



A Comparative Analysis of Machine-learning Models for Solar Flare Forecasting: Identifying High-performing Active Region Flare Indicators

Suvadip Sinha¹ , Om Gupta^{1,2} , Vishal Singh^{1,2} , B. Lekshmi^{1,3} , Dibyendu Nandy^{1,2} , Dhruvadya Mitra⁴ , Saikat Chatterjee⁵ , Sourangshu Bhattacharya⁶ , Saptarshi Chatterjee⁶ , Nandita Srivastava^{1,7} , Axel Brandenburg^{4,8} , and Sanchita Pal^{1,9}

¹ Center of Excellence in Space Sciences India, Indian Institute of Science Education and Research Kolkata, Mohanpur 741246, West Bengal, India
dnandi@iiserkol.ac.in

² Department of Physical Sciences, Indian Institute of Science Education and Research Kolkata, Mohanpur 741246, West Bengal, India

³ Max Planck Institute for Solar System Research, Göttingen, D-37077 Germany

⁴ Nordita, KTH Royal Institute of Technology and Stockholm University, Hannes Alfvéns väg 12, SE-10691 Stockholm, Sweden

⁵ KTH Royal Institute of Technology, Stockholm, Sweden

⁶ Indian Institute of Technology, Kharagpur, India

⁷ Udaipur Solar Observatory, Physical Research Laboratory, P.O. Box 198, Badi Road, Udaipur 313001, India

⁸ The Oskar Klein Centre, Department of Astronomy, Stockholm University, AlbaNova, SE-10691 Stockholm, Sweden

⁹ Department of Physics, University of Helsinki, P.O. Box 64, FI-00014 Helsinki, Finland

Received 2021 December 22; revised 2022 May 30; accepted 2022 June 14; published 2022 August 12

Abstract

Solar flares create adverse space weather impacting space- and Earth-based technologies. However, the difficulty of forecasting flares, and by extension severe space weather, is accentuated by the lack of any unique flare trigger or a single physical pathway. Studies indicate that multiple physical properties contribute to active region flare potential, compounding the challenge. Recent developments in machine learning (ML) have enabled analysis of higher-dimensional data leading to increasingly better flare forecasting techniques. However, consensus on high-performing flare predictors remains elusive. In the most comprehensive study to date, we conduct a comparative analysis of four popular ML techniques (k nearest neighbors, logistic regression, random forest classifier, and support vector machine) by training these on magnetic parameters obtained from the Helioseismic and Magnetic Imager on board the Solar Dynamics Observatory for the entirety of solar cycle 24. We demonstrate that the logistic regression and support vector machine algorithms perform extremely well in forecasting active region flaring potential. The logistic regression algorithm returns the highest true skill score of 0.967 ± 0.018 , possibly the highest classification performance achieved with any strictly parametric study. From a comparative assessment, we establish that magnetic properties like total current helicity, total vertical current density, total unsigned flux, R_VALUE , and total absolute twist are the top-performing flare indicators. We also introduce and analyze two new performance metrics, namely, severe and clear space weather indicators. Our analysis constrains the most successful ML algorithms and identifies physical parameters that contribute most to active region flare productivity.

Unified Astronomy Thesaurus concepts: Solar flares (1918); Solar active region magnetic fields (1975); Solar active regions (1974); Solar activity (1475); Space weather (2037)

1. Introduction

Solar flares are sudden bursts of electromagnetic radiation from the solar atmosphere, mainly in the extreme ultraviolet and X-ray regimes. They are classified into different categories based on the peak X-ray flux recorded in the 1–8 Å band by the Geostationary Operational Environmental Satellite (GOES). The X-class flares are the most powerful with peak fluxes $\geq 10^{-4} \text{ W m}^{-2}$, followed by M-class flares with peak fluxes $\geq 10^{-5} \text{ W m}^{-2}$. These classes of flares strongly influence the near-Earth space weather and present a bigger potential hazard to human space endeavors than flares with lower peak intensities, which, in decreasing order of intensity, belong to the C, B, and A classes.

From previous studies we know that solar flares originate in active region (AR) structures, where the magnetic flux system becomes energized due to rapid flux emergence, instability, or topological changes of the magnetic configuration via reconnection

processes (Forbes 2000; Priest & Forbes 2002; Schrijver 2007; Leka & Barnes 2003a, 2003b; Nandy et al. 2003; Hahn et al. 2005; Jing et al. 2006). A solar AR with a potential or near-potential magnetic field builds up magnetic nonpotential energy (or free magnetic energy) upon being sheared and twisted. A fraction of this free energy is dissipated during a flare event (e.g., Schrijver et al. 2008), and a typical large solar flare can release large quantities of energy of the order of 10^{32} – 10^{33} erg. Simultaneously, solar energetic particles are also released into the solar wind. Solar flares are often accompanied by coronal mass ejections (CMEs), which pose serious threats if directed toward the Earth. Earlier studies have shown that the magnetic characteristics of ARs (Yeates et al. 2010; Pal et al. 2018, 2017) and filaments determine their propensity to flare and produce associated CMEs (Sinha et al. 2019).

Solar flares (and CMEs) induce extreme space weather conditions that have the potential to harm satellites and impact communication and navigation sectors. The most energetic solar flares can cause electric power grid failures and radio communication blackouts, impact astronaut health, and expose air passengers to harmful doses of radiation (Hapgood 2011;

Schrijver 2015; Schrijver et al. 2015; Eastwood et al. 2017). Proactive measures to mitigate the physical and economic impact of space weather are therefore much sought after, of which early-warning systems are of foremost interest. While physical model based studies have demonstrated a strong potential for success in recent times toward predicting long-term solar activity variations over decadal timescales (Bhowmik & Nandy 2018; Nandy 2021; Nandy et al. 2021), physical model based assessment of AR flaring probability remains elusive.

The creation of knowledge toward predicting solar flares had been initiated with statistical approaches applied to observational data well before machine-learning (ML) techniques found favor. In a set of pioneering studies with vector magnetogram data Leka & Barnes (2003b) and Barnes et al. (2007) conducted a multi-parametric statistical study to distinguish between flaring and flare-quiet ARs based on discriminant analysis.

One of the early applications of ML in solar physics was the automatic real-time detection of solar flares from H α images (e.g., Fernandez Borda et al. 2002; Qu et al. 2003). Very soon, efforts were directed toward the forecast of solar flares. A number of ML methods were trained on sunspot-associated data to forecast solar flares (Li et al. 2007; Qahwaji & Colak 2007; Benvenuto et al. 2018; Cinto et al. 2020). Colak & Qahwaji (2009) used neural networks to make multiclass forecasts based on sunspot area and McIntosh classification data. Line-of-sight full-disk magnetogram data from the Solar and Heliospheric Observatory (SOHO) Michelson Doppler Imager (MDI) presented the next opportunity in the development of solar flare forecasting methods and several advances were made by using features calculated from them. Decision tree classifiers, learning vector quantization, ordinal logistic regression (LR), support vector machines (SVMs), and AdaBoost methods were experimented with by Yu et al. (2009), Song et al. (2009), Yuan et al. (2010), Huang et al. (2010), and Lan et al. (2012). Ahmed et al. (2013) applied a cascade correlation neural network and used feature evaluation algorithms to remove redundant features and showed that a smaller set of parameters yielded comparable results to the entire set. Huang et al. (2018) combined the line-of-sight magnetograms from MDI and the Solar Dynamics Observatory (SDO) Helioseismic and Magnetic Imager (HMI) to create an extensive data set and evaluated the performance of a convolutional neural network (CNN) on these data.

Following the launch of SDO in 2010, its HMI instrument (Scherrer et al. 2012) started providing one of the most advanced unhindered full-disk vector magnetograms. To facilitate AR-based event forecasting, the Spaceweather HMI Active Region Patch (SHARP) data product (Bobra et al. 2014) provides cutouts of automatically tracked magnetic flux concentrations on the solar disk. Using SHARP data, Bobra & Couvidat (2015) implemented an SVM algorithm to distinguish between ARs producing an M- or X-class flare (in the next 24 hr) and those not producing any flare or producing low-intensity flares. Bobra & Couvidat (2015) presented a significant improvement in the performance of AR parameter based ML algorithms, primarily due to the availability of continuous, high-quality HMI vector magnetogram data for deriving input magnetic features.

These recent advances piqued the interest of both the solar physics and computer science communities heralding a close interdisciplinary collaboration in solar flare forecasting. Liu et al. (2017) attempted a multiclass classification using a random forest;

Nishizuka et al. (2017) and Florios et al. (2018) compared various ML algorithms, which included SVM, multilayer perceptrons, random forest, and the k nearest neighbors (KNN) algorithm; Nishizuka et al. (2018, 2020) trained a deep neural network for binary classification; and Campi et al. (2019) used hybrid LASSO and random forest algorithms on features derived during the FLARECAST project. In a recent study, Ribeiro & Gradwohl (2021) used LightGBM for flare forecasting and showed a nice comparison with existing ML models. Classification using KNN was attempted by Hamdi et al. (2017) for univariate time series and by Filali Boubrahimi & Angryk (2018) for multivariate time series. Decision trees were used by Ma et al. (2017) for multivariate time series. Liu et al. (2019) implemented time series classification by training long short-term memory (LSTM) neural networks on SHARP features and flare history parameters. A similar approach was followed by Jiao et al. (2020), who built classification models on an LSTM regressor. Chen et al. (2019) compared LSTM models trained on SHARP parameters and autoencoder-derived features. Using wavelet analysis and features derived from SDO HMI magnetograms, support vector regression was applied to forecasting of the X-ray flux by Muranushi et al. (2015) and Boucheron et al. (2015), while Al-Ghraibah et al. (2015) attempted classification using relevance vector machines. Zernike moments calculated from images were also used for binary classification with SVM (Raboonik et al. 2017; Alipour et al. 2019). Strong-field high-gradient polarity inversion line (PIL) features derived from SHARP images were used by Sadykov & Kosovichev (2017) for classification comparing SVM and a graphical method, while Wang et al. (2019) used SHARP parameters weighed with a PIL mask to improve individual parameter performance on a random forest classifier (RFC). Dhuri et al. (2019) and Hazra et al. (2020) studied the time evolution of various magnetic parameters and the correlations between them. They trained and tested LR, SVM, gradient boost, random forest, multilayer perceptron, KNN, and a naïve Bayes classifier on SHARP feature data with good performance.

