








How to Train Your Flare Prediction Model: Revisiting Robust Sampling of Rare Events

Azim Ahmadzadeh¹, Berkay Aydin¹ , Manolis K. Georgoulis² , Dustin J. Kempton¹ , Sushant S. Mahajan³ , and Rafal A. Angryk¹ 

¹ Georgia State University, Atlanta, GA 30302, USA; aahmadzadeh1@cs.gsu.edu

² RCAAM of the Academy of Athens, Soranou Efessiou 4, Athens, GR-11527, Greece

³ Institute for Astronomy, University of Hawai'i at Mānoa, Honolulu, HI, USA

Received 2021 January 31; revised 2021 March 1; accepted 2021 March 5; published 2021 May 17

Abstract

We present a case study of solar flare forecasting by means of metadata feature time series, by treating it as a prominent class-imbalance and temporally coherent problem. Taking full advantage of pre-flare time series in solar active regions is made possible via the Space Weather Analytics for Solar Flares (SWAN-SF) benchmark data set, a partitioned collection of multivariate time series of active region properties comprising 4075 regions and spanning over 9 yr of the Solar Dynamics Observatory period of operations. We showcase the general concept of temporal coherence triggered by the demand of continuity in time series forecasting and show that lack of proper understanding of this effect may spuriously enhance models' performance. We further address another well-known challenge in rare-event prediction, namely, the class-imbalance issue. The SWAN-SF is an appropriate data set for this, with a 60:1 imbalance ratio for GOES M- and X-class flares and an 800:1 imbalance ratio for X-class flares against flare-quiet instances. We revisit the main remedies for these challenges and present several experiments to illustrate the exact impact that each of these remedies may have on performance. Moreover, we acknowledge that some basic data manipulation tasks such as data normalization and cross validation may also impact the performance; we discuss these problems as well. In this framework we also review the primary advantages and disadvantages of using true skill statistic and Heidke skill score, two widely used performance verification metrics for the flare-forecasting task. In conclusion, we show and advocate for the benefits of time series versus point-in-time forecasting, provided that the above challenges are measurably and quantitatively addressed.

Unified Astronomy Thesaurus concepts: [Solar flares \(1496\)](#); [Solar physics \(1476\)](#); [Classification \(1907\)](#)

1. Introduction

Data collected for addressing real-world problems are almost never clean and ready to use, no matter how carefully a screening process was carried out. Inevitably, there are some challenges inherited, either by the nature of the studied subject or by the data collection strategy, that should be identified, understood, and dealt with accordingly. Class imbalance and temporal coherence are two of such challenges that are present in many nonlinear dynamical systems such as earthquake prediction, fraud detection, and weather forecasting. This study revisits their impact in another natural manifestation of these issues, namely, solar flare prediction.

Solar flares are sudden and substantial enhancements of radiation spanning over the entire electromagnetic spectrum, including its high-energy part (extreme-ultraviolet, X-rays, γ -rays). They occur at local scales in the Sun and pose a threat to humans and equipment in space. Since 1974, X-ray flares are automatically detected and classified by the National Oceanic and Atmospheric Administration's (NOAA) GOES satellites in the 1–8 Å wavelength range. Based on peak soft X-ray flux in this range, flares are logarithmically classified as A, B, C, M, and X, from weaker to stronger, starting from 10^{-8} W m^{-2} . Therefore, an X-class flare is generally 10 times stronger, in terms of peak flux, than an M-class flare and 100 times stronger than a C-class flare. Within each class there is a subclass denomination from 1 to 9.⁴ When the X-ray background level is high, as typically occurs at times of elevated solar activity, A- and B-class flares are often difficult or impossible to discern, while C-class flares and above are mostly detected, particularly

above level C2 (Wagner 1988; Veronig et al. 2004; Aschwanden & Freeland 2012). The most intense flares, namely, the M and X classes, are commonly targeted for prediction owing to their potentially adverse space weather ramifications. In spite of more than 20 yr of research and meaningful advances, flare prediction remains a largely outstanding problem (e.g., Leka et al. 2018).

Some of the major challenges the flare-forecasting researchers are up against are rooted in the rarity of the events of interest, the high dimensionality of observational data, and the dynamic behavior of the Sun. Below we briefly review these challenges and a number of relevant studies.

Extreme Class Imbalance. The frequency distribution of the peak X-ray fluxes of solar flares is a robust power law with a dynamical range spanning several orders of magnitude. A common interpretation of flares views them as stochastic manifestations of self-organized criticality (see, e.g., the comprehensive review of Aschwanden et al. 2016, and numerous references therein).

A statistical analysis of the flares reported by the NOAA during solar cycle 23 (1996–2008) shows that around 50% of active regions produce at least one C-class flare, with approximately 10% of them producing at least one M-class flare and less than 2% producing at least one X-class flare (Georgoulis 2012b). Solar cycle 24 (2009 to present), according to the data set created by Angryk et al. (2020), exhibits a weaker major flare crop at a similar number of active regions to solar cycle 23, further highlighting the importance of effectively treating class imbalance.

Point-in-time versus Time Series Forecasting. Solar flare forecasting has been humanity's first attempt toward space weather forecasting. As such, numerous magnetic properties

⁴ https://www.nasa.gov/mission_pages/sunearth/news/X-class-flares.html

and forecast methods have been proposed since the early 1990s (see, e.g., Georgoulis 2012a; Barnes et al. 2016, for nonexhaustive reviews), while studies facilitating several predictive properties were performed by many others (see, e.g., Leka & Barnes 2003b, 2007; Qahwaji & Colak 2007; Barnes & Leka 2008; Welsch et al. 2009; Georgoulis 2012b; Bobra & Couvidat 2015). The first comprehensive comparison between multiple flare prediction methods was undertaken by Barnes et al. (2016), while comparative evaluations between operational and pre-operational methods were implemented by Leka et al. (2019a, 2019b) and Park et al. (2020).

A quick perusal of the voluminous literature, however, will show that the majority of these methods correspond to point-in-time forecasting, namely, to using the instantaneous value of one or more parameters in order to produce a binary or probabilistic flare forecast over a preset forecast horizon. But flares are an inherently dynamical phenomenon, with clear pre-flare and post-flare phases, characterized by certain evolutionary trends (e.g., Benz 2008; Fletcher et al. 2011). Because of this, time series of candidate flare forecast parameters should be used, rather than isolated points in time. Quite likely, the level of difficulty in assimilating time series of predictive parameters in flare forecasting was the main reason for utilizing (possibly over)simplified point-in-time forecasts in many cases. This compromise, however, may have hampered further progress: early works, as well as recent ranking efforts of predictive flare parameters, show that previous flare history is a significant factor to consider (Leka & Barnes 2003b; Barnes & Leka 2008; Falconer et al. 2012; Campi et al. 2019; Leka et al. 2019b; Park et al. 2020), hinting toward the need to study the pre-flare temporal evolution in terms of both parameter time series and flare history.

Nonrepresentative Data Sets. While there is no shortage of ground- and space-based telescopes mapping the magnetic field of the Sun’s photosphere over the past 30 yr (Cacciani et al. 1990; Balasubramaniam & West 1991; Scherrer et al. 1995, 2012; Lites 1996; Mickey et al. 1996; Spirock et al. 2001; Tsuneta et al. 2008), and new instruments are just about to be commissioned (Rimmele et al. 2020; Solanki et al. 2020), it is clear that (i) the temporal span of space-based vector magnetographic data is still limited to one solar cycle and (ii) training forecast methods on certain parts of a solar cycle is not necessarily optimal for forecasting other parts of the same and/or different cycles, due to the continuously modulating background of magnetic activity (see, e.g., the discussion on varying flare occurrence, or climatology, in McCloskey et al. 2018; Leka et al. 2019b). This heterogeneity, coupled with random undersampling for majority-class events, causes the sampled subsets of data to become nonrepresentative of the overall flare population. Therefore, in many cases, the performance of forecasting models may critically depend on the sampling strategy.

Each of the challenges mentioned above requires a proper treatment. Some correspond to the data collection phase, some to the cleaning and preprocessing phase, and some to the learning phase. While many studies have revisited these challenges (to name a few in geo- and space science domains, see Woodcock 1976; Bloomfield et al. 2012; Camporeale 2019), there is an obvious emphasis by interested communities on the performance of forecast models. Less obvious, however, is that this emphasis often comes at the expense of the models’ robustness. Typical forecast metrics used for performance

verification in flare-forecasting models are the true skill statistic (TSS) and realizations of the Heidke skill score (HSS; HSS2)—see Section 5. Recent works with apparently high, but perhaps not particularly robust, performance include Hamdi et al. (2017) (TSS ≥ 0.8 or even ≥ 0.9 for a 24 hr prediction of flares greater than M1.0) and Nishizuka et al. (2017) (TSS ≥ 0.9 for similar forecast settings). In other well-cited studies (for a comparison, see the reports in Bobra & Couvidat 2015; Barnes et al. 2016), the highest TSS that maintains a similar HSS2 varies around ≈ 0.6 . We therefore understand that without a major change in the learning process, exceedingly high TSS values may simply be a result of suboptimal preprocessing. Some factors crucial to a model’s performance verification are preprocessing steps, sampling strategies, and the choice of performance evaluation metrics. A model’s performance varies when any of these factors undergoes changes. However, a robust model is impacted only minimally relative to others as a result of these changes. That is, a robust model is less prone to overfitting, a known phenomenon that describes a model that is too closely fit to a limited set of data points and therefore does not generalize well. Throughout this study, we frequently draw a comparison between robustness and performance regarding different practices.

The rest of this paper is organized as follows: In Section 2, we first briefly introduce the SWAN-SF, the data set we run our experiments on. This data set allows us to showcase the concept of the class imbalance (Section 2.2) and temporal coherence (Section 2.3) of data. We also discuss data normalization (Section 2.4) and hyperparameter tuning (Section 2.5) of models. Section 3 then discusses the choice of the forecast data set from the SWAN-SF, while in Section 4 we present some of the major treatments regarding class-imbalance (Section 4.3) and temporal-coherence (Section 4.4) issues. Before presenting our experiments and the results, we briefly review in Section 5 the performance verification metrics we use. In Section 6, we discuss the details and results of a number of experiments we designed for examining the above-mentioned challenges and illustrate the impact of the corresponding treatments. We conclude in Section 7 by listing the key lessons learned from these experiments.

2. Benchmark Data Set and Challenges

In this section, we briefly discuss the SWAN-SF benchmark data set of Angryk et al. (2020) and the major issues intrinsic to this and similar data sets. If not properly treated, these issues may well lead to overestimation of the forecasts’ performance.

