The Astrophysical Journal Supplement Series, 277:60 (14pp), 2025 April https://doi.org/10.3847/1538-4365/adb9e3

Outlier Detection and Removal in Multivariate Time Series for a More Robust Machine Learning–based Solar Flare Prediction

Junzhi Wen[1], Azim Ahmadzadeh[2], Manolis K. Georgoulis[3,4], Viacheslav M. Sadykov[5], and Rafal A. Angryk[1]

[1] Department of Computer Science, Georgia State University, Atlanta, GA 30302, USA; jwen6@student.gsu.edu

[2] Department of Computer Science, University of Missouri-St. Louis, St. Louis, MO 63103, USA

[3] Research Center for Astronomy and Applied Mathematics of the Academy of Athens, 11527 Athens, Greece

[4] Johns Hopkins University Applied Physics Laboratory, Laurel, MD 20375, USA

[5] Physics & Astronomy Department, Georgia State University, Atlanta, GA 30302, USA

Received 2024 July 18; revised 2025 February 5; accepted 2025 February 20; published 2025 April 4

Abstract

Timely and accurate prediction of solar flares is a crucial task due to the danger they pose to human life and infrastructure beyond Earth's atmosphere. Although various machine learning algorithms have been employed to improve solar flare prediction, there has been limited focus on improving performance using outlier detection. In this study, we propose the use of a tree-based outlier detection algorithm, Isolation Forest (iForest), to identify multivariate time-series instances within the flare-forecasting benchmark data set, Space Weather Analytics for Solar Flares (SWAN-SF). By removing anomalous samples from the nonflaring class (N-class) data, we observe a significant improvement in both the true skill score and the updated Heidke skill score in two separate experiments. We focus on analyzing outliers detected by iForest at a 2.4% contamination rate, considered the most effective overall. Our analysis reveals a co-occurrence between the outliers we discovered and strong flares. Additionally, we investigated the similarity between the outliers and the strong-flare data and quantified it using Kullback–Leibler divergence. This analysis demonstrates a higher similarity between our outliers and strong-flare data when compared to the similarity between the outliers and the rest of the N-class data, supporting our rationale for using outlier detection to enhance SWAN-SF data for flare prediction. Furthermore, we explore a novel approach by treating our outliers as if they belong to flaring-class data in the training phase of our machine learning, resulting in further enhancements to our models' performance.

Unified Astronomy Thesaurus concepts: Solar flares (1496); Space weather (2037); Time series analysis (1916); Outlier detection (1934); Support vector machine (1936); Classification (1907); Solar activity (1475)

1. Introduction

Outlier detection, also known as anomaly detection, is an integral research branch in data mining and machine learning and has been extensively studied in many application domains. By finding outliers, insightful and actionable information can be obtained for researchers to better understand the underlying nature of the data and make better decisions. For instance, an anomalous pattern in public health data may reflect the symptoms of a new disease (V. Chandola et al. 2007; Y. Gao et al. 2022). In computer networks, an unusual pattern could be the signal of a suspicious activity (V. Kumar 2005). Similarly, an outlier in credit card transaction data may indicate a fraud or theft event (E. Aleskerov et al. 1997). Also, an abnormal pattern in spatiotemporal environmental data could forebode a rare but extreme event such as an earthquake or a storm (C. C. Aggarwal 2017). Detecting outliers is important because it can also help clean the data for future machine learning applications. Real-world scientific data are not as good and clean as one might wish them to be and frequently suffer from the presence of outliers. The existence of outliers could be due to several factors, including the inherent variability of the domain, strategy of data collection or labeling, instrument malfunction, and so on (H. J. Escalante 2005). As machine learning has been rapidly developing and applied to increasingly larger data sets, it has become increasingly harder to manually check every data instance collected. In classification tasks, such as our flare predictions, outliers in training data may mislead the classifier, hence resulting in poor classification performance. Therefore, outlier detection techniques need to be considered for machine learning tasks in order to get improved model performance.

Solar flares are sudden and impulsive releases of magnetic energy in the lower solar atmosphere, mainly in the form of electromagnetic radiation. The vast majority of solar flares take place in active regions (ARs), which are the areas on the Sun where magnetic flux density is exceptionally strong. Flares can be associated with coronal mass ejections (CMEs) and solar energetic particle (SEP) events. When flares occur, electromagnetic radiation is emitted over the entire electromagnetic spectrum, from radio to gamma rays, for large events (e.g., V. M. Sadykov et al. 2017). Based on the peak soft X-ray flux observed by the X-ray sensor on board the Geostationary Operational Environmental Satellite in units of watts per square meter and in the wavelength range from 0.1 to 0.8 nm, solar flares can be classified into five classes denoted by letters A, B, C, M, and X. In this classification, A represents the weakest class (10^{-8}–10^{-7} W m^{-2}) and X represents the strongest class ($\geqslant 10^{-4}$ W m^{-2}). The intensity of events within a class is denoted by a numerical suffix ranging from 1 up to, but excluding, 10 and more, except for X-class flares. This numerical value also serves as a factor for the event's strength within the class. For example, an X2.0 flare is twice as powerful as an X1.0 flare, while an X3.0 flare is 3 times stronger than an X1.0 flare and only 50% more powerful than an X2.0 flare. Each flare class is 10 times stronger than the preceding one. For example, an X1.0 flare is 10 times stronger than an M1.0-class flare and 100 times stronger than a C1.0-class flare. The radiation emitted by intense flares, excluding further repercussions by CMEs and SEPs, can interfere with radio communication and navigation signals (see the NOAA radiation scales (R-scales))[6] and can even be threatening to humans beyond Earth's atmosphere. Moreover, there is no early warning for solar flares given the light speed–propagating electromagnetic emission. Therefore, predicting solar flares is a task of capital importance for space weather prediction.

Machine learning methods on flare prediction have now outnumbered conventional statistical methods (M. K. Georgoulis et al. 2024), with many of these methods using time-series data that can help machine learning models generalize better with the additional information provided (L. E. Boucheron et al. 2015; R. Ma et al. 2017; Y. Chen et al. 2019). However, none of these methods, to our knowledge, paid significant attention to the discovery of outliers as a mechanism of improving prediction performance. In this study, we investigate the existence of outliers in nonflaring-class (N-class) data instances and their impact on the performance of a machine learning algorithm applied to the Space Weather Analytics for Solar Flares (SWAN-SF; R. A. Angryk et al. 2020), which is a multivariate time-series (MVTS) data set created from Solar Cycle 24 observations. We suspect that the timing of possible precursor events leading to a solar flare may not always align with the human-imposed prediction intervals, such as the 12 hr observation and 24 hr prediction, used in SWAN-SF. This discrepancy in timing could lead to instances where flares occur earlier or later than anticipated, resulting in their mislabelling as nonflaring data. These instances, which we refer to as outliers in our study, have the potential to disrupt machine learning predictions.