With rapid developments in the field of ML and image processing, it became possible to process images directly using CNNs. Jonas et al. (2018) used vector magnetic field data from HMI as well as multiwavelength image data of the chromosphere, transition region, and corona to train a single-layer CNN, and obtained results comparable to those of Bobra & Couvidat (2015). Zheng et al. (2019), Li et al. (2020), and Bhattacharjee et al. (2020) used line-of-sight magnetograms to train deep CNN models.

The underlying nonunique and nondeterministic nature of the triggering mechanisms without well-defined parametric thresholds makes flare forecasting a challenging task making the problem suitable for multiparametric statistical approaches and computational ML algorithms applied to large databases. Attempts to supplement vector magnetogram data with extreme ultraviolet images have not yielded significant improvement. On comparing CNN models trained with and without multiwavelength image data from the Atmospheric Imaging Assembly (AIA) on board SDO, Jonas et al. (2018) found that the best-performing model was the one not provided with AIA data as input. Similarly, the implementation of CNNs has to be developed further for application in flare forecasting. Bhattacharjee et al. (2020) found that the CNN output has spurious dependencies on the magnetogram dimensions.

We limit our comparative analysis to well-studied and successful ML algorithms (limited to parametric approaches

alone for efficiency) to determine their relative performance. This is achieved by applying these algorithms to the largest, single-instrument database suitable for this purpose, i.e., the HMI vector magnetic field observations.

Over the last two decades a wide range of ML algorithms have been applied to forecasts of solar flares. The input data to such algorithms are, most commonly, several AR magnetic parameters derived from magnetograms, magnetogram images, and time series data of magnetic parameters. These works have been able to achieve reasonably accurate forecasts of whether an AR is going to flare or not and, if it does, which class the flare lies in. Furthermore, these works have attempted to extract which magnetic parameters are best correlated with flaring. In general the obtained results are independent of the algorithms used—unsigned current density, unsigned flux, and current helicity have come up as key parameters in most of the previous studies. However, their relative ranking in terms of which contribute most to the flare potential has not been rigorously explored.

In this paper, we compare several ML algorithms to find out which offers the best flare forecasting capability. Bobra & Couvidat (2015) and subsequent studies have shown that the magnetic twist parameter in the SHARP database does not perform well for machine classification whereas earlier physics-based works suggest that twist is a flare-relevant parameter (Linton et al. 1996; Nandy et al. 2003; Hahn et al. 2005; Nandy 2008). We therefore include a global indicator of magnetic twist in our analysis. Furthermore we introduce two new performance metrics, the severe space weather (SSW) and clear space weather (CSW) indicators, to distinguish between these two equally important conditions. Our analysis, detailed in the following sections, is based on the highest number of unique ARs used in training ML algorithms, from the beginning of the SDO era to 2020 December, covering the entire solar cycle 24.

2. Data Selection

Based on their intensities, flares are categorized into five classes: A, B, C, M, and X in ascending order of intensity. In this study, all the AR information is collected from the *hmi.sharp_cea_720s* series (Bobra et al. 2014) but for a longer time of observation. We build our data set considering all the ARs that appeared on the Sun from 2010 May to 2020 December. We divide the ARs into two groups—the positive or flaring class and the negative or nonflaring class. The positive class is defined such that it consists of ARs that have produced at least one M- or X-class flare in their lifetime. In contrast, the negative class is formed by those ARs that produce only low-intensity flares (\leq B-class) or do not flare at all.

We use the XRT Flare Catalog based on the Hinode Flare Catalogue (Watanabe et al. 2012) and the GOES flare catalog to collect information on flare events of the past 10 years, such as flare timing, flare intensities, associated ARs (NOAA number), and their positions on the solar disk. GOES flare events are collected from the *sunpy.instr.goes* module of the Python SunPy library (The SunPy Community et al. 2020). Our data set covers the entire solar cycle 24 from 2010 May to 2020 December. We find that not all events in the XRT catalog match the GOES event list because of slight differences in flare peak times. For each event in the XRT catalog we search for a similar event in the GOES catalog within a time window of 4 minutes centered at the flare peak time of the XRT event. If a

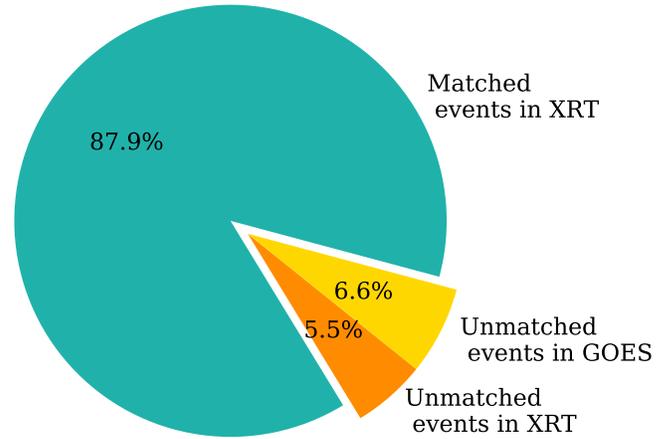


Figure 1. Distribution of matched and unmatched events between XRT and GOES flare catalogs.

similar flare event is found in the GOES catalog with the same flare class and NOAA number, we call it a match. The pie chart in Figure 1 depicts the number of matched and unmatched events in the two catalogs. Following this, a total of 668 matched and 80 unmatched events in the XRT catalog are obtained with flare intensity \geq M-class. Manual inspection of these 80 unmatched events with more lenient conditions, for example, by allowing flare peak time differences up to 15 minutes, results in a further reduction of the number of unmatched events by 57. The negative data set is prepared by excluding flare-associated (of intensity C class and higher) ARs from all recorded SHARP regions during our observational time domain.

All magnetic parameters representing the flaring AR are either collected from the SHARP header keywords or calculated from the vector magnetic field data, 24 hr before the flare peak time. For the negative/nonflaring class, we choose the magnetogram observation at the central time snap of their entire lifespan on the visible disk. In addition, if an AR's position vector from the Sun center makes an angle greater than 70° with the line of sight, we discard that region from our analysis, which is a standard method to avoid high projection effects. We implement this 70° positional filter in the last step of our data preparation process so that it only restricts our domain of analysis, not the domain of our observation. In other words, this ensures that any AR producing an M/X-class flare outside this 70° angular region is not included in our negative class.

Following all these selection criteria, our final positive class contains 503 flaring events and the negative class consists of 3358 nonflaring events. Note that in our positive class, recurrent flare events are treated as separate events with different entries. In contrast, each nonflaring SHARP region has a single entry in the negative class.

3. Methods and Analysis

Preparing the input data and properly training the model are the most crucial steps in working with ML algorithms (Ahmadzadeh et al. 2021). SHARP data comes with various magnetic parameters, calculated from the vector magnetic field maps of the ARs. Previous studies have shown the importance of these derived parameters in characterizing AR properties and complexities (Hagyard et al. 1984; Leka & Barnes 2003c, 2003d; Georgoulis & Rust 2007; LaBonte et al. 2007; Moore et al. 2012). Bobra & Couvidat (2015) identified 13 such parameters that they

Table 1
Details of AR Parameters Extracted from SHARP Data

S. No.	Keyword	Description	Normalized F-score
1	R_VALUE	Sum of unsigned flux near PIL	1.000
2	TOTUSJZ	Total unsigned vertical current	0.927
3	TOTABSTWIST*	Total absolute twist calculated over strong-field ($ B \geq 300$ G) regions	0.898
4	TOTUSJH	Total unsigned current helicity	0.833
5	USFLUX	Total unsigned flux	0.707
6	AREA_ACR	Area of strong-field pixels in the AR	0.706
7	MEANPOT	Mean photospheric magnetic free energy	0.695
8	TOTFZ*	Sum of z-component of Lorentz force	0.579
9	TOTBSQ*	Total magnitude of Lorentz force	0.573
10	SAVNCP	Sum of the modulus of the net current per magnetic polarity	0.510
11	TOTPOT	Total photospheric magnetic free energy density	0.456
12	SHRGT45	Fraction of area with shear $>45^\circ$	0.451
13	EPSZ*	Sum of z-component of normalized Lorentz force	0.393
14	ABSNJZH	Absolute value of the net current helicity	0.393
15	AVG90PABSTWIST*	Average absolute twist for pixels having twist more than 90th percentile value	0.027
16	VTWIST*	Standard deviation of twist within an AR	0.013
17	AVGABSTWIST*	Average absolute value of twist	0.011
18	AVGTWIST*	Average value of twist	0.000
19	MEANALP	Mean value of flux-weighted twist	0.000

Note. An asterisk (*) denotes a parameter that is not readily available in the SHARP header keywords and is calculated explicitly from the SHARP vector magnetic field data.

