Draft version November 23, 2021
Typeset using LaTeX **preprint** style in AASTeX63

Decreasing False Alarm Rates in ML-based Solar Flare Prediction using SDO/HMI Data.

Varad Deshmukh,[1] Natasha Flyer,[2] Kiera van der Sande,[3] and Thomas Berger[4]

[1]*Dept. of Computer Science, University of Colorado Boulder, Boulder, CO, 80309 USA*
[2]*Flyer Research LLC, Boulder, CO 80303 USA*
[3]*Dept. of Applied Mathematics, University of Colorado Boulder, Boulder, CO, 80309 USA*
[4]*Space Weather Technology, Research, and Education Center, University of Colorado at Boulder 3775 Discovery Drive, Boulder, CO 80303*

ABSTRACT

A hybrid two-stage machine learning architecture that addresses the problem of excessive false positives (false alarms) in solar flare prediction systems is investigated. The first stage is a convolutional neural network (CNN) model based on the VGG-16 architecture that extracts features from a temporal stack of consecutive Solar Dynamics Observatory (SDO) Helioseismic and Magnetic Imager (HMI) magnetogram images to produce a flaring probability. The probability of flaring is added to a feature vector derived from the magnetograms to train an extremely randomized trees (ERT) model in the second stage to produce a binary deterministic prediction (flare/no flare) in a 12-hour forecast window. To tune the hyperparameters of the architecture a new evaluation metric is introduced, the "scaled True Skill Statistic". It specifically addresses the large discrepancy between the true positive rate and the false positive rate in the highly unbalanced solar flare event training datasets. Through hyperparameter tuning to maximize this new metric, our two-stage architecture drastically reduces false positives by $\approx 48\%$ without significantly affecting the true positives (reduction by $\approx 12\%$), when compared with predictions from the first stage CNN alone. This, in turn, improves various traditional binary classification metrics sensitive to false positives such as the precision, F1 and the Heidke Skill Score. The end result is a more robust 12-hour flare prediction system that could be combined with current operational flare forecasting methods. Additionally, using the ERT-based feature ranking mechanism, we show that the CNN output probability is highly ranked in terms of flare prediction relevance.

varad.deshmukh@colorado.edu
natasha.flyer@gmail.com
kiera.vandersande@colorado.edu
thomas.berger@colorado.edu

Keywords: Solar flares, magnetograms, convolutional neural networks, extremely randomized trees

1. INTRODUCTION

Solar flares are the electromagnetic radiation outbursts accompanying Solar Magnetic Eruptions (SMEs) in the outer solar atmosphere (Fletcher et al. 2011). The resulting X-ray and EUV radiation ionizes Earth's upper atmosphere and can cause radio, radar, and Global Navigation Satellite System (GNSS) signal interference on the sunlit side of the Earth. In addition to flares, SMEs can also result in "Coronal Mass Ejections" (CMEs), large magnetic plasma clouds launched into interplanetary space at speeds on the order of 10^3 km s^{-1} (Webb & Howard 2012). If CMEs impact the Earth's magnetosphere, they can trigger geomagnetic storms which can produce a variety of impacts including large-scale inner magnetospheric currents, global ionospheric disturbances, increased drag on satellites in Low-Earth orbit, and, in more severe cases, geoelectric fields that can destabilize or damage electric power transmission grids (Lucas et al. 2020). SMEs and the shock waves propagating in front of the associated CMEs can also accelerate charged particles to relativistic energies, resulting in "radiation storms" that propagate along the heliospheric magnetic field (Reames 2013) to endanger astronauts in space and potentially damage spacecraft avionics. Historically, solar flares were the first phenomenon discovered to be associated with SMEs (e.g., Carrington 1859) and the electromagnetic radiation from flares is the first indication we receive of an eruption on the Sun. Indeed, the size of SMEs is still generally described by the intensity of the X-ray irradiance from the associated flare. Thus while the ultimate goal is to predict SMEs, we follow common usage and apply the term "solar flare prediction" to describe the goal of our investigation.

Currently, operational solar flare prediction is accomplished using manual classification of sunspot active region (AR) shape, size, and complexity in visible-light images of the solar photosphere. The classifications are then associated with historical 24-, 48-, and 72-hour probabilities of producing X-ray flares of a given magnitude via look-up tables (McIntosh 1990) and modified by human forecasters to take into account factors such as emerging flux or imminent collisions with other ARs. This subjective and largely "climatological" forecasting method has demonstrated only limited success in predicting SMEs during the 3-day forecast windows (Sharpe & Murray 2017; Crown 2012). For several decades, efforts have been made to automate the flare prediction process with computer-based classification and prediction systems in order to improve upon the current process. In recent years, these efforts have taken advantage of the rapid innovation in "machine learning" (ML) techniques developed for commercial image classification purposes. Qahwaji & Colak (2007) review some of the early attempts to employ machine learning to the solar flare prediction problem, and a recent series of papers compares the prediction skill of the manual method and several current automated models (Barnes et al. 2016; Leka et al. 2019a,b; Park et al. 2020).

The majority of automated flare prediction systems rely primarily on the properties of sunspot ARs derived from measurements of the one-dimensional line-of-sight (LOS or "longitudinal") component of the magnetic field in the solar photosphere. This is primarily because there are now almost 30 years of space-based full-disk photospheric magnetic field images ("magnetograms") taken on cadences of order 10-min or less that provide continuous, consistent, and high quality data. However, the primary magnetic reconnection that triggers SMEs does not take place in the photosphere (e.g., Simões et al.

2015), and with few exceptions (e.g., Sudol & Harvey 2005) photospheric magnetograms show no significant changes before and after SMEs. Ideally one would use magnetic field measurements in the upper solar atmosphere (the chromosphere and corona) to better predict eruption triggering, but there are not yet any reliably available, consistently high-quality, magnetic field measurements in the upper solar atmosphere. Thus solar physicists have concentrated on searching for physical properties derivable from photospheric magnetograms that correlate to a high probability of imminent eruption (e.g., Schrijver 2016; Kusano et al. 2020). Recently, the full-disk *vector* (i.e., three-dimensional) magnetograms from NASA's Solar Dynamics Observatory (SDO, Chamberlin et al. 2012) Helioseismic and Magnetic Imager (HMI, Scherrer et al. 2012) instrument have enabled the creation of a large database of derived AR magnetic field quantities called the SHARP parameters (Bobra et al. 2014a). Since 2015, many ML solar flare prediction systems have used the SHARP parameters, or a subset of parameters, as the primary feature vector input to supervised learning architectures (e.g., Bobra & Couvidat 2015; Bobra & Ilonidis 2016; Florios et al. 2018; Chen et al. 2019; Deshmukh et al. 2020). The SHARP AR image cutout and feature set has recently been expanded to include data from the predecessor instrument to SDO/HMI, the SOHO/Michelson Doppler Imager (MDI; Scherrer et al. 1995), to create the SMARPs dataset (Bobra et al. 2021).