It is important to clarify that the objective of this study is not to report on the most accurate machine learning algorithm for solar flare prediction or intended for operational purposes. Our attention is focused on discovering whether noise in the SWAN-SF data, introduced through human-imposed labeling of MVTS solar flare data, is actually present and impacting data-driven flare prediction. For this reason, we are interested in how different levels of outlier removal in our training data (Partition 1 from SWAN-SF) affect the solar flare prediction process. To do this, we detect outliers using a tree-based outlier detection algorithm named Isolation Forest (iForest) and then compare the performance of one of the most popular machine learning algorithms, support vector machine (SVM), on a flare classification task with and without outlier detection.

To further justify our reasoning behind using outlier detection to clean nonflaring data and reduce the extreme imbalance in the SWAN-SF data, we focus on qualitative and quantitative analysis of our outliers at the 2.4% contamination rate, which our experimental results suggest to be promising for the automated SWAN-SF data cleaning.

The rest of the paper is structured as follows: In Section 3, we present some previous machine learning work on solar flare detection and provide details about the SWAN-SF data set and the iForest algorithm. In Section 4, we describe the experiments we designed, and present and discuss the results obtained. In Section 5, we conduct an analysis of the outliers detected by the iForest algorithm. Finally, in Section 7, we draw conclusions and suggest potential future research directions.

2. Machine Learning for Solar Flare Prediction

Various machine learning techniques have been implemented for solar flare prediction. Y. Yuan et al. (2010), M. G. Bobra & S. Couvidat (2015), N. Nishizuka et al. (2017), and V. M. Sadykov & A. G. Kosovichev (2017) have all utilized SVM to predict flares, with M. G. Bobra & S. Couvidat (2015) additionally employing a feature selection algorithm and then a parameter ranking to determine the most useful among 25 metadata parameters with a physical background deemed meaningful for flare prediction. M. G. Bobra & S. Ilonidis (2016) predict CMEs in a similar way, by employing SVM in combination with a feature selection algorithm to identify the most distinguishable features derived from photospheric vector magnetic field data. The application of k-nearest neighbors (k-NN) classifiers for flare classification has also been explored by X. Huang et al. (2013), S. M. Hamdi et al. (2017), and R. Ma et al. (2019). R. Ma et al. (2019) introduced the use of k-NN for clustering time-series data using a similarity measure known as dynamic time warping. R. Li et al. (2008) utilized the SVM combined with the k-NN (i.e., the SVM-KNN method) for solar flare prediction and found a better performance compared to using the SVM alone. R. Ma et al. (2017) attempted to predict solar flares by employing time-series decision trees and produced a ranking of parameters regarding their importance for prediction. Other machine learning algorithms, such as random forest (C. Liu et al. 2017; K. Florios et al. 2018), k-means (M. K. Georgoulis et al. 2021), Gaussian process (E. Camporeale et al. 2017), and ensemble learning (T. Colak & R. Qahwaji 2009; J. A. Guerra et al. 2015), have also been applied to flare forecasting.

Deep learning models have also been used to improve flare prediction. N. Nishizuka et al. (2018) developed a deep neural network (DNN) model called Deep Flare Net (DeFN) to predict flares occurring in the following 24 hr in each studied AR. Y. Chen et al. (2022) addressed the issue of data scarcity by employing generative adversarial networks to generate synthetic time-series data of M- and X-class flares, resulting in improved performance with increased training data. A. Ji et al. (2022) explored the enhancement of solar flare prediction by using multiple DNNs. X. Wang et al. (2020) utilized long short-term memory (LSTM) and incorporated time-series data to capture the temporal information, achieving improved skill scores. Z. Jiao et al. (2020) developed a mixed LSTM regression model to predict the maximum solar flare intensity and identified the most effective observation window for solar flare prediction as 24 hr before the forecast issue time. Z. Sun et al. (2022) investigated the performance of LSTM, convolutional neural network, and their stacking ensembles and observed that LSTM trained on two solar cycles achieves higher skill scores.

To the best of our knowledge and understanding, none of the above studies has leveraged machine learning from the perspective of outlier detection. The novelty of our study lies in the identification of outliers using a machine learning algorithm, pursuing the provision of cleaner data for subsequent classification processes.

[6] https://www.swpc.noaa.gov/noaa-scales-explanation

Table 1
Sample Sizes and Imbalance Ratios in Each Partition in SWAN-SF

Partition	Time Span	Class Distribution					Imbalance Ratio	
		X	M	C	B	N	X:N	MX:NBC
1	2010 May–2012 Mar	165	1089	6416	5692	60,130	1:364	1:58
2	2012 Mar–2013 Oct	72	1392	8810	4978	73,368	1:1019	1:62
3	2013 Oct–2014 Mar	136	1288	5639	685	34,762	1:256	1:29
4	2014 Mar–2015 Mar	153	1012	5956	846	43,294	1:283	1:43
5	2015 Mar–2018 Aug	19	971	5753	5924	62,688	1:3299	1:75

3. Background

3.1. SWAN-SF Dataset

SWAN-SF is a benchmark data set introduced by R. A. Angryk et al. (2020), which entirely consists of MVTS data. It provides a unified testbed for solar flare prediction algorithms. The data set contains 4098 MVTS data instances from ARs of 13,641 flare reports spanning over 8 yr of solar AR data from Solar Cycle 24 (2010 May–2018 December). Each MVTS data instance represents a 12 hr observation window of 51 flare-predictive parameters. Within the observation window, each time series is collected with the full 12 minute cadence of the Solar Dynamics Observatory/Helioseismic and Magnetic Imager definitive vector magnetogram data, resulting in five values per hour. Consequently, each observation-window time series for each AR comprises 60 data values. The data instances in SWAN-SF are collected through a sliding-window methodology with a 1 hr step size. An MVTS data instance is labeled by the class of the strongest flare reported within a 24 hr prediction window right after the observation window (namely, with zero latency). If no flare happens or only A-class flares are reported within the prediction window, the data instance is labeled as a flare-quiet instance, denoted by N.