found to be most useful in describing the flare potential. However, they estimated the magnetic twist using the parameter called MEANALP, whose poor performance led them to exclude any contribution of the magnetic twist from their classification. But it has been shown that a high twist in the magnetic flux rope can store nonpotential magnetic energy and often leads to eruption of the flux rope via the kink instability (Nandy et al. 2008). Motivated by these physical arguments, we calculate six new magnetic parameters related to AR twist: TOTABSTWIST, AVG90PABSTWIST, VTWIST, AVGABSTWIST, AVGTWIST, and MEANALP (see Table 1 for descriptions). We incorporate these six twist-related features along with those 13 parameters used by Bobra & Couvidat (2015). Here we assume the force-free field approximation for the estimation of magnetic twist—also known as the alpha parameter. The vertical twist parameter α_z at each pixel of an AR magnetogram is given as

$$\alpha_z = \mu_0 \left(\frac{J_z}{B_z} \right), \quad (1)$$

where J_z and B_z are the z-components of the current density and magnetic field, respectively, and μ_0 is the permeability of free space. All these 19 features and their descriptions are listed in Table 1. We perform univariate feature selection analysis with ANOVA Fisher statistics (F-statistics) using the Python scikit-learn library to finalize our set of input magnetic features by eliminating those that are not very useful for this classification. The obtained F-score (see Bobra & Couvidat 2015 for the calculation of F-score) for each feature is represented in Figure 2. It is quite surprising that all the magnetic twist related parameters (including MEANALP) are insignificant according to the F-statistics except for TOTABSTWIST, which ranks third in the list. This indicates that flare potential is more closely coupled to gross/extrinsic quantities than to magnetic properties at individual pixels. We exclude the last five features having the lowest normalized F-scores in Figure 2, i.e., AVG90PABSTWIST, VTWIST, AVGABSTWIST,

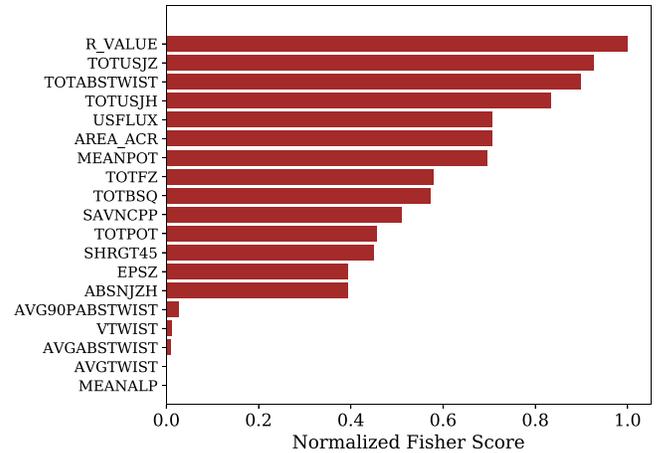


Figure 2. Normalized F-score ranking of the magnetic field parameters.

AVGTWIST, and MEANALP (descriptions in Table 1). All further analyses are done with the remaining 14 features, which contain 10 SHARP keyword parameters and 4 derived parameters including 1 newly introduced parameter TOTABSTWIST.

Henceforth, each AR is represented by a single data point in the 14-dimensional feature space except those ARs that have produced multiple M/X-class flares. The latter are accounted separate events for each M/X-class flare. Our whole data set, consisting of 3861 events, is randomly divided into two groups, training and testing, with a population ratio of 4:1, respectively. We arrange the data such that the ratio of flaring to nonflaring events is the same for both the training set and the test set. We preprocess the data by normalizing it such that the processed data has zero mean and unit standard deviation. For this normalization we solely use the training data set and then apply the same population mean and standard deviation to normalize the test data set. To make our classification more robust and independent of any bias, we randomly shuffle our data set to make 20 similar but differently distributed representative pairs of training and testing data. We denote each such pair with D_i ,

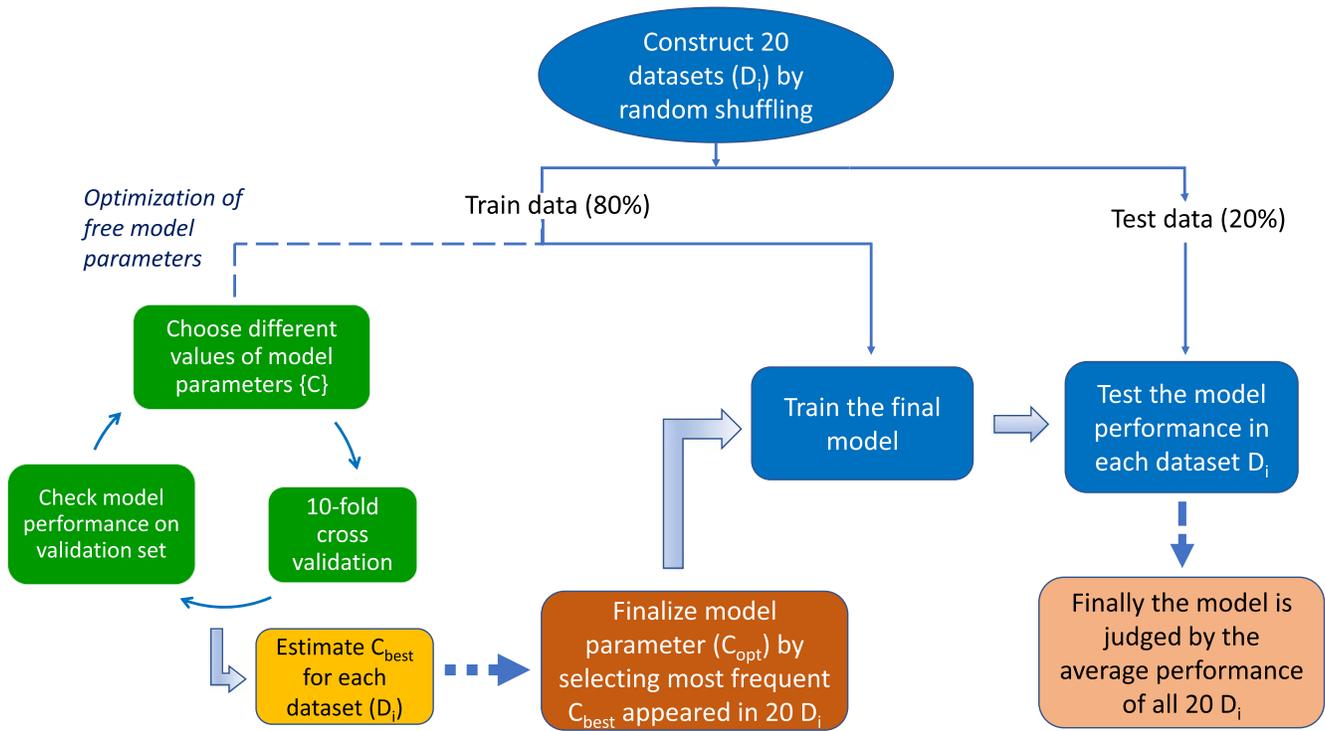


Figure 3. Schematic diagram of our method of analysis.

where i is a running index between 1 and 20. Each ML algorithm is evaluated by its average performance over these 20 D_i . A schematic diagram of our analysis method is shown in Figure 3. The following two sections describe how we quantify model performance and compare the different ML algorithms.

3.1. Performance Metric

Typically, the performance of an ML model is evaluated from the confusion matrix. It is a 2×2 matrix whose elements are the number of correctly forecasted positive-class events (TP), the number of correctly forecasted negative-class events (TN), the number of events falsely forecasted as positive-class (FP), and the number of events falsely forecasted as negative-class (FN). In general, there are various parameters that can be derived from the confusion matrix such as accuracy, recall, F1-score, etc., but their suitability depends on the particular problem. Simple accuracy is defined as the ratio of the number of correct forecasts to the total number of forecasts. In our data set, the number of flaring ARs (positive class) is much less than the number of nonflaring (negative-class) ARs, which means our data set is highly imbalanced. Hence we cannot simply use the accuracy measure to evaluate the models. To deal with this problem we use the macro accuracy (MAC) (Pereira et al. 2018) and true skill statistic (TSS) (Woodcock 1976) values to evaluate our ML models. Bloomfield et al. (2012) showed that the TSS is unaffected by class imbalance and gives an unbiased result. The MAC and TSS are defined as

$$\text{MAC} = \frac{1}{2} \left(\frac{\text{TP}}{\text{TP} + \text{FN}} + \frac{\text{TN}}{\text{FP} + \text{TN}} \right), \quad (2a)$$

$$\text{TSS} = \frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FP}}{\text{FP} + \text{TN}}. \quad (2b)$$

The MAC is the average of the accuracy of each individual class; hence its value lies between 0 and 1. On the other hand

the TSS has two components: the first is the positive-class accuracy and the second is the probability of false forecasts for the negative class. Using the TSS, we penalize the model's performance commensurately by subtracting the false-alarm ratio from the positive-class accuracy. This shows the usefulness of the TSS in the present problem as we are more interested in correctly predicting flaring ARs with a minimal number of false detections. The value of the TSS ranges from +1 to -1 and we optimize our models to maximize the TSS.

