusion (Zeiler & Fergus 2014) is the most well-known method. Occlusion masks the input image with a gray patch at different locations and sees how much the prediction score of the ground truth class drops. The prediction score drop varies with location, forming a heatmap, with large values indicating the positions of the features important to the CNN's correct prediction. One drawback of the occlusion method is that it is computationally expensive. Another drawback is that the attribution depends on the size and shape of the patch, which need to be tuned for sensible results. Therefore, this type of approach is not used in our work.

Another type of approach is gradient-based methods, the basic idea being that the gradient of the predicted score of a certain class with respect to the input reveals the contribution of each dimension of the input. Saliency map (Simonyan et al. 2013), one of the earliest gradient-based methods, is simply the absolute value of the gradients. The intuition is that the magnitude of the derivative indicate which pixels need to be changed the least to affect the class score the most (Simonyan et al. 2013). Deconvolution Network (Zeiler & Fergus 2014) and Guided Backpropagation (Springenberg et al. 2015) modified the backpropagation rule of ReLU nonlinearity. Integrated Gradients (Sundararajan et al. 2017) integrated the gradients along the path from a reference image to the target image. Formally, the integrated gradient along the i-th dimension for an input x and a baseline x' is

$$L_i^c(x;x') = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F_c(x' + \alpha \times (x - x'))}{\partial x_i} \, \mathrm{d}\alpha, \tag{15}$$

where $F_c(x)$ is the model output for class c with input x. One desirable property of Integrated Gradients, known as completeness, is that the pixels in the attribution map add up to the difference of prediction scores of the target and the reference image, i.e., $F(x) - F(x')$. DeepLIFT (Shrikumar et al. 2017) and its gradient-based interpretation (Ancona et al. 2018) can be seen as the gradient with modified partial derivatives of non-linear activations with respect to their inputs. Grad-CAM (Gradient-weighted Class Activation Mapping) (Selvaraju et al. 2017) accredits decision-relevant signatures by generating a saliency map, highlighting pixels in the input image that increase the confidence of the network's decision for a particular class. More formally, the Grad-CAM heatmap L^c for class c with respect to a particular convolutional layer is given by the positive part of the weighted sums of the layer's activation maps A_k, i.e.,

$$L^c = \mathrm{ReLU}\left(\sum_k \alpha_k^c A^k\right), \tag{16}$$

with weights α_k^c given by the spatial average of partial derivatives of the class-specific score y^c with respect to the class activation map as

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}, \tag{17}$$

where Z is a normalization constant. Intuitively, a class activation map is weighted more if the pixels therein make the CNN more confident to decide the input belongs to class c.

In solar flare prediction, Yi et al. (2021) applied Grad-CAM to full-disk MDI and HMI magnetograms on CNNs and found the polarity inversion line is highlighted as an important feature by CNN. In this paper, we observe the same trend for the CNN trained on SHARP and SMARP. Taking a step further, we evaluated other attribution methods, providing a more complete view on significant features identified by modern attribution methods.

4. RESULTS

4.1. *Does data from another solar cycle help?*

One major goal of this paper is to examine the utility of using SMARP and SHARP together. We set an experimental group and a control group and contrast their 24-hour "strong-vs-quiet" flare prediction performance. The control group consists of models that train, validate, and test exclusively on SHARP data. We refer to this type of dataset as `SHARP_ONLY`. Compared to the control group, models in the experimental group have the training set enriched by SMARP data, while the validation and the test set are kept the same. We call this type of dataset `FUSED_SHARP`. The only difference between `SHARP_ONLY` and `FUSED_SHARP` is that models using `FUSED_SHARP` have access to data from a previous solar cycle in the training phase. Symmetrically, we design `SMARP_ONLY` and `FUSED_SMARP` to examine the utility that SHARP brought to SMARP. Specifically, the four types of datasets are generated as follows:

Table 4: Sample sizes of the processed datasets

	Train		Validation		Test	
	Positive	Negative	Positive	Negative	Positive	Negative
SHARP_ONLY	1774	1774	665	665	410	410
FUSED_SHARP	5377	5377	1663	1663	410	410
SMARP_ONLY	2849	2849	860	860	892	892
FUSED_SMARP	5084	5084	1474	1474	892	892

Table 5: Test set performance of LSTM and CNN on 24-hour "strong-vs-quiet" flare prediction. The two datasets within each comparison group share common test sets. The 1-σ error is calculated from 10 random experiments. Bold fonts indicate the experiments in which the mean of the metric on the fused dataset is higher than that on the single dataset.

		Group 1		Group 2	
	Dataset	FUSED_SHARP	SHARP_ONLY	FUSED_SMARP	SMARP_ONLY
	Model				
ACC	CNN	0.906+/-0.036	0.922+/-0.017	**0.901+/-0.028**	0.877+/-0.031
	LSTM	**0.950+/-0.012**	0.942+/-0.016	**0.905+/-0.025**	0.900+/-0.024
AUC	CNN	0.980+/-0.009	0.981+/-0.006	**0.963+/-0.017**	0.950+/-0.020
	LSTM	**0.990+/-0.004**	0.986+/-0.004	**0.966+/-0.015**	0.963+/-0.015
TSS	CNN	0.812+/-0.071	0.843+/-0.034	**0.802+/-0.056**	0.754+/-0.061
	LSTM	**0.900+/-0.023**	0.884+/-0.032	**0.810+/-0.050**	0.800+/-0.049
BSS	CNN	0.649+/-0.152	0.714+/-0.064	**0.628+/-0.114**	0.520+/-0.121
	LSTM	**0.799+/-0.036**	0.775+/-0.047	**0.626+/-0.107**	0.586+/-0.108

1. Dataset SHARP_ONLY: 20% of all the HARPs are randomly selected to form a test set. 20% of the remaining HARPs are randomly selected to form a validation set. The rest of the HARPs belong to the training set. In each split, negative samples are randomly selected to match the number of positive samples.

2. Dataset FUSED_SHARP: The test set and the validation set are the same with SHARP_ONLY. The remaining HARPs and all TARPs are combined into the training set. In each split, negative samples are randomly selected to match the number of positive samples.

3. Dataset SMARP_ONLY: 20% of all the TARPs are randomly selected to form a test set. 20% of the remaining TARPs are randomly selected to form a validation set. The rest of the TARPs belong to the training set. In each split, negative samples are randomly selected to match the number of positive samples.

4. Dataset FUSED_SMARP: The test set and the validation set are the same with SMARP_ONLY. The remaining TARPs and all HARPs are combined into the training set. In each split, negative samples are randomly selected to match the number of positive samples.

The tally of samples produced by a particular random splitting and undersampling is shown in Table 4. On each of the four types of datasets, LSTM and CNN are fitted on the training set, validated on the validation set, and evaluated on the test set. We reiterate that LSTM uses 24-hour-long time series of parameters before the prediction period begins, whereas CNN uses the static point-in-time magnetogram right before the prediction period begins.

Table 5 shows the results of the "strong-vs-quiet" active region prediction using LSTM and CNN. For LSTM, a consistent improvement on the fused datasets (FUSED_SHARP and FUSED_SMARP) is observed in terms of the mean of all metrics. This aligns with the fact that more data are typically desired to improve the generalization performance

Table 6: Paired t-tests for significant improvement of test set performance on the fused datasets as measured by different metrics. The alternative hypothesis H_1 claims that metric S on the fused dataset (FUSED_SHARP or FUSED_SMARP) is greater than the respective single dataset (SHARP_ONLY or SMARP_ONLY), which is tested against the null hypothesis H_0 claiming otherwise. The bold font p-values are less than 0.05 and considered to be significant.

	H_1	$S_{\text{FUSED\_SHARP}} > S_{\text{SHARP\_ONLY}}$		$S_{\text{FUSED\_SMARP}} > S_{\text{SMARP\_ONLY}}$	
		p-value	t	p-value	t
Metric S	Model				
ACC	CNN	0.885787	-1.292359	**0.001862**	3.881137
	LSTM	**0.016544**	2.514074	**0.026797**	2.219666
AUC	CNN	0.589845	-0.233881	**0.001399**	4.070352
	LSTM	**0.000459**	4.842485	**0.033930**	2.074572
TSS	CNN	0.885787	-1.292357	**0.001862**	3.881135
	LSTM	**0.016544**	2.514079	**0.026796**	2.219673
BSS	CNN	0.889419	-1.314583	**0.000482**	4.806837
	LSTM	0.054812	1.775082	**0.000099**	6.014784

of deep learning models because they are overparameterized and can easily overfit on small datasets. For CNN, an improvement is observed on FUSED_SMARP over SMARP_ONLY, but not on FUSED_SHARP over SHARP_ONLY. This indicates that the lower image quality in SMARP has a negative effect on CNN's performance.

The statistical significance of the improvement on the fused datasets is tested using a one-sided paired t-test with significance level 95%. Table 6 shows the t-statistics and the associated p-values of the paired t-tests. The bold font p-values are less than 0.05 and considered to be significant. For LSTM, the fused datasets are better than the single datasets in a statistically significant way in almost all settings. The only exception is BSS on FUSED_SHARP, whose p-value is only slightly larger than 0.05. For CNN, across all metrics, statistically significant improvement is observed for FUSED_SMARP over SMARP_ONLY, but not for FUSED_SHARP over SHARP_ONLY. This indicates that adding SHARP magnetograms into SMARP during training helps CNN to better predict flares, but not the other way around. One potential reason is SMARP magnetograms have a lower signal-to-noise ratio than SHARP magnetograms, which may have negatively affected CNN. The LSTM, on the other hand, uses the keyword metadata, which could suppress the effect of noise during summarizing magnetograms, providing information in a sufficiently good quality that does not offset the improvement induced by the increased training sample size.

