A&A 674, A159 (2023)
https://doi.org/10.1051/0004-6361/202245742

Astronomy & Astrophysics

Comparing feature sets and machine-learning models for prediction of solar flares

Topology, physics, and model complexity

V. Deshmukh[1], S. Baskar[1], T. E. Berger[2], E. Bradley[1,3], and J. D. Meiss[4]

[1] Department of Computer Science, University of Colorado Boulder, 430 UCB, Boulder, CO 80309, USA
e-mail: varad.deshmukh@colorado.edu
[2] Space Weather Technology, Research, and Education Center, University of Colorado Boulder, 3775 Discovery Dr., Boulder, CO 80303, USA
[3] Santa Fe Institute, 1399 Hyde Park Rd, Santa Fe, NM 87501, USA
[4] Department of Applied Mathematics, University of Colorado Boulder, 526 UCB, Boulder, CO 80309, USA

Received 20 December 2022 / Accepted 25 April 2023

ABSTRACT

Context. Machine-learning methods for predicting solar flares typically employ physics-based features that have been carefully chosen by experts in order to capture the salient features of the photospheric magnetic fields of the Sun.
Aims. Though the sophistication and complexity of these models have grown over time, there has been little evolution in the choice of feature sets, or any systematic study of whether the additional model complexity leads to higher predictive skill.
Methods. This study compares the relative prediction performance of four different machine-learning based flare prediction models with increasing degrees of complexity. It evaluates three different feature sets as input to each model: a "traditional" physics-based feature set, a novel "shape-based" feature set derived from topological data analysis (TDA) of the solar magnetic field, and a combination of these two sets. A systematic hyperparameter tuning framework is employed in order to assure fair comparisons of the models across different feature sets. Finally, principal component analysis is used to study the effects of dimensionality reduction on these feature sets.
Results. It is shown that simpler models with fewer free parameters perform better than the more complicated models on the canonical 24-h flare forecasting problem. In other words, more complex machine-learning architectures do not necessarily guarantee better prediction performance. In addition, it is found that shape-based feature sets contain just as much useful information as physics-based feature sets for the purpose of flare prediction, and that the dimension of these feature sets – particularly the shape-based one – can be greatly reduced without impacting predictive accuracy.

Key words. Sun: flares – Sun: magnetic fields – sunspots – solar-terrestrial relations – methods: data analysis

1. Introduction

Solar flares are produced during magnetic eruptions from sunspot active regions (ARs) or filament channels on the Sun. The high-energy plasma emitted in conjunction with these eruptions can cause catastrophic events on Earth, with potentially trillions of dollars in associated economic losses. Forecasting these events is of obvious importance since, with enough notice, it is possible to mitigate some of their impacts. In current operational practice, flare forecasts are produced by human experts using established classification systems (McIntosh 1990; Hale et al. 1919) to categorize ARs into various classes. Forecasts for the probability of flaring in the next 24, 48, and 72 h are then constructed by consulting lookup tables of flaring rates for each category derived from historical records (Crown 2012). These climatological forecasts are of limited utility as they provide insufficient accuracy or reliability for actionable flare warnings to be issued. Over the past two decades, machine-learning (ML) solutions to this problem have been explored in an effort to provide higher predictive skill. After being trained on corpora of observations of the Sun to learn correlations between the data and known instances of solar flares, these models can be applied to new observations to generate flare forecasts.

The most commonly used observations in solar-flare forecasting are magnetic field images called magnetograms that are captured by the Helioseismic and Magnetic Imager (HMI) on board the Solar Dynamics Observatory (SDO). The SDO, located in a geosynchronous orbit above White Sands, New Mexico, has been operational since 2010, measuring the photospheric vector magnetic field on the visible hemisphere of the Sun by exploiting the Zeeman effect in solar spectral lines. In addition to the full-disk vector magnetic field data returned every 12 min, the HMI science team creates cutouts of ARs as they rotate across the visible disk. These cutouts are called Spaceweather HMI Active Region Patches (SHARPs; Bobra et al. 2014). These SHARPs records contain values for the radial, polar, and azimuthal components B_r, B_θ, and B_ϕ of the magnetic field at each pixel of the cutout image, along with a number of aggregate properties, such as the total unsigned magnetic flux and the total vertical electric current (Bobra et al. 2014). These physics-based metadata have been developed by solar physics researchers in the search for parameters relevant to the prediction of flares.

The SHARPs images and metadata have been used in a variety of ML methods. Some researchers have trained models directly on the SHARPs magnetogram images (for example, Huang et al. 2018; Park et al. 2018; Zheng et al. 2019, 2021; Li et al. 2020; Abed et al. 2021). However, it is well known in the ML literature that "featurizing" data – preprocessing it to extract higher-level properties that are salient in a given context – can be advantageous. The notion of salience is problem-specific; in computer vision, for instance, useful features might be edges or polygons. Most ML work to date has used the physics-based features in the SHARPs metadata as the preferred feature set for the flare forecasting problem. A wide range of models have been trained on this feature set: linear discriminant analysis (Leka & Barnes 2007), logistic regression (LR; Yuan et al. 2010), least absolute shrinkage and selection operator regression (Campi et al. 2019), support vector machines (Bobra & Couvidat 2015), random forests or extremely randomized trees (Nishizuka et al. 2017; Campi et al. 2019) – and, in recent years, deep-learning models such as MLPs (Nishizuka et al. 2018), long short-term memories (LSTMs; Chen et al. 2019), autoencoders (Chen et al. 2019), and Convolutional Neural Networks (CNNs; Deshmukh et al. 2022).

However, the performance of these ever-more-complex prediction models has so far not been markedly better than the current climatology-based systems used in operational settings (Barnes et al. 2016; Leka et al. 2019a,b). Moreover, increasing model complexity may not actually be an advantage in this application. It is possible that this is due simply to data limitations: in general, more complex ML models require larger training sets to effectively extract the patterns needed for prediction. This issue is exacerbated when correlations are deeply embedded in complex data for complex situations. A second issue is the nature of the training data. Until recently, ML-based flare forecasting work had not moved beyond the original SHARPs data: that is, the images themselves (the vector $\boldsymbol{B}$ values at each pixel) and the associated physics-based metadata. Choosing features that are scientifically meaningful makes good sense, of course, but ML methods can sometimes leverage attributes that are not obvious to human experts. For example, a recent *Nature* paper reports on using ML to discover a previously unknown correlation between geometrical and topological attributes of knots (Davies et al. 2021). A third and related concern is the dimensionality of the training data. This, too, is a well-known issue in the ML community, but there has been little exploration of it in the context of ML-based flare forecasting methods. Existing approaches have used the complete SHARPs feature set– or minor physics-based augmentations (for example, Sinha et al. 2022) – but it may well be the case that a carefully crafted subset of these features would work as well, or perhaps even better, thus reducing computational cost for a given forecast model.

In this paper, we aim to investigate the issues raised above. Our first objective is to explore whether there are useful ways to move beyond the set of physics-based attributes in the SHARPs metadata. Conjecturing that the shape of the magnetic structures in the photosphere could provide important clues about the evolving complexity of the field leading up to an eruption, we focus specifically on the abstract spatial properties of the magnetograms extracted using a mathematical technique called topological data analysis (TDA). Our preliminary studies using MLPs – also called fully connected networks (FCNs) – suggested that features like this could perform as well as a basic set of SHARPs features (Deshmukh et al. 2020, 2021). This was significant because the SHARPs feature set has been designed by solar physics experts, while the topological features are mathematical spatial attributes that are unrelated to physics. However, our previous study used a single ML model and the comparison only involved a subset of the SHARPs features. In this paper, we perform a more extensive comparison using a wider range of ML models, employ a new method of quantifying topological changes in the data, and compare the resulting flare predictive skill to a more comprehensive physics-based feature set.

Our second objective is to make a systematic comparison of the predictive performance of ML flare-forecasting models with varying complexity. The specific question we address is how increasing the model complexity affects the skill of a 24-h solar flare prediction. To that end, we study four ML models: logistic regression (LR), extremely randomized trees (ERTs), MLPs, and long short-term memory (LSTM) systems. Here, we define complexity informally as the number of free parameters that are optimized during the training of the model. For example, logistic regression requires calculating a scaling weight for each element of the input feature vector, whereas the MLPs and LSTMs that we use are complex networks with thousands of nonlinear, weighted connections that are adjusted during training. Comparing different ML models is not a trivial task given the complexity of the training process. Each model has a number of free parameters, known as hyperparameters, that control the learning process: for example, the learning rate in neural networks, or the maximum number of trees in an ERT. A systematic and fair comparison requires that the hyperparameters be optimized for each model and each test problem. To that end, we use an established hyperparameter tuning framework (Liaw et al. 2018) to optimize the performance of each model. Finally, we use several standard metrics to compare the predictive skill of these models on a different set of similarly labeled active regions. The results demonstrate that the simpler models (LR and ERT) actually perform better than the more complicated MLP and LSTM models.

We also demonstrate the impact of dimensionality reduction of the various feature sets on flare prediction skill. Feature sets often contain correlations and redundancies: for example, the mean and total values of some derived quantity. Removing these redundancies can both speed training and increase predictive skill. Here, we use Principal Component Analysis (PCA) to show that models trained on the most significant principal components of each feature set, that is, those features explaining 98.5% of the variance, largely equaled the performance of models trained on the full set of features. This is an advantage since the amount of data required to successfully train a machine-learning model grows with its dimensionality. We include this work in the study because while PCA dimensionality reduction is a well-known technique in ML, it is not clear how PCA will apply to the different feature sets developed for this study.