A common challenge for all ML-based solar flare prediction models is the relative dearth of large, space-weather important (defined as X-ray class M1 or above on the NOAA radio black-out scale[1]) flares on which to train the models. Large SMEs and their associated large flares are relatively rare compared to the many smaller flares that occur in any given AR over the course of its evolution. Thus for any given sequence of AR magnetograms, there will be many more "non-flare" magnetograms, i.e, magnetograms that do not have an M1 or larger flare within the next k hours, where k is the forecasting window (typically 24, 48, or 72 hours), than there are "flare" labelled magnetograms. This fact, combined with the fact mentioned above that photospheric magnetograms show only minor changes before and after flares of any size implies that ML models will naturally train to predict no flaring, achieving high accuracy scores at the expense of sensitivity[2]. This training set imbalance can be addressed in several ways. Balancing training sets by removing non-flare examples (e.g., Chen et al. 2019) improves sensitivity, but such a model is likely to be difficult to optimize for operational space weather forecasting where the incoming real-time data stream is naturally extremely unbalanced. Other studies have addressed training set imbalance using oversampling of flare magnetograms during training (e.g., Zheng et al. 2021) or data augmentation: creating artificial flare magnetograms using image processing techniques such as affine transformations of real flare magnetograms or employing Generative Adversarial Networks (GANs; Zheng et al. 2019). Data augmentation has been successful in improving the training of ML image classification models (e.g., Wang et al. 2017), but our experiments with augmentation via affine transformation of flare magnetograms did not show improved skill over non-augmented training datasets (see Sec. 4). Another method of addressing training set imbalance that preserves the original dataset is to overweight the loss function used to train the network weights to penalize false negatives (missed flare detection) more severely than false positives. Models trained in this way can achieve high skill metrics in testing, but tend to overpredict flares, resulting in unacceptably high False Alarm Rates (FARs).

[1] See https://www.swpc.noaa.gov/noaa-scales-explanation for definitions of the NOAA space weather scales.
[2] See Jolliffe & Stephenson (2012) for formal definitions of these binary categorical forecasting metrics.

We have recently undertaken the development of ML models for prediction of solar flares based on Convolutional Neural Network (CNN) architectures that analyze spatial and, when used in recurrent architectures, temporal evolution of magnetic structure prior to flaring. Here we present a preliminary CNN model that analyzes SDO/HMI radial field ("B_r") magnetograms from the SHARP AR cutout series to produce a short-term (12-hour) probabilistic flare prediction. We achieve temporal correlation analysis by feeding the CNN several magnetograms in a temporal sequence as a single "multi-layer" input. This "temporal stacking" CNN input has been found to be more successful than the more traditional recurrent Long Short-Term Memory architecture models (Hochreiter & Schmidhuber 1997). We use loss function weighting to compensate for training set imbalance and tune hyperparameters using a new "scaled True Skill Statistic" (TSS_{scaled}) metric to optimize the flare/no-flare threshold of the CNN model. We address the relatively high FAR by developing a hybrid architecture that employs an additional extremely randomized trees (ERT) model. The ERT uses the flaring probability for a given magnetogram time series from the CNN stage as an additional feature added to derived features including the SHARPs parameters from the given set.

2. DATA

For this model, we use vector magnetogram image cut-outs observed by the Helioseismic and Magnetic Imager (HMI) instrument on-board the Solar Dynamics Observatory telescope Pesnell et al. (2012). These cut-outs — called as Spaceweather HMI Active Region Patch or *SHARPs* — have been tracking active regions on the surface of the Sun visible to the SDO since 2010 (Bobra et al. 2014b). For our dataset, we choose all magnetogram images, in the Cylindrical Equal Area (CEA) projection, across all recorded active regions from 2010 to 2017, at a cadence of 3 hours. This gives us a total of 157095 images. Each image includes metadata that contains features associated with the magnetogram including physics-based attributes extracted from the raw magnetic field data and deemed important by solar physicists. X-ray irradiance data from the NOAA Geostationary Observational Environmental Satellite (GOES) provide the location, intensity, and the onset, peak and termination times of recorded flares. Solar flares, based on the logarithm of the magnitude of their 1–8 Å X-ray irradiance, are classified into five major categories — A, B, C, M and X (in increasing order of magnitude). The first three classes usually are considered as minor flares, while the remaining two are major flares, and therefore of higher importance. Combining the SHARPs metadata and the GOES flare data, we can determine if a magnetogram produced a major flare (M- or X-class) in the next k hours. Since we are interested in short-term predictions that could generate solar flare warnings we set $k = 12$ for this study. Each magnetogram image is labeled as 1 if it produced an M/X flare within the following 12 hours, 0 otherwise. Since major flares are rare, this labeling results in an extremely imbalanced dataset. 1561 (≈1%) of the total magnetograms are labeled positive (flaring), and the rest of the 99% are labeled negative. Such a highly imbalanced dataset poses a challenge for training ML models.

Existing CNN flare prediction models (e.g., Huang et al. 2018; Park et al. 2018; Zheng et al. 2019; Li et al. 2020; Abed et al. 2021; Zheng et al. 2021) use balanced datasets for training and evaluating the models. The balanced datasets in these works are either generated by undersampling the majority class (lower intensity or no-flares) or oversampling the minority class (higher intensity flares) for both the training and testing sets. As mentioned, data augmentation is used successfully in image classification research to balance datasets. We applied this technique to augment the minority class of the training set by applying simple rotation and polarity swapping to generate new flaring

magnetograms. This reduced the dataset imbalance to 1:10. However we found that our model showed no improvement in flare prediction skill as measured by, e.g., the TSS of the test set. As a result, we decided not to include data augmentation in our experiments. To our knowledge the only ML flare prediction study to train on imbalanced data was Huang et al. (2018), and we observe that their model suffers from a high false positive rate similar to our CNN implementation described below.

The magnetogram image dataset is not directly usable as-is with deep learning models. Each image cut-out is variable in size, whereas the convolutional neural network we model requires fixed input dimensions across the entire dataset. There are multiple ways to transform all images to a standard size; in this work, we choose to perform affine transformations, which is a linear transformation that preserves lines and parallelism in an image, but not distances. We use the standard `OpenCV` package to convert all SHARPS radial magnetograms to a 128×128 pixel format. We find that these are the smallest set of dimensions that require less memory and processing time without affecting the quality of predictions.

To train and evaluate our architecture, we split our dataset into 70% training, 10% validation and 20% testing sets. The splitting is based on the active region number, so that all images of any given active region are present in the same set. The splitting is randomized 10 times using 10 random seeds; the model is trained, tuned and tested one randomized dataset at a time, and the statistics of the performance score reported across these 10 trials. With this arrangement, the total number of samples in the testing set is approximately 24000 samples, the positive samples varying from 137–347 and negative samples between 22187–24532. The positive and negative samples for the individual splits are shown in Table 6 in the appendix.

2.1. *Feature sets*

We use the SDO HMI radial field (B_r) magnetograms as input to a CNN model, as well as a source to extract numerical subsets of features that are used to train an extremely randomized trees (ERT) model, as described in Section 3. Here, we discuss the two types of numerical features extracted from the magnetogram data.

Physics-based Features

The first subset of features are the standard attributes of an active region cut-out available in the metadata of the SDO/HMI SHARPs dataset. These attributes, such as the area, total magnetic flux, magnetic shear, total vertical current, current helicity, etc., are predominantly derived from the spatial and/or extensive properties of the vector magnetic field in a given magnetogram image. A complete list of these features is available in Table 1.