Aside from the numerical metrics, we provide graphical evaluation results for Group 1 (FUSED_SHARP and SHARP_ONLY) in Figure 7, and Group 2 (FUSED_SMARP and SMARP_ONLY) in 8. A trend of over-forecasting for high probabilities and under-forecasting for low probabilities is observed in some cases but such effect is minor considering the size of the error bars. In reliability diagrams, all models have points closer to the diagonal, indicating high reliability. In ROC plots, it is observed that LSTM achieves higher AUC on the fused datasets (FUSED_SHARP and FUSED_SMARP) than on the single datasets (SHARP_ONLY and SMARP_ONLY). For CNN, similar improvement is also observed in the comparison of FUSED_SMARP and FUSED_SHARP, whereas the ROCs are almost indistinguishable for FUSED_SHARP and SHARP_ONLY. In skill score profiles, the TSS for LSTM trained on fused datasets are at the same level as that trained on single datasets. For CNN, on the other hand, FUSED_SHARP display an disadvantage against SHARP_ONLY, whereas FUSED_SMARP displays an advantage over SMARP_ONLY. This verifies the observations made from metrics. In all cases, the skill score profiles are high and relatively flat, indicate the robustness of the performance to the change of thresholds within a wide range of the varying threshold.

4.2. *Does LSTM perform better than CNN?*

This section provides forecast verification to the LSTM and CNN. We use the same evaluation results for 10 experiments in each setting mentioned in Section 4.1, but present them in a way that makes it easier to compare LSTM and CNN. We note the differences between our verification set-up and that in an operational setting:

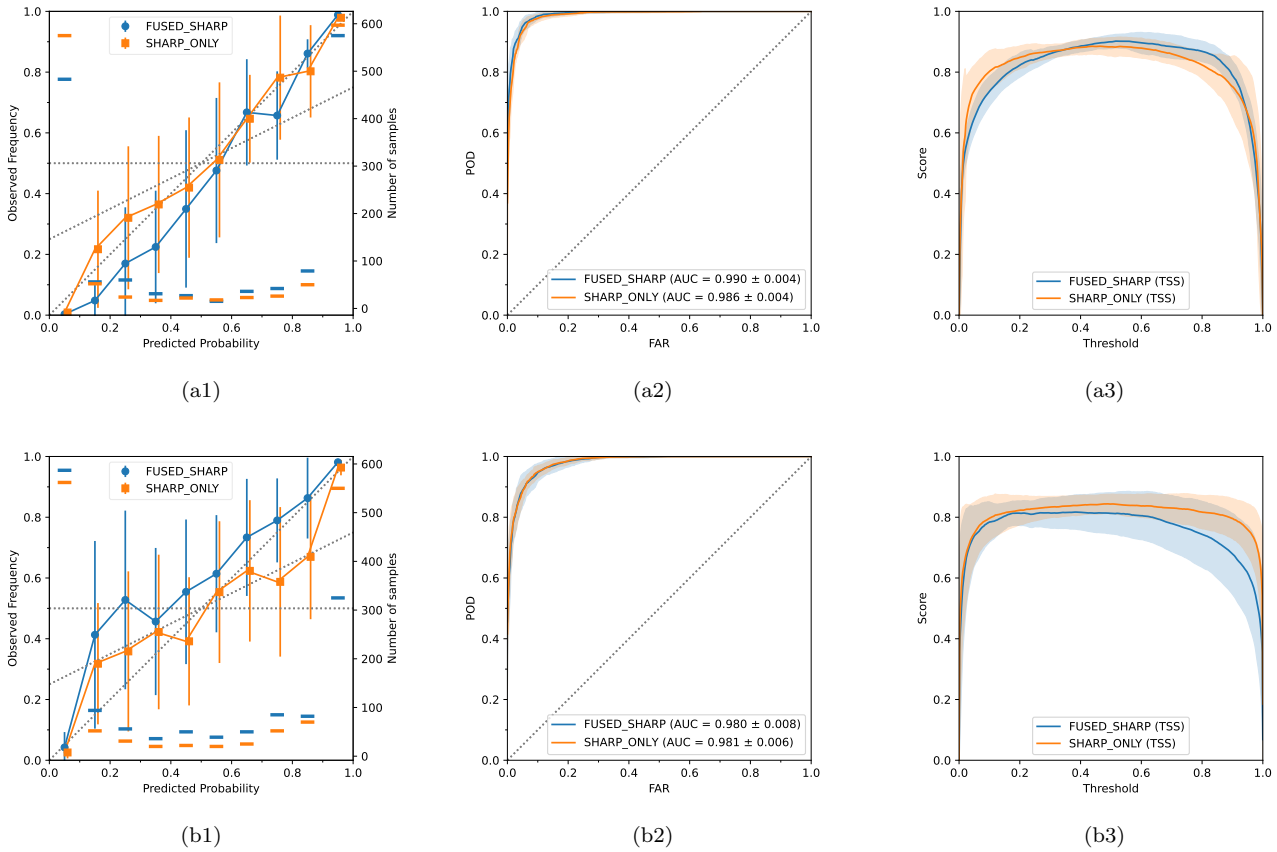


Figure 7: Verification plots on SHARP test data to compare models trained on `FUSED_SHARP` and `SHARP_ONLY`. Shown in (a1) – (a3) are reliability diagram, ROC, and SSP for LSTM. Shown in (b1) – (b3) are the same plots but for CNN. In each panel, the blue/orange curve is the test performance for the model trained on `FUSED_SHARP`/`SHARP_ONLY`. In each graph, solid curves and error bars (or shaded area) indicate respectively the means and the standard deviations calculated from 10 random experiments. In each reliability plot, the short horizontal bars indicate the number of samples in each probability bin, and the two curves are separated horizontally to prevent error bars from overlapping.

1. In terms of data, the test set of our sort has lots of samples removed based on their active regions, observational data, and flare activities. About 1/5 of tracked active region time series in the evaluation period (May 2010 – December 2020) are selected. Within each active region series, only samples with good quality observation and certain flaring patterns are selected (detailed in Section 2.2). Negative samples (flare-quiet active regions) are significantly downsampled to match the number of positive samples (strong-flare-imminent active regions). In contrast, operational forecasts do not discard any sample unless absolutely necessary.

2. In terms of outcomes, the forecast of our sort is independent for individual active regions, with the prediction result available every 96 minutes (i.e., MDI cadence) for valid active regions. In contrast, the end goal of an operational forecast is a full-disk forecast. For operational forecasts built upon active region based forecasts, the predictions for all active regions on the solar disk are aggregated to compute the full-disk prediction. In addition, operational forecasts are typically issued at a lower frequency (e.g., every 6 hours), but in a consistent manner.

The verification results in this section should be interpreted with the above differences in mind.

It can be seen from Table 5 that the LSTM generally scores higher than CNN in terms of mean performance. We performed paired t-test to validate this observation. The results in Table 7 confirm that LSTM scores significantly higher ($p < 0.01$) than CNN across all metrics on all datasets except for `FUSED_SMARP`. On `FUSED_SMARP`, although we cannot claim statistical significance, LSTM's performance is slightly better or at the same level with CNN as is observed from Table 5.

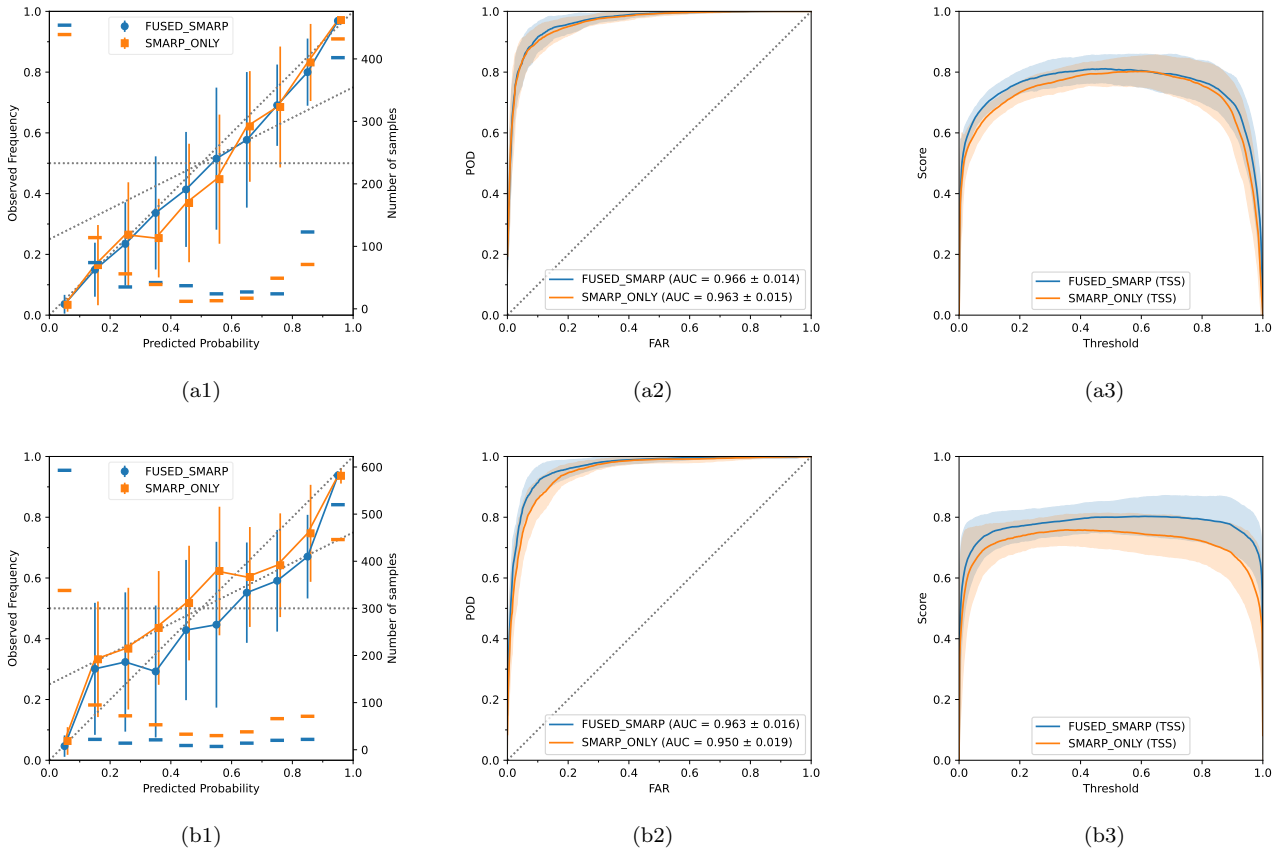


Figure 8: Same as Figure 7 but for SMARP test data to compare models trained on `FUSED_SMARP` and `SMARP_ONLY`.