2.1. SWAN-SF Benchmark Data Set

Multiple flare prediction studies (Bloomfield et al. 2012; Bobra & Couvidat 2015; Barnes et al. 2016) and the European Union FLARECAST project (Benvenuto et al. 2018; Florios et al. 2018; Campi et al. 2019) emphasize machine learning for flare prediction, but they use point-in-time measurements. Given different periods of observation and partitioning/sampling strategies, we are unable to determine whether the potential differences in accuracy or skill score values in these studies reflect the intrinsic stochasticity in flare occurrence, the specific preprocessing and sampling strategies carried out, the implementation of machine-learning models, or perhaps a combination of the above. This is the core reason behind the development of the Space Weather Analytics for Solar Flares

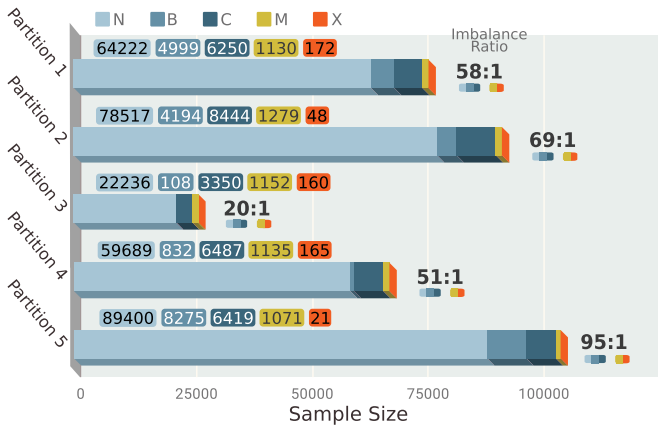


Figure 1. Stacked bar plot of multivariate time series populations for each of the five flare classes across different partitions of the SWAN-SF benchmark data set. Flare classes X, M, C, and B are taken and validated from GOES classification, while N denotes flare-quiet and GOES A-class instances. The annotated populations are stemming from the current time series slicing strategy of the SWAN-SF with a 1 hr step size, a 12 hr observation window, and a 24 hr prediction window. The sliced multivariate time series are labeled with the class of the strongest reported flare, if any, within the respective prediction window.

(SWAN-SF) benchmark data set, composed entirely of multivariate time series. Its purpose is to facilitate unbiased flare forecasting and hopefully make strides toward nonincremental future improvements in forecasting.

The data points of the SWAN-SF benchmark data set are labeled by five different flare classes, namely, GOES X, M, C, B, and a nonflaring class denoted by N. Therefore, N includes flare-quiet instances and GOES A-class events. We hereafter omit GOES from the above classes, for brevity. This data set comprises five temporally segmented partitions and is designed in such a way that each partition includes approximately the same number of X- and M-class flares (Figure 1). The data set contains various time series parameters derived from solar photospheric magnetograms along with NOAA’s flare history of active regions. Magnetograms and their metadata are provided by the Solar Dynamics Observatory’s (Pesnell et al. 2012) HMI Active Region Patches (HARP) data product (Hoeksema et al. 2014). The magnetic field parameters are physics based and were originally taken by the Space weather HMI Active Region Patches (SHARP) data product (Bobra et al. 2014)), but they were recalculated for validation purposes and enhanced with parameters not present in SHARPs (see Table 1 in Angryk et al. 2020).

The data points in this data set are time series slices of parameters extracted from solar active regions in a sliding fashion. That is, for a particular flare with a unique ID, k equal-length multivariate time series are collected from a fixed period of time in the pre-flare history. This is called the *observation window*, T_{obs} . Denoting s_i as the starting point of the i th slice of the multivariate time series, the $(i+1)$ th slice starts at $s_i + \tau$, where τ is the step size in the sliding process. Each of the extracted multivariate time series is assigned a label, determined by the class of the strongest flare recorded during a fixed-size temporal window that follows T_{obs} . This window is the *prediction window*, T_{pred} , and starts precisely at the end of T_{obs} . In the SWAN-SF, T_{obs} and T_{pred} span over 12 and 24 hr, respectively, with $\tau = 1$ hr. Placing the prediction window right after the observation window implies a zero latency (i.e., a

finite time before a given forecast comes into effect) in our tests, for simplicity.

2.2. Class Imbalance of Data

We characterize a data set as class imbalanced when the population of one or more data classes are significantly smaller than those of the majority classes. The data points of the smaller group are called the *minority* instances. If forecasting is intended for them, they are also called *positive* instances. Conversely, the other group’s data points are called the *majority*, or *negative*, instances, in case minority instances are the forecast targets. The SWAN-SF benchmark data set exhibits extreme class imbalance: the imbalance ratios for each partition of the data set and for using GOES M- and X-class flares as the minority class are annotated in Figure 1.

Class imbalance is a well-known concept in the machine learning and statistics communities (Kubat & Matwin et al. 1997; Japkowicz & Stephen 2002; Ganganwar 2012; Krawczyk 2016) but has also been addressed in many other domains, including atmospheric sciences and flare forecasting (Woodcock 1976; Bloomfield et al. 2012; Jolliffe & Stephenson 2012; Bobra & Couvidat 2015). However, the complexity of the forecasting problem sometimes causes this important issue to go relatively unnoticed, resulting in a large variance in the reported performance of the models, as we briefly pointed out in the Introduction.

The class-imbalance issue can be roughly explained as follows: classification models, in general, aim to optimize their cost functions by minimizing the total number of misclassifications. Given the significantly higher density of the majority class, a correct classification on the decision boundary, where two or more classes overlap, is typically accompanied by multiple misclassified majority instances. The support vector machine (SVM) classifier (Vapnik 1963), for example, calculates an optimal hyperplane that makes this separation (Borges 1998; Ben-Hur & Weston 2010). Due to the imbalance between the classes, a hyperplane that is (ideally) supposed to pass through the decision boundary will be shifted farther into the region of the minority class (classifying almost all the data points as the majority class) to reduce the total number of incorrect classifications. This results in higher true negatives (i.e., correct classification of CBN-class flares) and lower true positives (i.e., correct classification of XM-class flares as per the example of Figure 1). In other words, a model in a class-imbalanced space always tends to favor the majority class. This is of particular concern because flare-forecasting research focuses on the minority, rather than majority, instances.

Another angle to this problem is the determination of appropriate performance measures. Many well-known performance metrics are significantly impacted by class imbalance (Hossin & Sulaiman 2015), including *accuracy*, *precision* (but not *recall*), and hence the *F1-score*. This is mainly because these measures ignore the number of misclassifications. For instance, a naive model that classifies all instances as the negative (majority) class may result in a very high (often asymptotic to 1.0) accuracy, while learning very little or nothing about the minority class. In Section 5, we review some popular metrics used in flare forecasting. Furthermore, in Section 4.3 we discuss the classical approaches to tackle the class-imbalance issue.

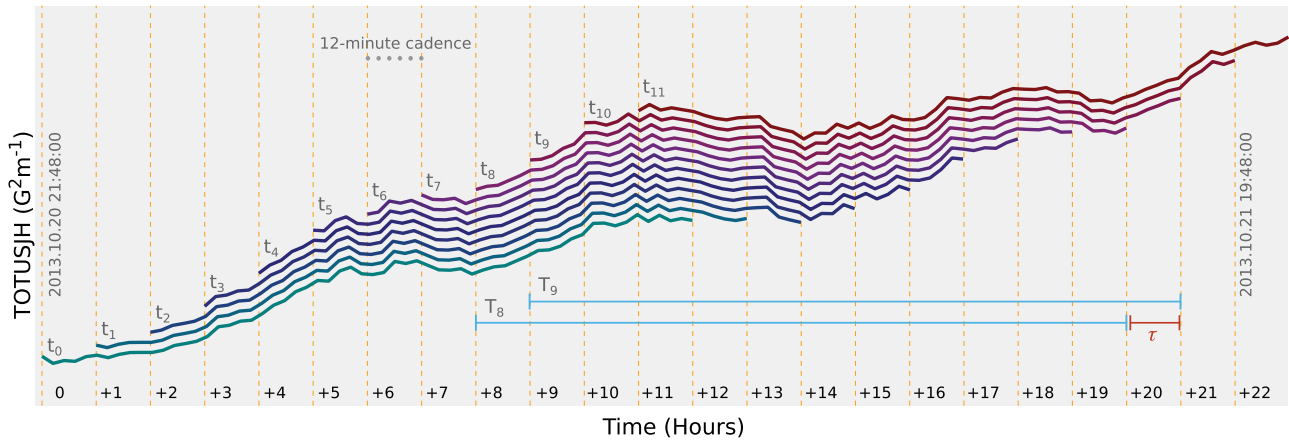


Figure 2. Twelve consecutive time series slices for the parameter *Total Unsigned Current Helicity* (TOTUSJH) corresponding to an M1.0-class flare associated with NOAA AR 11875 (HARP 3291). Each time series spans over 12 hr of observation, with a 12-minute cadence. The blue interval segments T_8 and T_9 , for example, show the domain of time series t_8 and t_9 , respectively, with τ representing the slicing step size, i.e., 1 hr. The vertical spacing of each time series slice along the y-axis has been added manually to make different time series discernible.

2.3. Temporal Coherence of Data

The sliding observation window mentioned in Section 2.1 is crucial for operational prediction on real-time data. Indeed, a continuous and uninterrupted observation is needed for training a forecast model. This continuity dictates the existence of partially overlapping slices of multivariate time series and, therefore, nonindependent data points. A characteristic example is provided in Figure 2, where 12 consecutive 12 hr slices of the parameter *Total Unsigned Current Helicity* (TOTUSJH), extracted from an active region with HARP number 3291, are visualized. From this example, one clearly notices that a physical parameter of an active region may not be expected to behave significantly differently from a slice to its adjacent one. Any two adjacent time series are identical for a period of 11 hr, and that accounts for more than 90% (precisely, $\frac{11}{12} = 0.92$) of each time series’ length. Of course, this is not specific to the particular parameter used in the example but is a characteristic inherited from the 1 hr time-stepped slicing methodology.

Let us now look at the example of Figure 2 more closely: these very similar slices, identical for 11 hr, will appear as a cluster of data points in the multidimensional feature space, located close to each other. This clustering is apparent in Figure 3, where we plot the *mean* versus the *standard deviation* of the TOTUSJH slices of Figure 2 for the same active region (red/darker circles) and compare them with assorted means and standard deviations from different active regions on the same parameter (blue/lighter circles). For an adequate comparison, in terms of flare label and solar cycle phase, all blue instances correspond to the same flare label (M1.0) and partition (partition 3). As expected, the *mean* values of the time series sampled from the same active region (red/darker circles) are confined to a very narrow subspace, i.e., limited to the interval (2300, 2400) ($G^2 m^{-1}$) of the *mean* dimension. Any forecast method treating this as a potentially useful pattern in the data is erring in this case, because this “pattern” is simply a by-product of the slicing strategy used to create the benchmark data. Classification algorithms are generally designed to decipher such issues. To avoid misleading them by introducing artificial clusters, nonetheless, data points from the same active region (such as the red points of Figure 3) should not be considered as entirely distinct and independent.