Shape-based Features

To complement the physics-based features, we also include some shape-based features extracted using topological data analysis (TDA), as proposed in Deshmukh et al. (2020). TDA is an approach to characterize the shape of data in terms of its homology, i.e. by counting the $j-$dimensional holes of an object. The counts of these j-dimensional holes are defined as Betti numbers $\beta = \{\beta_0, \beta_1, \beta_2, ...\beta_{d-1}\}$, where d is the dimension of the manifold that the data lies in. β_0 counts the number of connected components in an object, β_1 the number of circular 2-dimensional loops, β_2 the total number of 3-dimensional voids, and so on. Since we are dealing with a 2-dimensional image for extracting the

Acronym	Description	Units
LAT_FWT	Latitude of the flux-weighted center of active pixels	degrees
LON_FWT	Longitude of the flux-weighted center of active pixels	degrees
AREA_ACR	Line-of-sight field active pixel area	micro-hemispheres
USFLUX	Total unsigned flux	Mx
MEANGAM	Mean inclination angle, gamma	$degrees$
MEANGBT	Mean value of the total field gradient	G/Mm
MEANGBZ	Mean value of the vertical field gradient	G/Mm
MEANGBH	Mean value of the horizontal field gradient	G/Mm
MEANJZD	Mean vertical current density	mA/m^2
TOTUSJZ	Total unsigned vertical current	A
MEANALP	Total twist parameter, alpha	$1/Mm$
MEANJZH	Mean current helicity	G^2/m
TOTUSJH	Total unsigned current helicity	G^2/m
ABSNJZH	Absolute value of the net current helicity	G^2/m
SAVNCPP	Sum of the absolute value of the net currents per polarity	A
MEANPOT	Mean photospheric excess magnetic energy density	$ergs/cm^3$
TOTPOT	Total photospheric magnetic energy density	$ergs/cm^3$
MEANSHR	Mean shear angle (measured using B_{total})	$degrees$
SHRGT45	Percentage of pixels with a mean shear angle greater than 45 degrees	$percent$
R_VALUE	Sum of flux near polarity inversion line	G
NACR	The number of strong LOS magnetic field pixels in the patch	N/A
SIZE_ACR	Projected area of active pixels on image	micro-hemispheres
SIZE	Projected area of patch on image	micro-hemispheres

Table 1. The SHARPs feature set, as available in the metadata of the SDO HMI dataset. Abbreviations: Mx is Maxwells, G is Gauss, Mm is Megameters, and A is Amperes. From Bobra et al. (2014b).

features, our Betti numbers are restricted to $\{\beta_0, \beta_1\}$. As in Deshmukh et al. (2020), we choose β_1 for our topology-based feature set.

On an image, TDA counts holes by first performing sub-level thresholding, i.e. keeping magnetic flux pixels below a chosen threshold and discarding the rest. The selected pixels connect to each other forming connected components (β_0) and loops with empty space between them (β_1). Repeating this process for 7 thresholds on the positive and negative flux structures on a magnetogram separately, we obtain the β_1 counts for each of the thresholds. We choose equally spaced magnitudes of magnetic flux thresholds for the positive and negative fluxes,

$$thresholds = \{20G,\ 420G,\ 820G,\ 1220G,\ 1620G,\ 2020G,\ 2420G\}.$$

This gives us a total of 14 TDA-based features, denoted by `flux_pos_t` and `flux_neg_t`, where $t \in thresholds$.

3. A TWO-STAGE MACHINE LEARNING PIPELINE

In this study, our primary machine learning model is a CNN. CNNs are an effective tool for extracting patterns from images which can subsequently be used in training the model to automatically classify the image content. This is advantageous as a way to avoid manual feature engineering of the dataset, or alternatively as a way to complement these manually engineered features. Because the pattern extraction is statistical in nature and not easily attributed to any particular heuristic, this process is sometimes referred to as "deep learning." Here we aim to combine the predictive power of the manually engineered features (SHARPs and topological) of the magnetograms with the features automatically extracted by a CNN model.

To do so, we propose a two-stage model architecture. The first stage implements a CNN architecture that is trained on the magnetogram images directly and classifies a flaring probability as the output. This flaring probability is then used as a feature along with the manually engineered features to train an extremely randomized trees (ERT) architecture. We choose this architecture on account of its design simplicity in being able to separate the prediction capabilities of the CNN features and the engineered features. An additional benefit of using the ERT architecture is its ability to rank the relevance of various features in terms of predicting flares. We discuss the two stages below.

3.1. *Stage I: Convolutional Neural Network*

The first stage of the model is a CNN adapted from the VGG-16 model - a deep CNN with 13 2-D convolution layers and 4 dense layers designed to classify images into 1000 pre-defined categories (Simonyan & Zisserman 2014). The input sample to our model is not a single magnetogram image, but a temporal stack of 4 consecutive magnetograms separated by a cadence of 3 hours. The input layer and the output layer of the VGG-16 model are modified to have 4 channels instead of 3, and two output nodes instead of 1000, respectively. The two output nodes respectively produce the probability of flaring and non-flaring in the next k hours for a given input sample, where k is the forecast window, which can be distinct from the 12-hour temporal stack we provide as input to the model. For this study however we set $k = 12$ to match the forecast window with the temporal span of the input data. Probabilistic output is desirable from a forecasting point of view. However, for comparison to most other automated solar flare prediction models that produce categorical flare/no-flare event predictions, we convert the probabilistic output into a categorical output by defining an optimal flare event threshold. Optimizing the threshold is accomplished using validation data, as described in Section 3.4. Some methods choose an arbitrary threshold of 0.5 to generate the categorical prediction (Leka et al. 2019a) while others use an automatically determined threshold based on optimizing the Receiver Operating Characteristic (ROC, Jolliffe & Stephenson 2012). Interestingly, this coincides with the threshold that maximizes the TSS score.

Four variations on the model architecture and the input data format were investigated:

1. C_1: Using an input stack consisting of the $[B_r, B_\phi, B_\theta]$ components of the vector magnetogram and setting the number of input channels of the VGG-16 model to 3.

2. C_2 Using a single component B_r at a single time as the input data per sample, setting the number of input channels of the VGG-16 model to 1.

3. C_3: Using the temporal stacked configuration $[B_{r,t}, B_{r,t-3}, B_{r,t-6}, B_{r,t-9}]$ as the input data per sample (as described above), setting the number of input channels of the VGG-16 model to 1. Each component in the temporal stack is operated on by the convolutions and dense layers

individually to generate 4 feature representations. An LSTM layer is introduced before the output layer to process the sequence of the 4 representations.

4. C_4: Using the temporal stacked configuration $[B_{r,t}, B_{r,t-3}, B_{r,t-6}, B_{r,t-9}]$ as the input data per sample (as described above), setting the number of input channels of the VGG-16 model to 4. All components of the temporal stack are treated as individual channels in the input layer. Such a setup leads to a single feature representation at the final layer that is acted upon by the softmax function, as opposed to the four representations generated from the temporal stack in C_3.

All four configurations were modeled in `Pytorch`. We use a weighted focal loss function ($\alpha = \frac{1}{11}$, $\gamma = 2$) (Lin et al. 2017) with an additional L_2 regularization weight decay factor $\beta = 0.001$. The weights of the models are updated using an `Adagrad` optimizer (Duchi et al. 2011), using an initial learning rate of 0.0001 and a batch size of 64. A cosine annealing learning rate scheduler is used for adjusting the learning rate through the training. The evaluation of each of these configurations (after hyperparameter tuning discussed below), is presented in the appendix in Table 5. Summarizing the results, we find that the B_r-only configuration C_2 performs slightly worse than the vector magnetogram configuration C_1. However, the temporal stacking configuration C_4 performs very similar to C_1. What this tells us is that from the perspective of the CNN, the B_r channel is sufficient for predicting flares and the other two components are superfluous. Additionally, comparing the two temporal stacking configurations, the LSTM model in C_3 does worse than using the temporal stack as channels as in C_4.