Because of the sliding-window methodology used for SWAN-SF, caution must be taken when dealing with the temporal coherence of data (A. Ahmadzadeh et al. 2021), which can be briefly described as follows: since temporally adjacent time series have over 91% of overlap (i.e., 11 out of 12 hours of the observation window, given the 1 hr sliding step), random sampling of data in order to create nonoverlapping training, validation, and test sets, will fail due to learner biases and possible overfitting. To properly deal with temporal coherence, we take advantage of the fact that the data set is already split into five nonoverlapping partitions, and each partition has approximately the same number of X- and M-class flares. Therefore, the training and testing data sets in our experiments are selected from different partitions to prevent bias. The details of the sample sizes for each partition in SWAN-SF are listed in Table 1.

SWAN-SF also exhibits extreme class imbalance, with a notable disparity between the number of strong-flare instances (referred to as the minority class) and weak-flare instances (referred to as the majority class) within each partition. For instance (Table 1), in Partition 1, there is an imbalance ratio of 1:364 between the X and N classes, and a ratio of 1:58 between the combined category MX and the combined class NBC. Throughout this paper, we consistently use similar terminology to denote the combination of two or more classes, where MX represents the concatenation of X and M classes, and NBC denotes the concatenation of C, B, and N classes. Addressing the pervasive issue of class imbalance is crucial in machine learning–based rare event prediction to ensure the reliability and effectiveness of predictive models (A. Ahmadzadeh et al. 2019a, 2019b, 2021; Y. Chen et al. 2022).

3.2. Isolation Forest

Isolation Forest (iForest, F. T. Liu et al. 2008) is an unsupervised algorithm widely used for anomaly detection. It works based on the idea that anomalies are more susceptible to isolation than normal data points. The algorithm builds an ensemble of trees, called Isolation Trees (iTrees), each of which isolates data points through recursive random partitioning. Collectively, these trees form the iForest, which is used to identify anomalies in new data.

Each iTree is constructed from a random subset of the data set. At each node, data points are split based on a randomly selected feature and a random split value. This recursive process continues until all data points are isolated into individual branches or until a predefined maximum depth is reached. An anomaly score is then assigned to each data point based on the average number of splits across the iForest. Anomalies are identified by the smaller number of splits (or partitions) required to isolate them, as fewer splits imply a shorter path from the root to the terminal node. This is because anomalies, by nature, deviate from normal data points and are easier to separate from the rest of the data set.

Due to its tree-based structure, iForest is simple, fast, effective, and easily interpretable. Additionally, it requires no prior knowledge of the data distribution or assumption about the underlying structure. These advantages have made iForest a popular and flexible choice for anomaly detection tasks across various domains (Z. Ding & M. Fei 2013; Y. Chen et al. 2019; K. Sadaf & J. Sultana 2020) and have contributed to its widespread adoption and steady growth since its introduction in 2008.

Despite its effectiveness, iForest has several limitations. Its performance can be sensitive to the choice of contamination rate, particularly when the actual proportion of outliers in the data set is unknown. The reliance on random partitioning also makes the algorithm less robust when using a small number of iTrees. Furthermore, iForest may struggle to capture complex correlations between features, which can limit its performance when feature interdependence is critical.

Nonetheless, we employ iForest in this study for preliminary experiments because the primary objective of the study is not to advocate for the superiority of iForest or any specific anomaly detection algorithm. Instead, our focus is to demonstrate the importance of outlier detection as a preprocessing step and to evaluate its impact on rare event forecasting tasks.

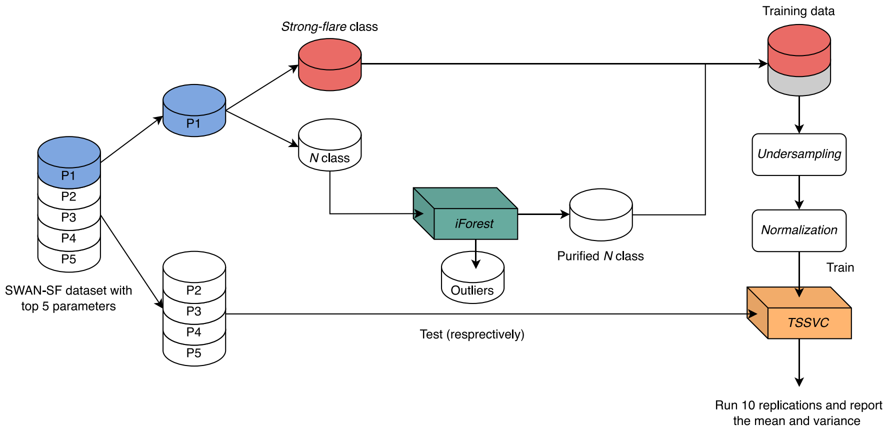

Figure 1. Data flow in our experiment. When the contamination rate for iForest equals 0, that means no data will be removed from the N class and that is our baseline. The red and gray cylinders may consist of different data in different experiments, which we talk about in Section 4.4. In N-N-X, the red cylinders consist of X-class flare data and the gray cylinder contains the data of N class, while the red cylinder consists of MX-class data and the gray cylinder consists of NBC-class data in N-NBC-MX.

3.3. TimeSeriesSVC

The classifier used for the binary classification in this study is TimeSeriesSVC (TSSVC) from tslearn (R. Tavenard et al. 2020). Tslearn is a Python package that provides machine learning tools for time-series analysis. It is built on Scikit-learn (F. Pedregosa et al. 2011), NumPy (C. R. Harris et al. 2020), and SciPy (P. Virtanen et al. 2020) libraries. The TSSVC is a time-series-specific SVM classifier that accepts all regular kernel functions and the global alignment kernels (M. Cuturi 2011), which is a kernel function designed for time series.

Similar to the traditional SVM, TSSVC also uses the soft margin constant C and the kernel coefficient γ as key hyperparameters. The parameter C balances the trade-off between minimizing training error and avoiding overfitting, while γ controls the influence radius of the support vectors. To ensure fair comparisons across different contamination rates and prevent hyperparameter tuning from affecting model performance and not being able to distinguish if the observed improvements are due to outlier removal or hyperparameter tuning, we consistently employ the pretuned configuration from A. Ahmadzadeh et al. (2021), which is a radial basis function kernel with $C = 100$ and $\gamma = 0.01$, for training each model.