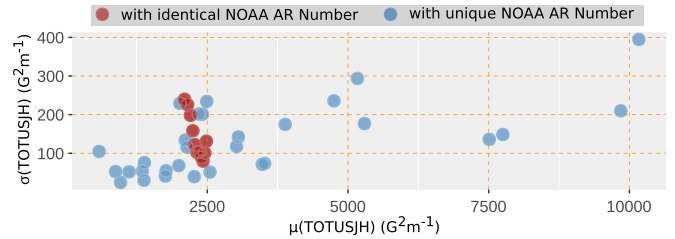


Figure 3. Standard deviation (σ) vs. mean (μ) of parameter TOTUSJH, for two different groups: one (red/darker circles) extracted from the consecutive slices of Figure 2, and another (blue/lighter circles) from a randomly selected set of time series slices for different active regions. Both groups correspond to M1.0-class flare labels and stem from Partition 3 of the SWAN-SF. The apparent clustering of red circles is due to the temporal coherence of consecutive time series.

The above discussion, as the reader may have already noticed, can be traced back to the well-known assumption that random variables must be *independent and identically distributed* (i.i.d.) for the models to make sense. Although this practice may not look like a challenge, it is easily overlooked on multiclass, multivariate time series data. Randomly sampling and shuffling the instances is often assumed to tackle the problem, but, as our example showed, this may not always be the case. More importantly, when such clusters are allowed to exist in both the training and validation sets, the high performance on the *unseen* data (of the validation set) may convince researchers that the classifier was able to generalize the learned patterns and forecast flares. Such a superficial performance is exemplified in Experiment D of Section 6. It is also important to note that as the dimensionality of the feature space increases (by including more statistical features), spotting such clusters becomes more and more difficult; in high-dimensional data, it is very likely that many dimensions are irrelevant but they mask the existing clusters (Parsons et al. 2004).

While this concept is generally known as the independence of random variables, we prefer to use our term, *temporal coherence*, for two reasons: first, while the typical solution for preserving the independence between data points is random sampling, this action does not resolve the issue in this special case. Second, we would like to avoid the generality of the term “independence” and instead point to the direct cause of the

issue, i.e., the temporal overlap. We therefore refer to a data set as *temporally coherent*⁵ if (a) it comprises partially identical samples and (b) this overlap is caused by the slicing methodology used to create the data set, regardless of its nature, or the topic of study.

Looking at the consecutive slices shown in Figure 2, it is evident that the two conditions of temporal coherence are satisfied in the SWAN-SF because (a) the temporally close time series are partially identical, i.e., t_i and t_{i+1} are identical within an 11 hr interval out of the 12 hr they each span over, and (b) this is caused by the slicing of parameter time series designed to ensure both continuity and detailed coverage, by allowing a time step much smaller than the observational window (i.e., 1 hr vs. 12 hr). Therefore, the SWAN-SF can be described as a temporally coherent data set, and the independence of data points should not be presumed. Note that we use the term “partially identical” (instead of “similar”) to emphasize that what defines the temporal coherence is the temporal overlap and not the correlation between time series’ values. The former is often a by-product of the slicing methodology in the data preparation process, whereas “similarity” generally refers to the homogeneity among observations.

2.4. Data Normalization

The ranges of all feature vectors of a data set are typically transformed into a unified range before the training phase starts. This preprocessing step is called normalization and is mainly performed because it allows models to distinguish between different patterns and structures independent of the physical units and dynamical ranges of each parameter. The simplicity of this concept often leads to overlooking the variable impact of different ways of normalization (Al Shalabi & Shaaban 2006; Yu et al. 2009). Regardless of which transformation function one may use (e.g., linear, nonlinear, or data-driven), normalization can be done “locally” or “globally.” A global normalization takes into account the global statistics, i.e., *min* and *max* of each feature/parameter over the entire data set, whereas a local transformation may consider local extrema of different subsets (e.g., partitions, or training and testing sets) to normalize each of those subsets separately. Although it is common practice to apply global normalization, for the SWAN-SF and similar data sets this should be put to the test owing to the modulating event occurrence rate throughout solar cycle 24. In such cases, a global normalization could impact some features negatively by transforming their values into a too narrow or too wide range. We illustrate this in Experiment E in Section 6.4, by showing the significantly different performance levels using different normalization strategies.

2.5. Hyperparameter Tuning

Temporal coherence affects also the way we tune the hyperparameters of our models. Any supervised learning model requires optimization of its hyperparameters in a data-driven manner. Since the hyperparameters should remain optimal over the entire data set (including the incoming data points, for an operational model), the training and validation sets must each be representative of the entire data set. In our data set, because of the temporal coherence, random sampling does not produce such subsets (see Experiment D of Section 6.3, which examines

the impact of temporal coherence). Hence, the tuning process remains confined to the partitions. In Section 4.4, while discussing a remedy for temporal coherence, we review the use of NOAA AR number that addresses this issue. However, it is still very likely that a model highly optimized on one partition corresponding to a particular phase of a solar cycle performs better than one that is optimized globally (Barnes et al. 2016; Leka et al. 2019a). We believe that hyperparameter tuning is a problem yet to be thoroughly investigated since at this point it is rather clear to us that flare forecasting has a dynamic behavior, for which ensemble models, which can be trained differently for different periods of a solar cycle, may be more appropriate (see, e.g., Guerra et al. 2015; Murray 2018; Guerra et al. 2020).

3. Derived Forecast Data Set

To use the SWAN-SF benchmark data set, two general approaches might be taken. One is to preprocess the time series and feed them directly to supervised models for prediction. The other is to extract a set of statistical features describing the time series and then train the supervised models on these descriptors. We choose the second approach in order to carry out our experiments without having to deal with the high dimensionality of the original data type, i.e., time series.

3.1. Feature Extraction

SWAN-SF is designed to rely entirely on time series forecasting of solar flares, if so desired. In a wealth of studies on flare forecast, only a relative minority of them have attempted to use time series (e.g., Leka & Barnes 2003a, 2003b; Barnes & Leka 2006; Welsch et al. 2009; Reinard et al. 2010; Cinto et al. 2020a). More recently, several studies moved toward time series forecasting (Ahmadzadeh et al. 2019b; Hostetter et al. 2019; Ma et al. 2020; Ji et al. 2020; Chen et al. 2020). In spite of a number of efforts, reports on which time series parameters perform better than others are not necessarily compelling. Since it is not the focus of this work to determine which time series characteristics will perform well, without loss of generality and in the interest of simplicity, we choose a limited set of four descriptive statistics of the time series for our experiments, namely, *median*, *standard deviation*, *skewness*, and *kurtosis*—previous studies cited above have also used them. Additionally, to allow comparison (at least partially) between point-in-time (such as Bobra & Couvidat 2015) and time series forecasting, and hence assess the potential benefits of using time series, we consider a point-in-time feature, namely, *last value*. This is precisely the last value of each time series. Except for the last experiment, where fully fledged time series forecasting is attempted, all other experiments exercise point-in-time forecasting using *last value*.

Prior to computing these statistical features, the data set requires a minimal preprocessing owing to some missing values, corresponding to <0.01% of the data. Preprocessing proceeds by first filling the missing values by linear interpolation and then using zero-one data transformation to normalize the data for our experiments. The obtained data set of extracted features has a dimensionality of 120, resulting from computing the five above-mentioned statistics (namely, the four descriptive statistics of the time series and their *last value*) on 24 of the physical parameters of the SWAN-SF benchmark data set. For several of these preprocessing steps,

⁵ The concept is here introduced in the context of data manipulation, not to be confused with “temporal coherence” met in optics or elsewhere.

we employed the open-source Python package MVTs-Data Toolkit (Ahmadzadeh et al. 2020).

3.2. Dichotomization of the Problem

To place the focus of our study on the challenges discussed in Section 2, we set up a simplified bi-categorical setting; we carry out experiments on binary-class data by merging X and M classes into a superclass called XM and, similarly, the classes C, B, and N into another superclass, denoted by CBN. The former becomes our minority (positive) class, while the latter constitutes our majority (negative) class.

4. Model and Treatments of Issues

In this section, we discuss some of the challenges in training machine-learning models, caused directly or indirectly by the two most important characteristics of our data set, namely, class imbalance and temporal coherence. In Section 4.1, we first justify why we use SVMs for analyzing the impact of different remedies for these two issues. Then, in Section 4.2, for those unfamiliar with this classifier, we present a high-level discussion on how it discriminates between different classes. In Sections 4.3 and 4.4, we review the basic remedies for the class-imbalance and temporal-coherence issues, respectively.

4.1. Control for Learning Algorithms

Without loss of generality and for keeping our focus on the above-mentioned issues, we control for the learning algorithms by limiting our experiments to one classifier, namely, SVMs. The generality is preserved because, first, the issue of temporal coherence can only be dealt with at the data level, during preprocessing, and does not involve the classifier. The class-imbalance issue, however, can be addressed either at the data level or at the algorithm level, or even both. Among different data-driven strategies (e.g., sampling and feature selection) or algorithm-driven strategies (one-class learning, cost-sensitive learning, ensemble learning), only cost-sensitive learning depends on the choice of the classifier (Ali et al. 2013). In cost-sensitive learning the class-imbalance issue is tackled by customization of the cost function so that it takes into account the imbalance ratio when penalizing misclassification. But even in this subset of remedies, although different classifiers may be impacted differently by the class-imbalance issue, they all will be impacted regardless (He & Garcia 2009; Krawczyk 2016). Therefore, the remainder of this paper can be reproduced with any other classifier without any change on the presented analyses.

4.2. SVM Classifier

SVMs are discriminative, supervised classifiers introduced by Vapnik (1963). The concept behind SVM is deeply rooted in the theory of statistical learning. The objective is to find an optimal d -dimensional hyperplane that separates the classes of data points, where d is the number of features, determining the dimensionality of the feature space. Such a hyperplane is said to be optimal if the average distance between the plane and all the data points on the decision boundary (i.e., the support vectors) is maximum, meaning that the two classes are best segregated. For more elaborate, multicategorical studies, a combination of multiple binary SVMs can be employed to

classify the data (see, e.g., Dietterich & Bakiri 1995; Hastie & Tibshirani 1997).