3.2. *Stage II: Extremely Randomized Trees (ERT) model*

While there have been CNN implementations proposed in recent years, the novelty in our approach is the combined use of the features extracted through convolutions together with engineered features based on the physics and the shape of the magnetogram. We generate a feature set that concatenates the output of the VGG-16 model (the probability of a flare) with these engineered features. These features are extracted from two main sources. The first is a set of physics-based features — called *SHARPs* — that are available in the SHARPs HMI data set as metadata (Bobra et al. 2014a). The second source is a set of features extracted using topological data analysis (TDA) (Zomorodian 2011), as applied to sunspot magnetograms in Deshmukh et al. (2020); Deshmukh et al. (2021). Concatenating the VGG-16 probability output, 20 SHARPs features and 14 TDA-based features, we have a complete feature set of 35 features which combines information from deep learning-based and feature engineered approaches.

We use this feature set to train an extremely randomized trees (ERT) model (Geurts et al. 2006) in the second stage. An ERT is a tree-based model built as a hierarchical structure of nodes that successively perform the operation of separating the dataset into two classes. The entire dataset is "fed" to the root of this tree and undergoes a sequence of splitting operations at the intermediate nodes. At each node, the incoming dataset is separated into two subsets — termed "left" and "right" — based on a feature thresholding criterion. That is, a random subset of m features is chosen from the entire candidate feature set, together with m random splits (one for each feature). The quality of each split at node n is s_n, determined by computing the reduction in some "impurity" metric of the dataset given by —

$$\Delta i(s_n, n) = i(n) - p_L \times i(n_L) - p_R \times i(n_R). \tag{1}$$

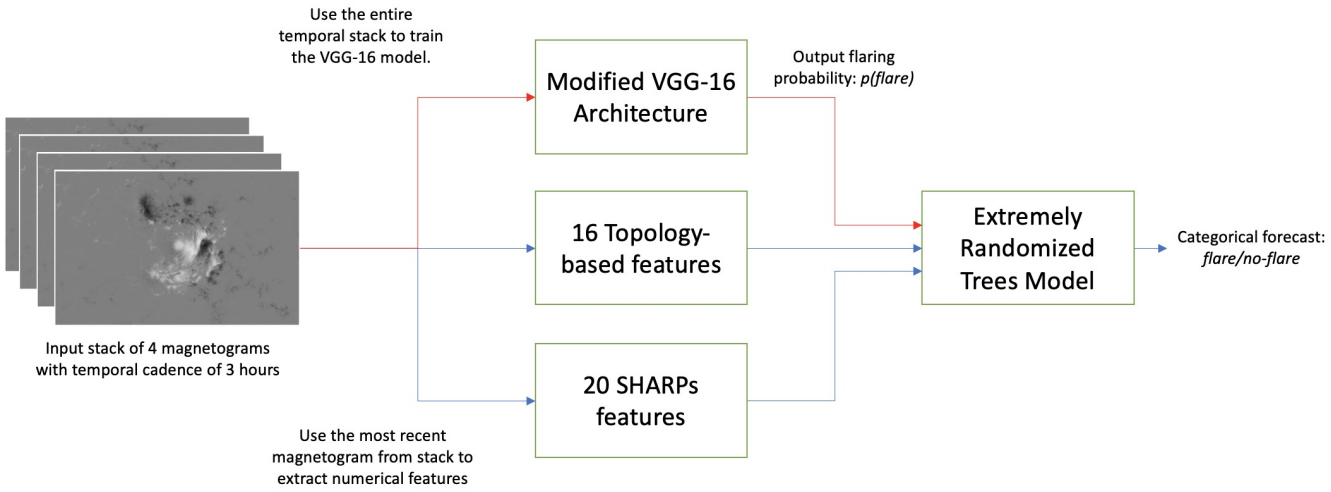


Figure 1. Our two-stage model for solar flare prediction. The input is a temporal stack of B_r magnetograms from SDO/HMI which is both fed to a custom CNN model and analyzed for feature vectors. The CNN model outputs the probability of flaring with the 12-hour forecast window and this probability is combined with the feature vectors to create a single feature vector input to the ERT model. The output of the ERT model is a binary event prediction.

Here, the impurity function $i(n)$ quantifies the degree of class intermixing for a given dataset input into node n. Correspondingly, the impurities for the left and the right subsets from the split are denoted by $i(n_L)$ and $i(n_R)$ respectively. $p_L = N_{n_L}/N_n$ and $p_R = N_{n_R}/N_t$ represent the proportions of the dataset arriving at node n of size N_n split into the left (size N_{n_L}) subset and right subset (size N_{n_R}) respectively. Of the m splits, the one that maximizes $\Delta i(s_n, n)$ is chosen. For the definition of impurity, we choose the standard Gini impurity index, as described in Raileanu & Stoffel (2004). Whether the two subsets are further subject to splitting at the next level is determined by an important hyperparameter in the training process known as the `min_impurity_decrease_index`, denoted by Δi_{min}. A dataset at any point in the tree is split further using an additional node if,

$$\Delta i(s_t, t) \geq \Delta i_{min}.$$

In the context of this problem, Δi_{min} determines how the model balances between the true positive rate (TPR) and false positive rate (FPR, also called the False Alarm Rate). A low value of Δi_{min} results in low FPR but a low TPR as well, whereas a high Δi_{min} raises the TPR at the cost of increased FPR as well. Just as with the threshold in the CNN stage, we tune this hyperparameter using a validation set (discussed in Section 3.4). The two-stage model is summarized in Fig. 1.

3.3. *Metrics*

With a categorical forecast (flare/no-flare), we can compute the standard confusion matrix on the testing set predictions: true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Using the entries of the confusion matrix, we study various metrics defined in Table 2. Most of these metrics are standard to flare prediction literature. A popular one among these is the true skill statistic score (TSS), equal to the difference TPR - FPR. TSS provides some utility to this problem because it is insensitive to dataset imbalance, and is a better indicator of the model

performance than the standard accuracy (Barnes et al. 2016; Bobra & Couvidat 2015). However, optimizing the TSS score often leads to an overforecasting model; models optimized on TSS tend to improve TPR at the cost of also slightly increasing the FPR. A slight increase in FPR can lead to a significant increase in the absolute false positives FP, since the number of negative samples is huge, thereby impacting other metrics like precision or F1, that are sensitive to FP.

Metric	Formula
Recall (TPR)	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
False Positive/Alarm Rate (FPR)	$\frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
True Skill Statistic (TSS)	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$
Heidke Skill Score (HSS)	$\frac{\mathrm{TP}\times\mathrm{TN}-\mathrm{FP}\times\mathrm{FN}}{(\mathrm{TP}+\mathrm{FP})(\mathrm{FP}+\mathrm{TN})+(\mathrm{TP}+\mathrm{FN})(\mathrm{FN}+\mathrm{TN})}$

Table 2. Metrics used for evaluating the binary forecasting models.