Table 7: Paired t-tests for significant improvement of LSTM over CNN in terms of different metrics S on the test set of the four datasets. The alternative hypothesis H_1 claims $S_{\text{LSTM}} > S_{\text{CNN}}$. The bold font p-values are less than 0.01 and considered to be significant.

Dataset	FUSED_SHARP		SHARP_ONLY		FUSED_SMARP		SMARP_ONLY	
Metric S	p-value	t	p-value	t	p-value	t	p-value	t
ACC	**0.001442**	4.050296	**0.007142**	3.028403	0.234866	0.754672	**0.001079**	4.245351
AUC	**0.003757**	3.429557	**0.002527**	3.682754	0.227978	0.779031	**0.000743**	4.501005
TSS	**0.001442**	4.050297	**0.007142**	3.028405	0.234865	0.754673	**0.001079**	4.245350
BSS	**0.005296**	3.213872	**0.002645**	3.653351	0.531965	-0.082481	**0.005781**	3.159315

We only present the graphical verification results for both models trained and tested `FUSED_SHARP`, given that SHARP is widely used and validated by a wealth of studies. For the results on other datasets, the visualization can be obtained by simply rearranging the same results shown in Figure 7 and 8.

The reliability diagram in Figure 9 shows that the probabilistic prediction given by LSTM is closer to the diagonal than CNN, and hence more reliable. The CNN exhibit a trend of under-forecasting especially when the predicted probability is less than 0.5. The histogram of predicted probability shows that probabilistic forecast by LSTM is "more confident", or has higher resolution, than LSTM, with most of the predicted probabilities close to 0 or 1.

The ROC in Figure 9 shows a clear advantage of LSTM over CNN, in the sense that it achieves a higher probability of detection with the same false alarm rate. This trend is also manifested in terms of AUC.

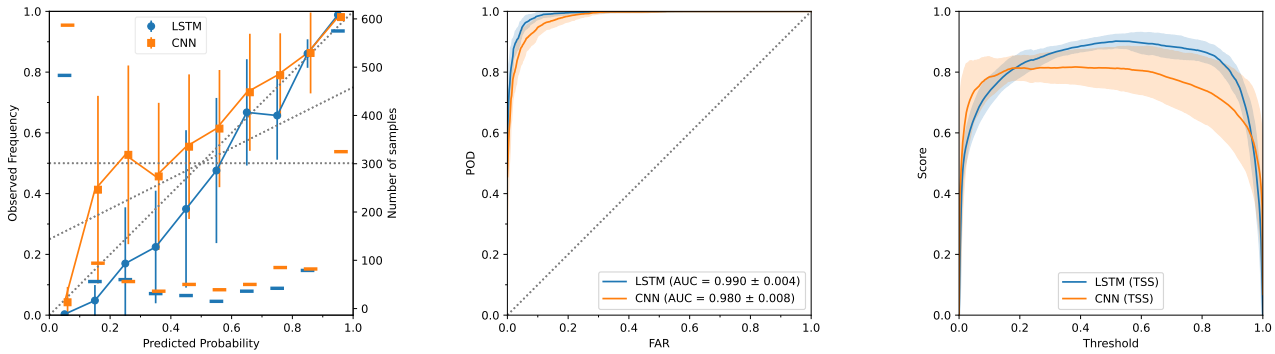


Figure 9: Verification plots of LSTM and CNN on `FUSED_SHARP`. Shown are the reliability diagram, ROC, and SSP, from left to right. This figure essentially extracts the blue curves (representing `FUSED_SHARP`) in both rows of Figure 7 and overlaps them together.

The SSP in Figure 9 shows LSTM achieves higher TSS on average for all thresholds within 0.2–0.9. It is also observed that the TSS for LSTM is maximized by a threshold very close to the climatological rate on the test set (which is 0.5 in our case), a necessary condition for a reliable predictor (Kubo 2019).

At the end of this section, we introduce a new interactive visual verification tool that we found useful in diagnosing the performance of a probabilistic forecasting method. The reliability diagram provides a concise summary of a probabilistic forecast. However, when it comes to diagnosing the method, it is often desired to pinpoint specific samples that contribute to a pattern (e.g. over- and under-forecast) observed in the reliability diagram. To this end, we propose to use a new interactive graphical tool, which we call the sorted probability plot (Figure 10). Samples in the verification dataset are first grouped by labels and then sorted by the predicted probabilities by a reference predictor. If only one predictor is available, the reference is that predictor. If multiple predictions by a group of predictors are available, the reference can be any predictor in the group, or the average thereof. The sorted probability plot can be reduced to the reliability diagram by binning the vertical axis and assigning the proportion of the points on the right section in each probability bin as its corresponding observed event rate. Figure 10 shows examples of this type of plot. Since each sample is preserved in the sorted probability plot, we can directly identify, for example, the samples that CNN is unsure about (i.e., samples with predicted probability covering a large range), the positive samples that are detected by CNN but missed by LSTM (i.e., samples on the right section with large CNN probabilities but low LSTM probabilities), etc. Further inspection of those samples will provide insights on the strength or the weakness of the prediction method.

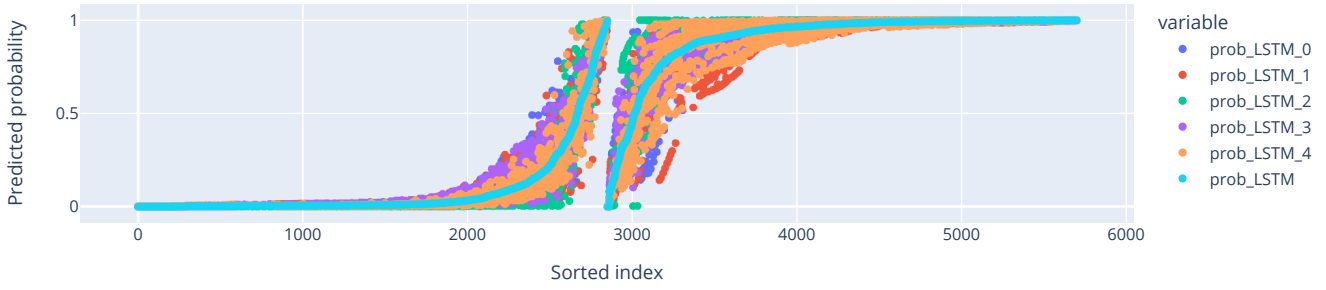
4.3. *Can CNN assist LSTM for a better prediction?*

In this paper, we only consider stacking methods to combine CNN and LSTM hoping for better predictive performance. We evaluate the test set performance of stacking methods using four different criteria:

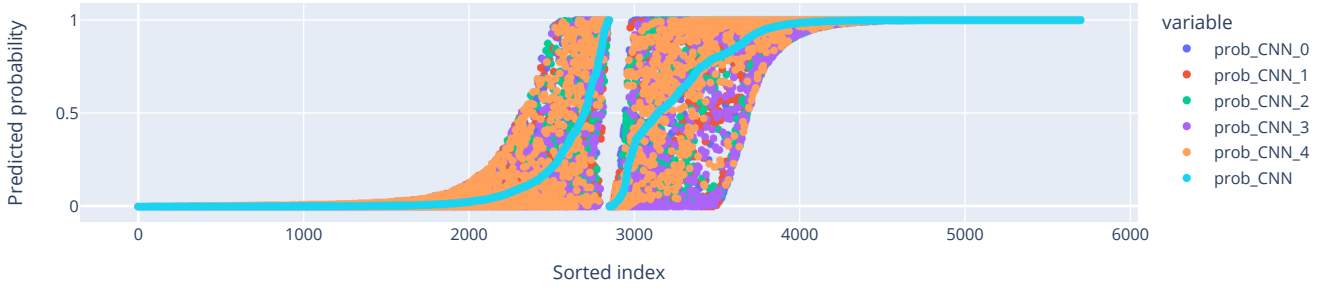
- `CROSS_ENTROPY`: weights are optimized to minimize cross-entropy loss on the validation set.
- `BSS`: weights are optimized to maximize BSS on the validation set.
- `AUC`: weights are optimized to maximize AUC on the validation set.
- `TSS`: weights are optimized to maximize TSS on the validation set.

Among these criteria, cross-entropy and negative BSS are known to be convex; TSS is neither convex nor concave; we observe AUC to be concave but we do not have proof other than empirical evidence. Criteria `HSS` and `ACC` are excluded from the evaluation since their stacking weights are the same as that of `TSS` due to the perfect correlation mentioned in Section 3.3.

To provide baseline performances, we include the evaluation results for the two base learners, `LSTM` and `CNN`. In addition to the above stacking methods, we consider two other meta-learning schemes:



(a) LSTM



(b) CNN

Figure 10: Sorted probability plot of (a) LSTM and (b) CNN. Samples in `SHARP_ONLY` are sorted first by labels and then by the mean probability predicted by five models. An interactive version integrating both plots and sorted by the average probability of LSTM and CNN is available at https://zeyusun.github.io/cv/sorted_probability.html.

- `AVG` outputs the average of predicted probabilities of two base learners.
- `BEST` (Džeroski & Ženko 2004) selects the base learner that performs the best on the validation set and applies it to the test set.

Splitting and undersampling are randomly performed 10 times on each of the four datasets `FUSED_SHARP`, `FUSED_SMARP`, `SHARP_ONLY`, and `SMARP_ONLY`. The test set TSS of the 10 random experiments for each criterion on each dataset are summarized as box plots in Figure 11. The optimal stacking weights for the four stacking ensembles are summarized in Figure 12.