Depending on the end-user application our priorities for detecting a specific class can change. For example, one may wish to identify only those regions that have the potential to flare with a high degree of confidence without worrying about misclassifying a nonflaring region as a flaring region. This motivates a new performance indicator, which we term the SSW metric, defined as

$$\text{SSW} = \frac{\text{TP} - \text{FN}}{\text{TP} + \text{FN}}. \quad (3)$$

Conversely, one may wish to focus on identifying nonflaring regions only with a high degree of confidence. For this we define another parameter called the CSW metric:

$$\text{CSW} = \frac{\text{TN} - \text{FP}}{\text{TN} + \text{FP}}. \quad (4)$$

It is important to note that SSW is only linked to flaring-class events. It indicates the correct identification ratio combined with a penalty for misidentification within the flaring class. Similarly, CSW deals with the nonflaring class only. The value of these two metrics lies between -1 and +1, where +1 indicates perfect identification of all the events within a specific class, whereas -1 indicates the scenario where all events are misclassified. A metric score of 0 denotes the scenario where half of the events in a specific class are correctly identified and

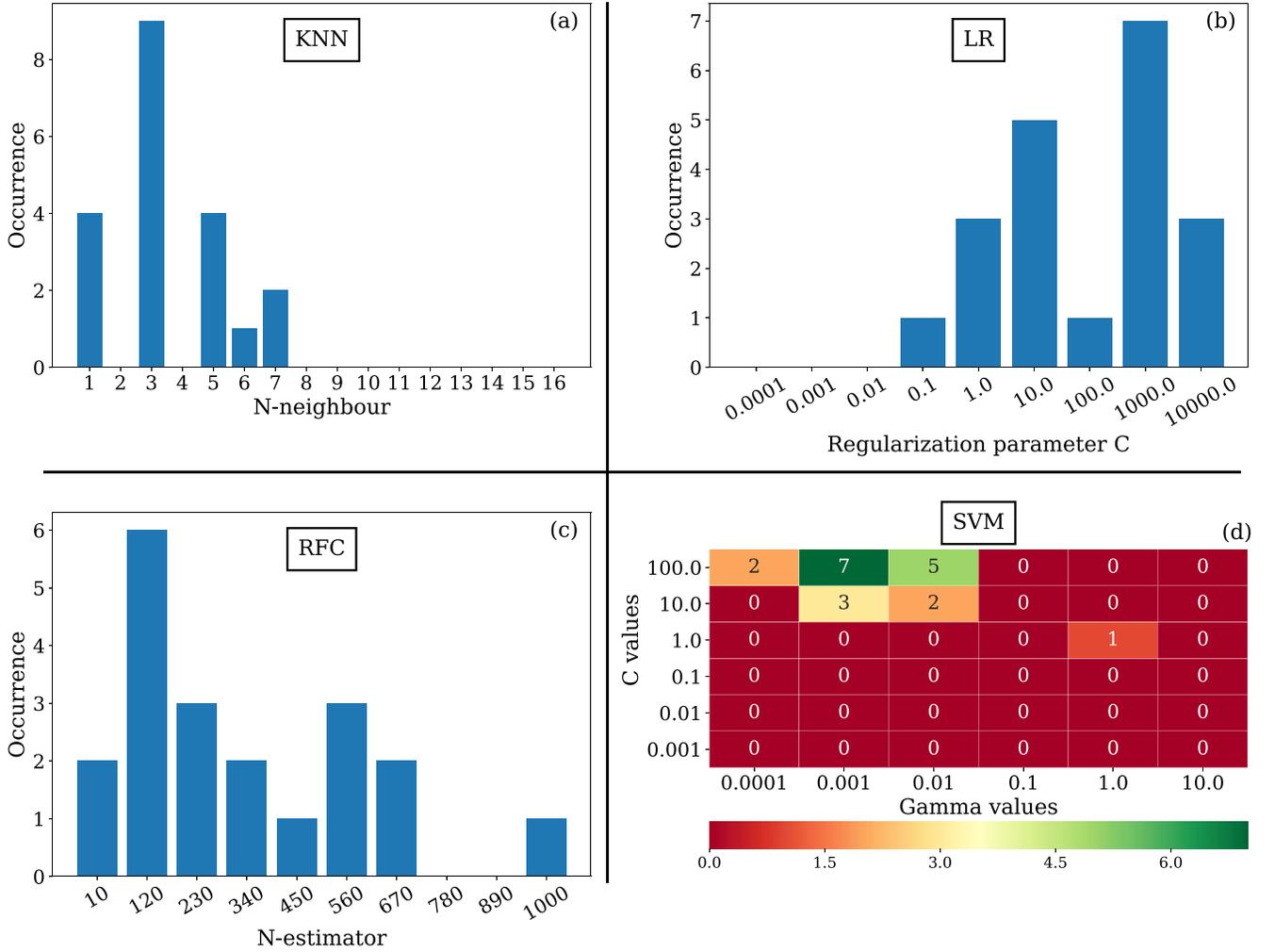


Figure 4. These plots represent the values of the optimal parameters, which give the best TSS on the validation data set for each experimental data set (D_i).

the other half are wrongly classified implying no useful classification capability.

Moreover the average value of these two performance metrics returns the TSS and can be demonstrated as follows:

$$\begin{aligned}
 \text{SSW} + \text{CSW} &= \frac{\text{TP} - \text{FN}}{\text{TP} + \text{FN}} + \frac{\text{TN} - \text{FP}}{\text{TN} + \text{FP}} \\
 &= \frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FN}}{\text{TN} + \text{FP}} + \frac{\text{TN}}{\text{TN} + \text{FP}} - \frac{\text{FP}}{\text{TP} + \text{FN}} \\
 &= \frac{\text{TP}}{\text{TP} + \text{FN}} - \frac{\text{FP}}{\text{TN} + \text{FP}} + \left(\frac{\text{TN}}{\text{TN} + \text{FP}} - 1 \right) \\
 &\quad + \left(1 - \frac{\text{FN}}{\text{TP} + \text{FN}} \right) \\
 &= 2\text{TSS} \Rightarrow \frac{\text{SSW} + \text{CSW}}{2} = \text{TSS}.
 \end{aligned}$$

3.2. Cross-validation

One of the most important aspects of any ML algorithm is the optimization of its hyperparameters, to achieve the best fit on the data set. If a classifier performs too well on the training data set, it might fail to capture the overall picture and can badly perform on the test data set—also known as overfitting. The optimization of hyperparameters is done by employing a

grid search algorithm for finding the optimal hyperparameters C_{opt} of the training component set of each D_i . We use a 10-fold cross-validation on the training data set of each D_i to avoid the issue of overfitting. This process divides the training set into 10 groups of equal sample sizes. Training the model on nine groups, the validation is done on the one remaining group of data points. This happens 10 times such that each data group is made the validation set once. The average validation TSS from this 10-fold cross-validation is used to decide the model hyperparameters for each data set D_i . We train our models with different values of the model hyperparameters and the optimal values of the hyperparameters C_{opt} are obtained for each D_i by maximizing the average validation TSS. Finally, we choose our operational model hyperparameter (C_{best}) by selecting the most frequently appearing C_{opt} among these 20 experimental sets D_i . This completes our model optimization process. Now we check the performance of the finalized model and judge the model based on the average test performance over 20 D_i .

3.3. ML Models

We use four popular supervised ML algorithms available in the Python scikit-learn library (Pedregosa et al. 2011). These four algorithms are KNN, LR, RFC, and SVM; they are discussed briefly in this section. In all four models, the model parameters are tuned properly to get the best achievable

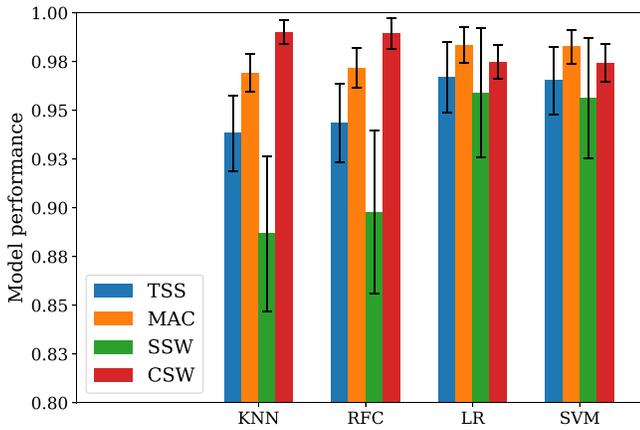


Figure 5. This plot depicts the classification performance of all four models.

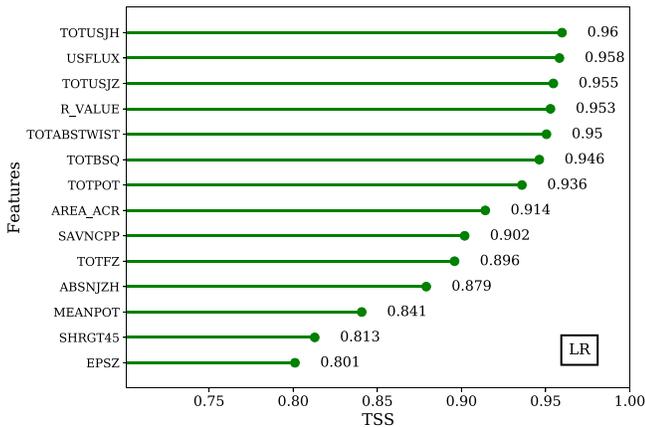


Figure 6. Average TSSs of individual features for LR. Each score is obtained by training the LR model with a single parameter as input data, averaged over the output of the 20 experimental data sets.

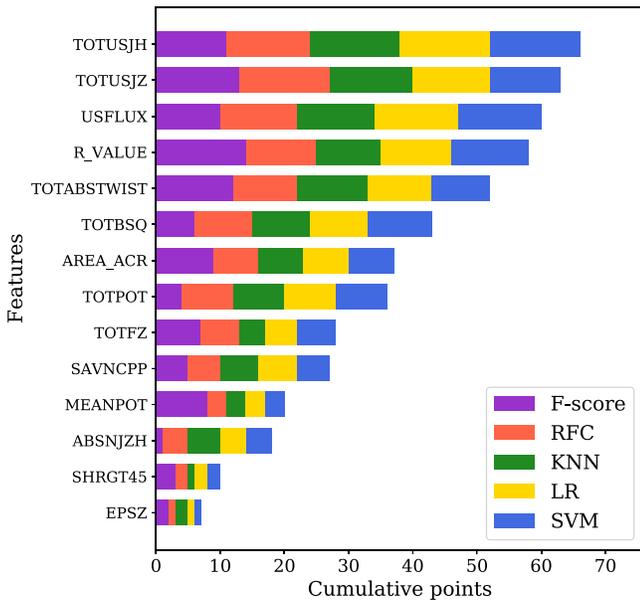


Figure 7. Feature ranking based on cumulative points obtained from F-statistics, RFC, KNN, LR, and SVM. The top-scoring feature in each model gets 14 points, while lowest-scoring feature gets one point. The points are added for each feature and then the features are ranked accordingly.

performance, which is the TSS in our case. The optimal model parameters of all four models are selected from the histogram plot of Figure 4. A general overview of these ML algorithms can be found in Bishop (2006) and Mehta et al. (2019).