4. Experiments and Results

4.1. Experiment Setup

The experiment design and data flow are summarized in Figure 1. In this section, we explain more details of several important steps during our experiments.

4.1.1. Train–Test Split

Since there are no ground truth labels for outliers in SWAN-SF, we assess the existence of outliers on the performance of the solar flare classification task. As mentioned in Section 1, among the five classes of flares, M and X are of the most importance. Consequently, a commonly followed approach in solar flare prediction studies adopts binary classification, designating the M and X classes as the positive class, while grouping the remaining classes into the negative class.

To ensure robust calibration and accurate evaluation of machine learning algorithms, a crucial step involves partitioning the SWAN-SF data set into distinct training and testing sets, which are readily offered by the five SWAN-SF partitions. In our experiments, we use Partition 1 for outlier detection and training the machine learning model, and Partitions 2–5 for testing. We first detect outliers from N-class data instances in the training data set using iForest, then we train our classifier with a cleaner data set by providing the classifier with the N class excluding the detected outliers.

4.1.2. Random Undersampling

Before training the classifier, we address the challenge of extreme class imbalance by implementing an undersampling strategy. Random undersampling is employed to establish a balanced 1:1 ratio between the minority class and the majority class. This involves selecting a subset of random instances from the majority class to match the quantity of the minority class. This, obviously, comes with the trade-off of potential information loss from the majority class. To ensure the reliability of our results and demonstrate the classifier's robustness, we repeat the undersampling process 10 times. In each iteration, a different set of random data instances from the majority class is selected, resulting in varied negative class samples for training. The outcomes from the 10 runs are aggregated, and we report both the average and the variance as indicators of the final model performance. This approach allows us to capture the overall effectiveness of the classifier while accounting for the variability introduced by different undersampling instances.

4.1.3. Normalization

Since different parameters have different ranges of values and such a difference will introduce bias during training, a normalization step is followed after undersampling. We follow a normalization procedure called local normalization in A. Ahmadzadeh et al. (2021). A min–max normalization will be applied to training and testing data separately using their own extrema. Furthermore, the size of training data for classification after undersampling will be small due to the initially small number of positive-class data instances (i.e., minority class). To reduce the impact of the curse of dimensionality, where the data needed to obtain a reliable result grows exponentially with the dimensionality, we choose to use five physical parameters that are suggested as the ones of

Table 2
Top Five Predictive SHARP Parameters and Their Ranking[a]

Keyword	Description	Formula	Rank
TOTUSJH	Total unsigned current helicity	$H_{c_{\rm total}} \propto \sum \quad |B_z \cdot J_z|$	1
TOTBSQ	Total magnitude of Lorentz force	$F \propto \sum B^2$	2
TOTPOT	Total photospheric magnetic free energy density	$\rho_{\rm tot} \propto \sum(\boldsymbol{B}^{\rm obs} - \boldsymbol{B}^{\rm pot})^2 dA$	3
TOTUSJZ	Total unsigned vertical current	$J_{z_{\rm total}} = \sum |J_z| dA$	4
ABSNJZH	Absolute value of the net current helicity	$H_{c_{\rm abs}} \propto |\sum B_z \cdot J_z|$	5

Note.
[a] M. G. Bobra & S. Couvidat (2015); A. Yeolekar et al. (2021).

most predictive significance (M. G. Bobra & S. Couvidat 2015; A. Yeolekar et al. 2021). Those parameters and their descriptions are listed in Table 2.

4.1.4. Contamination Rate Selection

As a contamination rate is required for iForest to work and the actual portion of outliers is unknown, we incrementally go through a set of contamination rates. The contamination rates are chosen within the range of 0%–30%, as a contamination rate of 30% is uncommon in real-world scenarios and we notice that a 30% removal of data instances becomes severe enough to detrimentally affect the machine learning process. All the contamination rates that we test can be seen on the *x*-axis in Figures 2 and 3.

4.2. Two Experiments

Two experiments are carried out to assess the impact of outliers on the machine learning process, which are named N-N-X and N-NBC-MX, respectively.

N-N-X means that we detect outliers from N-class data instances and construct the training data for binary classification using X and N classes only with X being the positive class and N being the negative class. This is the simplest case since X and N are the two most extreme classes and it is easier for the classifier to distinguish between them. We start with this experiment because we want to reduce the influence of intermediate classes and explore the benefits of the outlier detection approach more illustratively on this best-separated data set.

N-NBC-MX is the experiment where a more realistic classification problem is investigated after detecting outliers. In this experiment, all five classes are involved with MX being the positive class and NBC being the negative class for the binary classification. After observing the improvement in N-N-X, this experiment is carried out to examine if this improvement is preserved in the most realistic case.

4.3. Performance Verification Metrics for Classification

Many measures have been developed to evaluate the deterministic performance of classifiers based on the four quantities of the confusion matrix: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). For rare event classification tasks, the testing data is always imbalanced. In such case, normal evaluation metrics, like accuracy ((TP + TN)/(TP + TN + FP + FN)), could be misleading (A. Ahmadzadeh et al. 2021). That is, when the testing data is significantly imbalanced (e.g., when the negative class is significantly larger than the positive class, like in our case), just simply predicting all the data instances as the majority class will produce a high value, but in fact, the model is not generalizing well. Therefore, choosing proper evaluation metrics is important for achieving reliable and generalized model performance.

In this study, we use two popular evaluation metrics in the space-weather community, the true skill statistic (TSS; D. S. Bloomfield et al. 2012) and the updated Heidke skill score (HSS2; M. G. Bobra & S. Couvidat 2015).

$$\mathrm{TSS} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} - \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{1}$$

$$\mathrm{HSS2} = \frac{2(\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FN} \cdot \mathrm{FP})}{\mathrm{P}(\mathrm{FN} + \mathrm{TN}) + \mathrm{N}(\mathrm{TP} + \mathrm{FP})}. \tag{2}$$

TSS, as shown in Equation (1), measures the difference between the TP rate (i.e., recall, defined as TP/(TP + FN)) and the false alarm rate (defined as FP/(FP + TN)). TSS ranges from −1 to 1 and the higher the value the better the performance, with −1 indicating that all predictions made by the classifier are correct if precisely reversed.