Figure 11 shows that stacking methods perform slightly better than the `BEST` meta-learner, especially on `FUSED_SMARP` and `SMARP_ONLY`. Of note, the wide error bars are partially due to the randomness originated in data sampling. To fairly compare the methods, we perform paired t-tests with significance level 0.05. It turned out stacking is significantly better than `BEST` in the following three settings: `BSS` on `FUSED_SMARP` ($p = 0.048$), `AUC` on `SMARP_ONLY` ($p = 0.025$), and `TSS` on `SMARP_ONLY` ($p = 0.013$).

We also note in Figure 11 that `BEST` unsurprisingly achieves better performance than `AVG` but is slightly worse than the better performing base learner `LSTM`, most noticeably on `FUSED_SHARP`. In fact, `BEST` decided that `CNN` is the better model in 3 out of 10 experiments on `FUSED_SHARP`. This is not unexpected because the "best" model on the validation set is not necessarily the best on the test set.

From Figure 12, we can see that α is greater than 0.5 in most experiments, with the median falling between 0.55 and 0.9 in all settings. This suggests that stacking ensembles generally depend more on LSTM than on CNN. The variance of α is large in some settings, especially for the `AUC` on `FUSED_SMARP`. The variance of convex criteria (`CROSS_ENTROPY` and `BSS`) is not smaller than that of nonconvex criteria (`TSS`), indicating that the local minima of non-convex loss functions is not the major source of variance. We suspect the major source of the variance comes from the data sampling

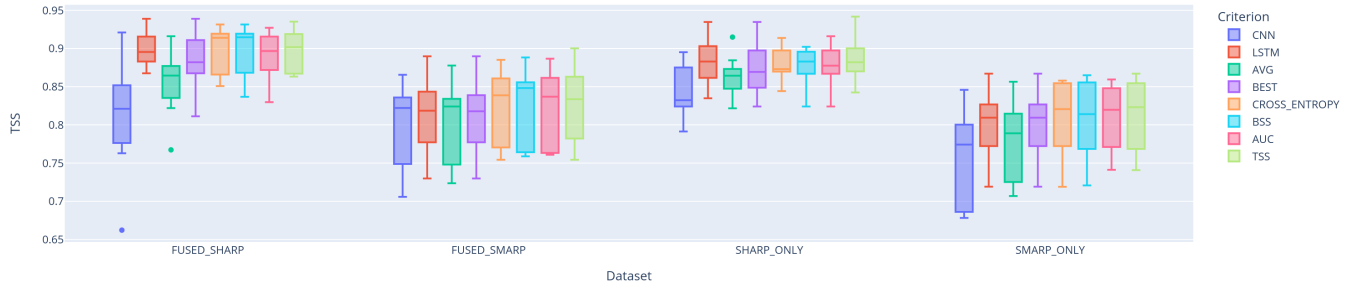


Figure 11: Test set TSS for base learners and meta-learners using different criteria.

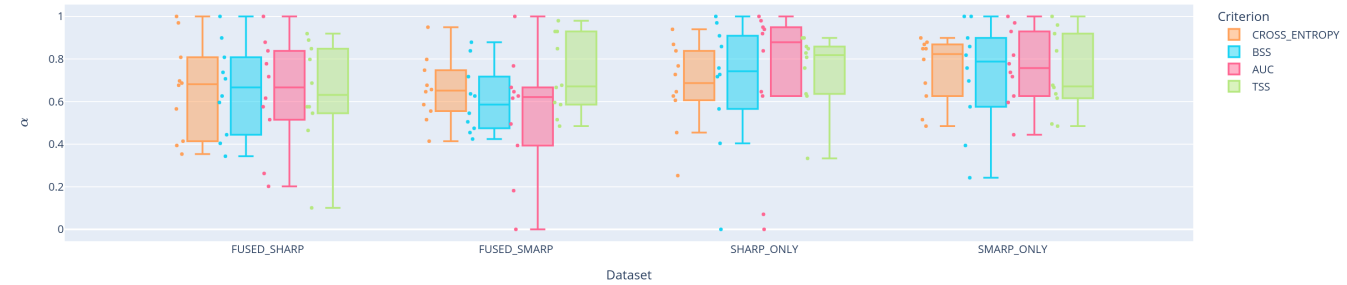
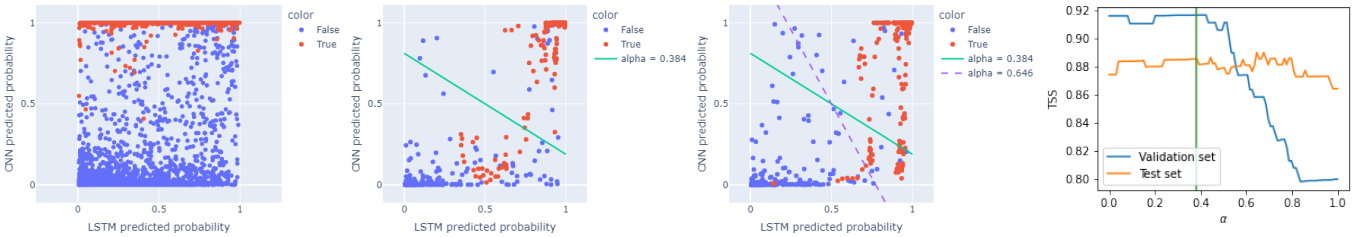


Figure 12: Stacking weight α fitted using different criteria on different datasets. All 10 values of α in an experiment setting are shown as points next to the corresponding box.

bias among experiments, which is, in turn, a collective consequence of the insufficient sample size, heterogeneity across active regions, and possibly a small amount of information leakage because the validation set is used both in the validation of base learners and the training of the meta-learner.

We inspect one experiment of stacking with criterion `TSS` and the results are presented in Figure 13. Figure 13 (a1)–(a3) show the predicted probabilities by LSTM and CNN of each instance in the training, the validation, and the test set. The points are colored by their labels, with red representing the positive class and blue representing the negative class. The green solid line in (a2) and (a3) shows the decision boundary by the meta-learner with α fitted on the validation set to maximize TSS. The points (p, q) on the upper right side of the boundary are classified as positive because they satisfy $r = \alpha p + (1-\alpha)q > 0.5$. In this experiment, the fitted $\alpha = 0.384$, suggesting the stacking ensemble relies more on the CNN than on the LSTM. The violet dashed line in (a3) is the decision boundary with α fitted on the test set, and hence can be seen as the oracle. It can be observed that the distribution of predicted probabilities on the validation set (a2) and the test set (a3) are similar but not exactly the same, which explains the difference between the estimated α and the oracle α. The distribution of predicted probabilities on the training data in (a1), on the other hand, looks completely different, with CNN achieving almost perfect separation. In fact, CNN overfitted on the in-sample data, as indicated by a significantly lower positive recall rate in (a2) and (a3). This validates the decision that meta-learners should not be fitted on the predicted probabilities of the same data used to train the base learners.

Figure 13(b) inspects the optimization process of the same experiment, in which the TSS is calculated on the validation set (a2) and the test set (a3) by scanning a fine grid of $\alpha \in [0, 1]$ with resolution 0.001. The green line is the α that maximizes the validation TSS curve, which equals 0.384. It is indeed observed the TSS is not convex with respect to α. In fact, the test set TSS has a lot of local maxima across a wide range of α. Still, its trend can be roughly estimated by the validation set TSS, and its value at the estimated α is higher than both ends of the curve, indicating an improved performance over any of the two base learners.



(a1) Train set (a2) Validation set (a3) Test set (b) Optimization of TSS

Figure 13: (a1)–(a3): CNN predicted probability (y-axis) vs. LSTM predicted probability (x-axis) for the train, the validation, and the test set. The green solid line in (a2) and (a3) is the decision boundary of the ensemble with meta-learner fitted on the validation set. The violet dashed line in (a3) is the same as the green line but fitted on the test set, and hence can be seen as the oracle. (b): TSS on the validation and the test set as functions of α. The green line shows the α that maximizes the validation TSS. The TSS on the left end with $\alpha = 0$ show the TSS of CNN, and the TSS on the right end with $\alpha = 1$ show the TSS of LSTM.

4.4. *What image characteristics can CNN inform?*

We use visual attribution methods to extract flare-indicative characteristics of magnetograms from trained CNNs. First, we use synthetic images to examine patterns that contribute to a positive decision of CNN. The results of synthetic images help us understand better the attribution maps of real magnetograms. Then, we apply visual attribution methods to image sequences of selected active regions that transition from a flare-quiescent state to a flare-imminent state. Setting the baseline to the first image in the sequence gives a time-varying attribution map that tracks magnetic field variations that contribute to the change in the predicted probability.

4.4.1. *Synthetic image*

We generate bipolar magnetic regions (BMR) used in Yeates (2020). Following that paper's notation, a location in the Heliographical coordinate system is denoted as (s, ϕ), where s denotes sine-latitude and ϕ denotes (Carrington) longitude. A BMR is represented as a scalar function of location, parameterized by amplitude B_0, polarity separation ρ (in radian), tilt angle γ (in radian) with respect to the equator, and size factor a. The untilted BMR centered at origin has the form

$$B(s,\phi) = -B_0 \frac{\phi}{\rho} \exp\left[-\frac{\phi^2 + 2\arcsin^2(s)}{(a\rho)^2}\right]. \tag{18}$$

We follow Yeates (2020) and fix a to be 0.56 to match the axial dipole moment of SHARP.

We sweep a grid of B_0, ρ, and tilt angle γ to generate a BMR dataset. We are interested in those images that are considered to be positive by CNN. Figure 14 shows some examples of them and their attribution results, from which patterns of positive predictions can be summarized. Guided Backpropagation heatmaps have both poles highlighted with the signs matching the polarities. Integrated Gradients produces heatmaps that are more concentrated to polarity centers and attribute more credits to the negative polarities. DeepLIFT produces similar heatmaps to those by Integrated Gradients. Grad-CAM's results are not as interpretable as the above methods. They seem to avoid the polarities and highlight the background and sometimes the polarity inversion lines.