To address this problem, we define a new metric for model optimization, $\mathrm{TSS}_{\mathrm{scaled}}$, given by

$$\mathrm{TSS}_{\mathrm{scaled}} = \mathrm{TPR} - \frac{\mathrm{TPR}_{\mathrm{max}}}{\mathrm{FPR}_{\mathrm{max}}}\mathrm{FPR}, \tag{2}$$

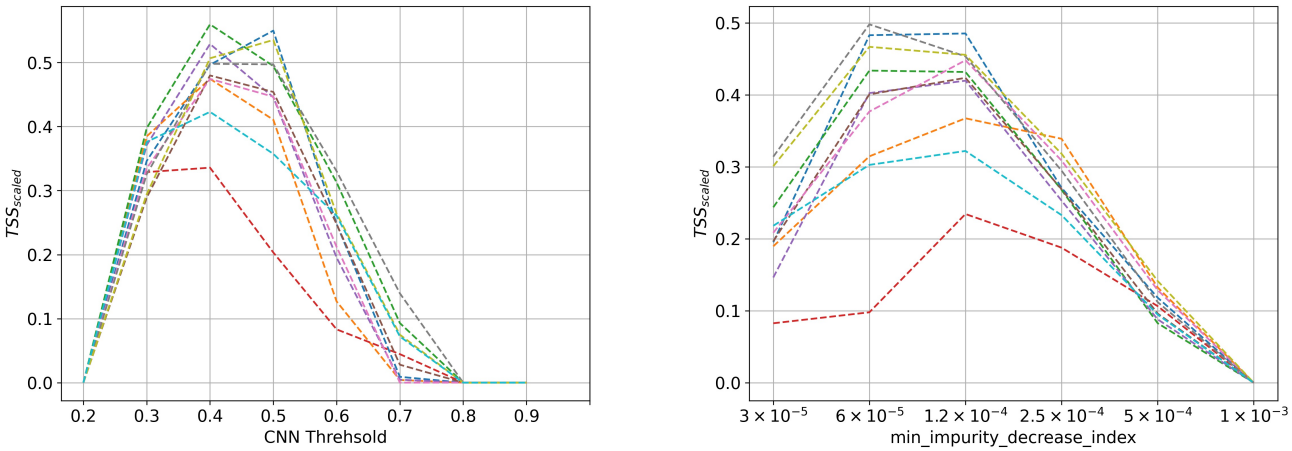
where $\mathrm{TPR}_{\mathrm{max}}$ and $\mathrm{FPR}_{\mathrm{max}}$ are the maximum values of the two metrics determined over the range of model hyperparameters. The scaling factor $\frac{\mathrm{TPR}_{\mathrm{max}}}{\mathrm{FPR}_{\mathrm{max}}}$ applied to the FPR term is a quantity greater than 1 for our hyperparameter choices that penalizes increase in false positives more than the TSS score. We therefore choose $\mathrm{TSS}_{\mathrm{scaled}}$ for selecting the model hyperparameters, which, as we will show in the next section provides a better balance between TPR and FPR.

3.4. *Hyperparameter Tuning*

Hyperparameter tuning is essential for optimizing machine leaning model performance. For both models, we identify the hyperparameter that significantly affects the TPR-FPR balance. In case of the first stage CNN-only model, it is the threshold for converting probabilistic to categorical forecast. In case of the second stage ERT model, it is the `min_impurity_decrease_index` parameter.

The process for choosing the optimal hyperparameters is performed individually on the CNN and ERT. The first step is to choose a set of suitable hyperparameter values to sample from in each case. The CNN stage is trained only once, determining the $\mathrm{TSS}_{\mathrm{scaled}}$ score by simply choosing different thresholds on the validation set. The ERT model is trained with all values of the chosen hyperparameter set.The setting that maximizes the chosen metric on the validation set — in this case, $\mathrm{TSS}_{\mathrm{scaled}}$ — is then determined.

This process is applied to both stages across the 10 randomly selected train-validation-testing set combinations. For the CNN-only stage, the tuning is performed over a range of eight even spaced thresholds in the interval $[0.2, 0.9]$. In the second stage, six values of `min_impurity_decrease_index` are chosen for tuning — $[3 \times 10^{-5}, 6 \times 10^{-5}, 1.2 \times 10^{-4}, 2.5 \times 10^{-4}, 5 \times 10^{-4}, 1 \times 10^{-3}]$ The results are shown in Fig. 2. It is clear that for the majority of the dataset combinations, a single hyperparameter maximizes $\mathrm{TSS}_{\mathrm{scaled}}$. For stage 1, this is a threshold of 0.4, for the second,



(a) Stage-1 CNN-only

(b) Stage-2 CNN+ERT

Figure 2. Hyperparameter tuning across multiple seeds for both stages of the hybrid flare prediction model. In both stages, hyperparameters that optimize the $\text{TSS}_{\text{scaled}}$ metric are determined.

`min_impurity_decrease_index`= 0.00012. Performing a similar optimization study for the TSS score instead yields an optimal threshold of 0.3 for stage 1 and an optimal `min_impurity_decrease_index`= 0.001. Note that if the CNN+ERT model were tuned with this latter value of 0.001, all 10 trials would have converged onto a $\text{TSS}_{\text{scaled}}$ score of zero. The TPR and FPR statistics on the validation set across the 10 trials for each of the these hyperparameter choices are shown in Table 3. It can be seen that optimizing over $\text{TSS}_{\text{scaled}}$ over optimizing TSS on average reduces FPR by a factor of approximately 2-4 while only reducing the TPR by a factor of approximately 1.2-1.4. The $\text{TSS}_{\text{scaled}}$ optimized hyperparameters offer a more favorable result in terms of the TPR-FPR balance. For our models, we therefore use the hyperparameters that optimize $\text{TSS}_{\text{scaled}}$ on the validation set.

Model stage	Optimized metric	Optimal hyperparameter	TPR	FPR
CNN-only	TSS	threshold = 0.3	0.90 ± 0.05	0.13 ± 0.02
CNN-only	$\text{TSS}_{\text{scaled}}$	threshold = 0.4	0.75 ± 0.09	0.06 ± 0.01
CNN+ERT	TSS	`min_impurity_decrease_index` = 0.001	0.90 ± 0.05	0.12 ± 0.02
CNN+ERT	$\text{TSS}_{\text{scaled}}$	`min_impurity_decrease_index` = 0.00012	0.65 ± 0.11	0.03 ± 0.01

Table 3. The mean and standard deviation values of TPR and FPR on the validation set (10% of the full dataset or $\approx 15,000$ samples) for the two stage models using TSS and $\text{TSS}_{\text{scaled}}$ metrics for optimizing hyperparameters. The statistics are generated over 10 trials with 10 different random dataset splits.

4. RESULTS

For the 10 dataset splits, we separately determine the confusion matrix on the test set for each of the two stages, and calculate the metrics discussed in Table 2. The optimal hyperparameters for each stage, as derived in Section 3.4, are used. We then evaluate the change in these various metrics between stages, i.e. using only the CNN and then appending the ERT to it. The raw values of six

metrics is shown in Fig. 3 in the form of box plots. Note also that TP and FP are presented in this plot instead of TPR and FPR. Looking at these two metrics in box plots Fig. 3(a) and (b), we observe that TP is slightly decreased with the use of the ERT architecture. On the other hand the FP values decrease significantly, thus reducing the over-forecasting nature of the model. It should also be observed that the FP box plots for the CNN-Only and CNN+ERT architecture are non-overlapping, demonstrating that the improvement is significant. The changes in TP and FP scores impact other metrics both positively and negatively. For example, the precision and the HSS score in Fig. 3(c) and (f) respectively are overall better for the CNN+ERT architecture, whereas the recall and TSS are overall slightly worse due to the dominance of TP in calculating these metrics. (Fig. 3(d) and (e)).