One potential issue with TSS is that it focuses on the balance between TP and TN (discrimination) and does not consider the prevalence between positive and negative classes. When the data is highly imbalanced, TSS can be heavily influenced by the majority class, leading to an overestimation of model performance. HSS2, on the other hand, emphasizes the model's ability to make correct positive predictions while minimizing false alarms (reliability). It measures the fractional improvement of prediction that the classifier has over one particular class of no-skill models, namely, a random chance model. The same as TSS, HSS2 ranges from −1 to 1, where 1 means a perfect model and −1 means the reverse assignment of labels to all data instances. Therefore, combining TSS and HSS2 provides a well-rounded assessment of a model's performance on imbalanced data by considering both discrimination and reliability.

4.4. Results

In the N-N-X experiment scenario, as illustrated in Figure 2, we observe a significant enhancement in model performance across all testing partitions. The intervention toward outlier removal, reflected in the increasing contamination rate, results in notable improvements in both TSS and HSS2. Removing detected outliers substantially increases TSS, nearly reaching 1 for all partitions except Partition 3. HSS2 shows a doubling or even multiple-fold increase. For example, in Partition 2 (Figure 2(a)), with a contamination rate of 2.1%, the average of TSS over 10 repetitions of random undersampling increases from 0.332 to 0.995, which indicates a 200% improvement,

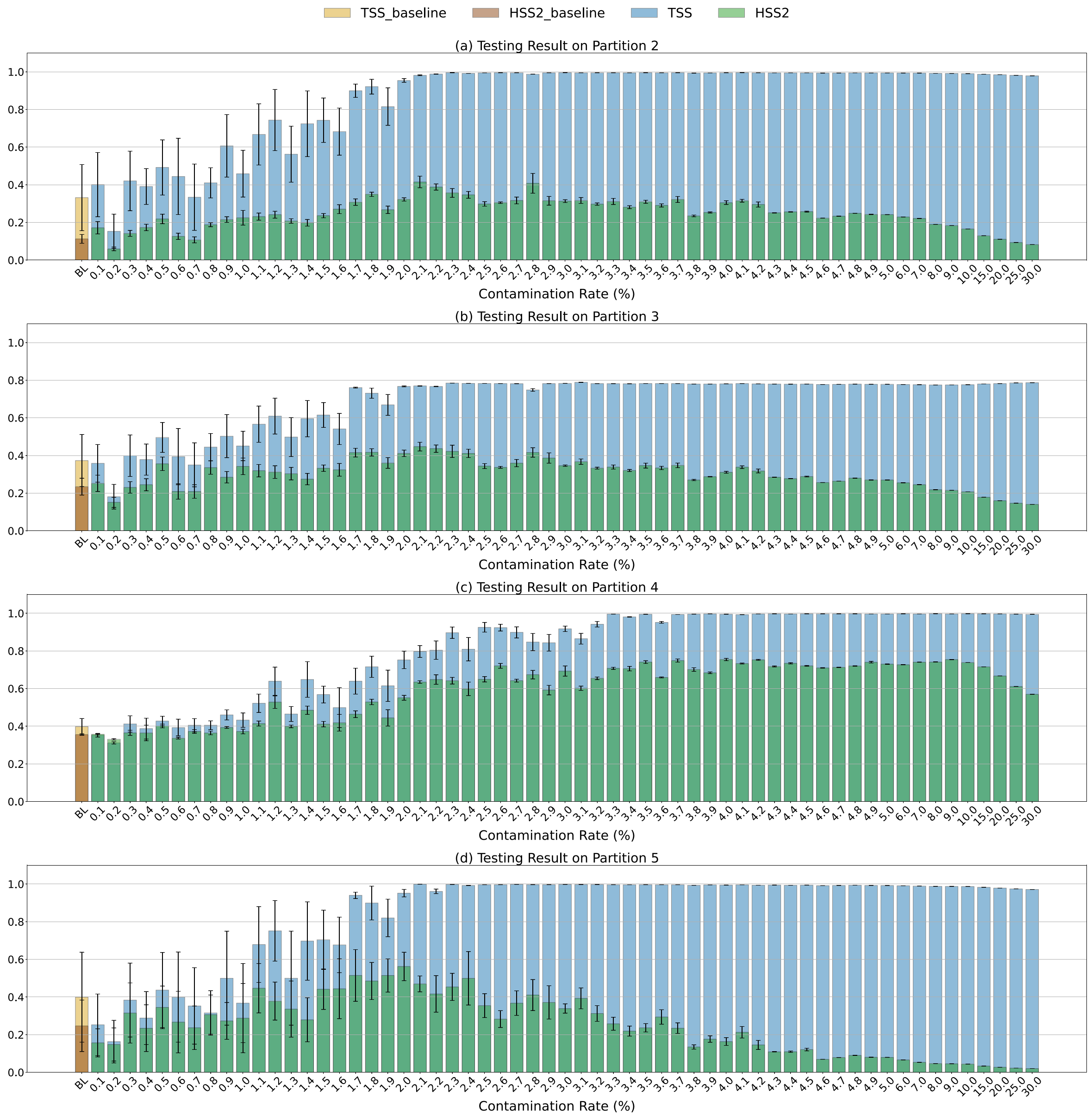

Figure 2. Results of the N-N-X experiment. The contamination rate, which configures iForest to detect a specific percentage of N-class data as outliers, is depicted on the *x*-axis. We compare our baseline model, trained on unaltered data and represented in different colors. Each bar's height represents the mean value, and the accompanying black error bar reflects the variance of the respective metric across 10 repetitions.

and the average HSS2 improves from 0.113 to 0.312, meaning a 176% improvement. Similarly, in Partition 4 (Figure 2(c)), at a contamination rate of 4.0%, TSS elevates to 0.997 from 0.398, marking a 151% improvement, and HSS2 reaches 0.75, surpassing the baseline average of 0.356 by 110%.

However, beyond a certain contamination rate, HSS2 starts to decrease, while TSS remains relatively stable. This could be due to the N-class data becoming less representative when too many samples are removed. Consequently, the decision boundary of the SVM classifier shifts away from the positive class, leading to a saturation of the TSS. Given the extreme class imbalance in the test partitions, too much removal of N-class data has a minimal impact on the false alarm rate, as the change in FP is offset by the large number of TN. This underscores why the TSS alone, as discussed in Section 4.3, may not fully capture model performance in imbalanced data sets. Although the TSS remains stable, the rise of FP causes a decrease in the HSS2, indicating a decline of the overall model performance.