2. Data

In this study, we use magnetic field images ("magnetograms") captured by the Helioseismic and Magnetic Imager (HMI) instrument on board the Solar Dynamics Observatory (SDO; Scherrer et al. 2012). The HMI full-disk magnetogram data as well as the SHARPs cutout images are available from the Stanford Joint Space Operations Center[1]. We use the `hmi.sharp_cea_720s` data series, which provides the Lambert Cylindrical Equal-Area (CEA) projection (Snyder 1987) of the magnetic field images. We use only the radial field component, B_r, for flare prediction, as obtained directly from the image data associated with each SHARPs data file. Use of B_r is standard

[1] http://jsoc.stanford.edu

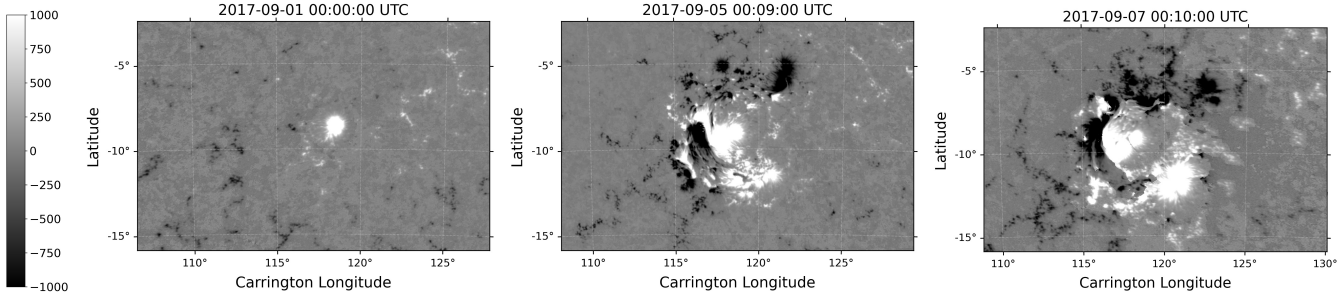

Fig. 1. Radial component of the magnetic field in NOAA active region AR 12673 (SHARP 7115). This region produced multiple M- and X-class flares during its rotation across the solar disk in September 2017. (a) Initial emergence of the field at 0000 UT on 1 September 2017; (b) Complex evolution at 0900 UT on 5 September 2017, approximately 24 h before producing an X17 flare, the largest flare of solar cycle 24; and (c) at 1000 UT on 7 September 2017, during an M-class flare.

practice in ML-based flare forecasting work because the radial surface field is by far the strongest component and it corrects for the variable line-of-sight angle between the Sun-Earth line and the solar surface across visible disk. Figure 1 shows SHARP #7115, a particularly active sunspot region that transited the solar disk in September of 2017. In each panel, white represents the positive polarity of the radial flux, while black corresponds to negative polarity. Panel (a) shows the radial flux captured shortly after the emergence of the AR where the configuration is simple, with one positive sunspot surrounded by scattered negative polarity "plage" fields. The AR evolves rapidly to develop a complex pattern of positive and negative polarity structures that exhibit close proximity of positive and negative polarity fields in a sheared or elongated pattern. This "sheared neutral line" pattern is frequently seen in large eruptions and indeed the largest flare of Solar Cycle 24 (an X17 class flare on 7 September 2017) was generated by this active region about 24 h after the observation shown in panel (b). Panel (c) shows the post-flare radial field configuration, which is still complicated, though perhaps less so than the field in the previous panel. Such a reduction in complexity is expected after peak flaring activity. However, the field in panel (c) is still highly sheared and this particular AR continued to produce large M- and X-class flares for at least another 5−10 days, including an extremely large eruption and X-class flare when the AR was on the west limb of the Sun on 10 September 2017.

For the study reported here, we use SHARPs data from May 2010 (early in the SDO mission) to December 2017 (near the end of Solar Cycle 24), using a one-hour cadence between magnetograms to reduce redundancy between consecutive data elements[2]. This data set contains 505 872 records. For training the ML models, each SHARPs record (images and metadata for a particular AR at a given time) is labeled based on whether the corresponding AR produced an M- or X-class flare in the next 24 h. Flare-related information is not part of the HMI metadata, so we use the NOAA Geostationary Operational Environment Satellite (GOES) X-ray Spectrometer (XRS) flare catalog[3], which contains records of the location on the solar disk, onset time, peak time, end time, and magnitude by class (A, B, C, M, or X) of all solar flares since 1975. For our period, which encompasses the weak activity levels of Solar Cycle 24, the GOES database reports only 509 M-class and 36 X-class flares. We label each SHARP record as "flaring" if an M1.0+ flare (one with an intensity above $10^{-5}\frac{W}{m^2}$) occurred from it within 24 h after the observation time; otherwise, it is labeled as "nonflaring". It is important to note that this means we categorize an AR magnetogram that may have produced a C9.9 flare as nonflaring while another AR that produces an M1.0 flare (essentially indistinguishable from a C9.9 flare) is labeled as flaring. This choice of a hard cutoff is necessary to produce a binary classification of flaring versus nonflaring active regions, but it obviously means that the ML model will be challenged with data from different classes that may have significant overlap in characteristics. Models using multiclass categorization or regression to scalar intensity values can avoid this difficulty, but we choose the binary "flare–no-flare" classification problem for this study to make it directly comparable to the large number of prior flare prediction studies that conduct similar binary classification predictions. Our binary M1.0 or greater flare–no-flare categorization yields a data set containing 3872 individual ARs, each identified by a unique SHARP number, with 453 273 SHARPs labeled as nonflaring and 5769 as flaring. It should be noted that the total number of labeled SHARPS is less than the total dataset number of 505 872 due to the removal of SHARPS that contain NaN values from off-limb data. This strong class imbalance (78 nonflaring magnetogram for every flaring magnetogram) in the training set is a major challenge for ML-based models but is also unavoidable in training for realistic conditions–artificially balancing the flaring and nonflaring event frequency in training produces models with unacceptably high false-positive rates.

Training, optimizing, and evaluating ML models requires splitting the data into training, validation, and testing sets, respectively. We assign all magnetogram data from a given AR to either the training or testing sets rather than assigning individual magnetogram data to each set randomly. This ensures that observations of the same AR at different points in its evolution do not appear in both the training and testing sets, thus preventing a possible artificial score improvement from testing the model on data it saw in training. Random assignment of magnetogram data from the same AR to training and test sets is a major error that produces leakage of training data into testing producing unnaturally high skill scores. Of the 3872 ARs in our data set, we assign 70% to the training set and 30% to the test set. The validation sets used for optimizing the models are produced as a fixed fraction of the training set as explained in Sect. 4.2. We produce ten different train-test data splits through randomized selections (again, ensuring that each AR is either in one or the other but not both

[2] Data from 2018–2020 were omitted from this study because the very few high-intensity flares that occurred during that period, which was at solar minimum, do not significantly add value to our current dataset; data from 2021 – present were not available at the time of the study, which we carried out in 2021 and 2022.

[3] https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-features/solar-flares/x-rays/goes/xrs/

sets), each generated with a different random number generator seed. These sets serve as ten separate trial runs of the experiment of training, optimizing, and testing our models, enabling statistical analysis of the results.

3. Featurization of magnetograms

The magnetogram data described in the previous section (for example, the value of B_r at every pixel in a SHARPs record) can be used directly to train models like Convolutional Neural Networks (CNNs), which were developed for the specific purpose of learning patterns in images. Alternatively, the magnetogram data can be processed to extract meaningful numerical attributes, or "features", of the magnetic field that can be assembled into "feature sets" and used to train other ML models, such as logistic regression models, Support Vector Machines (SVMs), Extremely Randomized Trees (ERTs), etc. Feature engineering–the task of crafting a set of attributes that help the ML methods work better–can be a challenge. The more salient the features are in the context of the task at hand, the more skill they give the method, but it is not always obvious which attributes to choose. Domain knowledge can be useful in this endeavor, but sometimes unexpected attributes can also be predictive, particularly in the context of complicated problems and rich data.

Here, we work with two different feature sets: (i) a set of standardized physics-based properties of the photospheric magnetic field used by many solar flare prediction methods and (ii) a set of abstract topological properties of the photospheric magnetic field. In addition we consider a third feature set that combines (i) and (ii).

3.1. Physics-based features

Each record in the SHARPs data set includes a set of metadata containing twenty values that characterize the corresponding AR at a particular time (Bobra et al. 2014)[4]. These attributes, developed and refined by the solar physics community (for example, Leka & Barnes 2007), are listed in Table 1.

Most of the SHARPs metadata are derived from the vector magnetic field observations and are designed to be useful predictors of solar flares. Two important examples are USFLUX (total unsigned magnetic flux) and the R_VALUE, or summed flux near a polarity inversion line, developed in Schrijver (2007). Both of these extensive parameters[5] increase significantly as an AR evolves to produce major flares making them useful features for prediction. It is also believed that indications of twist in the magnetic field above the photosphere, as parameterized by the total twist parameter MEANALP or total unsigned vertical current TOTUSJZ, for example, have predictive potential (for example, Nandy et al. 2008). It should be noted that some of the metadata parameters are simple extensive scalars such as AREA_ACR or location-related parameters such as the latitude and longitude of the flux-weighted center of the AR image (LAT_FWT and LON_FWT). While the area of an active region on the Sun is a sensible correlate for major flare productivity, location on the disk, particularly after cylindrical equal-area projection of the images, would not seem to be relevant. Nevertheless, all parameters listed in Table 1 are included in the ML model runs discussed below. All metadata listed in Table 1 are calculated automatically for each SHARP record by the data processing pipeline at the Joint Space Operations Center at Stanford, from which all data for this study were downloaded.

3.2. Shape-based features

The evolution of the magnetic fields in the Sun during the lead-up to major flares manifests as an increase in the topological complexity of the structures on the magnetogram, as seen in both the mixing of strong positive and negative polarity regions and highly sheared polarity inversion lines. This observation, which is visually obvious in Fig. 1, plays a critical role in the qualitative classifications used in operational space-weather forecasts. The McIntosh classification system used at the NOAA Space Weather Prediction Center, for instance, is based on characteristics like the size and number of umbras and penumbras in a given AR. These spatial characteristics are, however, absent from the SHARPs metadata shown in of Table 1, most of which are derived from vector operations (for example, div and curl) on the magnetic field, or scalar values such as means and totals. Thus it seems natural to explore whether features that characterize shape would be useful in ML-based flare forecasting methods–not only because this is what human forecasters use in their classifications, but also because AR shape in the photosphere has fundamental, meaningful connections to the coronal magnetic field physics leading up to an eruption.

Topology is the fundamental mathematics of shape: it distinguishes sets that cannot be deformed into one another by continuous transformations. Part of this shape classification – homology – corresponds to the number of connected components, 2D holes, 3D voids, etc., of a set. These numbers are known as the Betti numbers, $\beta_0, \beta_1, \beta_2, \ldots$, where β_k is the number of k-dimensional "holes" (Carlsson 2009). Topological data analysis (TDA), also called computational topology, operationalizes this framework for situations where one has only finitely many samples of an object, for example when working with digital images. The strategy for employing TDA in solar-flare forecasting was first developed in Deshmukh et al. (2020). We offer a brief review of the procedure for analyzing shape from magnetograms into feature vectors suitable for input to ML-based flare prediction models below; for more detail, please see Deshmukh et al. (2020).