Metric	% average change in metric between stages
Recall (TPR)	-12 ± 6.9
False Positive/Alarm Rate (FPR)	-48 ± 12.4
Accuracy	3 ± 0.7
Precision	69 ± 16.7
TSS	-8 ± 7.0
HSS	56 ± 35.7

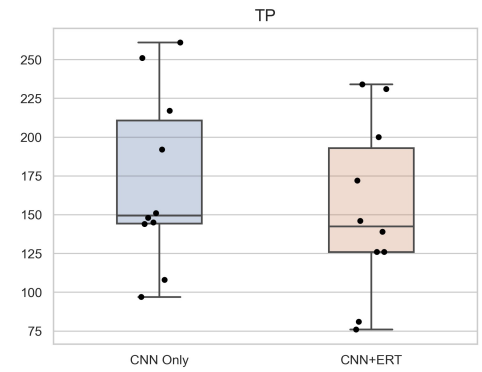
Table 4. Percent change in metrics of using the 2-stage model (CNN+ERT) over using a single stage CNN-only model, along with the standard deviation, summarized over 10 dataset experiments.

Table 4 shows the average percentage improvement across all the dataset splits, which summarizes the results in Fig. 3. The true positive rate is decreased (on average) by 12%, while the false positive rate improves by 48%. This impacts the derived metrics in different ways. For example, the more popular TSS metric is decreased by an average of 8%, and similarly the recall is decreased by an average of 12%. On the other hand, we see very large improvements in precision (≈69%) and HSS (≈56%). Thus, our two-stage model provides a prediction that can be more reliably incorporated into solar flare forecasting processes. We stress that operational flare forecasts are never dependent on a single model – they always incorporate multiple factors including climatological predictions, model predictions, and changing conditions evaluated in real time by forecasters.

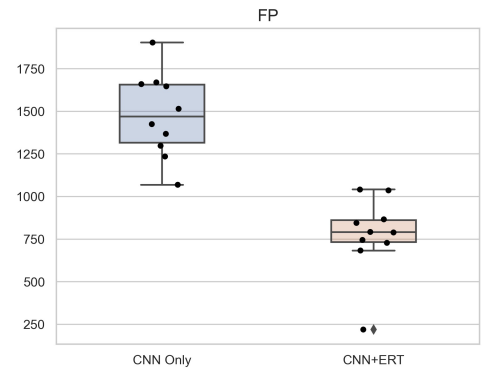
4.1. *Feature Ranking*

It is useful to know which features in our feature set play an important role in the model prediction. We use the Gini impurity index extracted from the ERT model to determine how much each feature is successful in separating the positive and negative labels across all the nodes of the tree. The relative rankings are shown in Fig. 4. While there are other ways for performing multivariate feature ranking, e.g. the linear discriminant analysis from Leka & Barnes (2007), extending our ERT prediction model to also perform feature ranking makes sense.

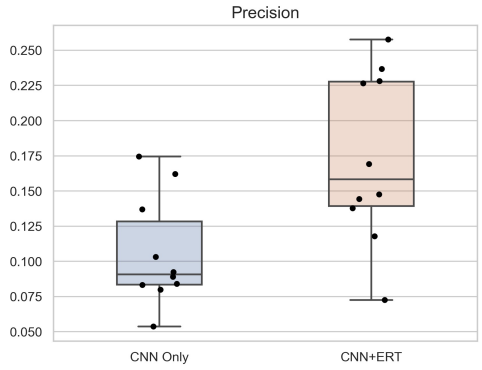
Fig. 4 shows that the CNN output probability `cnn_prob` ranks highest amongst all the features. In the top 10 features there is a mix of topological and physics-based features. The `R_VALUE` feature from the HMI feature-set ranks highly (second only to the CNN probability), since this corresponds to the magnetic flux from the neutral line where magnetic reconnection (and thus SMEs) occur. With



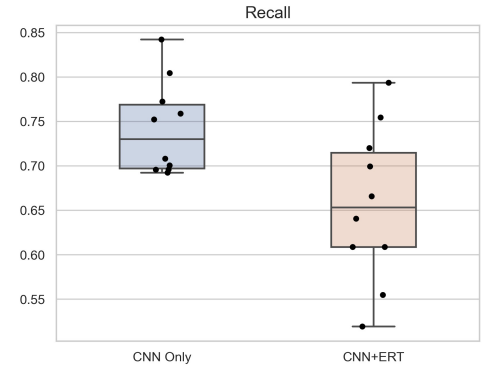
(a) TP



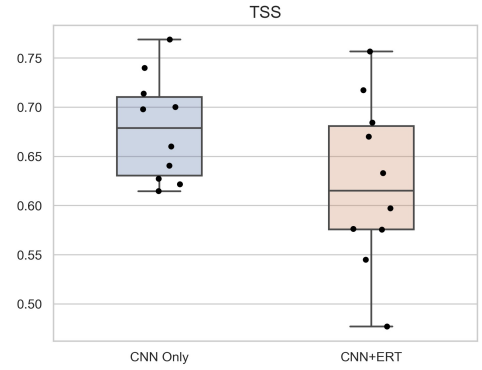
(b) FP



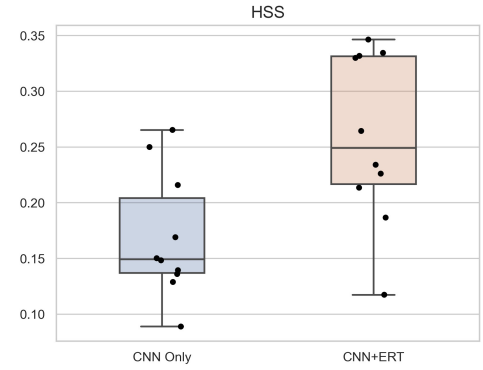
(c) Precision



(d) Recall



(e) TSS



(f) HSS

Figure 3. Performance comparison between CNN-Only and CNN+ERT models across six different metrics on the testing set. For each metric boxplot, the 10 dots shown represent the individual score of each of the 10 dataset splits.