4.4.2. *The emergence of preflare signatures in the active region evolution*

We focus on the attribution results on SHARP as opposed to SMARP because the former has magnetograms of higher resolution and better quality. We choose the CNNs that are trained on `SHARP_ONLY` as opposed to `FUSED_SHARP` because the former is observed to generalize better in Section 2. To get results that reflect the generalization performance, we need to make sure that active regions of interest are out-of-sample, i.e., in the test set. To evaluate any active region of interest in SHARP, we perform 5-fold cross-validation on `SHARP_ONLY`, so that every active region is associated with a CNN that has never seen the active region in training. In addition, we do not enforce the flare-based sample selection rule and random undersampling, so that any active region magnetogram of good quality is included in this study. Of note, attribution methods in this study act in a frame-by-frame manner. Integrated Gradients and DeepLIFT require

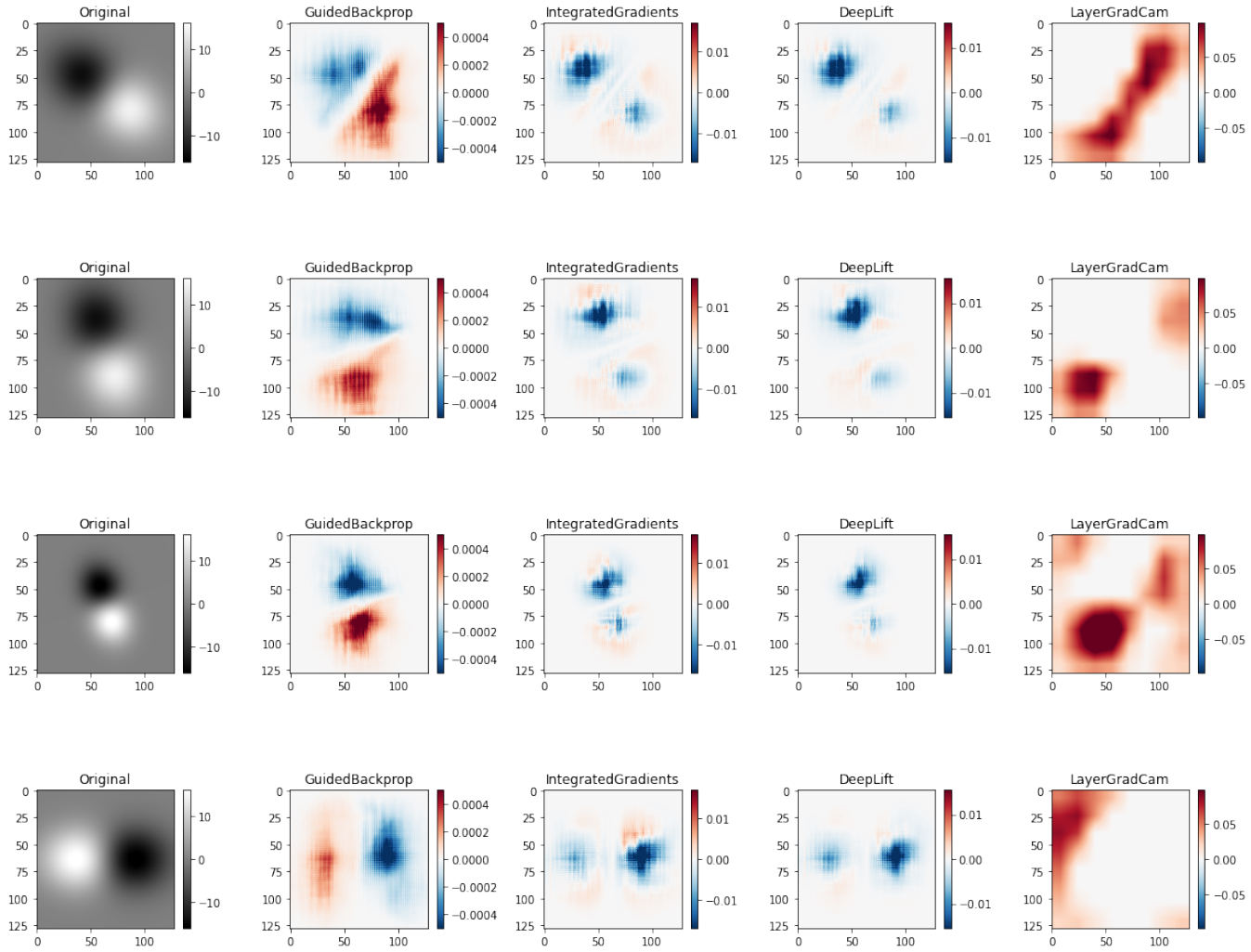


Figure 14: Examples of synthetic bipole images and attribution maps.

a reference image, which we select as the first image in the sequence. No other temporal information is exploited by these attribution methods.

As case studies, we consider four HARP sequences that transition from a flare-quiescent state to a flare-imminent state. The HARP numbers and the associated NOAA AR active region numbers are: HARP Figure 15 shows the labels and predicted probabilities of the four sample sequences.

Figure 16 shows the last image of the four HARP sample sequences. The attribution maps of the same size as the input of the CNN (128× 128 pixels) are upsampled to the original resolution of the SHARP magnetogram using the `resize` method in the Python package `skimage.transform` with 2nd-order spline interpolation. The attribution maps of DeepLIFT and Integrated Gradients are similar. As such, only the results of the former are shown. The results for Integrated Gradients can be accessed online with the link shown in the caption.

In Figure 16, the attribution maps of Guided Backpropagation are observed to be more concentrated in strong fields compared to that of Deconvolution. The reference image of DeepLIFT and Integrated Gradients are chosen as the first sample in each sequence. From these two methods, the change of the prediction scores is attributed to the change of magnetic configuration of the last frame relative to the first frame, with red pixels indicating positive contribution and blue pixels indicating negative contribution. Since the predicted event probability of the last frame is higher than the first frame for all HARPs (Figure 15), the red pixels outweigh the blue pixels in the attribution maps of DeepLIFT and

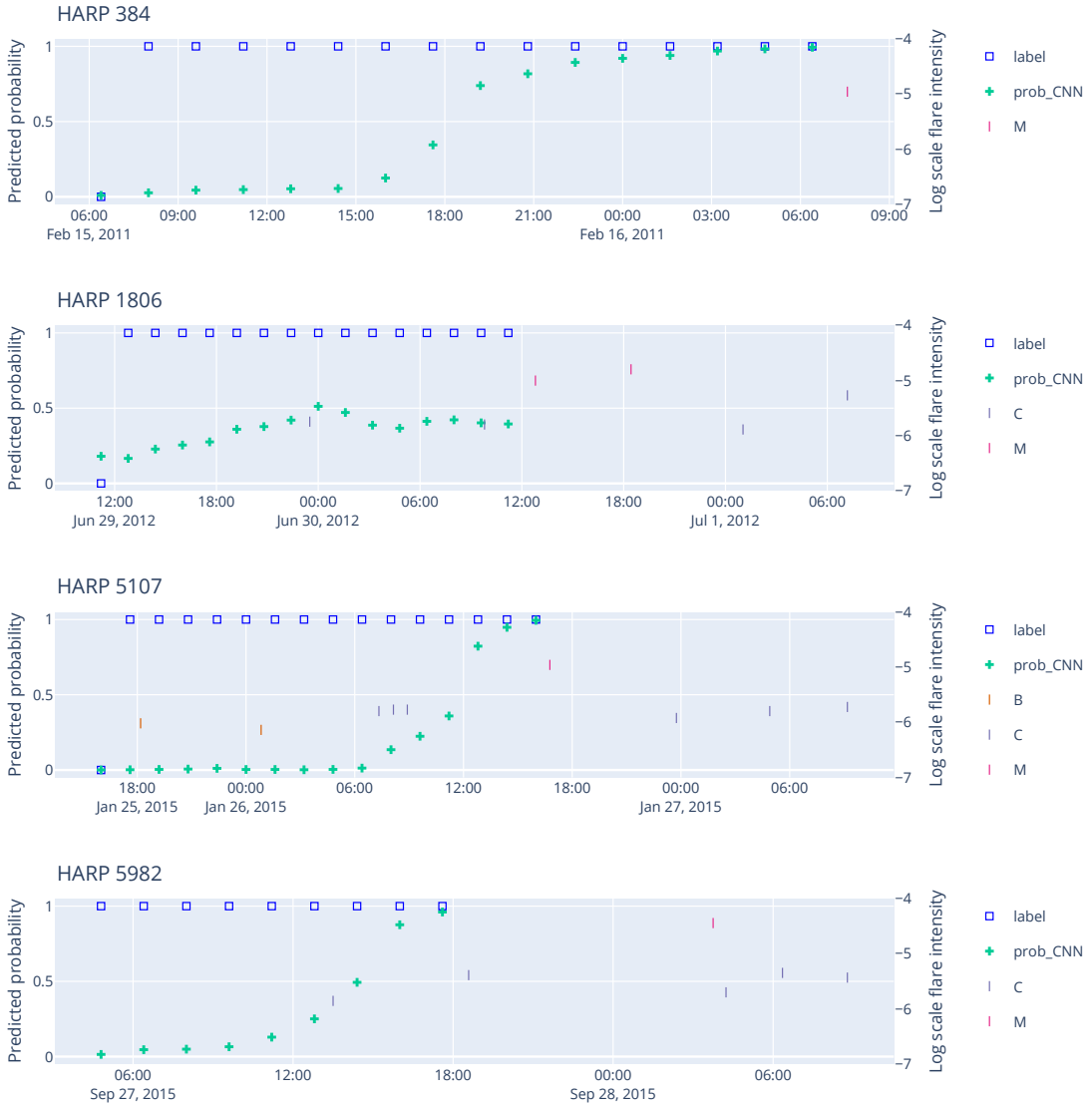


Figure 15: CNN predictions of part of time series of in HARP 384, 1806, 5107, and 5982. The labels are shown as blue open boxes and predicted probabilities as green plus symbols. The point-in-time instance is labeled as positive if an M1.0+ flare occurred in the future 24 hours in that active region. GOES flare events during and 24 hours within the sample sequence are shown as short vertical bars, with y-coordinates indicating flare intensities (peak flux in W/m^2) on a log scale.