1. KNN: This is an instance-based ML technique that uses instances of training data to compute the machine classification based on a simple majority vote of the K number of nearest neighbors of each point (Fix & Hodges 1951). When the data set is not large, as in our case, using the KNN classifier poses no disadvantage as it does not create an internal model, which might otherwise use a large memory space. In our model, the weights assigned to each neighbor are equal and the nearest neighbors are calculated using Euclidean distance. The best KNN model is obtained by finding the optimal K , i.e., the optimal value of the number of nearest neighbors to maximize the TSS output. We search for the optimal K value between 1 and 16 in the 20 data sets (D_i) to get the maximum TSS. Figure 4(a) shows a histogram plot of optimal K values for the 20 different data sets. Since $K = 3$ has the highest number of occurrences, it becomes our final choice.
2. LR: This classifier, also known as the log-linear classifier, is a linear classification model that uses the sigmoid function to classify data into discrete categories (Mehta et al. 2019). This makes it extremely suitable for binary classification problems. Our model uses regularized LR, and is implemented using the LR classifier available in scikit-learn. The only free parameter of this model, which we use, is the regularization parameter C , and the most favorable value is estimated from within the range $[0.0001, 10,000.0]$, varied with logarithmic increments. In Figure 4(b) we can see that occurrence is at its maximum at $C = 1000$; hence we choose 1000 as the optimal C parameter.
3. RFC: This classifier consists of a large number of individual decision tree classifiers that operate as an ensemble (Tin Kam Ho 1995). Each decision tree is trained on a subset of the entire data set. Generally, decision trees tend to overfit the data and exhibit high variance. Random forests are constructed so as to decrease the variance. The overall prediction is generated by taking an average of the constituent tree predictions, which tends to cancel out some prediction errors from individual trees. Thus, a large number of uncorrelated trees can produce largely accurate ensemble predictions. Our model uses the RFC available with the scikit-learn package, and the best forest is created by varying the number of trees (also called the “ N -estimator”) in the forest from 10 to 1000. We can see from Figure 4(c) that occurrence is at its maximum for the N -estimator = 120. Therefore, we select 120 as the optimal value of the number of trees.
4. SVM: This is a powerful classification technique (Cortes 1995), and has previously yielded the best results among various ML models, when applied to solar flare prediction based on SHARP parameters. SVM works by creating a decision boundary, marked by a subset of

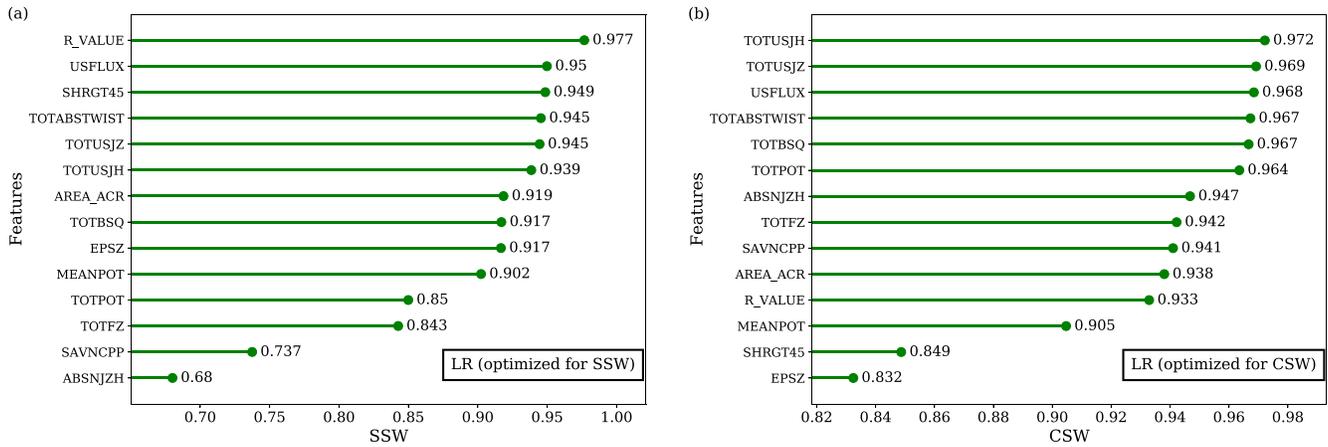


Figure 8. The left panel (a) shows the feature ranking for the LR model when the model is optimized for the metric SSW and panel (b) on the right shows the feature ranking when the LR model is optimized for the CSW metric.

Table 2

Performance of Classifiers Trained with the Best Hyperparameters Deduced via Grid Search and 10-fold Cross-validation over 20 Randomly Shuffled Data Sets

Classifier Name	Average Performance Measure in 20 Trials when the Models Are Optimized for TSS				An Example of Confusion Matrix Elements			
	SSW	CSW	TSS	MAC	TN	TP	FN	FP
KNN	0.887 ± 0.040	0.990 ± 0.006	0.938 ± 0.019	0.969 ± 0.010	663	110	6	3
RFC	0.898 ± 0.042	0.989 ± 0.008	0.944 ± 0.020	0.972 ± 0.010	664	101	6	2
LR	0.959 ± 0.033	0.975 ± 0.009	0.967 ± 0.018	0.983 ± 0.009	669	94	1	9
SVM	0.956 ± 0.031	0.974 ± 0.010	0.965 ± 0.017	0.983 ± 0.009	656	105	2	10

Note. The confusion matrix elements correspond to a test data set in D_i whose TSS is closest to the determined mean.

training points called support vectors, to separate positive and negative events in the training data. It uses a kernel function to map the data points to higher-dimensional space. Our model uses a Gaussian radial basis function as the kernel and assigns the class weight in a way that is inversely proportional to the class frequency to handle the class imbalance problem. The kernel coefficient γ and the regularization parameter C are varied within the ranges $[0.0001, 10.0]$ and $[0.001, 100.0]$, respectively, to get the best SVM model by comparing the TSSs. From Figure 4(d) we can see the optimal values of C and γ are 100 and 0.001, respectively, as this combination produces the highest TSS in 7 out of 20 D_i .

4. Results

Each ML classifier is trained with the finalized hyperparameters (C_{best}) on the training set of each D_i , and then the trained model is applied on the test set in that D_i . The average and standard deviation of the performance metrics over these 20 test sets are reported in Table 2. We find that all of these models work reasonably well for identifying flaring and nonflaring ARs. The performance of both LR and SVM is very similar, better than that of KNN and RFC. The average TSSs of LR and SVM are 0.967 and 0.965, respectively. Therefore, we claim that LR and SVM are equally good in terms of performance. For further analysis we primarily focus on LR because of its marginally

higher TSS value. The comparison of the four ML classifiers is depicted in Figure 5. We achieve a remarkable MAC of 0.983 for both LR and SVM. SSW is also much higher for LR (and SVM) than for RFC or KNN, which indicates the suitability of LR (and SVM) in highly active space weather circumstances. Also the close values of SSW and CSW tell us of the unbiased nature of the model predictions. Clearly, LR/SVM is a better choice over KNN/RFC for having very similar SSW and CSW scores. An example of confusion matrix elements corresponding to a seed value of D_i with the TSS close to the mean value is also shown in Table 2.

To understand which AR parameters are more useful in determining the flaring capability of an AR, we train our models with the 14 AR parameters individually. The outcome of this experiment for LR is presented in Figure 6, where all these AR parameters are plotted along the y-axis in ascending order of their individual TSSs. This implies that the topmost parameter on the y-axis is the most significant one having the highest individual classification capability and as we move downward, we find parameters of lesser importance.

The ranking of input features based on the individual TSSs depends on the ML model used, and can moderately differ for different models. For a particular ML algorithm, feature ranking may also depend on the model hyperparameters. Hence to get a more general global ranking of features, we follow a marking scheme in which we assign points (ranging from 1 to 14) to each parameter based on its individual TSS

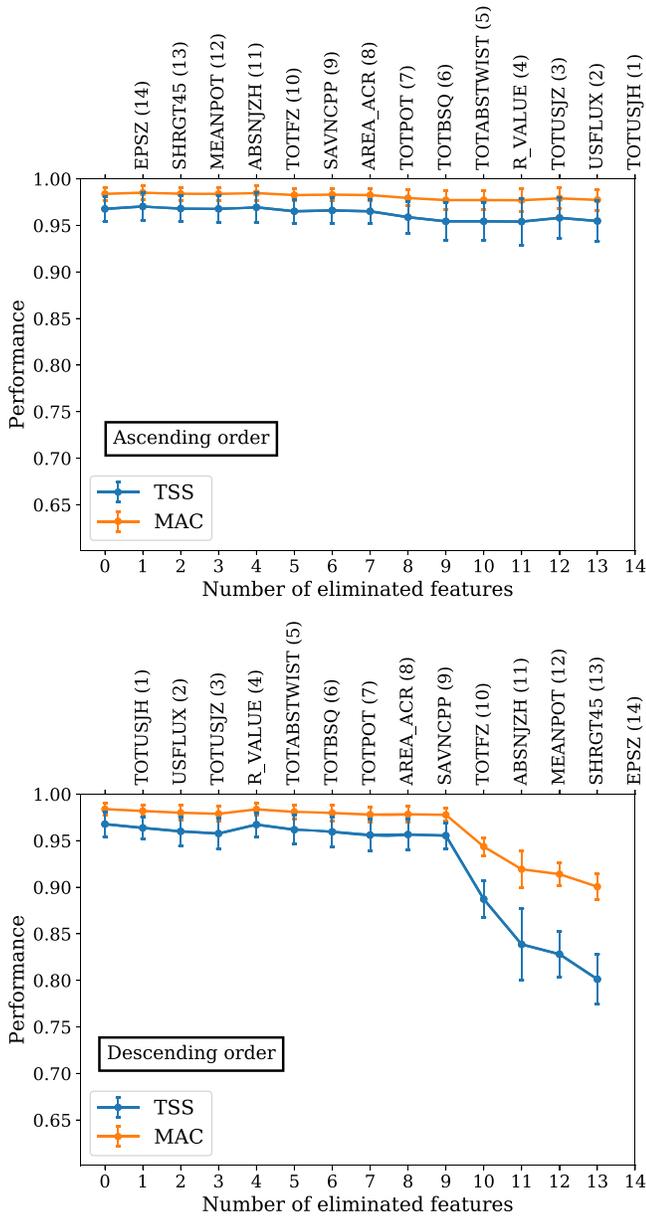


Figure 9. Variation of LR performance with number of input features. The experiment is performed over all 20 experimental data sets (D_i). In each step, the eliminated feature along with its rank is indicated on the top x -axis whereas the bottom x -axis indicates the total number of eliminated features at that step.

ranking for each of the four models and the univariate F-score ranking. For example, for LR 14 points are assigned to TOTUSJH for its highest TSS, whereas EPSZ gets one point based on Figure 6. Finally we add up all the points for each feature from the different models to get a cumulative ranking as shown in Figure 7.