The SVM classifier owes its popularity primarily to its efficient learning of nonlinear decision surfaces, thanks to its support vectors and the transformation functions (*kernels*). Different kernels can be used to provide better transformations of data into new feature spaces where the data points are potentially more accurately separable. One of the most popular kernels is *radial basis function* (RBF; Cristianini & Shawe-Taylor 2000) and was found to be more effective than other kernels for this study (see Section 6). Like any function, kernels have one or more variables that need to be specified a priori. For RBF, these are the desired smoothness of the decision surface and the radius of influence of support vectors on forming the decision surface, C and γ , respectively. All kernels share the smoothness C as a trade-off between performance and simplicity of the decision surface. The a priori specification of SVM hyperparameters C and γ is the required hyperparameter tuning in this case (see also Burges 1998, for a tutorial). Although this step is crucial for training an accurate and robust model and needs to be carefully carried out on every variation of the preprocessed data, in this study we only tune the hyperparameters once (on the entire data set) and utilize their optimal values for all experiments. This simplification is made because, again, we are interested not in finding the best models but in highlighting the advantages and disadvantages of utilizing different preprocessing and pretraining practices.

4.3. Treatments of Class Imbalance

In Section 2, we discussed the class-imbalance issue and pointed to the SWAN-SF as an extremely class-imbalanced data set. In the following, we review some of the simplest, but effective, approaches for dealing with this issue. Later in Section 6, we verify these approaches and, using several experiments, compare the impact that each of these treatments has on the robustness of the trained models.

4.3.1. Undersampling and Oversampling

A simple approach for tackling class imbalance in model training is to enforce class balance by either *undersampling* (that is, taking out instances from the majority class) or *oversampling* (that is, replicating instances of the minority class). This can achieve a nearly 1:1 balance ratio between the negative and positive samples in the training phase. This solution, however, comes at a cost: when undersampling, we leave out a significant portion of the data during training, preventing the model from taking full advantage of the entire data collection. When oversampling, on the other hand, we add replicates of the existing instances, causing the model to memorize the patterns (hence overfitting), rather than generalizing. Bearing in mind the potential negative impacts caused by either of these practices, some studies recommend undersampling over oversampling (Drummond et al. 2003). That said, one should decide about employing these models only by trial and error on their specific data sets. It is crucial to note that a series of more advanced, synthetic sampling techniques aiming to mitigate these impacts are long-standing (see, e.g., Chawla et al. 2002, and references therein). The key unknown with such methods, however, is the reliability of the generated samples, i.e., the difficulty of assessing how well the synthetic data align with the actual distributions. In our experiments, we

Table 1

Variations of Undersampling (US) and Oversampling (OS) Approaches Applied to Partition 3 of the SWAN-SF Data Set, Showing the Population of Classes (Δ) and Expansion/Shrinkage Factors (\square) in Each Scenario

Method		X	M	C	B	N	Method Description
	\blacktriangle	160	1152	3350	108	22236	
US1	Δ	160	1152	171	5	1135	Preserves climatology in subclass level
	\square	(1.00)	(1.00)	(0.05)	(0.05)	(0.05)	
US2	Δ	160	160	106	106	106	X-based undersampling; enforces a subclass balance
	\square	(1.00)	(0.14)	(0.03)	(0.98)	(0.00)	
US3	Δ	1152	1152	768	768	768	M-based undersampling; enforces a subclass balance
	\square	(7.20)	(1.00)	(0.23)	(7.11)	(0.03)	
OS1	Δ	3133	22560	3350	108	22236	Preserves climatology in subclass level
	\square	(19.58)	(19.58)	(1.00)	(1.00)	(1.00)	
OS2	Δ	1225	8824	3350	108	6592	Similar to OS1, but it suppresses N
	\square	(7.66)	(7.66)	(1.00)	(1.00)	(0.30)	
OS3	Δ	5025	5025	3350	3350	3350	C-based oversampling; enforces a subclass balance
	\square	(31.41)	(4.36)	(1.00)	(31.02)	(0.15)	
OS4	Δ	33354	33354	22236	22236	22236	N-based oversampling; enforces a subclass balance
	\square	(208.46)	(28.95)	(6.64)	(205.89)	(1.00)	

Note. In the header, the initial population of each class is shown (\blacktriangle). Methods US1, US2, and US3 are compared in Experiment F of Section 6.5. US2 and OS3 are used in Experiments A and B of Section 6.2, respectively. These quantities tabulated here do not represent the number of unique flare instances in this time period, but the number of data points collected by means of a sliding observation window with a 12 hr observation window and a 24 hr forecast window.

restrict ourselves to the simpler strategies because they (1) establish lower bounds for the performance boost that can be obtained by sampling and (2) exhibit the sampling nuances more clearly in the absence of complex assumptions that are embedded in advanced synthetic sampling strategies.

While both undersampling and oversampling, in their simplest settings, seem fairly straightforward and easy to implement, one should be particularly careful when applying them to a multiclass data, such as the flare prediction data set. This holds despite the fact that we later (in Section 3.2) convert our data set to a bi-categorical class problem. The concern is that different possibilities exist depending on whether a balance at subclass level is also desired (i.e., $|X| = |M|$ and $|C| = |B| = |N|$). In case of undersampling, therefore, we need to decide which minority subclasses should be considered the “base” class. Letting X be the base class, we must first undersample from M-class flares to balance X and M classes and then undersample from the majority class (CBN). This yields a balanced data set at both superclass and subclass levels. For convenience, we call this undersampling method an *X-based undersampling*, since the population of X class is preserved. Similarly, other sampling methods can be introduced. Some examples of different sampling methods on the SWAN-SF data set are shown in Table 1.

A quick look in Table 1 shows that the choice of the sampling method plays a critical role in tackling the classification problem. Each sampling method has to contort the flare climatology (namely, the mean occurrence frequency of flares in the time span of the data set) in order to achieve a desired balance. This change affects the distribution of samples in the feature space by making the decision boundary either denser or sparser. A denser decision boundary makes it more difficult for the classification’s hyperplane to separate the positive and negative instances correctly. Conversely, a sparser decision boundary presents an easier classification problem, i.e., with fewer misclassifications. In the bi-categorical case (XM vs. CBN), the decision boundary will most likely be where C and M classes overlap. Knowing, for example, that from the two superclasses the C- and M-class instances are

more similar to one another (than C to X, or M to B and N), it is expected that the classification problem will become somewhat easier if the chosen sampling strategy reduces the populations of either of these two classes. A good example of this situation is the oversampling method OS3 in Table 1. There, X-class flares are replicated more than M-class flares (by a factor of 31.41 vs. 4.36), while the number of C-class flares is kept unchanged. The decision boundary, therefore, becomes sparser than it was in the unsampled data, with relatively more X- and B-class instances reducing the overall likelihood for misclassification in the training phase, hence having an easier problem in hand. Similarly for undersampling methods, both US2 and US3 yield denser decision boundaries than US1. That the sampling method US1 indeed simplifies the flare-forecasting problem is verified in Experiment F of Section 6.5 and illustrated in Figure 5.

Looking at Table 1, it is also important to note that in order to achieve a particular balance sometimes both undersampling and oversampling should take place simultaneously. For example, in US3, while C- and N-class instances are under-sampled, X- and B-class instances are replicated roughly over 7 times, to produce the desired balance. The opposite also takes place in OS2 and OS3 methods, where N-class instances are under-sampled while others are oversampled.

4.3.2. Misclassification Weighting

Another known remedy for the class-imbalance problem is penalizing misclassification of each class differently. The SVM, like most machine-learning algorithms, can incorporate different weights in its cost function. For details on how this is implemented mathematically, we refer the interested reader to Ben-Hur & Weston (2010). In this study, wherever we use misclassification weighting, we adjust the weights using the formula $w_j = \frac{n}{k \cdot n_j}$, where n is the total sample size, n_j is the sample size of class j , and k represents the number of classes. To give an example, in a bi-categorical data set with two classes, A and B, where $|A| = 100$ and $|B| = 900$, the weights would be $w_A = \frac{1000}{2 \times 100} = 5$ and $w_B = \frac{1000}{2 \times 900} = 0.56$.

Therefore, misclassification of the minority instances (i.e., A) will be penalized 5 times more than that in the balance case, whereas for the majority instances this weight actually reduces the penalty by about one-half. The impact of this approach on tackling the class-imbalance issue is compared with the previously listed solutions in Experiment C of Section 6.2.

4.4. Treatment of Temporal Coherence

Temporal coherence, as discussed in Section 2.3, appears to be due to the way the data were collected and manifests itself invariably when splitting the data into the training, validation, and testing sets. To obtain a more reliable analysis on the performance and robustness of models, we often repeat this splitting process several times to verify how susceptible a model is to overfitting, and whether it has generalized well or instead memorized the patterns. This practice is called *cross validation* (Stone 1974; Geisser 1975). Cross validation is a family of statistical techniques, typically used to determine the generalization power of a model. In this process, a subset of the data, named a *validation set*, is used for validation of the trained model to ensure its generalization over memorization. This subset must have no intersection with either the training set or the test set (which is reserved only for the final evaluation). There are a number of different cross-validation techniques such as *k*-fold, leave-*p*-out, stratified, or purely random (Burman 1989). Random sampling is an important part of these techniques. Disregarding the temporal coherence of the data and thus using random sampling in order to obtain the validation set may result in an artificial boost in performance and, more importantly, obscure the evidence of overfitting, i.e., the significant difference between the performance in the training and the validation phase.

To avoid this pitfall, we customize the cross-validation technique such that it selects training and testing samples from different time-segmented partitions of the data set. This, in fact, is the main reason why the SWAN-SF data set is structured in multiple nonoverlapping partitions. By choosing the training and testing samples from different partitions of SWAN-SF, we prevent the model from being tested on time series that are partially identical to some of those it was trained on. Experiment D of Section 6.3, puts this to test.

In addition to the above cross-validation practice, one could use the NOAA AR numbers or HARNUMs (i.e., unique identifiers of HARP series). This approach is already proposed in the literature (e.g., Bobra & Couvidat 2015; Florios et al. 2018; Campi et al. 2019), and it restricts random sampling to different tracked active regions, rather than different slices of multivariate time series possibly extracted by the same active regions. NOAA identifies and tracks active regions on the Sun's photosphere assigning NOAA AR numbers whenever possible. This is a valid treatment of temporal coherence; however, definitive HARNUMs often include multiple NOAA active regions or, in case of near real-time HARNUMs, different HARNUMs can be merged. In these cases, one needs to filter or untangle these complex active regions to properly tackle the temporal coherence in training, testing, and validating. Such actions are beyond the scope of the current study; therefore, we simply draw different, random samples from different partitions.