Integrated Gradients. The Grad-CAM results roughly reveal the position of the strong fields and polarity inversion lines.

Figure 17 shows the contour plots of attribution maps overlaid on magnetograms of the four HARP series. The contours enclose areas with large absolute values of Integrated Gradients in the last frame of each series, with red/blue contours indicating the region contributing positively/negatively to the increase in predicted probability. A general pattern is that the flux is emerging in red contours and canceling in blue contours. From the attribution maps, we can explain the increase in prediction scores as the consequence of the emerging flux outweighing the canceling flux. In Figure 17(d), it is interesting to note that the emerging polarity inversion line in the penumbra of the leading polarity (on the right/west part of the active region) is picked up as a preflare signature by the largest red contour. However, this PIL is actually an artifact of magnetogram due to the projection effect when the magnetic vector's inclination relative to the line-of-sight surpasses 90° (Leka et al. 2017). This effect is significant at the end of the HARP 5982

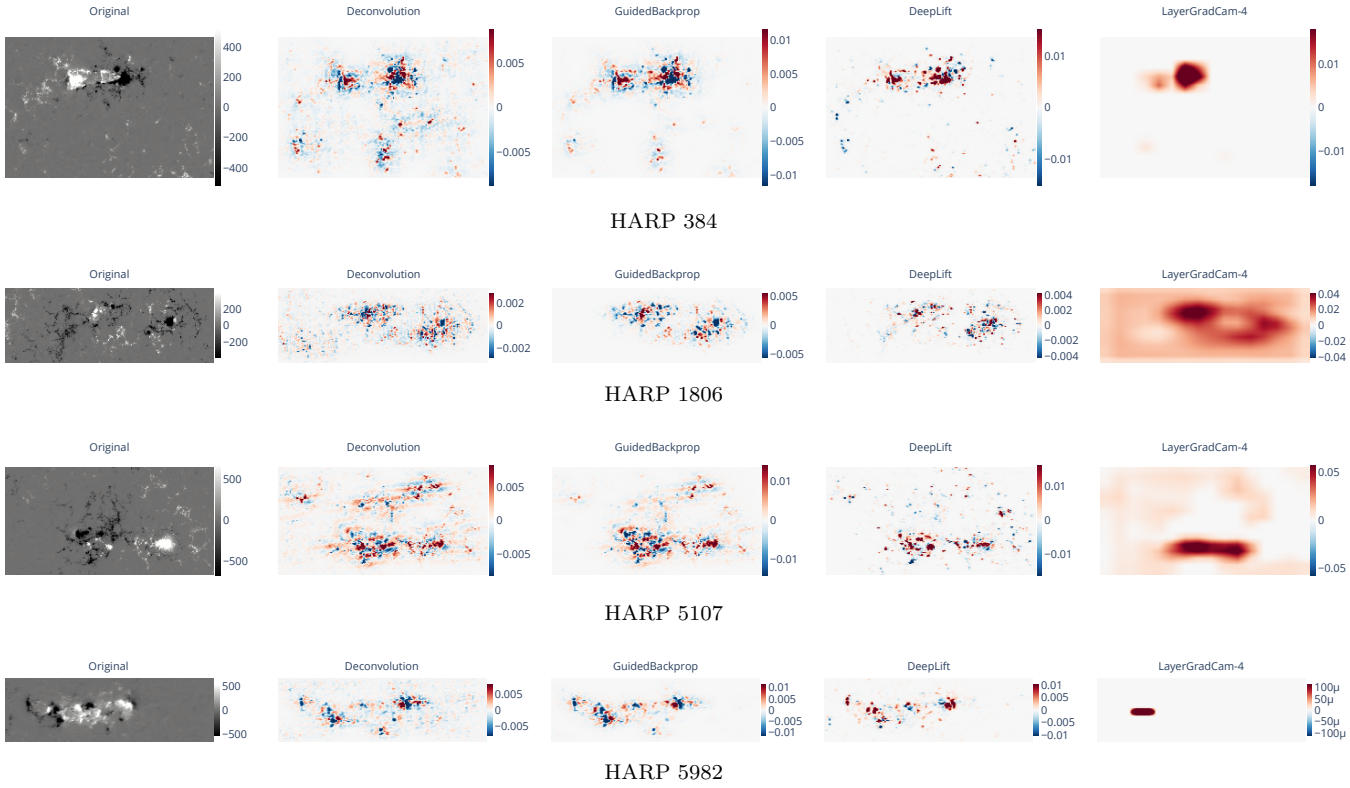


Figure 16: Attribution results of Deconvolution, Guided Backpropagation, DeepLIFT, and Grad-CAM on the last magnetogram in the sample sequences of HARP 384, 1806, 5107, and 5982. DeepLIFT chooses the first sample in the sequence as the reference. "LayerGradCam-4" means Grad-CAM with respect to the output of the fourth, or the second to last, convolutional layer. The interactive movie of heatmaps on all 9 samples in HARP 5982 using more attribution methods can be accessed at https://zeyusun.github.io/attribution/captum_movie_first.html.

series. The magnetic field is highly inclined in the penumbra of the leading polarity as the flux rope is elevating from the surface the on the limb of the solar disk (Figure 18). This shows that the CNN trained to associate magnetogram and flaring activities is not able to discern the polarity artifact by itself. This suggests that the model could be potentially improved if we feed the location information to the model to help CNN correct such artifact.

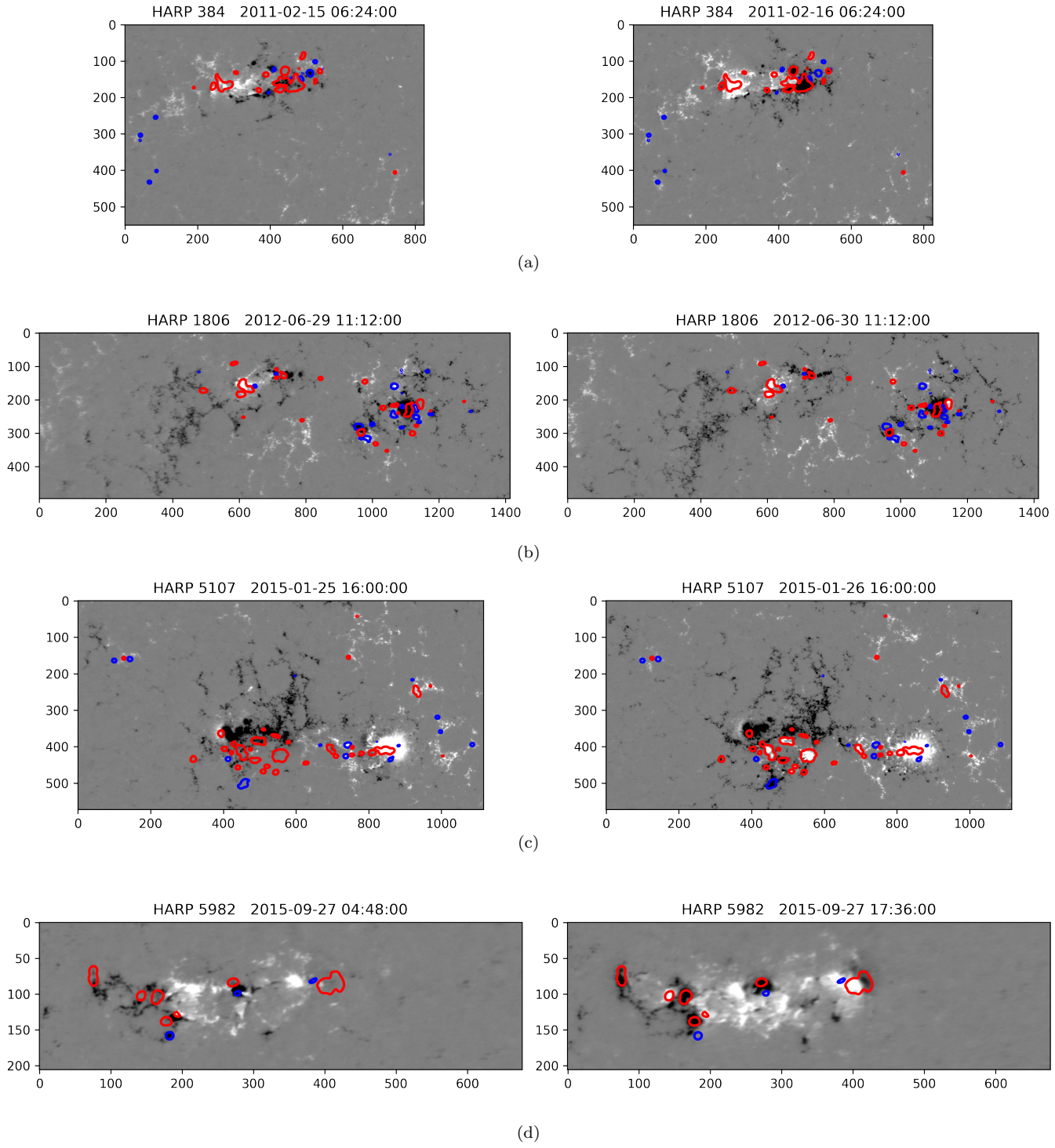
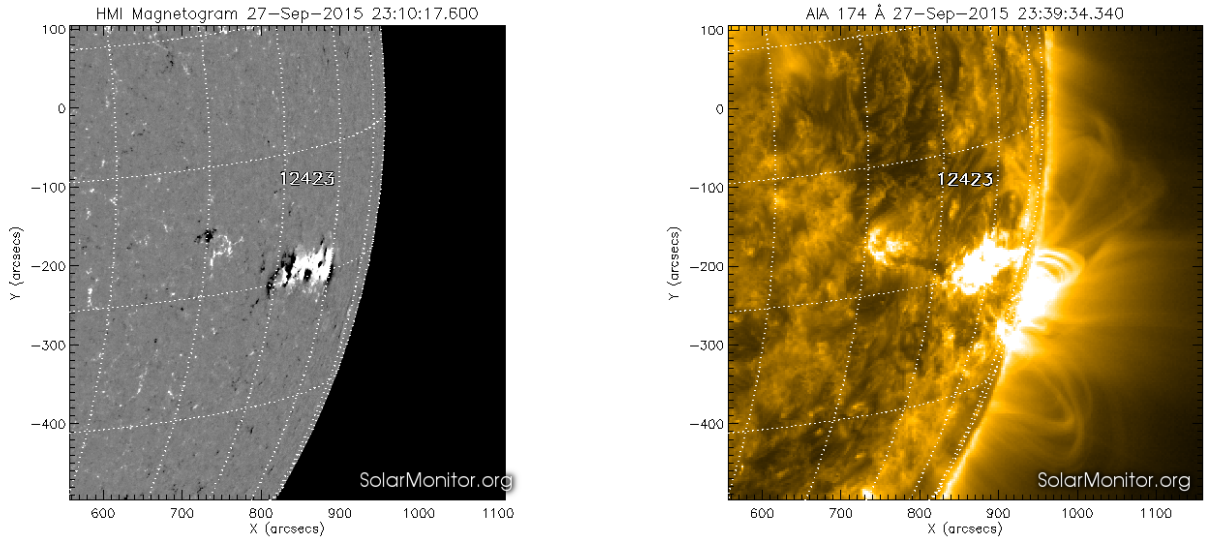