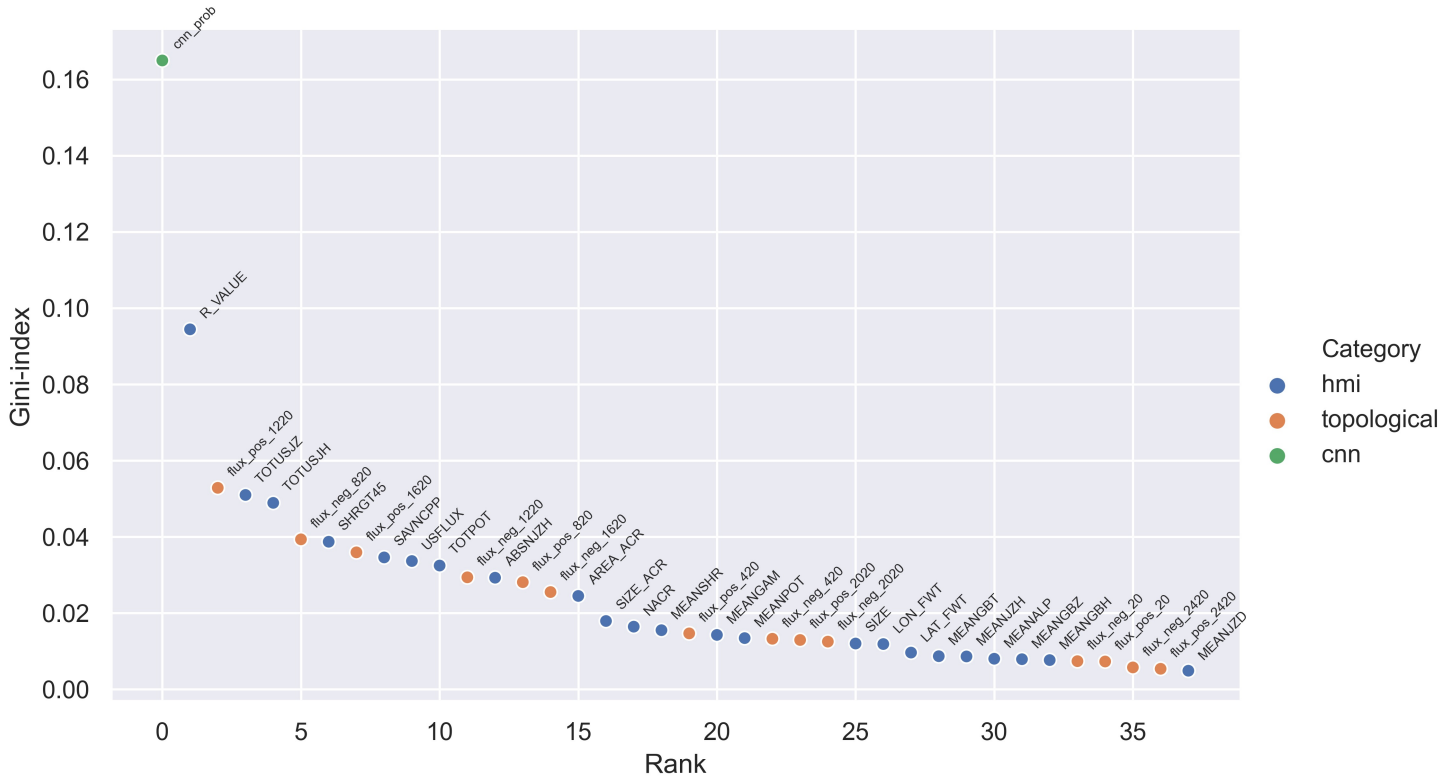


Figure 4. Feature ranking using the Gini impurity index from the ERT model.

regard to the topological feature set, the β_1 counts in the range of 800G to 1600G are shown to be important.

5. CONCLUSIONS

Extreme dataset imbalance is a significant challenge for machine learning-based solar flare prediction. To address this problem, various methodologies have been adopted in previous literature. A common approach is to balance the dataset, either through oversampling the minority class or undersampling the majority class. Some approaches weight the loss function of the model that penalizes mispredictions on the minority class more in relation to the majority class. Finally, many methods optimize models to maximize metrics such as the True Skill Statistic (TSS). This study presents a systematic evaluation of some of these approaches, uncovering their limitations and proposing modeling and evaluation strategies for overcoming them. To that end, we have proposed a two-staged machine learning model for predicting M1.0+ class flares in the next 12 hours. The first stage is a state-of-the-art VGG-16 convolutional neural network model that outputs a flaring probability by extracting features from raw magnetogram images. The output of this model is then used as input to an extremely randomized trees model in the second stage, along with various engineered physics-based and topological features extracted from the magnetogram.

Our first important contribution is the impact evaluation of various dataset manipulations and modeling strategies on the performance of a CNN-only model (i.e. the first stage VGG-16 model). Primary among the dataset manipulations is the dataset augmentation. We find that performing augmentation of the minority class (using standard rotation and polarity swapping) on the dataset

for this model yields no improvement in predictive skill. It should be noted that, unlike some studies which incorrectly augment both training and testing sets, we perform augmentation only on the training set. In addition, we also study the predictive performance of using a temporal sequence of the B_r component (with proper modeling) as opposed to all three components — B_r, B_θ and B_ϕ — of the vector magnetogram image to train the VGG-16 model. Our findings show that using the B_r sequence is just as predictive as the full stack, indicating the redundancy of other components. Finally, when modeling on a temporal sequence of image data, we show that using an LSTM layer at the end of the VGG-16 model performs worse than using the sequence as channels to the VGG-16 input layer.

The second focus of this paper is the use of binary categorization metrics for evaluating flare prediction models. While the standard TSS metric is often used, tuning hyperparameters to optimize on the TSS metric alone can lead to a model that highly over-forecasts, i.e. one with many false positives. To address this, we propose a modified alternative metric — $\mathrm{TSS}_{\mathrm{scaled}}$, which reduces the false positive rate in optimized models.

Our third major contribution from this paper is the use of the ERT model in the second stage that trains on a feature set that includes the output probability from the VGG-16 model from the first stage together with various engineered features. This two-stage design offers various advantages. First, this combines the prediction power of the automatically learned features from magnetogram images by the VGG-16 with the engineered features shown to be skillful in flare prediction in earlier studies. This can be considered as a comprehensive way to extract as much information from the magnetogram data as possible. Secondly, as we show, the two-stage model has significantly lower false positive rates compared to the VGG-16 model alone: it reduces the false positives (≈48%) without significantly reducing the true positives (≈12%). Finally, the ERT model provides a way to rank the forecasting capability of various features (VGG-16 output probability, physics-based, topological). In our ERT ranking, two features considerably outrank the others. The most highly ranked is the VGG-16 output probability, indicating that the first-stage model is skillfully predictive of flares. The second-most ranked feature is the `R_VALUE` parameter — the total flux in the polarity inversion line — a feature designed for discriminating flaring from non-flaring active regions (Schrijver et al. 2005).

In this work, we have explored numerous ways of extracting information from the photospheric magnetic field for the purpose of predicting flares. As in previous studies, the overall skill of even our best model does not significantly exceed the modified climatological forecasts developed by human forecasters (Leka et al. 2019b). We conclude that the information contained in photospheric magnetic field measurements alone is insufficient to predict SMEs with significantly more skill than a basic climatological prediction. In future studies we plan to include chromospheric and coronal observations from the SDO/Atmospheric Image Assembly (AIA) instrument for training machine learning solar flare prediction models. This will be a challenging task since there are no standard AIA features (analogous to the SHARPs features for HMI) with which to form feature vector inputs. Our focus will therefore be on developing deep learning CNN models that can efficiently extract predictive information from multi-wavelength time series of AIA images.

ACKNOWLEDGMENTS

This study was funded by a grant from the NASA Space Weather Science Applications Program (Grant No. 80NSSC20K1404) and by a grant from the National Science Foundation (Grant No. AGS 2001670).