One way to create a shape from finite data is to "fill in the gaps" between the samples by treating two points as connected if they lie within some distance ϵ of one other. Building an approximating object in this fashion, TDA computes the Betti numbers, then varies ϵ and repeats the process. The dependence of β_k on ϵ provides a rich morphological signature that captures the shape of an object at multiple resolutions. One can construct an even-richer representation by tracking the ϵ value at which each feature is formed, and at which it is destroyed or merges with another. This results in a set of (birth, death) values of ϵ for each feature (component, 2D hole, 3D void, etc.). This methodology has proved to be quite powerful; it has been successfully applied to a range of different problems ranging from coverage of sensor networks (de Silva & Ghrist 2007), to structures in natural images (Ghrist 2008), neural spike train data (Singh et al. 2007), and even the large-scale structure of the universe (Xu et al. 2019).

Our strategy for employing TDA in solar-flare forecasting is somewhat different from the ϵ-connectivity approach. Conjecturing that the shapes of the level sets of a magnetogram

[4] It should be noted that the original 16 parameters shown in Bobra et al. (2014) have been expanded over time to the set of 20 shown here.

[5] Extensive features are those that scale in proportion to the size of the underlying object.

Table 1. SHARPs physics-based feature set.

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	microhemispheres
USFLUX	Total unsigned flux	Mx
MEANGAM	Mean inclination angle, γ	degrees
MEANGBT	Mean value of the total field gradient	G Mm^{-1}
MEANGBZ	Mean value of the vertical field gradient	G Mm^{-1}
MEANGBH	Mean value of the horizontal field gradient	G Mm^{-1}
MEANJZD	Mean vertical current density	mA m^{-2}
TOTUSJZ	Total unsigned vertical current	A
MEANALP	Total twist parameter, α	Mm^{-1}
MEANJZH	Mean current helicity	G^2 m^{-1}
TOTUSJH	Total unsigned current helicity	G^2 m^{-1}
ABSNJZH	Absolute value of the net current helicity	G^2 m^{-1}
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 deg.	percent
R_VALUE	Sum of flux near polarity inversion line	Mx

Notes. Values for these 20 features, together with error estimates, are available for each magnetogram in the SDO SHARPs database. Abbreviations: A and mA are Amperes and milli-Amperes, respectively; Mm is megameters, G is Gauss and Mx is Maxwells.

are of central importance in this problem, we use the magnetic field intensity B_r as the variable parameter, rather than a distance ϵ. We threshold the SHARPs image, keeping only the pixels where the magnetic field intensity falls at or below some value (that is, sublevel thresholding) then compute the topology of the resulting object. By varying the threshold, and tracking the birth and death of each feature, we obtain a signature that captures the morphological richness of a magnetogram in a manner that factors in the spatial structure of field strength.

We build what is technically known as a "cubical complex" (Kaczynski et al. 2004) from the pixels in the SHARPs image whose B_r values fall below some threshold. Pixels are connected in such a complex if they share an edge or a vertex. Since magnetograms are 2D images, only connected components and 2D holes (which manifest as noncontractible loops in the thresholded image) make sense–there is no higher-dimensional structure. Counting these gives β_0 and β_1 for the given threshold field, and this computation is then repeated for a range of B_r thresholds.

Figure 2 demonstrates this procedure for an artificial 11×11 image. Each pixel in panel (a) is color-coded according to intensity. Given an intensity value, a sublevel thresholded image corresponds to those pixels with a magnitude at or below the threshold. The gray regions in panels (b–f) represent such images for thresholds $B_r \in [0, 4]$. Pixels that share an edge or a vertex in a thresholded image become a component and increment β_0. Empty regions (black) in the interior of a thresholded image that are surrounded a loop of connected gray pixels become holes, incrementing β_1. In panel (b), where $B_r = 0$, the gray, thresholded image contains two components separated by the empty, black region where the intensity is larger than zero; thus $\beta_0 = 2$. There is a single loop in the image (the red curve) that encloses the empty region, so it is noncontractible and $\beta_1 = 1$. Increasing the threshold to $B_r = 1$, as in panel (c), causes the structure to enlarge, shrinking the hole and splitting it into two; thus $\beta_1 = 2$.[6] The two components from panel (b) merge in panel (c), and for the remainder of the thresholding process, the number of components remains one, so $\beta_0 = 1$. Upon raising the threshold, as shown in panel (d), the dominant hole splits into seven while the single-pixel hole remains intact, bringing the number of holes to eight ($\beta_1 = 8$). At $B_r = 3$ in panel (e), two holes disappear; thus $\beta_1 = 6$. Finally, for $B_r = 4$ in panel (f), the number of holes is reduced to three as three of the holes are filled in by new image pixels. If the threshold were raised to five (not shown), all pixels would be filled so that the image would have $\beta_0 = 1$ and $\beta_1 = 0$.

In this study, we use the values of β_1 as a function of the threshold B_r to represent the topological complexity of a magnetogram. For the example with discrete thresholds $B_r = (0, 1, \ldots, 5)$, this gives the vector $\beta_1 = (1, 2, 8, 6, 3, 0)$. A similar vector for β_0 would give additional information about the image. For the example in Fig. 2, however, the number of components is not too interesting since – apart from the brief appearance of a second component at $B_r = 0$ – there is only one component for all thresholds. This holds true for the number of components for actual magnetograms: β_0 does not add much information to the analysis. For this reason, we exclusively use β_1 below.

A number of different representations have been developed by the TDA community to capture the information about the scale and complexity of the topology of data. These include "bar codes" (Ghrist 2008), "persistence diagrams" (Edelsbrunner et al. 2000), "persistent rank functions" (Robins & Turner 2016), and "persistence images" (Adams et al. 2017), all of which are graphical representations of the birth and death thresholds for each feature in a single point cloud. The CROCKER plot[7] introduced in Topaz et al. (2015) extends this to capture information about

[6] The green loop is in the image since gray pixels are connected at vertices.

[7] A pseudo-acronym for "contour realization of computed k-dimensional hole evolution in the Rips complex".

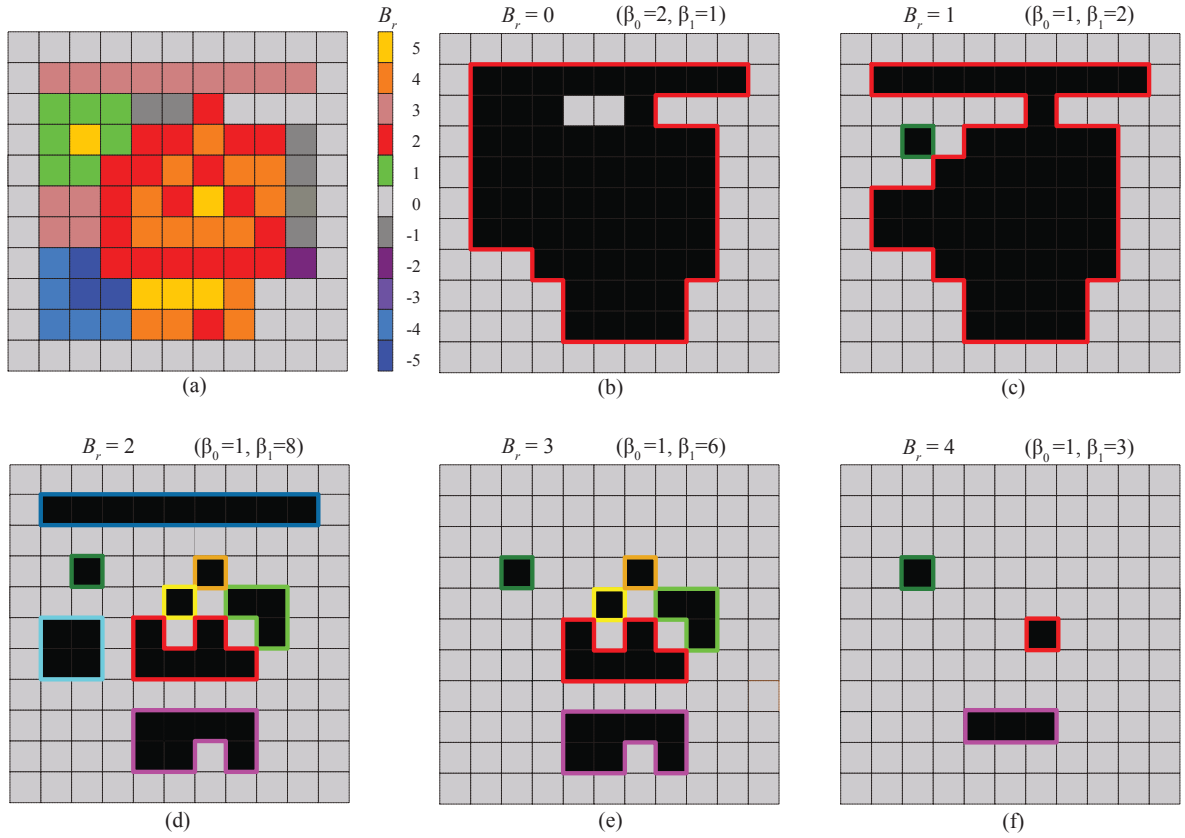

Fig. 2. Topological data analysis for the example "image" in panel (a). The gray pixels in panels b–f represent the cubical complex of the image for five sublevel thresholding values, $B_r = 0, 1, 2, 3$, and 4. For each complex, the (β_0, β_1) values are given. The colored loops represent the holes in the thresholded images.

the temporal evolution of the Betti numbers, which is important for flare prediction. Related methods are discussed in Xian et al. (2021). Figure 3 shows an example for two ARs. Here the ordinate is the threshold value, B_r, and the abscissa is time. The value shown at each (t, B_r) point is β_1 for the AR at the current time for the indicated threshold. The colors change logarithmically with β_1, as shown in the color bars.

In panel (a) of Fig. 3, for AR #6950, β_1 falls to zero (white color) at a relatively low threshold, $B_r \approx 1300$ G. This active region did not flare. By contrast for AR #7115 in panel (b), the complexity of the structure rapidly grows in days 5−6, so that $\beta_1 \gtrsim 10$ for thresholds up to $B_r \approx 2300$ G and β_1 is nonzero up to $B_r \approx 3500$ G during days 8−10. As shown in the panel, this AR had its first M-class flare early in day 7, nearly two days after the jump in β_1, and there is an X-class flare at the beginning of day 9. The topological structure appears to exhibit a strong and potentially predictive correlation with these flares.