Figure 17: Highly attributed pixels in the last frame by Integrated Gradients on four select HARPs shown in rows. In (a), the left/right panel shows the first/last magnetogram in the sample sequence of HARP 384. The magnetograms are in the SHARP resolution, with ticks on the axes indicating pixels. Pixel values saturate at ±500 Gs. The red/blue contours on the right panel (last frame) highlight the areas with strong positive/negative Integrated Gradients relative to the first frame. The same contours are mapped to the left panel (first frame) for contrast. The contours are drawn on the attribution map smoothed with a Gaussian kernel with a standard deviation of 3 pixels. Figures in (b), (c), and (d) are similar to (a) but for other HARPs. The movies showing the entire samples sequence of can be accessed at, e.g., https://zeyusun.github.io/attribution/contours/5107/contour_movie.gif for HARP 5107.



(a) HMI magnetogram

(b) AIA 171 Å image

Figure 18: Line-of-sight magnetic field (a) and coronal image (b) of HARP 5982 (NOAA AR 12423) at 23:10:17 on Sep 27, 2015. Images are taken from https://solarmonitor.org/. Note that the image title of (b) should be "AIA 171 Å" instead of "AIA 174 Å".

5. CONCLUSIONS AND DISCUSSION

In this paper, we took advantage of a recently published dataset (Bobra et al. 2021) and examined the improvement in the flare predictive performance of two deep learning models, LSTM and CNN, when trained on the fused datasets. When tested on SMARP, both models showed significant improvement. When tested on SHARP, LSTM showed significant improvement. The results of the controlled comparative studies indicate such an improvement is due to the significantly increased sample size from the other solar cycle. Then, in our setting of flare prediction, we verified the performance of LSTM and CNN using skill scores, reliability diagrams, ROC, and skill score profiles. The comparison showed that LSTM is generally a better model than CNN. After that, we explored the possibility of combining LSTM and CNN for a better prediction performance in the framework of a meta-learning paradigm called stacking. The results showed that in some settings, the stacking model outperforms the best member in the ensemble. Lastly, we applied visual attribution methods to CNN. The results demonstrate the utility of visual attribution methods in identifying flare-related signatures in active regions, including the flux emergence and new polarity inversion lines. The attribution map on one particular region on the limb of the solar disk revealed one limitation of CNN and suggested potential modifications for improvement.

The questions raised in Section 4 are arguably broad and general. We have taken one particular path to partially address each question. To inspire future studies, we provide additional comments and discussions related to these questions.

Does more data help flare prediction?—We took a straightforward approach to add the new data in the training set and train the models as usual. Based on our experiments, this simple approach generally brings an improvement in flare prediction performance. However, such improvement is not observed for CNN when the additional magnetogram dataset comes from SMARP. Except for the difference in magnetogram quality, the way we fuse the two datasets could also be a reason for the performance decrease, which will be discussed in the next paragraph. We also note a slightly more advanced idea inspired from transfer learning (see, e.g., Weiss et al. 2016, for a survey): train on the additional data first, then switch to the original data. A brief experiment did not demonstrate the improvement, so this direction is not pursued. The SMARP dataset is the best effort the HMI team has made so far to extend the SHARP dataset backward for a solar cycle. The additional data could bring a broader impact, under the condition that the differences between the two databases are well accounted for.

On the fusion of data derived from MDI and HMI—There are exciting ongoing efforts in image super-resolution that transforms the MDI magnetograms to high-resolution HMI-style magnetograms (Shneider 2019; Jungbluth et al. 2019). By doing that, we do not have to compromise the image quality of SHARP magnetograms to make them useable with SMARP magnetograms. Rather, we can super-resolve SMARP magnetograms. The improved overall image quality will potentially retain more details of the active region and assist the discovery of flare precursors.

On problem setting and dataset design for flare prediction—Our setting of flare prediction is somewhat unique: a balanced classification problem that discriminates active regions that produce at least one flare of size greater than M1.0, from active regions that remain quiet within ± 24 hours from the flare issuance. The motivation of this problem setting is that we are trying to make the learning process as easy as possible, so that models can really learn something rather than being confused by extreme class imbalance ratio and complex flaring patterns. On the flip side, this setting has to be changed if we are going to apply the model in an operational setting.

Performance comparison between LSTM and CNN—The keywords used by LSTM are derived from magnetograms. In that sense, the data used by CNN contains complete information of the data used by LSTM. However, our experiments show that LSTM generally has better performance. There are many potential reasons that CNN does not perform better than, or as well as LSTM: (1) CNN takes in uniformly sized images whose size and aspect ratio are distorted. (2) CNN only uses the image of the last frame, whereas LSTM uses historical data; (3) CNN learns the features by itself, wheres LSTM uses hand-crafted parameterizations that are known to be relevant to flaring activity; (4) CNN uses subsampled images with information loss, whereas parameters are derived from full resolution images; (5) CNN has more parameters and more prone to overfitting (which reflects on the lower training loss but not validation loss of CNN in many experiments).

On flare forecast verification—(1) *Interpretation of the predicted probabilities.* Unlike mechanistic models, the probability reported by neural networks is not physically grounded, but purely a statistic learned from the data distribution. A consequence of this is that we are allowed to adjust the categorizing threshold for the probability in favor of any skill score of interest. (2) *A new graphical verification tool.* In addition to the widely used graphical tools of ROC, Reliability Diagram, and SSP, we advocate the use of sorted probability plot in flare forecast verification for richer information. A plot of this type not only reveals (overall) characteristic patterns as a whole, but also preserves the information of each individual sample. When integrated with interactive features, these plots are powerful tools to pinpoint, either attribute or blame, samples to a certain global pattern. (3) *Methodical comparison.* For meaningful results, models should be compared with the same evaluation dataset. Paired t-test should be used to claim if a model is better than others. This is the reason that all comparisons in this paper are made among the experiments, and no skill scores by other research are quoted or compared.

On stacking—In our experiments, stacking performs similarly to the "select best" strategy but not significantly better in most settings. However, Guerra et al. (2020) observed most ensembles achieved a better skill score (between 5% to 15%) than any of the members alone. We think the difference is that they consider more base learners, some of which involve human intervention, whereas we consider only two base learners, both of them machine learning models. On the positive side, stacking is noticeably better than average, which aligns with Guerra et al. (2020)'s observation. A promising direction is to incorporate other meta-features, so that the meta-learner is able to assign different weights to base learners in different situations.

Choice of the baseline in attribution methods—Some visual attribution methods require reference input, such as Integrated Gradients and DeepLIFT. One naive choice is an image with all values equal to zero. Images of this sort imply a lack of pattern. These are the baselines mostly used for interpretation in computer vision tasks like object detection. In our case, the images are magnetic field component measurements, which can take on positive or negative values and a wide dynamic range, unlike normal images in real life. We choose the first image in the sequence as the reference, so that the visual attribution methods can attribute the change of prediction scores to the change of magnetic field configuration, which is of actual interest. There are other choices of baselines. One example is input images with Gaussian noise. Using this type of reference may recover the sensitivity of the network's prediction to local changes. Also, the integration can take a different path other than simply linearly interpolating the reference and the input on the original image space, i.e., the 2D cartesian plane. A natural choice of path is the time series of active region magnetogram. The Riemann sum to approximate the path integral should be sufficiently accurate since the SHARP has a cadence of 12 minutes and the change in the magnetogram appears to be continuous. The Integrated Gradients

calculated with this approach have a temporal dependency on each point-in-time in the sequence and could reveal more information of the evolution.

The authors would like to thank K. D. Leka for valuable discussions on the polarity artifacts of the line-of-sight component of the photospheric magnetic field.