We further optimize the LR model by tuning the model hyperparameters for maximizing the SSW and CSW metrics to see how the ranking of input magnetic features changes for these two newly introduced space weather metrics. The left panel in Figure 8 shows the ranking of input features according to their individual SSW scores and the right panel shows the feature ranking with respect to the CSW score. We can see that the ranking of R_VALUE , SHRGT45, and EPSZ goes down significantly on optimizing our LR model with CSW instead of SSW. Both R_VALUE and SHRGT45 shift downward in

ranking by 10 due to this change in performance metric. On the other hand, features like ABSNJZH, SAVNCPP, TOTPOT, and TOTFZ perform better on CSW, reflecting an upward shift in the feature ranking (Figure 8).

To study the dependence of model performance on the number of input AR features, we train our LR model by eliminating input features one by one and check the model performance at each step. The elimination is done by following both the ascending and descending order of ranking based on the individual TSS, the result of which is represented in Figure 9. We can see that when we eliminate the features in ascending order of their ranking the model performance does not change much. This is expected because the more important features are eliminated at the last steps. On the other hand, for the descending order we see a drastic fall in model performance when the number of eliminated features increases beyond 9. This indicates a significant loss of correlation with the output labels at each step beyond this point. The plateau in the descending-order plot of Figure 9 is only possible if the top-ranked features are highly correlated among themselves, causing no significant loss of information when these features are thrown out. A study by Hazra et al. (2015) also confirms the correlation among integrated magnetic features, showing a connection between AR magnetic properties and coronal X-ray flux. The correlogram presented in Figure 10 confirms this, with all top eight features, excluding R_VALUE , being highly intercorrelated. One possible reason behind this high correlation could be that they are extrinsic features, or in other words, that their values depend on the size of the AR as they represent the sums of physical quantities over the entire AR. As correlated features do not provide new information, we group features with correlation constants >0.9 and train our model by picking up the top-performing feature from each group. Following this scheme we select six features: TOTUSJH, R_VALUE , TOTFZ, SAVNCPP, MEANPOT, and SHRGT45. When trained with these features only, the LR classifier gives average TSS and MAC values of 0.962 and 0.981, respectively (with SSW and CSW of 0.956 and 0.968), which are close to our primary model performance with 14 features.

5. Conclusions

With the advancement of new technologies, especially in satellite-based telecommunications and navigational networks, a significant fraction of our technological assets have become increasingly vulnerable to space weather disturbances. This has resulted in growing demand for reliable space weather forecasts. Solar flares strongly influence space weather, which is why we address the problem of predicting solar flares using their source region characteristics. In this work, we have built a high-performance operational LR classifier that can differentiate solar ARs based on their flaring capabilities. We have compared four supervised ML models, all of which perform quite well in classifying ARs into positive/flaring and negative/nonflaring categories. The method we follow is statistically unbiased due to the use of 20 randomly shuffled replicas of the primary data set for measuring model performance. The LR classifier delivers the highest average TSS of 0.967 ± 0.018 closely followed in performance by the SVM classifier.

While a direct comparison of model performance between our algorithms with that in previous studies may not be appropriate due to subtle differences in the data selection

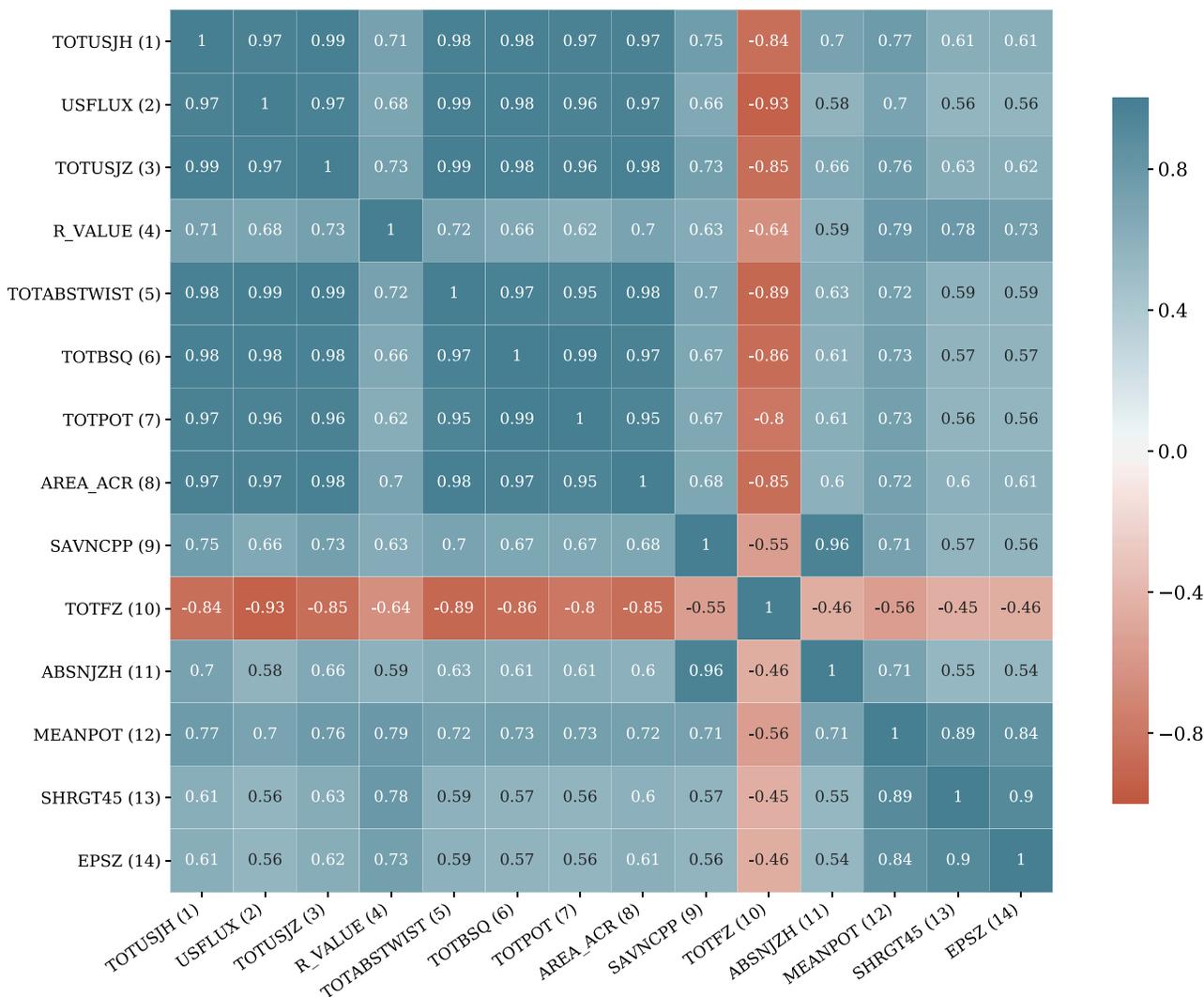


Figure 10. Correlation matrix for the input features, which are arranged in the order of their individual TSS ranking with LR. The ranks are shown in brackets next to the feature keywords.

scheme and the size of the database used, we do note that in the context of the TSS, we achieve a higher performance score relative to earlier classification attempts with supervised ML algorithms (e.g., Bobra & Couvidat 2015; Nishizuka et al. 2017; Florios et al. 2018).

One possible reason for our achieving a high TSS could be the exclusion of C-class events in the data preparation stage. The distribution of the top five input features in our data set is shown in Figure 11, where we can see a clear separation between two clusters of data points for the two different classes. This ensures that our data set is easily separable with two distinct classes in feature space. Other possible reasons could be the different event selection scheme and the larger temporal coverage of our data set and also it is important to note that each entry in the negative class comes from a different SHARP region ensuring no repetition of AR patches in the nonflaring class.

In addition to achieving a high TSS, we find that a global indicator of magnetic twist, estimated by the feature TOTABSTWIST, plays an important role in predicting AR flare potential. Although TOTABSTWIST comes in fifth position of the cumulative feature ranking, other twist-related

parameters, including VTWIST and MEANALP, are not found to play a significant role.