5. Verification Measures

For a comprehensive analysis of different measures and their interpretations, we encourage the interested reader to consult Jolliffe & Stephenson (2012). In this section, however, we only briefly review the two metrics we equipped our experiments with, namely, TSS and HSS2, and justify why we did so.

TSS (Hanssen & Kuipers 1965) measures the difference between the probability of detection (i.e., *true-positive rate*), $\frac{tp}{p}$, and the probability of false alarm (i.e., *false-positive rate*), $\frac{fp}{n}$, where $p = tp + fn$ and $n = fp + tn$ are the numbers of the positive and negative instances, respectively. In other words, $TSS = \frac{tp}{p} - \frac{fp}{n}$. It ranges from -1 to $+1$, where -1 indicates that the model is wrong in all of its predictions, 0 means that the model has no skill and reflects the success of *random-guess* models (i.e., models that randomly assign labels to the instances), and $+1$ represents a perfect model that correctly assigns labels to all instances.

Advantage: The strengths of TSS can be summarized as follows: (1) $TSS = 0$ if and only if $tp \cdot tn = fp \cdot fn$, and this simple rule equates three seemingly different models, models that only assign positive labels, models that only assign negative labels, and random-guess models. (2) TSS includes all four elements of the confusion matrix, i.e., tp , fp , tn , and fn . (3) It is unbiased to the imbalance ratio (Bloomfield et al. 2012). That is, given a fixed model (i.e., with an unchanging performance), regardless of the imbalance ratio, it returns a constant value representing its performance.

Disadvantage: Using TSS has a drawback as well. Statistical models whose differences between *true-positive rate* and *false-positive rate* are the same are not always equally good. To give a numerical example, we can calculate TSS on two models' performance: suppose we have 5100 instances and model A gives the confusion matrix [$tp = 80$, $fn = 20$, $fp = 0$, $tn = 5000$] and model B returns [$tp = 90$, $fn = 10$, $fp = 500$, $tn = 4500$]. While according to TSS A and B perform equally good, i.e., $TSS(A) = TSS(B) = 0.8$, model A does not misclassify any minority instance ($fp = 0$), whereas in B, for every correct prediction of the minority instances (tp), it makes, on average, ≈ 5.5 incorrect predictions of that class (fp).

The updated Heidke skill score (HSS2; Balch 2008) is the other metric that quantifies the performance of a model by comparing it to the random-guess model. This is formulated as $\frac{2((tp \cdot tn) - (fn \cdot fp))}{p(fn + tn) + n(tp + fp)}$. Similar to TSS, HSS2 ranges within the interval $[-1, 1]$, with 0 indicating that there is no difference between the model's performance and a random guess. Any other positive value indicates how much better than random the model of interest performs. Decreasing negative values reflect a higher similarity to a model that misclassifies all instances. The lower bound of HSS2 approaches -1 , as the imbalance ratio approaches 1:1.

Advantage: HSS2 returns 0 in all cases that TSS returns 0 . Therefore, it has the first two advantages of TSS while evaluating performance from a different angle. As per the numerical example given above that showed a shortcoming of TSS, HSS2 strongly favors model A over B. Precisely, $HSS2(A) = 0.89$ puts A much higher than B in terms of their performance, where $HSS2(B) = 0.23$. *Disadvantage:* Unlike TSS, HSS2 is biased to imbalance ratio.

A key point in assessing metrics' appropriateness, especially in the context of rare-event classification, is their biases to the class-imbalance ratio. To highlight this, we present a

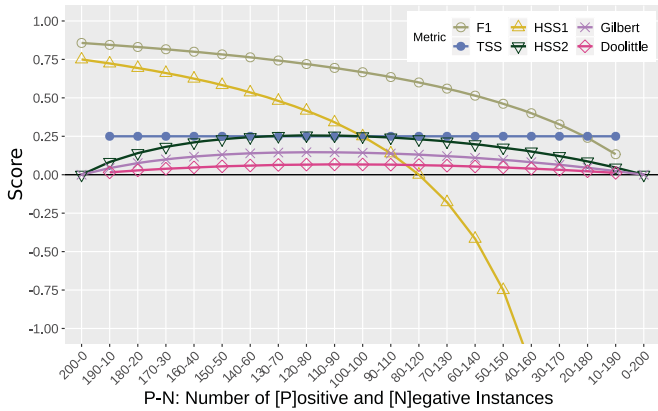


Figure 4. Comparison of several metrics’ bias to the class-imbalance ratio, with the (unbiased) TSS metric in filled blue circles. For a model that correctly predicts 75% of positive instances and 25% of negative instances, behavior of the performance metrics is monitored as the class imbalance transforms from ($p = 0, n = 200$) to ($p = 200, n = 0$), with a step size of 10 for p and n .

comparative test designed as follows: assuming that a model correctly predicts 75% of positive instances and 25% of negative instances, we monitor the behavior of a few performance metrics as we gradually change the negative-to-positive, class-imbalance ratio from 200:0 to 0:200, with a step size of 10. In addition to TSS and HSS2, we include a few other popular metrics as well to emphasize the unique feature of TSS. These metrics are *F1-score*, HSS1, Gilbert’s success ratio, and Doolittle index (see their definitions in Jolliffe & Stephenson 2012). Looking at the results illustrated in Figure 4, HSS1 and *F1-score* metrics imply that the performance deteriorates as the imbalance ratio changes, which is spurious because the model’s performance is assumed to remain unchanged throughout all 21 trials. Meanwhile, HSS2, Gilbert’s success ratio, and Doolittle index show an improvement in performance as the imbalance ratio decreases (toward the center), which is also an artifact of their susceptibility to the class-imbalance change. Among many similar metrics, TSS is one of the few that remains unbiased to this change while being informative (e.g., recall that there is another metric that is also unbiased but carries much less information about the performance and cannot be used as a stand-alone measure).

Concluding from the above discussion, in all experiments carried out in Section 6, we invariably provide TSS and show HSS2 only if the class-imbalance ratio remains unchanged. These two metrics have already been reported in many flare-forecasting studies with the same justification (e.g., Barnes & Leka 2008; Wilks 2011; Bobra & Couvidat 2015). Also critically important to keep in mind is that a higher TSS value does not necessarily imply a better forecast model under the class-imbalance condition, as it may be coupled with a very low HSS2, and hence a lack of robustness. In case of class balance, TSS and HSS2 (and the original HSS1) have identical values.

We conclude this section by mentioning that on the topic of flare forecasting several studies have extensively discussed the different aspects of the verification process. We avoid repeating them here but refer the interested reader to Barnes & Leka (2008), Bloomfield et al. (2012), Jolliffe & Stephenson (2012), Leka et al. (2019b), and Cinto et al. (2020b).

6. Experiments and Results

In this section, we present the experiments conducted to showcase the challenges discussed previously in the frame of the overarching flare-forecasting task. We reiterate that the objective of this study is not to achieve a robust model with high performance but to compare models trained differently. Therefore, although any new preprocessing of the data (e.g., using different normalization, data splits, or sampling techniques) requires retuning of the hyperparameters, without loss of generality we rely on our pretuned hyperparameters for SVM, that is, $C = 1000$, $\gamma = 0.01$, with an RBF kernel. These are the results of our pretuning of the hyperparameters using a grid search, by training the SVM on partitions 1, 2, and 3 and validating it on partitions 4 and 5. We compare the models’ performance mainly using the TSS values achieved.

Throughout Experiments Z, A, B, C, D, E, and F, we utilize *last-value* of the time series as the single feature. In the last experiment (Experiment G), however, we use our four descriptive statistics, namely, *median*, *standard deviation*, *skewness*, and *kurtosis*, to compare their discriminative power against *last-value* of the time series. This allows comparing time series with point-in-time forecasting. It is beyond the scope of this work to optimize solutions using more statistical features, so a more thorough feature selection process is deferred for future studies.

6.1. Baseline

To establish a baseline for the experiments, a model first needs to learn from the available data without any special treatment in the data input process or the model configuration. With this in mind, we start with Experiment Z.

Experiment Z: Baseline. This corresponds to the straightforward training of the SVM on all instances of one SWAN-SF partition and testing on another. We try this on all possible partition pairs, resulting in 20 different trials, to expose a possible variable performance for different partition choices. The results are shown in Figure 5 (top plot; line with “ ∇ ” markers, under label “None”). The baseline TSS values across all partition pairs have a mean $\mu_{\text{TSS}(Z)} = 0.18$, with a standard deviation $\sigma_{\text{TSS}(Z)} = 0.11$.

6.2. Impact of Class-imbalance Issue

Experiments A, B, and C compare the three different sampling approaches discussed in Section 4.3 in tackling the class-imbalance problem. The experiments share a common setup: the SVM is independently trained and tested on all permutations of partition pairs. We reiterate that each sampling method is only applied during the training phase, as modulating the sampling of the test set distorts reality and does not reflect the true operational performance of the model. To gauge the confidence of a model’s performance when a sampling method is employed, we repeat the experiment 10 times and report the mean and standard deviation values of the achieved TSS.

Experiment A: Undersampling. In the training phase, the model takes in a subset of a training partition generated by an X-based undersampling method (US2 from Table 1). This enforces a 1:1 balance not only in the superclass level (i.e., $|XM| = |CBN|$) but also in the subclass level (i.e., $|X| = |M|$ and $|C| = |B| = |N|$). The trained model is then tested against all other partitions. In the top plot in Figure 5 (line with “ \square ” markers, under label “Undersampling”), the consistent and