Interestingly, the model performance for Partition 4 (Figure 2(c)) exhibits a distinct trend compared to the other testing partitions. As the contamination rate increases, the performance rises and stabilizes before declining when the contamination rate becomes unrealistic. This discrepancy may be attributed to differences in data distributions collected during various phases of the solar cycle. Solar activity varies over time, potentially causing distinct distributions in different data partitions. Consequently, the classifier's performance may vary across partitions, depending on the period during which the data was collected. We hypothesize that Partition 4 may have a clearer separation between N and X classes, allowing the outlier removal process to maintain the decision boundary until normal data points begin being identified as outliers. Additionally, the model demonstrates increased robustness at

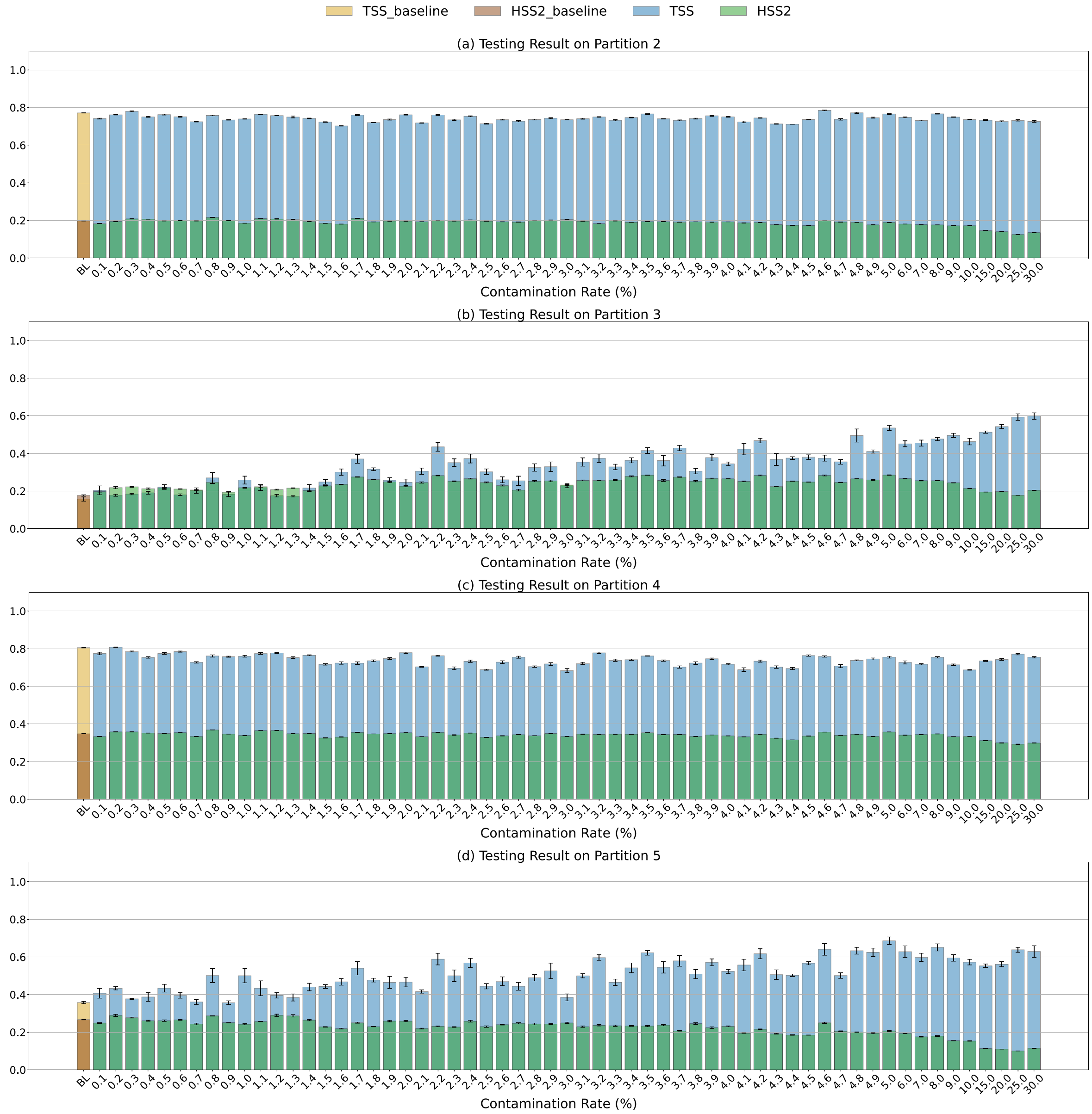

Figure 3. Same as Figure 2, but for the N-NBC-MX experiment.

higher contamination rates, as evidenced by reduced variance in performance across 10 repetitions of random sampling.

In the N-NBC-MX experiment, shown in Figure 3, we observe continued improvements in the more complex classification task, particularly in Partitions 3 and 5. For instance, in Partition 3 (Figure 3(b)), at a contamination rate of 5.0%, the average TSS increases by 239%, from 0.158 to 0.535, while the average HSS2 improves by 60%, rising from 0.177 to 0.283. Similarly, in Partition 5 (Figure 3(d)), at a contamination rate of 0.8%, the average TSS grows from 0.358 to 0.501, and the average HSS2 increases from 0.267 to 0.287.

Similar to the N-N-X experiment, the trends show an increase and stabilization in TSS, while HSS2 increases initially but decreases at higher contamination rates. Notably, the rise in TSS for Partitions 3 and 5 is accompanied by a larger variance, while the variance of HSS2 remains consistent. This could be due to the imbalance in the testing data. Each time the decision boundary shifts due to random undersampling, the components of the confusion matrix are affected. However, when the first term in Equation (1) fluctuates, the second term remains relatively stable due to the large value of TN. HSS2 is less sensitive to such changes because of the multiplicative factors in its formula (Equation (2)), resulting in more stable values. Consequently, TSS demonstrates greater variance compared to HSS2.