REFERENCES

Abed, A. K., Qahwaji, R., & Abed, A. 2021, Advances in Space Research, 67, 2544, doi: https://doi.org/10.1016/j.asr.2021.01.042

Barnes, G., et al. 2016, Astrophysical Journal, 829, 89, doi: 10.3847/0004-637X/829/2/89

Barnes, G., et al. 2016, The Astrophysical Journal, 829, 89, doi: 10.3847/0004-637x/829/2/89

Bobra, M. G., & Couvidat, S. 2015, The Astrophysical Journal, 798, 135, doi: 10.1088/0004-637x/798/2/135

Bobra, M. G., & Ilonidis, S. 2016, ApJ, 821, 127, doi: 10.3847/0004-637X/821/2/127

Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014a, Solar Physics, 289, 3549, doi: 10.1007/s11207-014-0529-3

Bobra, M. G., Wright, P. J., Sun, X., & Turmon, M. J. 2021, ApJS, 256, 26, doi: 10.3847/1538-4365/ac1f1d

Bobra, M. G., et al. 2014b, Solar Physics, 289, 3549, doi: 10.1007/s11207-014-0529-3

Carrington, R. C. 1859, Monthly Notices of the Royal Astronomical Society, 20, 13

Chamberlin, P., Pesnell, W. D., & Thompson, B., eds. 2012, The Solar Dynamics Observatory (New York, NY: Springer)

Chen, Y., et al. 2019, Space Weather, 17, 1404, doi: 10.1029/2019SW002214

Crown, M. D. 2012, Space Weather, 10, doi: 10.1029/2011SW000760

Deshmukh, V., Berger, T., Meiss, J., & Bradley, E. 2021, Proceedings of the AAAI Conference on Artificial Intelligence, 35, 15293. https://ojs.aaai.org/index.php/AAAI/article/view/17795

Deshmukh, V., et al. 2020, Journal of Space Weather and Space Climate, 10, 13, doi: 10.1051/swsc/2020014

Duchi, J., Hazan, E., & Singer, Y. 2011, J. Mach. Learn. Res., 12, 2121–2159, doi: 10.5555/1953048.2021068

Fletcher, L., Dennis, B. R., Hudson, H. S., et al. 2011, Space Sci Rev, 159, 19, doi: 10.1007/s11214-010-9701-8

Florios, K., et al. 2018, Solar Physics, 293, 28, doi: 10.1007/s11207-018-1250-4

Geurts, P., Ernst, D., & Wehenkel, L. 2006, Machine Learning, 63, 3

Hochreiter, S., & Schmidhuber, J. 1997, Neural Comput., 9, 1735–1780, doi: 10.1162/neco.1997.9.8.1735

Huang, X., Wang, H., Xu, L., et al. 2018, The Astrophysical Journal, 856, 7, doi: 10.3847/1538-4357/aaae00

Jolliffe, I., & Stephenson, D. 2012, Forecast Verification: A Practitioner's Guide in Atmospheric Science (Wiley)

Kusano, K., Iju, T., Bamba, Y., & Inoue, S. 2020, Science, 369, 587, doi: 10.1126/science.aaz2511

Leka, K. D., & Barnes, G. 2007, Astrophysical Journal, 656, 1173, doi: 10.1086/510282

Leka, K. D., et al. 2019a, Astrophysical Journal, 243, 36, doi: 10.3847/1538-4365/ab2e12

—. 2019b, Astrophysical Journal, 881, 101, doi: 10.3847/1538-4357/ab2e11

Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, The Astrophysical Journal, 891, 10, doi: 10.3847/1538-4357/ab6d04

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollar, P. 2017, in Proceedings of the IEEE International Conference on Computer Vision (ICCV)

Lucas, G. M., Love, J. J., Kelbert, A., Bedrosian, P. A., & Rigler, E. J. 2020, Space Weather, 18, doi: 10.1029/2019SW002329

McIntosh, P. S. 1990, Solar Physics, 125, 251, doi: 10.1007/BF00158405

Park, E., Moon, Y.-J., Shin, S., et al. 2018, The Astrophysical Journal, 869, 91, doi: 10.3847/1538-4357/aaed40

Park, S.-H., Leka, K. D., Kusano, K., et al. 2020, ApJ, 890, 124, doi: 10.3847/1538-4357/ab65f0

Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, SoPh, 275, 3, doi: 10.1007/s11207-011-9841-3

Qahwaji, R., & Colak, T. 2007, Sol Phys, 241, 195, doi: 10.1007/s11207-006-0272-5

Raileanu, L. E., & Stoffel, K. 2004, Annals of Mathematics and Artificial Intelligence, 41, 77, doi: 10.1023/B:AMAI.0000018580.96245.c6
Reames, D. V. 2013, Space Sci Rev, 175, 53, doi: 10.1007/s11214-013-9958-9
Scherrer, P. H., Bogart, R. S., Bush, R. I., et al. 1995, Solar Physics, 162, 129, doi: 10.1007/BF00733429
Scherrer, P. H., Schou, J., Bush, R. I., et al. 2012, Solar Physics, 275, 207, doi: 10.1007/s11207-011-9834-2
Schrijver, C. J. 2016, Astrophysical Journal, 820, 1, doi: 10.3847/0004-637x/820/2/103
Schrijver, C. J., DeRosa, M. L., Title, A. M., & Metcalf, T. R. 2005, ApJ, 628, 501, doi: 10.1086/430733
Sharpe, M. A., & Murray, S. A. 2017, Space Weather, 15, 1383, doi: 10.1002/2017SW001683
Simões, P. J. A., Graham, D. R., & Fletcher, L. 2015, A&A, 577, A68, doi: 10.1051/0004-6361/201424795
Simonyan, K., & Zisserman, A. 2014, arXiv e-prints, arXiv:1409.1556. https://arxiv.org/abs/1409.1556
Sudol, J. J., & Harvey, J. W. 2005, The Astrophysical Journal, 635, 647, doi: 10.1086/497361
Wang, J., Mall, S., & Perez, L. 2017, arXiv:1712.04621v1
Webb, D. F., & Howard, T. A. 2012, Living Rev. Solar Phys., 9, doi: 10.12942/lrsp-2012-3
Zheng, Y., Li, X., Si, Y., Qin, W., & Tian, H. 2021, Monthly Notices of the Royal Astronomical Society, 507, 3519, doi: 10.1093/mnras/stab2132
Zheng, Y., Li, X., & Wang, X. 2019, The Astrophysical Journal, 885, 73, doi: 10.3847/1538-4357/ab46bd
Zomorodian, A. 2011, in Computational Topology (Providence, RI: American Mathematical Society)

APPENDIX

Configuration	Optimal threshold	TPR	FPR	Accuracy	TSS	HSS	ROC AUC	PR AUC
C_1: $[B_r, B_\phi, B_\theta]$	0.4	0.83	0.02	0.98	0.81	0.19	0.967	0.43
C_2: B_r	0.4	0.84	0.03	0.97	0.82	0.16	0.965	0.43
C_3: B_r stack w/LSTM	0.4	0.80	0.04	0.96	0.76	0.11	0.975	0.43
C_4: B_r stack as channels	0.4	0.79	0.02	0.98	0.76	0.18	0.974	0.46

Table 5. Performance of the VGG-16 model variants discussed in Section 3.

Random seed	P	N	Architecture	TP	TN	FP	FN
Seed 100	286	23345	CNN-only	217	21977	1368	69
			CNN+ERT	200	22662	683	86
Seed 200	207	23768	CNN-only	145	22343	1425	62
			CNN+ERT	126	22979	789	81
Seed 300	137	22283	CNN-only	97	21214	1069	40
			CNN+ERT	76	22064	219	61
Seed 400	347	22760	CNN-only	261	21525	1235	86
			CNN+ERT	231	22015	745	116
Seed 500	228	22787	CNN-only	192	21117	1670	36
			CNN+ERT	172	21942	845	56
Seed 600	156	24532	CNN-only	108	22628	1904	48
			CNN+ERT	81	23496	1036	75
Seed 700	325	22187	CNN-only	251	20889	1298	74
			CNN+ERT	234	21395	792	91
Seed 800	217	24007	CNN-only	151	22360	1647	66
			CNN+ERT	139	22966	1041	78
Seed 900	207	22425	CNN-only	144	20765	1660	63
			CNN+ERT	126	21697	728	81
Seed 1000	184	23509	CNN-only	148	21994	1515	36
			CNN+ERT	146	22643	866	38

Table 6. Comparison results for the CNN-Only and CNN w/ ERT architectures