We have also introduced two new performance indicators, termed SSW and CSW, which are useful in comparing model performance depending on the operational space weather condition one wishes to lay more emphasis on. For example, when the solar activity is high, we may wish to get a reliable all-clear forecast for executing specific time-critical tasks that are susceptible to space weather. So, depending on the application and operational space weather scenario, SSW and CSW can provide more meaningful operational intelligence than the TSS alone. We can also get an estimate of the model bias toward a specific forecast by examining the difference between SSW and CSW. With these two indicators, we see that KNN and RFC are more biased toward the negative class (as CSW is much higher than SSW) than SVM and LR. Because of the larger size of the negative class, a classifier’s forecast may become biased toward it. But our analysis shows that classifiers such as LR and SVM can be suitably optimized to minimize the class imbalance problem significantly.

We have also studied the relative importance of input features in terms of their ability to classify flaring and nonflaring ARs. Based on the global ranking of Figure 7, we

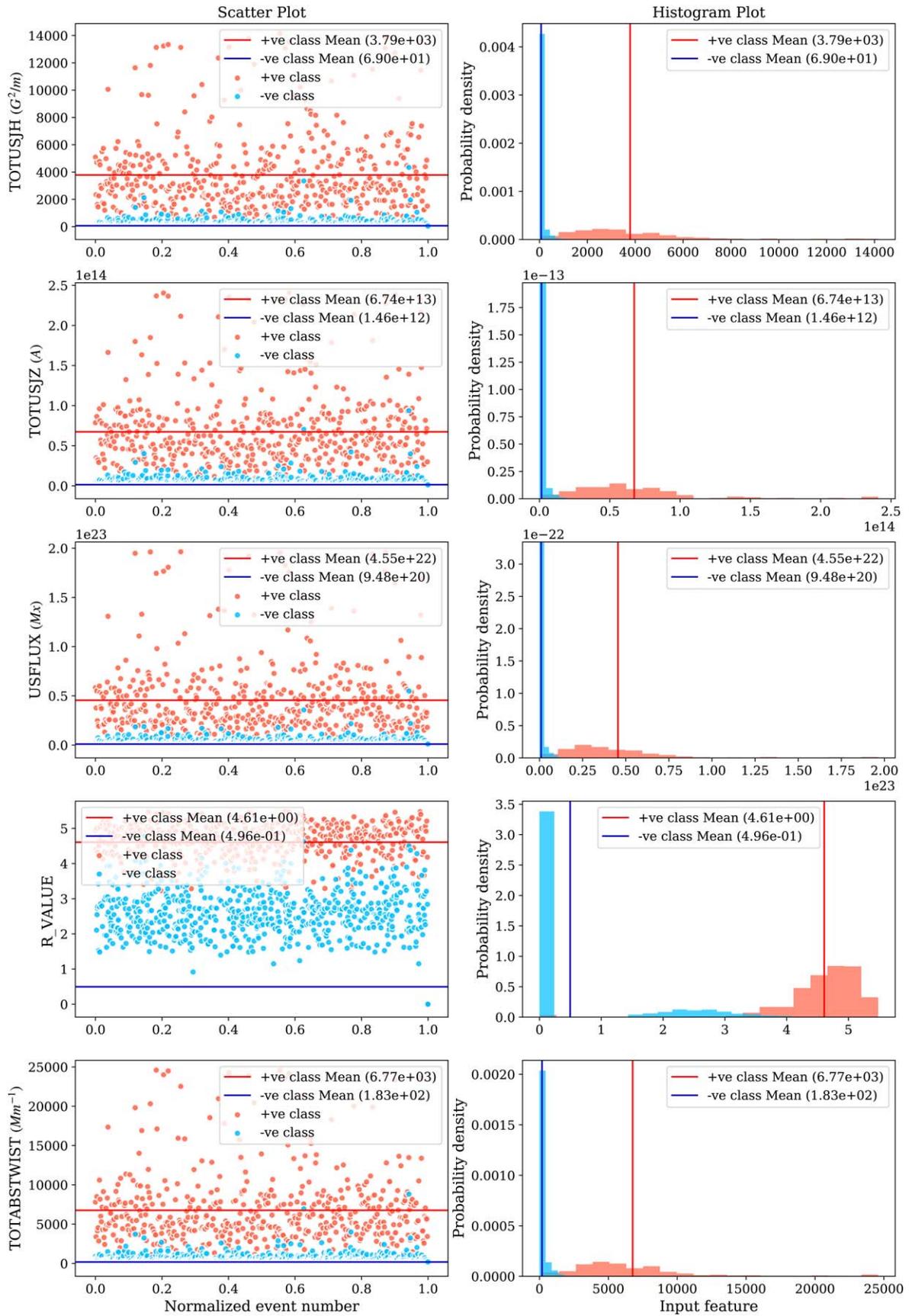


Figure 11. Distribution of magnetic features in positive and negative classes. The left column shows scatter plots of the top five input features according to the cumulative feature ranking. The x -axis of the scatter plots is the normalized event number, which is the number of events divided by the total number of events in that class. The mean values for both classes are shown. In the right column, histogram plots of the probability density are shown for the corresponding features.

have identified key magnetic features that are responsible for the flare potential of an AR. The total unsigned current helicity, the total unsigned vertical current, the total unsigned magnetic flux, the flux near the strong-field high-gradient neutral line, and the total absolute twist are the major deciding factors for AR flare potential. It is important to note that all of the highly ranked features in Figure 7 denote extensive or net properties of an AR, except for R_VALUE . This reaffirms previous findings (Welsch et al. 2009; Hazra et al. 2020) that extensive parameters contribute more to forecasting algorithms than intensive parameters. The only nonextensive feature that performs well is R_VALUE , indicating that it contains some unique information regarding flaring potential.

Our analysis shows that for a given classifier, the ranking of input magnetic features differs based on the choice of performance metric. For example, the ranking of R_VALUE and SHRG45 drops from 1st to 11th and from 3rd to 13th, respectively, when we switch to CSW from SSW as the model-optimizing metric. The reason behind this downward shift in feature ranking can be explained by examining the feature distribution plot. We have investigated this trend further by analyzing the feature distribution and found well-separated peaks in the frequency distribution for binary classes; we believe this might be the reason for these features performing well in discriminating between the classes. For the nonflaring class, these distributions are sparse and skewed toward the central peak of the flaring class. This reduces the ability for perfect identification of nonflaring events, especially those distributed near the flaring-class mean. As the CSW determination involves only the nonflaring class the ranking of these features that are not well separated from the mean of the other class shifts down. In Figure 11 (fourth row, right panel) we can see that R_VALUE shows a bimodal distribution leading to a further increase in variance for the nonflaring class. This bimodal distribution of R_VALUE is due to the absence of strong-gradient magnetic PILs in many nonflaring ARs. On the other hand, features whose ranking improves on optimizing with CSW have in general a sharp peak in the distribution for the nonflaring class with very small variance. This helps these features (ABSNJZH, SVANCPP, TOTFZ, and TOTPOT) to correctly identify nonflaring events.

We also find that model performance has very low dependency on the number of input features especially when the input features are highly correlated. We have shown that high model performance could be maintained even with a smaller set of input magnetic features, selected carefully to reduce internal correlation. Our work brings to the fore key properties of parameter-based ML flare forecasting that can be utilized in future works to develop more robust flare forecasting models. Finally, we anticipate our comprehensive analysis will lead to operational flare forecasting with higher efficiency and higher precision.

The Center of Excellence in Space Sciences India at IISER Kolkata is funded by the Ministry of Education, Government of India. The authors acknowledge funding from the Ministry of Education SPARC Project grant SPARC/2018–2019/P746/SL to IISER Kolkata, Nordita/Stockholm University, the KTH Royal Institute of Technology, and IIT Kharagpur. S.S. thanks the University Grants Commission, Government of India, for a senior research fellowship. O.G. thanks the Department of Science and Technology, Government of India, for an

INSPIRE scholarship. D.N. thanks the Wenner-Gren Foundation for a visiting professorship at Nordita, Stockholm University. Nordita is sponsored by NordForsk. All the magnetic field data are either collected or calculated from SDO/HMI SHARP data, maintained by the Joint Science Operation Center. The authors have utilized the drms open-source software package (Glogowski et al. 2019) to access data from HMI. The authors acknowledge use of data from the XRT Flare Catalog and GOES flare database to acquire information on flaring events and their corresponding ARs. All ML algorithms have been implemented using the scikit-learn package (Pedregosa et al. 2011) on Python.

Data Availability

The input magnetic feature data set used for this study is freely available on Zenodo: <https://doi.org/10.5281/zenodo.5498347>.

ORCID iDs

Suvadip Sinha  <https://orcid.org/0000-0001-7229-5192>
 Om Gupta  <https://orcid.org/0000-0001-8470-7289>
 Vishal Singh  <https://orcid.org/0000-0002-9911-2285>
 B. Lekshmi  <https://orcid.org/0000-0003-0666-7650>
 Dibyendu Nandy  <https://orcid.org/0000-0001-5205-2302>
 Dhrubaditya Mitra  <https://orcid.org/0000-0003-4861-8152>
 Saikat Chatterjee  <https://orcid.org/0000-0003-2638-6047>
 Sourangshu Bhattacharya  <https://orcid.org/0000-0001-5220-1881>
 Saptarshi Chatterjee  <https://orcid.org/0000-0002-7817-9851>
 Nandita Srivastava  <https://orcid.org/0000-0002-0452-5838>
 Axel Brandenburg  <https://orcid.org/0000-0002-7304-021X>
 Sanchita Pal  <https://orcid.org/0000-0002-6302-438X>