On the other hand, no significant improvements in performance or robustness are observed for the other testing partitions. This may be attributed to the greater similarities between the instances of M and C classes in those partitions. While outlier removal from the N class helps better distinguish the MX and N classes, it may increase misclassifications within the C class, leading to an unchanged confusion matrix and, consequently, no improvements in model performance.

It is also interesting to note that, with more classes involved, the performance of the baseline model in N-NBC-MX is better than in N-N-X, contrary to our initial expectations. The reason may be that in N-NBC-MX, there is more training data, mitigating the curse of dimensionality. This is also the reason

Table 3
The Imbalance Ratio in Training Partition (Partition 1) for Different Contamination Rates

Conte. Rate	Outliers Count	X:N	Size of N*	MX: NBC	Size of N*BC
0.0%	0	1:364	60,130	1:58	72,238
1.0%	601	1:360	59,529	1:57	71,637
2.4%	**1444**	**1:355**	**58,686**	**1:57**	**70,794**
5.0%	3007	1:346	57,123	1:55	69,231
10.0%	6013	1:328	54,117	1:52	66,225
15.0%	9020	1:309	51,110	1:49	63,218
20.0%	12,026	1:291	48,104	1:46	60,212
25.0%	15,033	1:273	45,097	1:44	57,205
30.0%	18,039	1:255	42,091	1:41	54,199

Note. *Conta.* stands for contamination. The notation N* denotes the purified N class after outlier detection, while N*BC signifies the combination of purified N, B, and C classes. The selected contamination rate of 2.4% is determined as the optimal value, shown in bold, and further details explaining this choice will be provided in Section 5.

why the baseline model in N-NBC-MX is more robust (less variance of TSS and HSS2) than the one in N-N-X across all partitions. Additionally, as shown in Table 3, we achieve improved performance without significantly altering the class ratio in training partition (Partition 1) for both classification tasks, highlighting the effectiveness of our approach in enhancing predictive capabilities while preserving the integrity of the original data.

5. Outlier Analysis

As previously discussed in Section 4.4, we observed that the removal of data instances identified by iForest as potential outliers leads to an enhancement in the performance of SVM for flare prediction. The next step is to investigate the origins of these outliers. Obviously, different outliers are detected at different contamination rates. To focus our analysis on the more influential outliers in machine learning, we select a contamination rate that achieves consistent improvement across all four testing partitions in the more realistic flare prediction experiment (i.e., N-NBC-MX). Since the model performance remains stable on Partitions 2 and 4, and the improvement of TSS is relatively more significant than the HSS2 (by percentage) on Partitions 3 and 5, our selection criterion prioritizes maximizing the improvement of TSS while optimizing HSS2 on Partitions 3 and 5. Empirically, a contamination rate of 2.4% is selected, leading to a detection of 1444 outliers originating from 37 ARs.

5.1. Co-occurrence between Outliers and Flares

We introduce a few definitions of ARs for a clearer discussion of our findings:

1. MX_ARs denotes ARs that hosted at least one M- or X-class flare during their lifetimes. By "lifetime," we hereafter mean the time interval during which magnetic properties could be obtained from these ARs as they traversed across the solar disk.
2. BC_ARs denotes the active regions in which, during their lifetime, at least one B- or C-class flare was reported as the strongest event.
3. NF_ARs denote the ARs in which, during their lifetime, no flare (B or above) was reported. We also refer to these instances as never-flaring ARs.

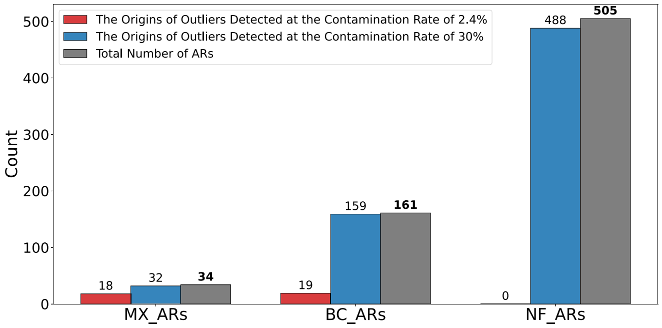

Figure 4. The count of distinct ARs containing the outliers detected at two contamination rates (2.4% vs. 30%) in Partition 1. The red bars represent the number of different ARs from which the detected outliers originate at a 2.4% contamination rate, considered the optimal one. In contrast, the blue bars indicate the count of different ARs associated with the detected outliers at a 30% contamination rate, the highest level we tested. For general context, the gray bars provide a total count of different ARs observed.

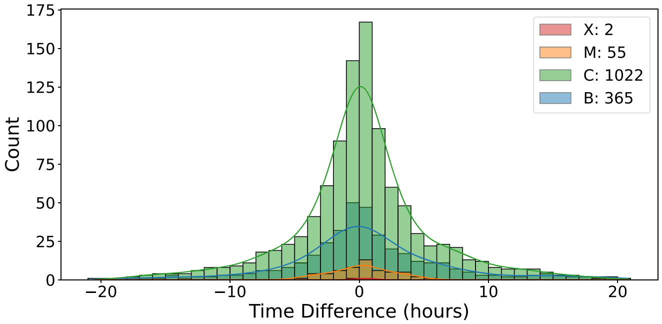

Figure 5. Time difference between 2.4% outliers from N-class samples in Partition 1 of SWAN-SF and their temporally nearest flares.

Grouping our selected outliers by AR type, as seen in Figure 4, all 1444 outliers at 2.4% contamination rate originate from MX_ARs and BC_ARs, i.e., ARs with a history of flaring events. Inspired by this finding, we further delve into the temporal aspect by calculating the time difference between those outliers and their closest flares. The time difference (TD) for an outlier is defined as the difference between the starting time of the observation window of the outlier, O_{start}, and the starting time of its temporally nearest flare, F_{start}:

$$TD = O_{\text{start}} - F_{\text{start}}. \tag{3}$$

TD can be positive or negative, signifying whether the closest flare occurred before or after the observation of the outlier. As shown in Figure 5, the time difference of the outliers follows a normal distribution centered near 0. For most outliers, there is a flare that happened within 20 hr either before or after the observation of the outlier; for those outliers closest to strong-class flares, the flares happened within 5 hr, suggesting a potential causal relationship between flares and outliers.