REFERENCES

Ahmadzadeh, A., Aydin, B., Georgoulis, M. K., et al. 2021, The Astrophysical Journal Supplement Series, 254, 23

Ancona, M., Ceolini, E., Öztireli, C., & Gross, M. 2018, in International Conference on Learning Representations

Barnes, G., Leka, K., Schrijver, C., et al. 2016, The Astrophysical Journal, 829, 89

Bengio, Y., & Grandvalet, Y. 2004, Journal of machine learning research, 5, 1089

Bloomfield, D. S., Higgins, P. A., McAteer, R. J., & Gallagher, P. T. 2012, The Astrophysical Journal Letters, 747, L41

Bobra, M. G., & Couvidat, S. 2015, The Astrophysical Journal, 798, 135

Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, Solar Physics, 289, 3549

Bobra, M. G., Wright, P. J., Sun, X., & Turmon, M. J. 2021, The Astrophysical Journal Supplement Series, 256, 26, doi: 10.3847/1538-4365/ac1f1d

Breiman, L. 1996, Machine learning, 24, 123

Campi, C., Benvenuto, F., Massone, A. M., et al. 2019, The Astrophysical Journal, 883, 150

Chen, Y., Manchester, W. B., Hero, A. O., et al. 2019, Space Weather, 17, 1404

Cohen, J. 1960, Educational and psychological measurement, 20, 37

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. 2018, arXiv preprint arXiv:1810.04805

Dua, D., & Graff, C. 2017, UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

Džeroski, S., & Ženko, B. 2004, Machine learning, 54, 255

Efron, B., & Tibshirani, R. 1997, Journal of the American Statistical Association, 92, 548

Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, Solar Physics, 293, 1

Freund, Y., & Schapire, R. E. 1997, Journal of computer and system sciences, 55, 119

Friedman, J. H. 2001, Annals of statistics, 1189

Guerra, J. A., Murray, S. A., Bloomfield, D. S., & Gallagher, P. T. 2020, Journal of Space Weather and Space Climate, 10, 38

Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, Space Weather, 13, 626

Hada-Muranushi, Y., Muranushi, T., Asai, A., et al. 2016, arXiv preprint arXiv:1606.01587

He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778

Hochreiter, S., & Schmidhuber, J. 1997, Neural computation, 9, 1735

Huang, X., Wang, H., Xu, L., et al. 2018, The Astrophysical Journal, 856, 7

Jonas, E., Bobra, M., Shankar, V., Hoeksema, J. T., & Recht, B. 2018, Solar Physics, 293, 1

Jungbluth, A., Gitiaux, X., Maloney, S. A., et al. 2019, arXiv preprint arXiv:1911.01490

Kingma, D. P., & Ba, J. 2014, arXiv preprint arXiv:1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, Advances in neural information processing systems, 25, 1097

Kubo, Y. 2019, Journal of Space Weather and Space Climate, 9, A17

Leka, K., & Barnes, G. 2003, The Astrophysical Journal, 595, 1296

Leka, K., Barnes, G., & Wagner, E. 2017, Solar Physics, 292, 36

Leka, K., Park, S.-H., Kusano, K., et al. 2019, The Astrophysical Journal Supplement Series, 243, 36

Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, The Astrophysical Journal, 891, 10

Liu, C., Deng, N., Wang, J. T., & Wang, H. 2017, The Astrophysical Journal, 843, 104

Liu, H., Liu, C., Wang, J. T., & Wang, H. 2019, The Astrophysical Journal, 877, 121

Liu, Y., Hoeksema, J., Scherrer, P., et al. 2012, Solar Physics, 279, 295

McCloskey, A. E., Gallagher, P. T., & Bloomfield, D. S. 2018, Journal of Space Weather and Space Climate, 8, A34

Murphy, A. H. 1973, Journal of Applied Meteorology and Climatology, 12, 595

Nishizuka, N., Kubo, Y., Sugiura, K., Den, M., & Ishii, M. 2020, The Astrophysical Journal, 899, 150

Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2018, The Astrophysical Journal, 858, 113

Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, The Astrophysical Journal, 835, 156

Nocedal, J., & Wright, S. 2006, Numerical optimization (Springer Science & Business Media)

Riley, P., Ben-Nun, M., Linker, J., et al. 2014, Solar Physics, 289, 769

Scherrer, P. H., Bogart, R. S., Bush, R. I., et al. 1995, SoPh, 162, 129, doi: 10.1007/BF00733429

Schou, J., Scherrer, P. H., Bush, R. I., et al. 2012, SoPh, 275, 229, doi: 10.1007/s11207-011-9842-2

Schrijver, C. J. 2007, The Astrophysical Journal, 655, L117, doi: 10.1086/511857

Selvaraju, R. R., Cogswell, M., Das, A., et al. 2017, in Proceedings of the IEEE international conference on computer vision, 618–626

Shneider, C. 2019

Shrikumar, A., Greenside, P., & Kundaje, A. 2017, in International Conference on Machine Learning, PMLR, 3145–3153

Silver, D., Huang, A., Maddison, C. J., et al. 2016, nature, 529, 484

Simonyan, K., Vedaldi, A., & Zisserman, A. 2013, arXiv preprint arXiv:1312.6034

Simonyan, K., & Zisserman, A. 2014, arXiv preprint arXiv:1409.1556

Springenberg, J., Dosovitskiy, A., Brox, T., & Riedmiller, M. 2015, in ICLR (workshop track)

Sundararajan, M., Taly, A., & Yan, Q. 2017, in Proceedings of the 34th International Conference on Machine Learning-Volume 70, 3319–3328

The SunPy Community, Barnes, W. T., Bobra, M. G., et al. 2020, The Astrophysical Journal, 890, 68, doi: 10.3847/1538-4357/ab4f7a

Wang, X., Chen, Y., Toth, G., et al. 2020, The Astrophysical Journal, 895, 3

Weiss, K., Khoshgoftaar, T. M., & Wang, D. 2016, Journal of Big data, 3, 1

Wilks, D. S. 2011, Statistical methods in the atmospheric sciences, Vol. 100 (Academic press)

Wolpert, D. H. 1992, Neural networks, 5, 241

Woodcock, F. 1976, Monthly Weather Review, 104, 1209

Yeates, A. R. 2020, Solar physics, 295, 1

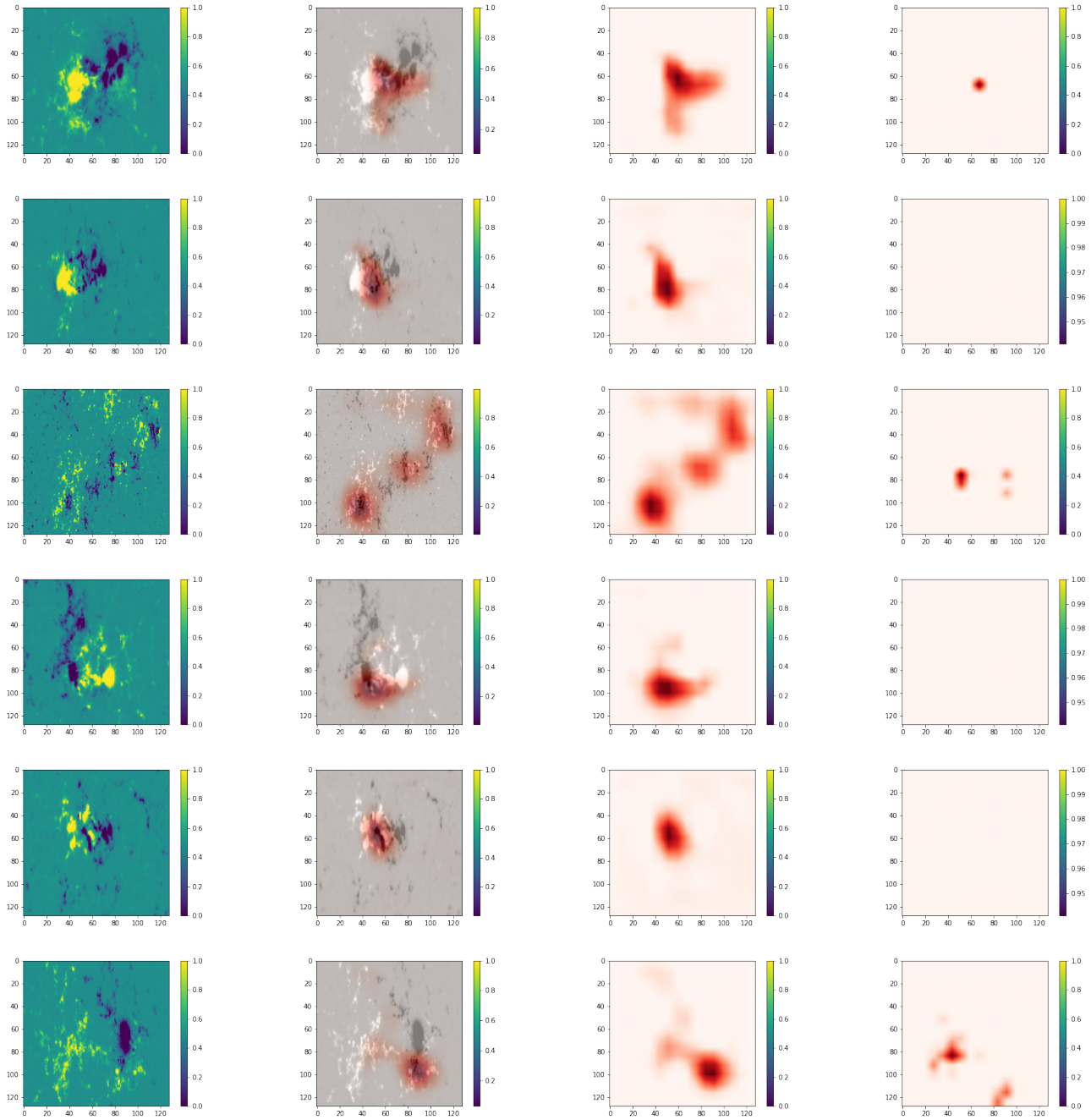
Yi, K., Moon, Y.-J., Lim, D., Park, E., & Lee, H. 2021, The Astrophysical Journal, 910, 8

Yuan, Y., Shih, F. Y., Jing, J., & Wang, H.-M. 2010, Research in Astronomy and Astrophysics, 10, 785

Zeiler, M. D., & Fergus, R. 2014, in European conference on computer vision, Springer, 818–833

Zheng, Y., Li, X., & Wang, X. 2019, The Astrophysical Journal, 885, 73

APPENDIX



(a) Input image (b) Grad-CAM overlaid (c) Grad-CAM map for CNN trained on `FUSED_SHARP` (d) Grad-CAM map for CNN trained on `SHARP_ONLY`

Figure 19: Grad-CAM visualization for positive class for CNNs. Shown are representative flaring active regions correctly detected by the CNN trained on `FUSED_SHARP` but missed by the CNN trained `SHARP_ONLY`.