The CROCKER plots in Fig. 3 show only the positive sublevel thresholds for B_r. We can also construct equivalent CROCKER plots for negative polarity shapes by choosing negative threshold values, though in this case it is appropriate to use "superlevel" instead of sublevel thresholding. Thus we would first include – for the threshold $B_r = 0$ – all pixels with $B_r \geq 0$, and then filter for increasingly negative values of B_r. For example, in Fig. 2a, this process will leave a hole surrounding the blue pixels for a superlevel threshold of $B_r = -3$.

The results suggest that CROCKER plots can be useful indicators of impending solar flares. To operationalize this in the context of machine learning, however, there is an additional challenge: ML models generally require data that have a fixed dimension, but B_r varies continuously. Various approaches to this "vectorization" problem have been proposed (Chazal et al. 2014; Bubenik 2015; Reininghaus et al. 2015; Kusano et al. 2016; Robins & Turner 2016; Adams et al. 2017; Carrière et al. 2017, 2020; Le & Yamada 2018). Here we choose 10 equally spaced thresholds with $B_r > 0$ for positive polarity thresholding and 10 thresholds with $B_r < 0$ for negative polarity thresholding, setting the maximum $|B_r| = 5000$ G[8]. The spacing of the chosen thresholds ensures a good balance between redundancy (that is, closely spaced, highly similar images with nearly identical β_1 counts) and adequate representation (avoiding a spacing so coarse as to miss important information). This process produces two 10-dimensional vectors: one for positive and one for negative B_r that give the number of "live" holes at each threshold. These 20 values make up the feature vector for the model comparison results presented in Sect. 5. Our preliminary experiments (Deshmukh et al. 2020) showed that the more complicated featurization methods such as persistence rank functions perform no better, for the purposes of this problem, than simpler approaches, indicating that the simple featurization effectively captures shape-based information that discriminates flaring and nonflaring magnetograms just as well as the more complicated approaches.

[8] Specifically the thresholds are {263 G, . . . , 4473 G, 5000 G} for sublevels and {−263 G, . . . , −4473 G, −5000 G} for superlevels.

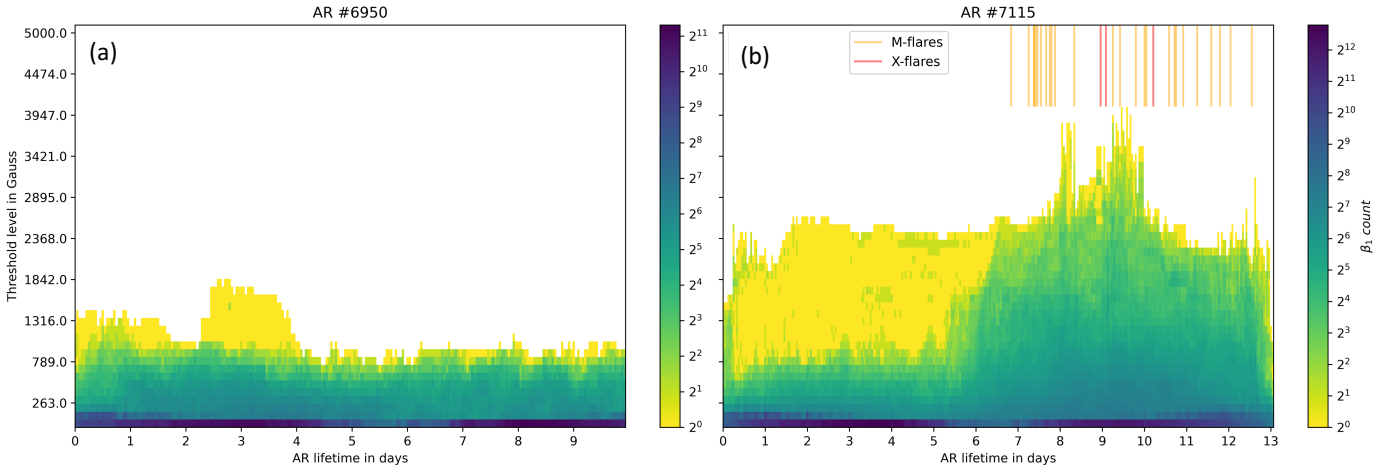

Fig. 3. Contours of β_1 as a function of time for sublevel thresholds $B_r = 0$ to 5000 G for two ARs. Panel (a) shows ten days of evolution for SHARP 6950 (NOAA AR 12636) where day 0 is 2300 UT on 14 February 2017. No flares were observed. Panel (b) shows thirteen days of evolution for SHARP 7115 (NOAA AR 12673) where day 0 is 0900 UT on 28 August 2017. In this case, the threshold for $\beta_1 \gtrsim 10$ jumps up about two days before the first M-class flare (indicated by orange vertical bar), and nearly four days before the first X-class eruption (red vertical bar) that occurred at 0857 UT on 6 September 2017. The B_r strength thresholds, $s_0 = 263$ G, $s_1 = 789$ G, ..., $s_8 = 4473$ G and $s_9 = 5000$ G, correspond to the tick marks on the ordinate. The breadth of the color bar scale is the result of the large number of small-scale holes from noise in the image.

4. Machine learning: models and framework

Evaluation and comparison of the predictive power of different ML methods and different feature sets require care. The first step, as described at the end of Sect. 2, involves creating a split of the data set into training, validation, and test sets that are representative of the problem at hand, but do not artificially boost the performance of the model. Secondly, it is important to include a range of ML models in the experimental framework to be able to claim whether one feature set is more useful than another. The models in this study, described in Sect. 4.1 below, were chosen to span ranges of complexity and strategies. Thirdly, ML methods have a number of hyperparameters that guide their training processes. Optimal values for these parameters depend on the model and the data, so a truly fair comparison requires individualized tuning. To that end, we use a k-fold cross-validation approach to automatically determine hyperparameters from the data as explained in Sect. 4.2. This systematic approach, which is in contrast to the trial-and-error grid search used in the majority of the flare-prediction literature, ensures a near-optimal selection of hyperparameters, especially those that are metric-sensitive and take on continuous values.

4.1. Machine-learning models

In our study, we use four different models: logistic regression, multilayer perceptrons, long short-term memories, and extremely randomized trees. In the following paragraphs, we give brief descriptions of these methods; for more details, please see (Murphy 2012; Goodfellow et al. 2016), or any other basic ML reference.

Logistic regression, perhaps the simplest of all models in the ML literature, uses a sigmoid function $h{:}\mathbb{R}^n \rightarrow [0, 1]$ to model the data:

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}. \quad (1)$$

The components of the vector $\boldsymbol{\theta}$ are the weights applied to each element of the input vector x. In the context of the flare-prediction problem, $h_\theta(x)$ represents the flaring probability, which we then convert to a categorical flare–no-flare output using a threshold of 0.5—if $h_\theta(x) < 0.5$ then x is classified as nonflaring, otherwise it is classified as flaring. Training this model is a matter of determining θ; we accomplish this using the LBFGS algorithm of Liu & Nocedal (1989). The only hyperparameter involved in this process is the class weight that is used in the gradient descent: the higher the weight for one class, the more the model is penalized for getting the classification wrong.

A multilayer perceptron or MLP is a type of feedforward artificial neural network containing multiple layers of nodes (neurons), with the outputs of each layer propagated forward to the next layer. These canonical ML models contain an input layer, some number of hidden layers, and an output layer. The output of each neuron is a nonlinear function of its inputs with a single free parameter, typically a multiplicative constant, that is "learned" during training using some optimization strategy (for example, gradient descent) on a suitable loss function: in this case, the weighted binary cross-entropy loss function, as described in Deshmukh et al. (2020). We employ the Adagrad optimizer (Duchi et al. 2011) to update the weights during gradient descent and an architecture with five dense layers containing 36, 24, 16, 8 and 2 nodes, respectively. This configuration was chosen using a manual grid search. There are two important hyperparameters here: the class weight, which plays a similar role as in the logistic regression model, and the L2 regularization constant in the loss function, which ensures that none of the neuron weights becomes too large, which would lead to overfitting.

Active regions evolve over time in ways that are meaningful from the standpoint of solar physics. Logistic regression and MLPs cannot leverage the information that is implicit in a sequence of magnetograms, as their forecasts are based on individual snapshots. For that reason, a thorough evaluation of ML-based solar flare forecasting should include methods that factor in the history of the observations. To that end, we use long-short term memories (LSTMs), which are designed to work with temporal sequences of data, using a feedback loop in a hidden layer that takes the state calculated in time step t_{n-1} and feeds it to the same network for the sample at time step t_n. The LSTMs used in our study contain the same number of hidden layers as the MLP discussed above, but with such a feedback loop incorporated in one of the hidden layers. This strategy for propagating

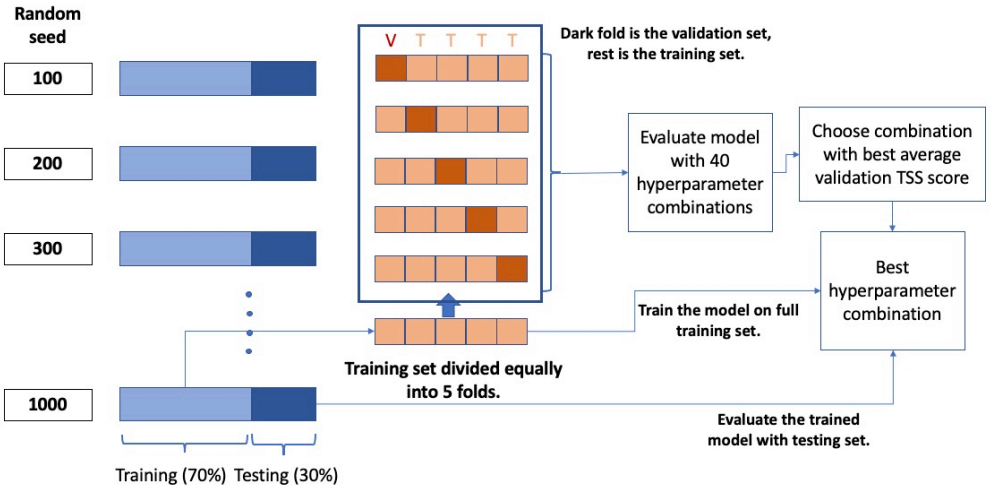

Fig. 4. Hyperparameter tuning workflow.

information forward in time is powerful, but it can complicate the training of these models. Simple gradient descent, for instance, can be problematic if there are long-term temporal dependencies in the data, since the gradient can decay as it is propagated through the time steps. To address this, LSTM nodes often incorporate mechanisms called "forget gates" that limit the number of steps through which information is propagated forward in time. This limit is an important choice, and one that is generally tuned by hand for a given model and data set. We use this approach and find that a sequence length of ten works well in our application. Like MLPs, LSTMs have two hyperparameters that significantly impact their performance: the weights in the binary cross-entropy loss function and the L2 regularization constant.