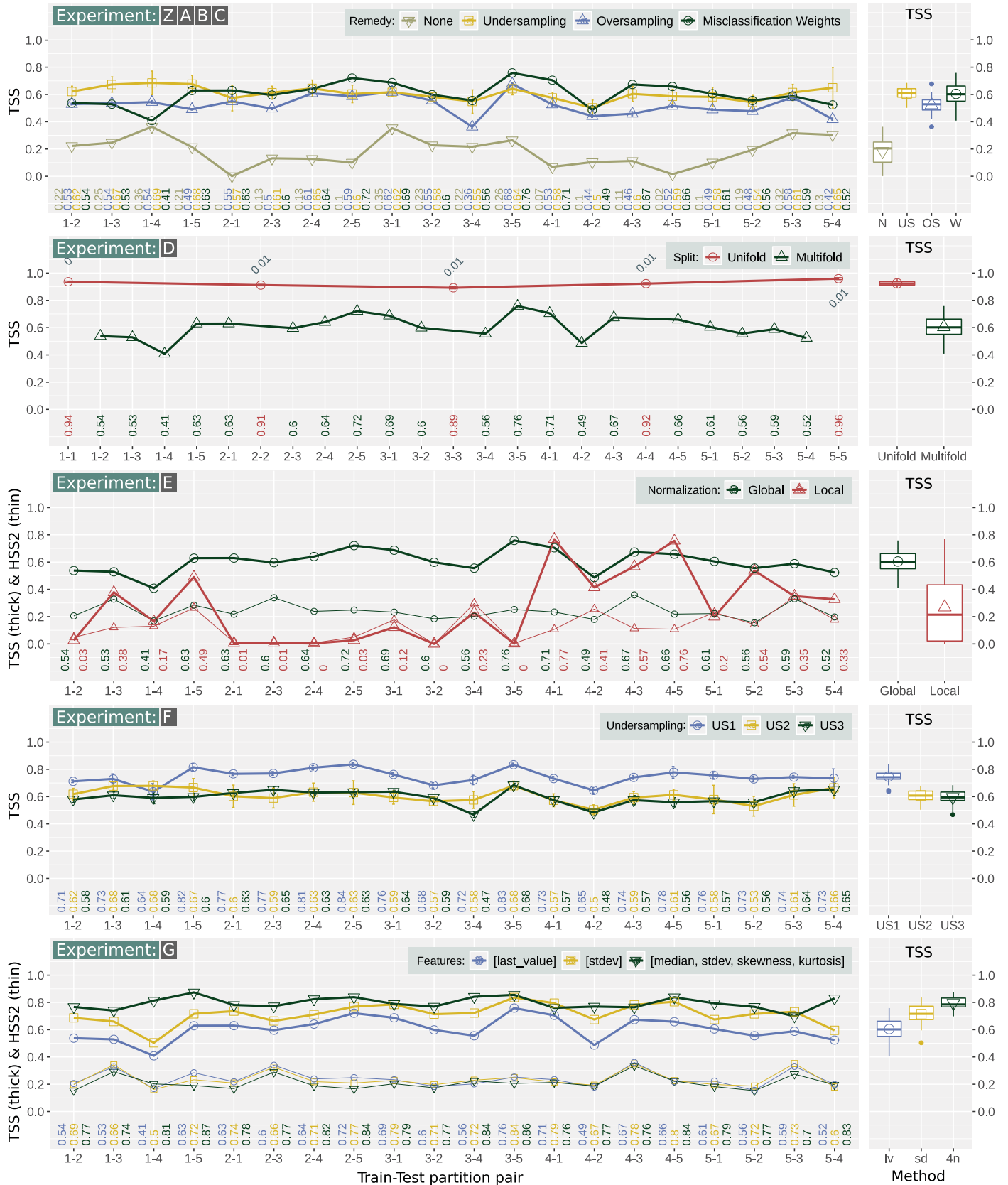


Figure 5. Performance of SVM (by means of TSS/HSS2), trained and tested on partition pairs of the SWAN-SF benchmark data set. Error bars show the standard deviation of individual TSS/HSS2 values, produced after 10 times of repeating each experiment. The TSS value corresponding to each partition pair is written above the horizontal axis. On the right side of each plot, the box plots depict the overall variation (interquartile range [IQR], median, and mean) of TSS across all pairs, for each scenario. The experiment(s) each plot corresponds to are mentioned in the upper left corner of each plot. In Experiments E and G, thinner lines correspond to HSS2, in addition to TSS.

significant impact of this remedy is evident, when compared to the baseline. We obtain a mean TSS $\mu_{\text{TSS}(A)} = 0.61$ with standard deviation $\sigma_{\text{TSS}(A)} = 0.05$, to be compared with 0.18 and 0.11, respectively, of the baseline Experiment Z.

Experiment B: Oversampling. Here we apply a C-based oversampling method (OS3 in Table 1) that also enforces a 1:1 balance in the subclass level. As shown in the top panel of Figure 5 (line with “ Δ ” markers, under label “Oversampling”), we obtain a mean TSS $\mu_{\text{TSS}(B)} = 0.52$ with standard deviation $\sigma_{\text{TSS}(B)} = 0.07$. Notice that undersampling and oversampling give similar TSS values, with a difference within applicable uncertainties.

Experiment C: Misclassification Weighting. We use the imbalance ratio of the superclasses as the weights (as discussed in Section 4.3.2). For instance, when working with Partition 3, since the majority-to-minority ratio is 20:1, we set the misclassification weights as follows: $w_{\text{XM}} = 20$ and $w_{\text{CBN}} = 1$. The top panel of Figure 5 (line with “ \circ ” markers, under label “Misclassification Weights”) shows that misclassification weighting typically scores near the top of the achieved TSS values ($\mu_{\text{TSS}(C)} = 0.60$ and $\sigma_{\text{TSS}(C)} = 0.09$), albeit within uncertainties from undersampling ($\mu_{\text{TSS}(A)} = 0.61$) and oversampling ($\mu_{\text{TSS}(B)} = 0.52$) remedies. This said, it is worth pointing out that this remedy is probably better suited toward producing more robust forecast models because it allows a data-driven tunability of applicable weights.

6.3. Impact of Temporal Coherence

In Section 2.3, we discussed the theoretical impact of random sampling, embedded in many cross-validation methods, on a temporally coherent data set. The following experiment is designed to quantify this impact.

Experiment D: Data Splits. This time, the SVM is trained and tested on two randomly chosen subsets of the same partition, with no overlap between the subsets, while preserving the climatology of the flares in each of them. More accurately, this is a stratified, k -fold cross validation using a random subsampling method with $k = 10$. The results are juxtaposed to those obtained by training SVM on one partition and testing it on another. In both scenarios, we equipped SVM with misclassification weights, to eliminate the need for an additional sampling layer. Therefore, the only determining factor is whether the instances are sampled from the same partition or not.

The second panel of Figure 5 illustrates this comparison. When SVM is trained and tested on a single partition (line with “ \circ ” markers, under label “unfold”), performance is boosted significantly with TSS averaging at $\mu_{\text{TSS}(D_u)} = 0.92$, with a standard deviation of $\sigma_{\text{TSS}(D_u)} = 0.03$. Training and testing on different partitions (line with “ Δ ” markers, under label “multifold”) show $\mu_{\text{TSS}(D_m)} = 0.60$, with a standard deviation $\sigma_{\text{TSS}(D_m)} = 0.09$. Subscripts u and m here correspond to unfold and multifold training/testing, respectively. The remarkable difference between the two scenarios is because the forecast models in unfold trials are both trained and tested on a temporally coherent data set. Although for the unfold scenario the training and testing sets are nonoverlapping, random sampling does not properly treat the temporal coherence, and consequently the models show such a high performance. Such results are occasionally misinterpreted as evidence of models’ robustness because of the consistency between the training and

testing results. However, as discussed in Sections 2.3 and 4.4, they are indications of overfitting due to memorization.

6.4. Impact of Normalization: Global and Local

Experiment E: Normalization. As we train and test SVM on a pair of partitions, we transform the feature space to a normalized space using a zero-one normalization method. This can be done either by means of features’ extrema of both partitions or by taking into account the extrema of each partition separately. We refer to the former as *global* normalization and to the latter as *local* normalization. It is worth noting that data normalization is an important preprocessing step in data cleaning, which needs a rigorous analysis of the data, including finding the outliers and obtaining the statistically meaningful extrema for each of the features. The current experiment only aims to compare the performance of the two normalization approaches for different partitions, hence phases of the solar cycle.

A comparison of different performances, shown in the third panel of Figure 5, reflects the significant changes in performance due to the continuously modulating magnetic activity during the 11 yr solar cycle. Mean TSS values of the global (line with “ \circ ,” under label “Global”) and local (line with “ Δ ,” under label “Local”) normalizations are $\mu_{\text{TSS}(E_g)} = 0.60$ and $\mu_{\text{TSS}(E_l)} = 0.27$, respectively. The relatively high standard deviation of $\sigma_{\text{TSS}(E_l)} = 0.26$ in the case of local normalization, compared to $\sigma_{\text{TSS}(E_g)} = 0.09$ for the global normalization, implies that the local normalization is generally inconsistent between the training and the testing partitions. Statistically, higher performance is achieved when global normalization is applied, although in some cases the opposite happens, according to TSS, HSS2, or both. Also, $\mu_{\text{HSS2}(E_l)} = 0.13$, $\mu_{\text{HSS2}(E_g)} = 0.24$, and $\sigma_{\text{HSS2}(E_l)} = 0.11$, $\sigma_{\text{HSS2}(E_g)} = 0.06$.

It is interesting to discuss the model performance for some specific partition pairs. For the local normalization instances, it seems that Partition 4 (which spans over the period from 2014 March through 2015 March) lays out a unique set of extrema with representative features for all partitions, which results in achieving a slightly better performance than global normalization. In contrast, Partition 2 (2012 February through 2013 October) seems to be holding the most restrictive range of values. A potential justification for this difference could be the presence of more extreme outliers in Partition 2, which corresponds to solar maximum, compared to Partition 4, which corresponds to the decay phase of solar cycle 24. This suggests that a careful outlier detection process in the data cleaning phase could also improve, or otherwise affect, the overall performance of forecast models.

While additional investigation of this effect is clearly warranted, one gathers that performance may be optimized when training and testing take place during similar levels of solar activity, when local normalization and global normalization converge on similar normalized values. This, however, is hardly tenable in operational settings, when forecast models perform on an always unknown day-by-day solar activity level.

6.5. Impact of Sampling

In Section 4.3, we showed that there are multiple variants of oversampling and undersampling strategies, a subset of which are listed in Table 1. We also presented how this affects flare distributions in each partition of the SWAN-SF benchmark.

Below, we test how different undersampling methods impact models' performance in terms of their TSS values, across different partitions. Testing different oversampling methods would essentially teach us similar lessons.

Experiment F: Different Undersampling Methods. We use all undersampling methods listed in Table 1, namely, US1, US2, and US3, in the training phase. Then we test the trained models against all other partitions. All three undersampling methods yield a 1:1 balance in the superclass level (i.e., $|XM| = |CBN|$). However, US1 preserves the climatology of flares in the subclass level, while US2 and US3 additionally enforce a 1:1 balance in the subclass level as well. US2 and US3 reduce the flare populations based on the number of X-class and M-class flares, respectively. We further employ global normalization only to keep the experiments' conditions constant. Our results are shown in Figure 5, the fourth panel from the top. From them, one sees a relatively similar, consistent performance, although the climatology-preserving undersampling, i.e., US1 (line with “°,” under label “US1”) seems to give a statistically higher performance compared to the other two methods. This is largely because in both US2 and US3 the number of flare-quiet instances are substantially reduced to meet the required criterion of 1:1 balance in the subclass level. As discussed in Section 4.3, this takes away many easy-to-predict instances from the models and makes the problem more challenging, hence the lower performance for US2 and US3.