References

- Ahmadzadeh, A., Aydin, B., Georgoulis, M. K., et al. 2021, *ApJS*, 254, 23
 Ahmed, O. W., Qahwaji, R., Colak, T., et al. 2013, *SoPh*, 283, 157
 Al-Ghraibah, A., Boucheron, L. E., & McAteer, R. T. J. 2015, *A&A*, 579, A64
 Alipour, N., Mohammadi, F., & Safari, H. 2019, *ApJS*, 243, 20
 Barnes, G., Leka, K. D., Schumer, E. A., & Della-Rose, D. J. 2007, *SpWea*, 5, S09002
 Benvenuto, F., Piana, M., Campi, C., & Massone, A. M. 2018, *ApJ*, 853, 90
 Bhattacharjee, S., Alshehhi, R., Dhuri, D. B., & Hanasoge, S. M. 2020, *ApJ*, 898, 98
 Bhowmik, P., & Nandy, D. 2018, *NatCo*, 9, 5209
 Bishop, C. M. 2006, *Pattern Recognition and Machine Learning* (New York, NY: Springer)
 Bloomfield, D. S., Higgins, P. A., McAteer, R. T. J., & Gallagher, P. T. 2012, *ApJL*, 747, L41
 Bobra, M. G., & Couvidat, S. 2015, *ApJ*, 798, 135
 Bobra, M. G., Sun, X., Hoeksema, J. T., et al. 2014, *SoPh*, 289, 3549
 Boucheron, L. E., Al-Ghraibah, A., & McAteer, R. T. J. 2015, *ApJ*, 812, 51
 Campi, C., Benvenuto, F., Massone, A. M., et al. 2019, *ApJ*, 883, 150
 Chen, Y., Manchester, W. B., Hero, A. O., et al. 2019, *SpWea*, 17, 1404
 Cinto, T., Gradwohl, A. L. S., Coelho, G. P., & da Silva, A. E. A. 2020, *SoPh*, 295, 93
 Colak, T., & Qahwaji, R. 2009, *SpWea*, 7, S06001
 Cortes, C., & Vapnik, V. 1995, *Machine Learning*, 20, 273
 Dhuri, D. B., Hanasoge, S. M., & Cheung, M. C. M. 2019, *PNAS*, 116, 11141
 Eastwood, J. P., Biffis, E., Hapgood, M. A., et al. 2017, *RiskA*, 37, 206
 Fernandez Borda, R. A., Mininni, P. D., Mandrini, C. H., et al. 2002, *SoPh*, 206, 347
 Filali Boubrahimi, S., & Angryk, R. 2018, in 2018 IEEE First Int. Conf. on Artificial Intelligence and Knowledge Engineering (AIKE) (Piscataway, NJ: IEEE), 162
 Fix, E., & Hodges, J. L. 1989, *Int. Stat. Rev.*, 57, 238

- Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, *SoPh*, **293**, 28
- Forbes, T. 2000, in *Encyclopedia of Astronomy and Astrophysics*, ed. P. Murdin, 2295 (Bristol: IOP), doi:10.1888/0333750888/2295
- Georgoulis, M. K., & Rust, D. M. 2007, *ApJL*, **661**, L109
- Glogowski, K., Bobra, M. G., Choudhary, N., Amezcuca, A. B., & Mumford, S. J. 2019, *JOSS*, **4**, 1614
- Hagyard, M. J., Smith, J. B., Teuber, J., & West, E. A. D. 1984, *SoPh*, **91**, 115
- Hahn, M., Gaard, S., Jibben, P., Canfield, R. C., & Nandy, D. 2005, *ApJ*, **629**, 1135
- Hamdi, S. M., Kempton, D., Ma, R., Boubrahimi, S. F., & Angryk, R. A. 2017, in 2017 IEEE Int. Conf. on Big Data (Big Data), (Piscataway, NJ: IEEE), 2543
- Hapgood, M. A. 2011, *AdSpR*, **47**, 2059
- Hazra, S., Nandy, D., & Ravindra, B. 2015, *SoPh*, **290**, 771
- Hazra, S., Sardar, G., & Chowdhury, P. 2020, *A&A*, **639**, A44
- Huang, X., Wang, H., Xu, L., et al. 2018, *ApJ*, **856**, 7
- Huang, X., Yu, D., Hu, Q., Wang, H., & Cui, Y. 2010, *SoPh*, **263**, 175
- Jiao, Z., Sun, H., Wang, X., et al. 2020, *SpWea*, **18**, e02440
- Jing, J., Song, H., Abramenko, V., Tan, C., & Wang, H. 2006, *ApJ*, **644**, 1273
- Jonas, E., Bobra, M., Shankar, V., Todd Hoeksema, J., & Recht, B. 2018, *SoPh*, **293**, 48
- LaBonte, B. J., Georgoulis, M. K., & Rust, D. M. 2007, *ApJ*, **671**, 955
- Lan, R.-S., Jiang, Y., Ding, L.-G., & Yang, J.-W. 2012, *RAA*, **12**, 1191
- Leka, K. D., & Barnes, G. 2003a, *ApJ*, **595**, 1277
- Leka, K. D., & Barnes, G. 2003b, *ApJ*, **595**, 1296
- Leka, K. D., & Barnes, G. 2003c, *ApJ*, **595**, 1277
- Leka, K. D., & Barnes, G. 2003d, *ApJ*, **595**, 1296
- Li, R., Wang, H.-N., He, H., Cui, Y.-M., & Zhan-LeDu. 2007, *ChJAA*, **7**, 441
- Li, X., Zheng, Y., Wang, X., & Wang, L. 2020, *ApJ*, **891**, 10
- Linton, M. G., Longcope, D. W., & Fisher, G. H. 1996, *ApJ*, **469**, 954
- Liu, C., Deng, N., Wang, J. T. L., & Wang, H. 2017, *ApJ*, **843**, 104
- Liu, H., Liu, C., Wang, J. T. L., & Wang, H. 2019, *ApJ*, **877**, 121
- Ma, R., Boubrahimi, S. F., Hamdi, S. M., & Angryk, R. A. 2017, in 2017 IEEE Int. Conf. on Big Data (Big Data) (Piscataway, NJ: IEEE), 2569
- Mehta, P., Bukov, M., Wang, C.-H., et al. 2019, *PhR*, **810**, 1
- Moore, R. L., Falconer, D. A., & Sterling, A. C. 2012, *ApJ*, **750**, 24
- Muranushi, T., Shibayama, T., Muranushi, Y. H., et al. 2015, *SpWea*, **13**, 778
- Nandy, D. 2008, in ASP Conf. Ser. 383, *Subsurface and Atmospheric Influences on Solar Activity*, ed. R. Howe, R. W. Komm, K. S. Balasubramaniam, & G. J. D. Petrie (San Francisco, CA: ASP), 201
- Nandy, D. 2021, *SoPh*, **296**, 54
- Nandy, D., Hahn, M., Canfield, R. C., & Longcope, D. W. 2003, *ApJL*, **597**, L73
- Nandy, D., Mackay, D. H., Canfield, R. C., & Martens, P. C. H. 2008, *JASTP*, **70**, 605
- Nandy, D., Martens, P. C. H., Obridko, V., Dash, S., & Georgieva, K. 2021, *PEPS*, **8**, 40
- Nishizuka, N., Kubo, Y., Sugiura, K., Den, M., & Ishii, M. 2020, *ApJ*, **899**, 150
- Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, *ApJ*, **835**, 156
- Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2018, *ApJ*, **858**, 113
- Pal, S., Gopalswamy, N., Nandy, D., et al. 2017, *ApJ*, **851**, 123
- Pal, S., Nandy, D., Srivastava, N., Gopalswamy, N., & Panda, S. 2018, *ApJ*, **865**, 4
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *JMLR*, **12**, 2825
- Pereira, R. B., Plastino, A., Zadrozny, B., & Merschmann, L. H. 2018, *Inf. Process. Manage.*, **54**, 359
- Priest, E. R., & Forbes, T. G. 2002, *A&ARv*, **10**, 313
- Qahwaji, R., & Colak, T. 2007, *SoPh*, **241**, 195
- Qu, M., Shih, F. Y., Jing, J., & Wang, H. 2003, *SoPh*, **217**, 157
- Raboonik, A., Safari, H., Alipour, N., & Wheatland, M. S. 2017, *ApJ*, **834**, 11
- Ribeiro, F., & Gradwohl, A. 2021, *A&C*, **35**, 100468
- Sadykov, V. M., & Kosovichev, A. G. 2017, *ApJ*, **849**, 148
- Scherrer, P. H., Schou, J., Bush, R. I., et al. 2012, *SoPh*, **275**, 207
- Schrijver, C. J. 2007, *ApJL*, **655**, L117
- Schrijver, C. J. 2015, *SpWea*, **13**, 524
- Schrijver, C. J., DeRosa, M. L., Metcalf, T., et al. 2008, *ApJ*, **675**, 1637
- Schrijver, C. J., Kauristie, K., Aylward, A. D., et al. 2015, *AdSpR*, **55**, 2745
- Sinha, S., Srivastava, N., & Nandy, D. 2019, *ApJ*, **880**, 84
- The SunPy Community, Barnes, W. T., Bobra, M. G., et al. 2020, *ApJ*, **890**, 68
- Song, H., Tan, C., Jing, J., et al. 2009, *SoPh*, **254**, 101
- Tin Kam Ho 1995, in Proc. 3rd Int. Conf. on Document Analysis and Recognition (Washington, DC: IEEE Computer Society), 278
- Wang, J., Liu, S., Ao, X., et al. 2019, *ApJ*, **884**, 175
- Watanabe, K., Masuda, S., & Segawa, T. 2012, *SoPh*, **279**, 317
- Welsch, B. T., Li, Y., Schuck, P. W., & Fisher, G. H. 2009, *ApJ*, **705**, 821
- Woodcock, F. 1976, *MWRv*, **104**, 1209
- Yeates, A. R., Attrill, G. D. R., Nandy, D., et al. 2010, *ApJ*, **709**, 1238
- Yu, D., Huang, X., Wang, H., & Cui, Y. 2009, *SoPh*, **255**, 91
- Yuan, Y., Shih, F. Y., Jing, J., & Wang, H.-M. 2010, *RAA*, **10**, 785
- Zheng, Y., Li, X., & Wang, X. 2019, *ApJ*, **885**, 73