The fourth model used in our work is a type of decision tree, a class of ML models that have a tree-like structure, where each branch represents a decision based on some attribute of the data and the leaves correspond to the salient classes for the problem at hand (flaring and nonflaring, in our case). The input data dictate the path taken through the tree during the classification process, eventually routing the outcome to one of these classes. A major advantage of this strategy is that its results are an indication of how well each individual feature is able to divide the data set: a major step toward explainability, a critical challenge in modern AI. The main disadvantage of decision trees is their tendency for overfitting. One can mitigate this by building an ensemble of trees using a randomly chosen subset of features at each branch point – a so-called "Random Forest" – and use the mean or mode of their predictions to classify a sample. The extremely randomized tree (ERT; Geurts et al. 2006) that we use in this paper is a variant of this approach. These models have three hyperparameters: the class weight; the minimum impurity decrease, which controls when nodes will split; and the number of trees in the ensemble, which mitigates overfitting.

This set of choices is intended to cover a large portion of existing ML methods, in terms of architecture (for example, tree-based or not) and complexity (number of free parameters), making it a sensible evaluation set. We discuss tuning of these models next.

4.2. Hyperparameter tuning

Since the performance of an ML model depends on both the data and the training process, and because that training process is governed by the hyperparameters, a fair comparison of two different ML models requires careful tuning of their hyperparameters. To do this, one evaluates the performance of a model on a subset of the data, called the validation set, for different hyperparameter combinations. The best-performing combination is then used to train the model on the training set before it is evaluated on the corresponding test set. Using distinct subsets of the data for these three purposes ensures that the processes are completely independent.

Instead of using a single validation set, a better strategy is to perform what is called a k-fold cross-validation. In this approach, the training set is divided randomly into k subsets, or "folds", of roughly equal size. For each evaluation of hyperparameters, one of the k folds acts as the validation set and the remaining $k-1$ folds are merged to become the training set. After training, the model is then tested on the 1-fold validation set. This process is repeated using each of the k folds, individually, and the success of the hyperparameter combination is judged using the mean of the model's performance metric across these runs.

This process is illustrated in Fig. 4. The first step is the standard random splitting of the data set, as described at the end of Sect. 2. We repeat this ten times using a 70%−30% ratio to generate ten trial runs, splitting by AR as described in Sect. 2, then generate k folds on each of those training sets. In ML practice, k is generally chosen as 3, 5, or 10. In our case, $k = 5$ works well to produce sufficiently long training and validation sets. A second important decision in this tuning process is the sampling method for the hyperparameter value combinations. Various strategies have been proposed for this in the ML literature: grid search, random sampling, Bayesian sampling, etc. Here, we employ the Bayesian optimization method of Martinez-Cantin (2014), that uses Gaussian processes to mix exploration and exploitation in optimizing the hyperparameter combinations. We employ the Python implementation `BayesOptSearch` and use the `ray.tune` Python library (Liaw et al. 2018) to implement the sampling and evaluation process. As shown in Fig. 4, the `BayesOptSearch` method iteratively samples 40 hyperparameter combinations, choosing the one that gives the best performance on the validation set, as measured by some metric that compares the forecast to the ground truth. (We use the true skill statistic, as described further in Sect. 5). For `BayesOptSearch`, we use the upper confidence bound as the acquisition function, setting the exploration parameter $\kappa = 5$ in order to balance exploitation and exploration in the sampling process; see Snoek et al. (2012) for details. That sampling process begins

with 10 uniformly chosen hyperparameter samples, continuing in steps of 10 to explore 40 hyperparameter combinations[9]. The model is then trained on the full training set using the best-performing hyperparameter values before being evaluated on the 30% test set.

Not all hyperparameters require this kind of complex, computationally intensive treatment; some of them can be effectively tuned using a manual grid search. For those, we employ a two-phase model-tuning strategy, first performing a hand optimization of hyperparameters such as the number of model layers, the nonlinear activation function, and the optimization function. Values for these hyperparameters, which are chosen from among a finite set of options, generally impact all metrics in a similar way. This is followed by the automated approach described above for tuning hyperparameters such as the loss function weights or the regularization penalty. Automatic tuning of these parameters is important for two reasons. Firstly, they take on continuous values, so hand-tuning becomes cumbersome. Secondly, their effects are not independent: changing one often improves one metric at the cost of another – a situation where an automated, systematic exploration of the search space can be especially appropriate. It is entirely possible, of course, to use the automated approach for all the hyperparameters, but that significantly increases the computation time.

Hyperparameter tuning is, as should be clear from the details above, a complex process. We offer these details here not only so that others can not only fully reproduce these results, but also use this framework in other applications that involve evaluation and comparison of flare-forecasting approaches. It is also worth mentioning that the metric that one uses to evaluate performance will affect the process, and in subtle ways: optimizing for pure accuracy will produce different results than balancing false positives against false negatives. This matter is discussed further below.

5. Results

In this section, we compare the relative prediction performance of the ML based flare-forecasting models described in Sect. 4 for the three feature sets covered in Sect. 3. Using the labeled data set described in Sect. 2, we carry out the hyperparameter tuning procedure outlined in the previous section on each model and feature-set combination, using the k-fold cross validation approach on the ten data sets, then train it with optimized hyperparameters on the corresponding training set. To evaluate the results, we run the model on the corresponding test set and compare its 24-h forecasts to the ground truth using four standard prediction metrics: accuracy; the true skill statistic or TSS (also known as the H&KSS), the Heidke skill score (HSS_2)[10], frequency bias (Bias), and F_1, which is the harmonic mean of precision and recall. These metrics, whose detailed formulae can be found in Jolliffe & Stephenson (2012), are derived from the entries of the contingency table – that is, the numbers of true and false positives and true and false negatives. In the context of this problem, a flaring magnetogram (that is, a magnetogram with an associated M1.0 or larger flare in the next 24 h) is considered as a positive while a nonflaring magnetogram is considered a negative. Since our data set includes 5769 of the former and 447 504 of the latter, accuracy is not a very useful metric here; a simple model that classified every input as nonflaring would be 98.7% accurate. The skill scores strike various balances between correctly forecasting the positive and negative samples. TSS ranges from $[-1, 1]$ and HSS_2 from $(-\infty, 1]$. In both cases, these scores are 1 when there are no false positives or false negatives, while a score of 0 means the model is doing only as well as a random forecast, essentially, an "always no-flare" forecast. Bias has the range $[0, \infty]$, where $\mathrm{Bias} < 1$ indicates under-forecasting (many false negatives), and $\mathrm{Bias} > 1$ implies over-forecasting (many false positives). F_1 has the range $[0, 1]$ with 1 indicating a perfect forecast score. These metrics, all of which are used broadly in the flare-prediction literature, span the various methods used to quantify the performance of ML models.

As discussed in Sect. 4, hyperparameters for each model must be individually tuned in order to provide a fair comparison of their performance. The choice of metric plays a subtle role here, since hyperparameters can have different effects on the various metrics, that is, tuning performance based on values of one can impact performance as measured by the others. The choice of metric is often left as a decision for the forecaster: some might wish to prioritize the TSS score, for example Deshmukh et al. (2021), while others might prefer a forecast that has lower false positive rate (Deshmukh et al. 2022) or is more reliable (Nishizuka et al. 2021). In the problem treated here, where the data set is highly imbalanced, an optimization based on accuracy would be a particularly bad choice, as it would lead to models defaulting to the always no-flare forecast. Here we use TSS, choosing hyperparameters that maximize its values via the k-fold cross-validation described above. TSS is a common choice in the flare-forecasting literature, as well as in the broader ML literature. It does come with limitations, however: optimizing the TSS score can lead to high false positives, thereby impacting some of the other metrics like precision, F_1, and Bias. This effect manifests in the results described below.

The experiments were carried out on an NVIDIA Titan RTX (24 GB, 33 MHz) GPU for the deep-learning models (MLP and LSTM), and on an Intel i9-9280X (3.30 GHz) CPU for the simpler models (logistic regression and ERT)[11]. Run times ranged from 2.5 s to train and 0.02 s to test each logistic regression model on the SHARPs feature set to 210 s and 9 s for the LSTM model using the combined feature set. The run time of the hyperparameter tuning procedure ranged from 40 s for each logistic regression model with the SHARPs feature set to just under two hours for each LSTM model with the combined feature set.

Table 2 compares the performance of the various models for the three feature sets: the traditional physics-based features that appear in the SHARPs metadata (Sect. 3.1), the shape-based attributes extracted from each magnetogram image using topological data analysis (Sect. 3.2), and a third set that combines the two.

A between-model comparison addresses the first research question posed in Sect. 1: whether model complexity is an advantage in the context of this problem. Table 2 suggests that the answer is no. Indeed, the general trend in the TSS scores shows that increased complexity slightly reduces performance. Indeed, even though the MLP and LSTM models have orders of magnitude more parameters, the simpler LR and ERT models perform marginally better, as judged by the TSS scores. (For the other metrics, there is no significant variation among the four models). This may simply be due to data limitations; recall that the higher the complexity of a ML model, the more

[9] That is, using `initial_random_steps` in `ray.tune`, with `max_concurrent_trials` = 10.

[10] We use the second common definition of the HSS score, denoted by HSS_2 as defined in Barnes et al. (2016, Eq. (6)).

[11] Different ML models lend themselves to different types of hardware, depending on how well they parallelize. The machine used to carry out these experiments affects only the run time, not the results.

Table 2. 24-h forecast performance of four ML models using three feature sets.