Concluding from this experiment, it becomes clear that different undersampling and oversampling methods give non-identical performances, and if one prefers to use undersampling or oversampling over misclassification weighting strategy, we recommend the climatology-preserving class balancing for better robustness on the SWAN-SF (and, apparently, similar benchmark data sets).

6.6. Impact of Time Series Features

We reserve the last experiment for presenting the benefit of using time series, rather than point-in-time values, for forecasting. To simulate the point-in-time effect, we employ the *last value* statistic that returns the last record of each time series of the SWAN-SF, converting a multivariate time series to a vector of *last values*.

Experiment G: SVM with Other Statistical Features. SVM is trained and tested on different partition pairs, using three sets of statistics extracted from the time series. These statistics sets are (i) {*last value*}, (ii) {*standard deviation*}, and (iii) {*median, standard deviation, skewness, kurtosis*}. In all three cases, misclassification weighting is used as a class-imbalance remedy. As the box plot in the bottom plot of Figure 5 shows, *standard deviation* (line with “□” markers, under label “[stdev]”) results in a statistically better performance than *last value* (line with “°” markers, under label “[last_value]”), while the third set (line with “∇” markers, under label “[median, stdev, skewness, kurtosis]”) seems to outperform *standard deviation*. This is a good indication that different characteristics of time series carry important, nonredundant information that may improve the reliability of a forecast model.

This said, note that HSS2 does not subscribe to such differences. This is a good example illustrating why the choice of metric must be specific to the objective of the task. Our interpretation of the apparent improvement here relies solely on how TSS reflects success. HSS2, on the other hand, does not

show any significant improvement among the three choices. Therefore, if the objective of a task is more aligned with what HSS2 measures, then the three approaches taken in this experiment are very similar, within the margin of error. To improve HSS2 as well as TSS, more time needs to be invested on tuning the SVM's hyperparameters. For the reasons mentioned at the beginning of Section 6, we did not tune SVM separately for each experiment. Also note that the utilized statistics are only chosen as a result of their general descriptive power and are not the outcome of a rigorous feature selection process. Finding a set of statistical features that optimally distinguish between metadata time series of flaring and nonflaring active regions is in itself a challenging task and goes beyond the scope of this work.

7. Summary, Conclusions, and Future Work

Working with multiclass, high-dimensional data is certainly challenging. From time to time and despite several revisions of these challenges, they tend to be overlooked by domain experts, understandably so, as the complexity of the original flare-forecasting problem may overshadow the challenges in the preprocessing of the data. We used the SWAN-SF benchmark data set as a reference in order to highlight some of the challenges. We pointed attention to an interesting characteristic of such data sets, which we called *temporal coherence*, inherited from the sampling and slicing methodologies. We also revisited the problem of class imbalance in flare forecasting and the impact of different remedies addressing this problem. In the context of temporally coherent, class-imbalanced data, we designed several different, nonoverlapping experiments showcasing these challenges and common preprocessing tasks to tackle them, in terms of normalization, sampling, and cross validation. Below, we summarize our key points and conclusions:

1. *On Normalization:* A global normalization of parameters (i.e., over both training and testing partitions) is preferred over a local normalization (i.e., separately over training and testing partitions). However, since the variance of the physical parameters changes significantly from one partition to another, following the changes within a solar cycle, a universal normalization (i.e., over the entire five partitions) should be avoided. In other words, the choice of the extrema should take into account the phase of the solar cycle in order to achieve the optimal forecasting capability. This said, we realize that the unavoidable local normalization in operational settings may have an impact on performance. In addition, our approach ignores the actual values of metadata parameters, some of which have also shown flare predictive ability in numerous previous studies cited in the Introduction.
2. *On Class Imbalance:* The class-imbalance problem needs to be dealt with properly, i.e., using simple or synthetic sampling approaches, weighting, or other means. Regarding sampling, any methodology applies only to the training set: altering the climatology of flares in the test and validation sets leads to unrealistic and overly optimistic performance that cannot be reproduced in operational settings. We recommend treating the class-imbalance problem by means of misclassification weighting, if the cost function of the utilized model allows.

3. *On Temporal Coherence:* Any random sampling employed for splitting data into training, validation, and test sets must take into account the temporal coherence of data that comes into play when the slicing time step is smaller than the observation window in time series forecasting. This is on top of fulfilling the nonoverlapping condition dictating that a given instance must appear in one and only one of the three sets. The time-segmented SWAN-SF data set with its nonoverlapping partitions can help models circumvent these issues. Another approach is to limit the random sampling to unique HARP series or NOAA active regions, as studied in detail by Campi et al. (2019), among others.
4. *On Performance Metrics:* In solar flare forecasting, class imbalance is intrinsic and at least some of the metrics chosen for verification of performance should not be susceptible to or biased by it. As previous studies have shown (e.g., Woodcock 1976; Bloomfield et al. 2012; Bobra & Couvidat 2015), TSS is not affected adversely by class imbalance. Hence, we also recommend using this metric for flare prediction, at least in binary (i.e., nonprobabilistic) forecasting. This said, HSS2 should also be used, as it quantifies how a model performs in comparison to the random-guess model. HSS2 is affected by class imbalance, however, in perfectly balanced data $HSS1 = HSS2 = TSS$. It is also possible that a specific task (with a particular objective) may call for a different, or even a new, metric.
5. *On Comparison of Models:* Different models (i.e., fitted algorithms) are not precisely comparable unless they (1) are trained on the same data set, (2) with identical normalization techniques, (3) with identical sampling strategies, and (4) with identical class-imbalance remedies, if any. The use of a benchmark data set allows satisfying the first condition toward producing comparable studies on flare forecasting. In this study, we highlighted the importance of the other three conditions.

There are many further interesting avenues to be discovered that we plan to exploit in future studies. Results reported here are based on a small set of five time series statistics (*median, variance, skewness, kurtosis, and last-value*). There was no consistent effort to find the best-performing features that could potentially boost the models' performance. Some of them could be based on, for example, the first and second derivatives of the time series, those that keep track of the general trends in time series, up-surges and down-slides, features that compare different parts of the time series, and many others. To be able to benefit from a long list of features, a feature selection algorithm is needed that must be carefully guided and augmented by domain expertise. Time series of the best selected features could, besides improving flare forecasting in operational settings, help solar physicists better understand the triggering mechanism(s) of solar flares. Interpretable machine-learning methods, rather than "black-box" solutions, could be paramount for this purpose and should be given priority (Lipton & Steinhardt 2018; Marcus 2018; Rudin 2019).

Hard-to-interpret methodologies often based on deep learning may, on the other hand, be problematic for solar flare forecasting, given the limited amount of data available, where millions of positive sample instances are typically needed (e.g., Goodfellow et al. 2016).

In such efforts, well-curated benchmark data sets have been found to be instrumental. With the SWAN-SF, for example, we have and will continue to run large numbers of experiments in a relatively short time and with considerable efficiency. Benchmark data sets, adhering to well-defined data collection and integration policies, can always be further enhanced and expanded, offering valuable services to interested communities. We hope that this and other recent works raise awareness within this and other domains dealing with class imbalance and temporal coherence in their respective forecasting problems (Ahmadzadeh et al. 2019a; He & Georgoulis 2019; Sadykov 2019; Nita et al. 2020). Interdisciplinarity is a key common element of such pursuits, so extended applicability may provide clues, or even viable solutions, for future diverse forecast efforts and other real-world problems.

We would like to thank the anonymous reviewer for the helpful feedback they provided on earlier drafts of the manuscript. This work was supported in part by two NASA grant awards (Nos. NNH14ZDA001N and 80NSSC20K1352) and two NSF grant awards (Nos. AC1443061 and AC1931555). The AC1443061 award has been supported by funding from the Division of Advanced Cyber infrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Astronomical Sciences within the Directorate for Mathematical and Physical Sciences, and the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences.

Appendix

For convenience, all acronyms and notations used in this work are listed in Tables 2 and 3.

Table 2
List of All Acronyms Used in This Manuscript

Acronym	Full Name
NOAA	National Oceanic and Atmospheric Administration
GOES	Geostationary Operational Environmental Satellite
SWAN-SF	Space Weather data ANalytics for Solar Flare; benchmark data produced by Georgia State University's Astroinformatics Cluster
SVM	Support vector machine; a family of statistical learning models
TSS	True skill score (Hanssen & Kuipers 1965)
HSS1	Heidke skill score (Heidke 1926)
HSS2	Updated Heidke skill score (Balch 2008)
GS	Gilbert's success ratio, also known as Gilbert skill score or equitable threat score (Gilbert 1884)
DI	Doolittle index (Doolittle 1885)
US _{<i>i</i>}	An undersampling method indexed <i>i</i> th in Table 1
OS _{<i>i</i>}	An oversampling method indexed <i>i</i> th in Table 1

Table 3
List of All Notations Used in This Manuscript

Notation	Description
X, M, C, B, A	Used both as the GOES flare class label and as the set of all instances of one class.
N	Used both as the nonflaring class label and as the set of all nonflaring instances.
XM	Used both as the flaring class label (instances from X and M) and as the set of all X- and M-class instances.
CBN	Used both as the nonflaring class (instances from C, B, and N) and as the set of all C-, B-, and N-class instances.
$ \cdot $	Set cardinality; e.g., $ X $ indicates the number of X-class flares.
T_{obs}	Temporal observation window; a time interval during which the values of a set of magnetic field parameters are observed.
T_{pred}	Temporal prediction window; a time interval that is labeled according to the strongest flare class recorded in that period.
τ	Temporal step size in the moving-window approach used for slicing the time series.
t_i	Time series with index i .
e	Statistical event, such as <i>flaring</i> or <i>nonflaring</i> .
$\mathcal{P}(e)$	Probability of the statistical event e .
p	Positive class label of an instance; in flare-forecasting context, it points to a data point that is predicted as flaring.
n	Negative class label of an instance; in flare-forecasting context, it points to a data point that is predicted as nonflaring.
tp	True positive (<i>hit</i>); a correct classification of an instance that is labeled as positive.
fp	False positive (<i>false alarm</i>); an incorrect classification of an instance that is labeled as negative.
tn	True negative (<i>correct rejection</i>); a correct classification of an instance that is labeled as negative.
fn	False negative (<i>miss</i>); an incorrect classification of an instance that is labeled as positive.