Model	Feature set	Param.	Accuracy	TSS	HSS_2	F_1	Bias
LR	SHARPs	21	0.87 ± 0.01	0.79 ± 0.01	0.13 ± 0.02	0.15 ± 0.02	11.44 ± 1.75
	Topological	21	0.87 ± 0.01	0.78 ± 0.02	0.12 ± 0.02	0.14 ± 0.02	11.88 ± 1.40
	Combined	41	0.87 ± 0.02	0.79 ± 0.02	0.13 ± 0.02	0.15 ± 0.02	11.76 ± 1.98
ERT	SHARPs	332	0.84 ± 0.01	0.79 ± 0.01	0.11 ± 0.01	0.13 ± 0.01	13.96 ± 0.81
	Topological	483	0.85 ± 0.02	0.76 ± 0.04	0.11 ± 0.02	0.13 ± 0.02	13.34 ± 1.93
	Combined	340	0.86 ± 0.02	0.77 ± 0.03	0.12 ± 0.02	0.14 ± 0.02	12.27 ± 2.12
MLP	SHARPs	2198	0.85 ± 0.02	0.76 ± 0.02	0.11 ± 0.02	0.13 ± 0.02	13.13 ± 2.42
	Topological	2198	0.85 ± 0.02	0.76 ± 0.02	0.11 ± 0.01	0.13 ± 0.01	13.56 ± 1.53
	Combined	2918	0.86 ± 0.03	0.76 ± 0.03	0.11 ± 0.02	0.13 ± 0.02	13.10 ± 1.78
LSTM	SHARPs	6662	0.87 ± 0.02	0.75 ± 0.02	0.12 ± 0.02	0.14 ± 0.02	11.90 ± 1.93
	Topological	6662	0.85 ± 0.02	0.75 ± 0.03	0.11 ± 0.02	0.13 ± 0.02	13.28 ± 2.11
	Combined	7382	0.86 ± 0.01	0.75 ± 0.02	0.12 ± 0.01	0.14 ± 0.01	12.00 ± 1.63

Notes. The third column shows the number of free parameters needed to classify a single data sample. For the ERT, this equals the average depth of the tree (the path taken by a data sample from the root to a leaf node in the tree). The listed metric score and error bars are determined by computing the sample mean and standard deviation of the scores from 10 experiments performed on 10 different data splits.

data is needed for training. If the training set is too small, the model will over-fit that data, causing it to fail to generalize well to the testing set. Simpler models avoid this trap. It is important to note that while a set of 460 000 samples might seem large, many of the images in the solar flare data set are similar, and almost all are nonflaring. Another possible hypothesis for these observations, as asserted in Florios et al. (2018), is that magnetogram data is not sufficient for improving the model performance beyond a certain limit by increasing the model complexity. In other words, the total amount of information – that is, the number of sufficiently diverse and useful samples – in this data set is inadequate for the purposes of the discussed ML models.

In addition, Table 2 shows that the topological features perform just as well as the SHARPs features. Moreover, the combined feature set does not provide any improvement, indicating that neither feature set provides a significant advantage over the other. This is an answer to our second research question. Abstract spatial properties of active region magnetograms–calculated using abstract universal algorithms that quantify shape from raw image data without any assumptions about the underlying mechanics – appear to impart equal skill to ML flare-forecasting models as the set of physics-based attributes developed by solar flare experts.

Determining whether data limitations are important is an open problem in current ML research. One way to approach it is to compare the values of the weighted binary cross-entropy loss function across the models; another is to observe the patterns in the convergence over the training process. Both are problematic for the more complex models in our study. ERTs do not generate a loss, nor do they have an iterative training process. For models that do have an iterative training procedure (MLP and LSTM), we carried out the second test, and found that both the validation and training losses reached asymptotes during the training process, suggesting – but of course not proving – that overfitting is not at issue. The first approach is not useful in the case of the LSTM because one-to-one loss comparisons are problematic in such models due to the variation in the number of samples in the training and testing sets that occurs when the original data are converted to temporal sequences by the feedback loop in the model. In the future, as more data is recorded by SDO/HMI, we hope to learn, by rerunning these experiments on longer, richer data sets, whether the effects described in this paragraph are artifacts of data limitations or something more fundamental.

In regard to optimization of models, optimizing TSS for this highly imbalanced data set leads to over-forecasting across all configurations, generating poor results for metrics such as F_1, HSS and Bias. Indeed, optimizing on different metrics gives different results. As an additional experiment, we regenerated Table 2 by optimizing each configuration on F_1 instead of TSS. Doing so reduced the over-forecasting, improving all of the metrics except for TSS, which decreased. The overall effect of this modification was due to a reduction in false positives, at the cost of a similar reduction in true positives. This, however, could be desirable for applications that have a high cost associated with the former. A more detailed discussion of optimizing metrics to achieve a lower false positive rate can be found in Deshmukh et al. (2022).

To assess the effects of the prediction horizon, we carried out a set of experiments using the best-performing model and feature combination from Table 2 (logistic regression with the SHARPs feature set) and generated forecasts for 3, 6, and 12 h in addition to our previous 24 h case. The results are shown in Table 3. As one would expect, the shorter the forecast window, the higher the accuracy, but the story for the other metrics is more complicated: TSS is roughly similar for all forecasting windows, while HSS_2, F_1, and Bias actually worsen as the forecast window shrinks. This counter-intuitive result is almost certainly due to the increasing fraction of negative samples: for each flare, fewer magnetograms will be labeled as "flaring within m hours" if m is smaller. This is, again, a side effect of tuning on the TSS score, as discussed above: its optimization leads to a high false positive rate for a severely imbalanced data set.

The LSTM is not only the most complex model in this study – by a factor of three, as judged by the number of free parameters – but also the only one that uses the AR history. In view of this, its lack of performance is particularly striking. The feedback loop in its architecture makes it difficult to deconvolve the effects of the temporal history and the number of parameters, so we cannot say for sure whether or not the former, alone, confers any advantage. Nevertheless, the additional complexity certainly does not appear to help.

Table 3. Logistic regression forecasts for different forecasting windows, measured in hours.

Forecasting window	Accuracy	TSS	HSS_2	F_1	Bias
24	0.87 ± 0.01	0.79 ± 0.01	0.13 ± 0.02	0.15 ± 0.02	11.44 ± 1.75
12	0.89 ± 0.02	0.82 ± 0.01	0.10 ± 0.03	0.11 ± 0.03	17.13 ± 4.10
6	0.91 ± 0.01	0.82 ± 0.02	0.07 ± 0.02	0.07 ± 0.02	25.17 ± 4.83
3	0.93 ± 0.01	0.79 ± 0.05	0.05 ± 0.01	0.05 ± 0.01	33.18 ± 4.60

Notes. Values and error bars are calculated as in Table 2.

Our final investigation concerns dimensionality reduction of the feature sets. To explore this, we perform principal component analysis (PCA) on the data sets (see Appendix A for the details), and then retune the model hyperparameters and repeat the train and test procedure using only nine, three, and eleven PCA vectors for the traditional, topological and combined feature sets respectively: so chosen to cover 98.5% of the variance in the three data sets, respectively. The number of PCA vectors for the SHARPs data set is in accord with the findings of Chen et al. (2019). What is more striking is that the topological data set can be reduced to only three features, indicating that the information in that feature set is highly anisotropic and suggesting that dimensional reduction has the potential to significantly reduce model complexity.

Figure 5 compares the TSS scores for the original data sets and the PCA-reduced data sets. For all three feature sets, the performance of the full and reduced-order models are similar for the logistic regression, MLP, and LSTM models. This extends to the other metrics as well (not shown). In other words, we can successfully simplify these ML models by reducing the number of features without sacrificing predictive skill. This is an obvious advantage as models that work with smaller feature sets have fewer free parameters that must be learned from the same training data. It is important to note that this advantage is highest for the topological feature set, which contains only three features. That is, by training a logistic regression model on only three features, we are able to achieve a simple model for solar flare prediction without sacrificing performance.

The ERT model, however, departs from this pattern, demonstrating a marked reduction in TSS scores for the PCA-reduced feature set. This effect is strongest for the topological feature set, followed by the SHARPs feature set, with the combined feature set seeing the smallest impact. A possible explanation for this performance degradation lies in the way ERTs are constructed. In a k-feature ERT, $\sqrt{k}$ randomly chosen features are typically used to determine the split at each branch point. A reduction in the number of features, then, confines the exploration of the ERT parameter space. This could result in branches that do not effectively separate the positive and negative samples, thereby impacting the model performance. Modifying the training process to use all k features for choosing the best split can ameliorate this problem. Indeed after doing this, we find that there is no statistically significant difference between the reduced and the full feature sets for the TSS scores of the topological and combined feature sets. However, the SHARPs feature set does not show a similar improvement upon increasing the number of features used in the split. This disparity suggests that the presence of shape-based features in the combined set helps make up for the shortcomings of the physics-based features. As shown in Appendix A, both SHARPs and topological features are highly weighted in the first principal component of the combined feature set.

6. Conclusions

Machine learning-based solar flare prediction has been a topic of interest to the space weather community for some time. Various ML models, ranging from simple ones like logistic regression to incredibly complex "deep-learning" models such as multilayer perceptrons (MLPs) and long short-term memories (LSTMs), have been used to map the correlation between physics-based magnetic field features and near-term flaring probability. There have been studies that compared different ML flare-forecasting models (Barnes et al. 2016; Nishizuka et al. 2017; Florios et al. 2018), but the models in those studies only used physics-based features derived from photospheric magnetograms. Furthermore, none of these approaches performed a systematic, automated hyperparameter tuning process to assure a fair comparison between models (Florios et al. 2018; Sinha et al. 2022 performed simple manual hyperparameter tuning).

Our first objective in this paper was to evaluate a new shape-based feature set, introduced in Deshmukh et al. (2020), composed of quantities extracted from magnetograms using topological data analysis. The results from these features are compared to the standard physics-based SHARPs feature set. Our results extend previous pilot studies (Deshmukh et al. 2020, 2021; Knyazeva et al. 2017) by using a more comprehensive SHARPs feature set, employing four different ML models, and using a systematic hyperparameter tuning methodology. This broader and deeper study confirmed that the shape-based features – calculated using abstract and universal algorithms without any assumptions about the underlying physics – impart just as much skill to ML flare prediction models as the set of traditional, heuristically derived physics-based attributes. Further, the combination of the shape-based and physics-based feature sets does not provide any improvement over the individual sets, confirming that neither set provides a significant advantage over the other. Using principal component analysis to find relevant subspaces of the feature sets, we studied the effect of dimensionality reduction on model performance. We found that the topological feature set afforded the largest dimensionality reduction, and that the PCA-reduced data sets performed just as well as the original feature sets for three of the four ML models. In a data-limited situation, this is a major advantage for the effectiveness of ML based solar-flare prediction methods, since more complex models generally require larger training sets.

In performing the feature set evaluations, we systematically compared a set of four ML models operating on feature vectors with increasing complexity to determine whether higher model complexity is correlated with higher predictive skill. Using an automated hyperparameter tuning approach to assure a fair comparison, we showed that increasing the complexity of the model did not improve the predictive skill of models developed for the M-class and above 24-h flare forecasting problem. It should be noted that we did not include convolutional neural networks as part of this study. CNNs are a form of automated shape-based

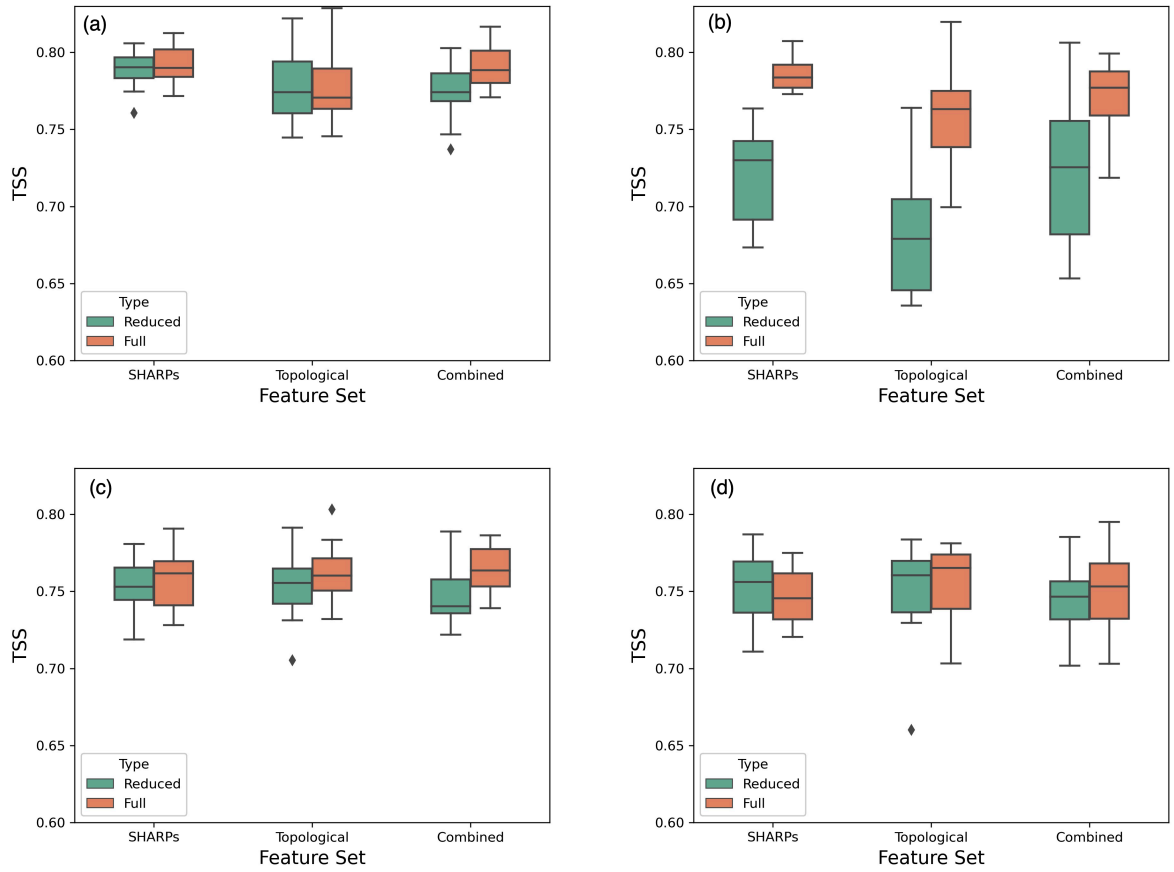

Fig. 5. TSS score comparison in the form of box-whiskers plots for four ML models. The models are trained on the three feature sets and their PCA reduced counterparts over ten trials: (a) Logistic Regression model; (b) ERT model; (c) MLP model; (d) LSTM model. The central line each box represents the median TSS score while the top and bottom edge of the boxes represent the 25 and 75 percentiles over the ten trials. The whiskers are the upper and lower bounds for the scores in each experiment set to be 1.5 times the interquartile range, and the dots represent the outliers defined as points outside of the specified whisker length. After PCA reduction, the SHARPs feature set is reduced from 20 to 9 features, the topological set from 20 to 3 features, and the combined set from 40 to 11 features. For all models except for the ERT, the reduced feature set does just as well as the complete feature set.

image analysis that are commonly used in large-scale classification problems; see, for example, Krizhevsky et al. (2017). It is difficult to directly compare the performance of CNN-based models to models operating on feature vectors; the level of complexity in these models, particularly for recurrent versions, is generally much higher than for the models used in this study and the computational time required for training and hyperparameter tuning is much longer. A separate comparative study of image-based flare prediction ML models, for example, CNN- vs. attention- vs. transformer-based models, would be interesting to conduct. Preliminary results with a hybrid CNN-ERT model analyzing HMI magnetograms for flare prediction is demonstrated in Deshmukh et al. (2022). In the future, we plan to develop image time series analysis models operating on both SDO/HMI magnetogram data and SDO/Atmospheric Imaging Assembly (AIA) coronal and chromospheric image data.

Acknowledgements. V. Deshmukh led the research resulting in this paper during his Ph.D. program in Computer Science at the University of Colorado at Boulder. The PCA analysis was performed by S. Baskar. This study was funded by a grant from the National Science Foundation (Grant No. AGS 2001670) and by a grant from the NASA Space Weather Science Applications Program (Grant No. 80NSSC20K1404). T.E. Berger was supported in part by a Grand Challenge grant from the University of Colorado at Boulder. We thank the anonymous reviewer for comments and suggestions that improved this paper significantly.

References

Abed, A. K., Qahwaji, R., & Abed, A. 2021, Adv. Space Res., 67, 2544
Adams, H., Emerson, T., Kirby, M., et al. 2017, J. Mach. Learn. Res., 18, 218
Barnes, G., Leka, K., Schrijver, C., et al. 2016, ApJ, 829, 89
Bobra, M., & Couvidat, S. 2015, ApJ, 798, 135
Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, Sol. Phys., 289, 3549
Bubenik, P. 2015, J. Mach. Learn. Res., 16, 77
Campi, C., Benvenuto, F., Massone, A. M., et al. 2019, ApJ, 883, 150
Carlsson, G. 2009, Bull. Am. Math. Soc., 46, 255
Carrière, M., Chazal, F., Ike, Y., et al. 2020, in Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics, eds. S. Chiappa, & R. Calandra (PMLR), Proc. Mach. Learn. Res., 108, 2786
Carrière, M., Cuturi, M., & Oudot, S. 2017, in Proceedings of the 34th International Conference on Machine Learning, eds. D. Precup, & Y. W. Teh (PMLR), Proc. Mach. Learn. Res., 70, 664
Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., & Wasserman, L. 2014, Proceedings of the Thirtieth Annual Symposium on Computational Geometry, SOCG'14 (New York: ACM), 474:474
Chen, Y., Manchester, W. B., Hero, A. O., et al. 2019, Space Weather, 17, 1404
Crown, M. D. 2012, Space Weather, 10, 6006
Davies, A., Velickovic, P., Buesing, L., et al. 2021, Nature, 600, 70
de Silva, V., & Ghrist, R. 2007, Algebr. Geom. Topol., 7, 339
Deshmukh, V., Berger, T., Bradley, E., & Meiss, J. 2020, J. Space Weather Space Clim., 10, 13
Deshmukh, V., Berger, T., Meiss, J. D., & Bradley, E. 2021, Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2–9, 2021 (AAAI Press), 15293

Deshmukh, V., Flyer, N., van der Sande, K., & Berger, T. 2022, ApJS, 260, 9
Duchi, J., Hazan, E., & Singer, Y. 2011, J. Mach. Learn. Res., 12, 2121
Edelsbrunner, H., Letscher, D., & Zomorodian, A. 2000, Proceedings 41st Annual Symposium on Foundations of Computer Science (IEEE), 454
Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, Sol. Phys., 293, 28
Geurts, P., Ernst, D., & Wehenkel, L. 2006, Mach. Learn., 63, 3
Ghrist, R. 2008, Bull. Am. Math. Soc., 45, 61
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (The MIT Press)
Hale, G., Ellerman, F., Nicholson, S., & Joy, A. H. 1919, ApJ, 49, 153
Huang, X., Wang, H., Xu, L., et al. 2018, ApJ, 856, 7
Jolliffe, I., & Stephenson, D. 2012, Forecast Verification: A Practitioner's Guide in Atmospheric Science, 2nd edn. (Wiley)
Kaczynski, T., Mischaikow, K., & Mrozek, M. 2004, Computational Homology (New York: Springer-Verlag)
Knyazeva, I. S., Urtiev, F. A., & Makarenko, N. G. 2017, Geomagn. Aeron., 57, 1086
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2017, Commun. ACM, 60, 84
Kusano, G., Hiraoka, Y., & Fukumizu, K. 2016, in Proceedings of The 33rd International Conference on Machine Learning, eds. M. F. Balcan, & K. Q. Weinberger (New York: PMLR), Proc. Mach. Learn. Res., 48, 2004
Le, T., & Yamada, M. 2018, in Advances in Neural Information Processing Systems, eds. S. Bengio, H. Wallach, H. Larochelle, et al. (Curran Associates, Inc.), 31
Leka, K., & Barnes, G. 2007, ApJ, 656, 1173
Leka, K., Park, S.-H., Kusano, K., et al. 2019a, ApJ, 243, 36
Leka, K., Park, S.-H., Kusano, K., et al. 2019b, ApJ, 881, 101
Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, ApJ, 891, 10
Liaw, R., Liang, E., Nishihara, R., et al. 2018, ArXiv e-prints [arXiv:1807.05118]
Liu, D. C., & Nocedal, J. 1989, Math. Progr., 45, 503
Martinez-Cantin, R. 2014, J. Mach. Learn. Res., 15, 3735
McInnes, L., Healy, J., Saul, N., & Großberger, L. 2018, J. Open Source Softw., 3, 861
McIntosh, P. S. 1990, Sol. Phys., 125, 251
Murphy, K. P. 2012, Machine Learning: A Probabilistic Perspective (The MIT Press)
Nandy, D., Mackay, D. H., Canfield, R. C., & Martens, P. 2008, J. Atmos. Solar-Terres. Phys., 70, 605
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, ApJ, 835, 156
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2018, ApJ, 858, 113
Nishizuka, N., Kubo, Y., Sugiura, K., Den, M., & Ishii, M. 2021, Earth Planets Space, 73, 1
Park, E., Moon, Y.-J., Shin, S., et al. 2018, ApJ, 869, 91
Reininghaus, J., Huber, S., Bauer, U., & Kwitt, R. 2015, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4741
Robins, V., & Turner, K. 2016, Phys. D: Nonlinear Phenom., 334, 99
Scherrer, P. H., Schou, J., Bush, R. I., et al. 2012, Sol. Phys., 275, 207
Schrijver, C. J. 2007, ApJ, 655, L117
Singh, G., Memoli, F., & Carlsson, G. 2007, in Eurographics Symposium on Point-Based Graphics, eds. M. Botsch, R. Pajarola, B. Chen, & M. Zwicker (The Eurographics Association), 91
Sinha, S., Gupta, O., Singh, V., et al. 2022, ApJ, 935, 45
Snoek, J., Larochelle, H., & Adams, R. P. 2012, in Advances in Neural Information Processing Systems, eds. F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Curran Associates, Inc.), 25
Snyder, J. P. 1987, Map Projections–A Working Manual (US Government Printing Office), 1395
Topaz, C. M., Ziegelmeier, L., & Halverson, T. 2015, PLoS One, 10, 1
Van der Maaten, L., & Hinton, G. 2008, J. Mach. Learn. Res., 9, 2579
Xian, L., Adams, H., Topaz, C., & Ziegelmeier, L. 2021, Found. Data Sci., 4, 1
Xu, X., Cisewski-Kehe, J., Green, S. B., & Nagai, D. 2019, Astron. Comput., 27, 34
Yuan, Y., Shih, F. Y., Jing, J., et al. 2010, Proc. Int. Astron. Union, 6, 446
Zheng, Y., Li, X., & Wang, X. 2019, ApJ, 885, 73
Zheng, Y., Li, X., Si, Y., Qin, W., & Tian, H. 2021, MNRAS, 507, 3519