ORCID iDs

Berkay Aydin  <https://orcid.org/0000-0002-9799-9265>

Manolis K. Georgoulis  <https://orcid.org/0000-0001-6913-1330>

Dustin J. Kempton  <https://orcid.org/0000-0002-1837-8365>

Sushant S. Mahajan  <https://orcid.org/0000-0003-1753-8002>

Rafal A. Angryk  <https://orcid.org/0000-0001-9598-8207>

References

- Ahmadzadeh, A., Aydin, B., Kempton, D., et al. 2019a, AGU Fall Meeting 2019 Abstracts, *NG22A-06*
- Ahmadzadeh, A., Aydin, B., Kempton, D. J., et al. 2019b, in 18th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), ed. M. Arif Wani et al. (Piscataway, NJ: IEEE), 1814
- Ahmadzadeh, A., Sinha, K., Aydin, B., & Angryk, R. A. 2020, *SoftX*, **12**, 100518
- Ali, A., Shamsuddin, S. M., & Ralescu, A. L. 2013, *Int. J. Adv. Soft Comput. Appl.*, **7**, 176
- Al Shalabi, L., & Shaaban, Z. 2006, in *Int. Conf. on Dependability of Computer Systems* (Los Alamitos, CA: IEEE Computer Society), 207
- Angryk, R. A., Martens, P. C., Aydin, B., et al. 2020, *NatSD*, **7**, 227
- Aschwanden, M. J., Crosby, N. B., Dimitropoulou, M., et al. 2016, *SSRv*, **198**, 47
- Aschwanden, M. J., & Freeland, S. L. 2012, *ApJ*, **754**, 112
- Balasubramaniam, K. S., & West, E. A. 1991, *ApJ*, **382**, 699
- Balch, C. C. 2008, *SpWea*, **6**, 1001
- Barnes, G., & Leka, K. 2008, *ApJL*, **688**, L107
- Barnes, G., & Leka, K. D. 2006, *ApJ*, **646**, 1303
- Barnes, G., Leka, K. D., Schrijver, C. J., et al. 2016, *ApJ*, **829**, 89
- Ben-Hur, A., & Weston, J. 2010, *Data Mining Techniques for the Life Sciences* (Clifton, NJ: Humana Press), 223
- Benvenuto, F., Piana, M., Campi, C., & Massone, A. M. 2018, *ApJ*, **853**, 90
- Benz, A. 2008, *LRSP*, **5**, 1
- 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
- Burges, C. J. 1998, *Data Min. Knowl. Discov.*, **2**, 121
- Burman, P. 1989, *Biometrika*, **76**, 503
- Cacciani, A., Varsik, J., & Zirin, H. 1990, *SoPh*, **125**, 173
- Campi, C., Benvenuto, F., Massone, A. M., et al. 2019, *ApJ*, **883**, 150
- Camporeale, E. 2019, *SpWea*, **17**, 1166
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, *J. Artif. Intell. Res.*, **16**, 321
- Chen, Y., Ji, A., Babajiyavar, P. A., Ahmadzadeh, A., & Angryk, R. 2020, in 2020 IEEE Int. Conf. on Big Data, ed. X. Wu et al. (Piscataway, NJ: IEEE), 4184
- Cinto, T., Gradwohl, A. L. S., Coelho, G. P., & da Silva, A. E. A. 2020a, *SoPh*, **295**, 93
- Cinto, T., Gradwohl, A. L. S., Coelho, G. P., & daSilva, A. E. A. 2020b, *MNRAS*, **495**, 3332
- Cristianini, N., & Shawe-Taylor, J. 2000, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge: Cambridge Univ. Press)
- Dietterich, T. G., & Bakiri, G. 1995, *J. Artif. Intell. Res.*, **2**, 263
- Doolittle, M. 1885, *Bull. Phil. Soc. Washington*, **7**, 122
- Drummond, C., Holte, R. C., et al. 2003, C4.5, *Class Imbalance, and Cost Sensitivity: Why Under-sampling Beats Over-sampling*, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.68.6858>
- Falconer, D. A., Moore, R. L., Barghouty, A. F., & Khazanov, I. 2012, *ApJ*, **757**, 32
- Fletcher, L., Dennis, B. R., Hudson, H. S., et al. 2011, *SSRv*, **159**, 19
- Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, *SoPh*, **293**, 28
- Ganganwar, V. 2012, *Int. J. Emerg. Technol. Adv. Eng.*, **2**, 42
- Geisser, S. 1975, *J. Am. Stat. Assoc.*, **70**, 320
- Georgoulis, M. K. 2012a, *The Sun: New Challenges* (Berlin: Springer), 93
- Georgoulis, M. K. 2012b, *SoPh*, **276**, 161
- Gilbert, G. K. 1884, *Am. Meteorol. J.*, **1**, 166
- Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (Cambridge, MA: MIT Press)
- Guerra, J. A., Murray, S. A., Shaun Bloomfield, D., & Gallagher, P. T. 2020, *JWSWC*, **10**, 38
- Guerra, J. A., Pulkkinen, A., & Uritsky, V. M. 2015, *SpWea*, **13**, 626
- Hamdi, S. M., Kempton, D., Ma, R., Boubrahimi, S. F., & Angryk, R. A. 2017, in 2017 IEEE Int. Conf. on Big Data, ed. Y.-N. Jian et al. (Piscataway, NJ: IEEE), 2543
- Hanssen, A., & Kuipers, W. 1965, *Mededelingen en Verhandelingen*, **81**, 2
- Hastie, T., & Tibshirani, R. 1997, *Advances in Neural Information Processing Systems* **10**, 507, <http://papers.nips.cc/paper/1375-classification-by-pairwise-coupling>
- He, H., & Garcia, E. A. 2009, *IEEE Trans. Knowl. Data Eng.*, **21**, 1263
- He, H., & Georgoulis, M. K. 2019, AGU Fall Meeting 2019 Abstracts, *SH34A-01*
- Heidke, P. 1926, *Geografiska Annaler*, **8**, 301
- Hoeksema, J. T., Liu, Y., Hayashi, K., et al. 2014, *SoPh*, **289**, 3483
- Hossin, M., & Sulaiman, M. 2015, *Int. J. Data Min. Knowl. Manag. Process.*, **5**, 1
- Hostetter, M., Ahmadzadeh, A., Aydin, B., et al. 2019, in 2019 IEEE Int. Conf. on Big Data, ed. C. Baru et al. (Piscataway, NJ: IEEE), 4960
- Japkowicz, N., & Stephen, S. 2002, *Intell. Data Anal.*, **6**, 429
- Ji, A., Aydin, B., Georgoulis, M. K., & Angryk, R. 2020, in 2020 IEEE Int. Conf. on Big Data, ed. X. Wu et al. (Piscataway, NJ: IEEE), 4218
- Jolliffe, I. T., & Stephenson, D. B. 2012, *Forecast Verification: A Practitioner's Guide in Atmospheric Science* (New York: Wiley)
- Krawczyk, B. 2016, *Prog. Artif. Intell.*, **5**, 221
- Kubat, M., Matwin, S., et al. 1997, in Proc. Fourteenth Int. Conf. on Machine Learning, ed. D. H. Fisher (Burlington, MA: Morgan Kaufmann), 179

- Leka, K., & Barnes, G. 2018, in *Extreme Events in Geospace*, ed. N. Buzulukova (Amsterdam: Elsevier), 65
- Leka, K. D., & Barnes, G. 2003a, *ApJ*, **595**, 1277
- Leka, K. D., & Barnes, G. 2003b, *ApJ*, **595**, 1296
- Leka, K. D., & Barnes, G. 2007, *ApJ*, **656**, 1173
- Leka, K. D., Park, S.-H., Kusano, K., et al. 2019a, *ApJ*, **881**, 101
- Leka, K. D., Park, S.-H., Kusano, K., et al. 2019b, *ApJS*, **243**, 36
- Lipton, Z. C., & Steinhardt, J. 2018, arXiv:1807.03341
- Lites, B. W. 1996, *SoPh*, **163**, 223
- Ma, R., Boubrahimi, S. F., Angryk, R. A., & Ma, Z. 2020, in *5th IEEE Int. Conf. on Big Data Analytics (ICBDA)* (Piscataway, NJ: IEEE), 94
- Marcus, G. 2018, arXiv:1801.00631
- McCloskey, A. E., Gallagher, P. T., & Bloomfield, D. S. 2018, *JSWSC*, **8**, A34
- Mickey, D. L., Canfield, R. C., Labonte, B. J., et al. 1996, *SoPh*, **168**, 229
- Murray, S. A. 2018, *SpWea*, **16**, 777
- Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, *ApJ*, **835**, 156
- Nita, G., Georgoulis, M., Kitiashvili, I., et al. 2020, arXiv:2006.12224
- Park, S.-H., Leka, K. D., Kusano, K., et al. 2020, *ApJ*, **890**, 124
- Parsons, L., Haque, E., & Liu, H. 2004, *ACM SIGKDD Explor. Newsl.*, **6**, 90
- Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. 2012, *The Solar Dynamics Observatory (SDO)* (New York: Springer US), 3
- Qahwaji, R., & Colak, T. 2007, *SoPh*, **241**, 195
- Reinard, A. A., Henthorn, J., Komm, R., & Hill, F. 2010, *ApJL*, **710**, L121
- Rimmele, T. R., Warner, M., Keil, S. L., et al. 2020, *SoPh*, **295**, 172
- Rudin, C. 2019, *Nat. Mach. Intell.*, **1**, 206
- Sadykov, V. M. 2019, AGU Fall Meeting 2019 Abstracts, SH34B-05
- Scherrer, P. H., Bogart, R. S., Bush, R. I., et al. 1995, *The Solar Oscillations Investigation-Michelson Doppler Imager* (Dordrecht: Springer), 129
- Scherrer, P. H., Schou, J., Bush, R. I., et al. 2012, *SoPh*, **275**, 207
- Solanki, S. K., del Toro Iniesta, J. C., Woch, J., et al. 2020, *A&A*, **642**, A11
- Spirock, T., Denker, C., Chen, H., et al. 2001, in *ASP Conf. Proc.* 236, *Advanced Solar Polarimetry—Theory, Observation, and Instrumentation*, ed. M. Sigwarth (San Francisco, CA: ASP), 65
- Stone, M. 1974, *J. R. Stat. Soc. B*, **36**, 111
- Tsuneta, S., Ichimoto, K., Katsukawa, Y., et al. 2008, *SoPh*, **249**, 167
- Vapnik, V. 1963, *Autom. Remote Control*, **24**, 774
- Veronig, A. M., Temmer, M., & Hanslmeier, A. 2004, *SoPh*, **219**, 125
- Wagner, W. J. 1988, *AdSpR*, **8**, 67
- Welsch, B. T., Li, Y., Schuck, P. W., & Fisher, G. H. 2009, *ApJ*, **705**, 821
- Wilks, D. S. 2011, *Statistical Methods in the Atmospheric Sciences* (New York: Academic)
- Woodcock, F. 1976, *MWRv*, **104**, 1209
- Yu, L., Pan, Y., & Wu, Y. 2009, in *2009 Int. Conf. on Computational Intelligence and Software Engineering* (Piscataway, NJ: IEEE), 1