Appendix A: Training set dimensionality reduction

As part of the training process, ML models learn which attributes of the input data, in which combinations, are meaningful. Their efficiency in doing so depends on the amount and complexity of the data, and also–importantly–on the way it is represented. In general, a larger feature set leads to increased model complexity, which in turn can result in overfitting and increased computational time (Goodfellow et al. 2016), as well as requiring larger training data sets. The nature of the individual features also matters. Features that are not salient, or that are redundant, hinder the learning process. For these reasons, it is important to minimize the number of features and maximize their relevance.

A good way to approach this problem is to apply dimensionality-reduction techniques to the feature space in order to find the most relevant subspaces. A variety of methods have been proposed for this, including principal component analysis (PCA), linear discriminant analysis, t-distributed stochastic neighbor embedding (Van der Maaten & Hinton 2008), uniform manifold approximation and projection (McInnes et al. 2018), etc. We use PCA here because of its simplicity and effectiveness. It determines an alternative basis to represent the data by iteratively constructing an orthogonal basis such that the variance of the data along the first dimension is maximal, the second dimension is in the direction of maximum variance orthogonal to the first, and so on. One then typically keeps the first l, say, principal vectors that together account for some chosen fraction of the total variance. This approach, applied to an n-dimensional feature set, effectively reduces the dimensionality to l. It should be noted that each of the l basis vectors may contain all of the original n values; we discuss this more below.

Applying this approach to the ten randomly shuffled training sets discussed in Sect. 2–first using the SHARPs feature set of Sect. 3.1, then the shape-based feature set of Sect. 3.2, and finally their combination–gives Fig. A.1. For the SHARPs feature set, the first principal component captures 38% of the variance. As the figure shows, adding a second principal component increases the total to 60%. The topological feature set is much more anisotropic: its first principal component captures almost 85% of the variance of the data set, a level that requires four principal components in the SHARPs feature set. We apply a threshold of 98.5% (the dashed line in Fig. A.1) to select nine, three, and eleven dimensions from the orthogonal PCA representation for the reduced-dimension versions of the traditional, topological, and combined feature sets, respectively.

A detailed analysis of the reduced features is revealing. The coefficients of the first principal component–the weights of each SHARPs attribute in the basis vector–for one of the ten training sets are shown in Fig. A.2(a). These results are to some extent consistent with the science: the `R_VALUE` and `USFLUX` attributes, which are known to be important for flare prediction, are weighted heavily in the first principal component of the SHARPs set. Note, though, that six other quantities are also weighted heavily in Fig. A.2(a); moreover, this first principal component does not capture much of the variance, so one should not over-interpret its weights. Rather, one must think about all of the principal components, acting together as a basis for the new feature space. (A feature that is totally absent from every principal component, of course, is certainly not salient, but a low-weighted feature in the first one could have a high weight in another). That being said, interpretation of Fig. A.2(b) is somewhat different because, as Fig. A.1 shows, the first principal component in the topological feature set captures a very large fraction of the variance. It is interesting to note that the topological features at lower magnetic fluxes are weighted more heavily in the first principal component, and with some symmetry across positive and negative fluxes. This makes sense in view of the nature of the threshold construction: a high threshold value removes much of the small-scale structure of the magnetogram, leaving only the highest-magnitude pixels. Finally, it is important to note that the first principal component from the combined feature set is a weighted combination of both SHARPs and TDA features. As we discuss in Sect. 5, this can have implications regarding model performance.

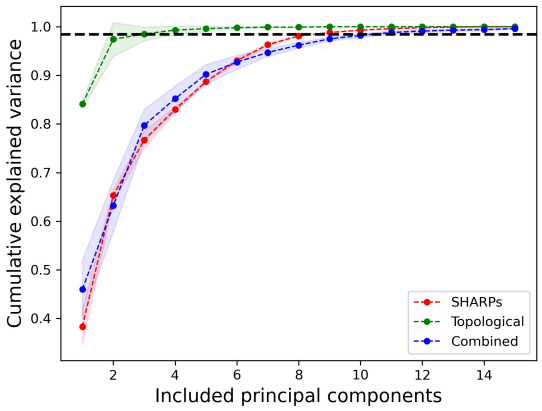

Fig. A.1. Cumulative explained variance plots. These are for the principal components for the three feature sets, determined from the ten training sets. The darker curves represent the medians of the explained variance, while the shaded regions around them indicate standard deviations. The dashed line marks the 98.5% level.

Of course, PCA only finds a new basis set for the feature space; it does not produce a feature ranking. The previous paragraph only explains how one principal component is represented in terms of the original features, in one particular split of the data. Even though this gives possible ties to the physics, it does not represent a comprehensive, formal analysis of the importance of a given feature in the classification problem.

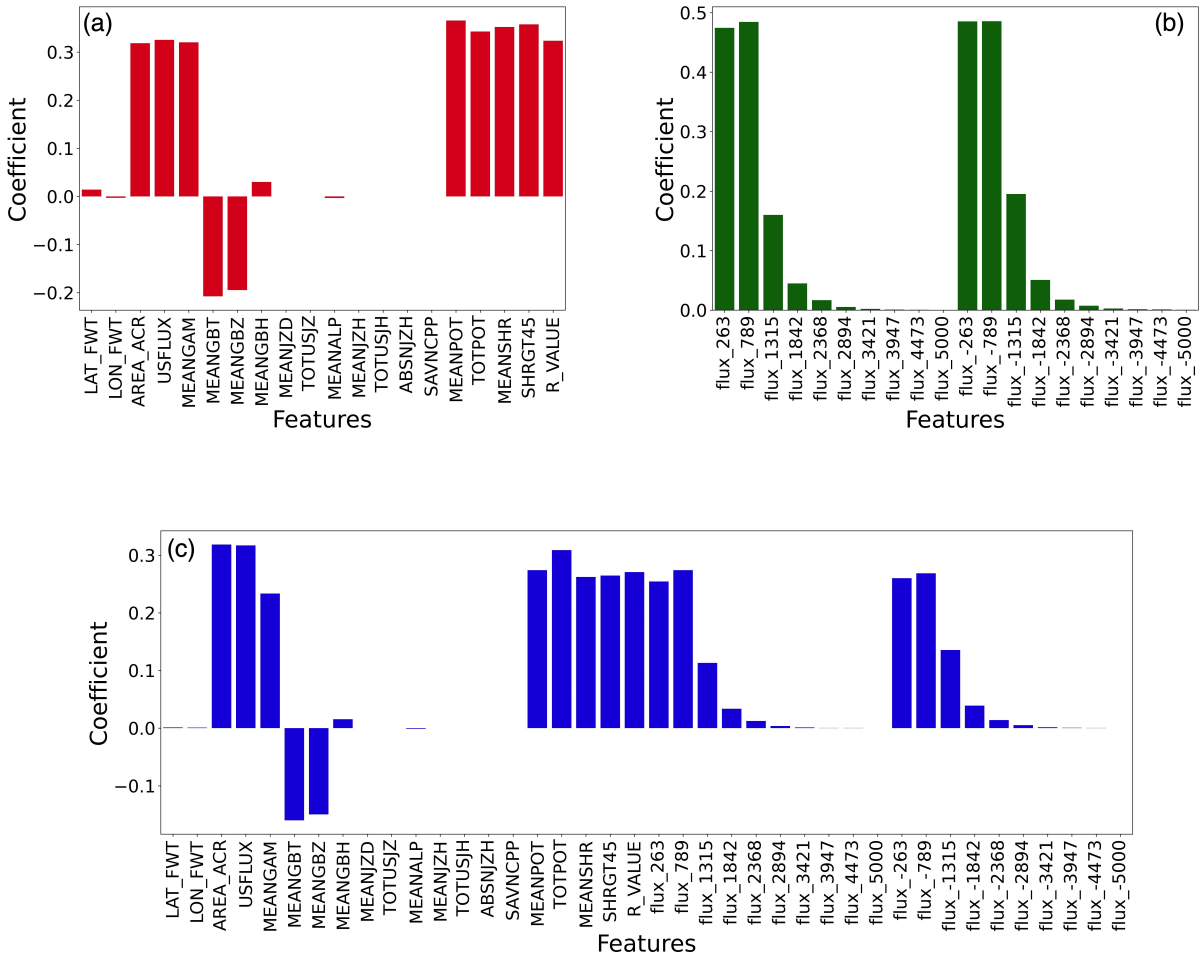

Fig. A.2. Weights of the features. (a) SHARPs, (b) topological, and (c) combined features in the first principal component of the corresponding feature set for one of the ten training sets examined in this paper. Labels of the topological features indicate the flux level of the threshold value used to construct those features.