Several plausible explanations account for the co-occurrence between outliers and flares. The first factor involves the relationship between magnetic energy and the flare process. Some of the magnetic properties of ARs (like TOTPOT) are expected to be directly associated with the free magnetic energy stored in them. According to M. J. Aschwanden et al. (2017), the ratio of

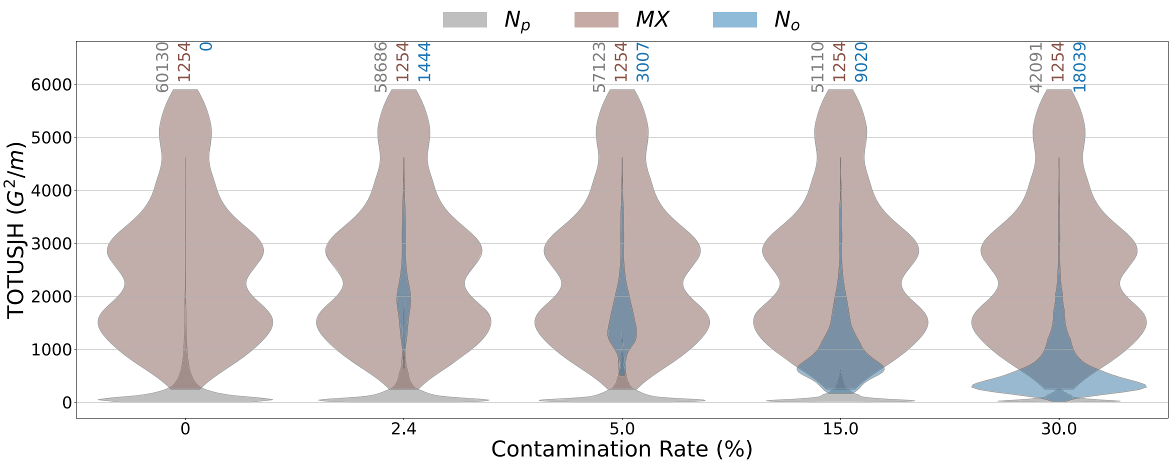

Figure 6. Violin plot depicting the median of TOTUSJH for each data instance in Partition 1. In this plot, N_p denotes N-class instances postoutlier removal, MX represents the combined M- and X-class data, and N_o signifies our outliers detected at a specific contamination rate by iForest. The sum of N_p and N_o constitutes the original N class, expressed as $N_p + N_o = N$. To enhance visualization, only a subset of contamination rates is displayed. The violin widths are adjusted proportionally for better visualization and not showing the actual data imbalance ratios between different groups, since N_p counts are much bigger than the other two groups. The actual counts of different groups are shown above the violins represented in the corresponding colors.

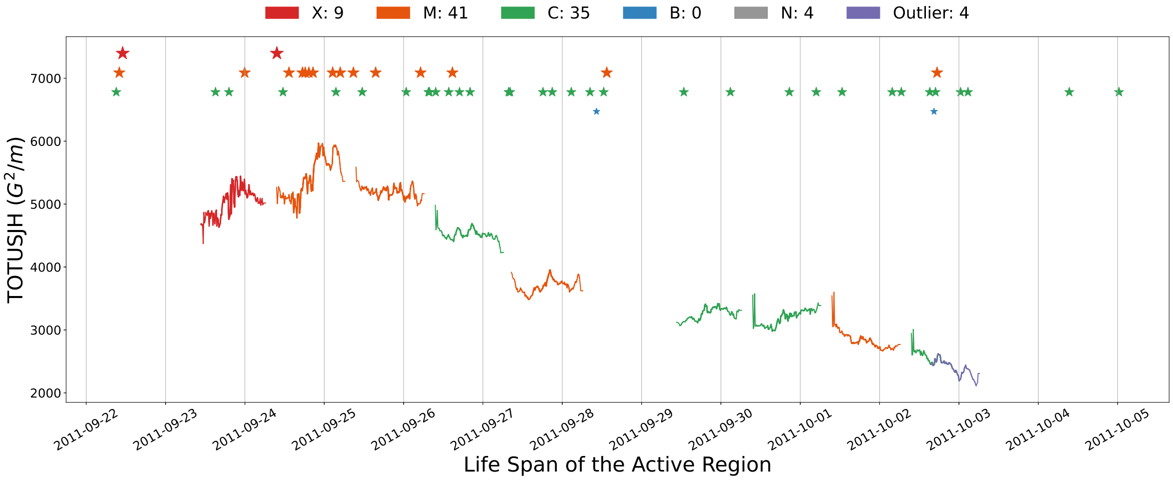

Figure 7. This figure shows the time series of TOTUSJH in an AR with HARP number 892 in Partition 1 of SWAN-SF. Each star corresponds to the peak time of a flare, with its color indicating the flare's class. The color of a time-series segment changes when the most intensive flare within its prediction window (24 hr after the last time stamp of the time-series segment) changes.

the energy radiated in soft X-rays to the free magnetic energy is $E_{\rm rad}/E_{\rm mag} = 0.004 \pm 0.130$, i.e., is accompanied with significant uncertainties. Consequently, while possessing enough magnetic energy to produce the M-class flare (which will correspond to the MX-flare category), the AR may produce several weaker C-class events (all of them being in the NBC-flare category). This may explain why the outliers mostly correspond to MX_ARs or BC_ARs for small contamination rates. The second important factor is the role of the soft X-ray background. The flare classes represent the peak fluxes of the whole-Sun-integrated soft X-ray emission rather than the emission from a particular AR. Therefore, the flare classes can stably be at the level of the C-class flares during high periods of solar activity. We suspect that this may contribute to the prevalence of C-class flares among the closest flares for these outliers.

5.2. Similarity between Outliers and Strong-flare Data

The observations discussed in Section 5.1 give rise to the thought that the presence of outliers may be because the influence of some strong flares last long enough to have impacts on the time series during the observation window of some N-class instances, which makes them more similar to the strong-flare instances. To investigate this hypothesis, we initially create a violin plot depicting the statistical characteristics of magnetic parameters for various data groups across different contamination rates. As an example, Figure 6 displays the violin plot representing the median values of TOTUSJH for distinct data groups. Notably, at a contamination rate of 2.4%, the outlier group (N_o) closely resembles the combined M- and X-class data (MX). However, as the contamination rate increases, N_o progressively diverges from MX and converges toward N_p. This observation provides further support for the declining performance observed in our experiments as the contamination rate grows.

We illustrate an example of outliers in Figure 7. As depicted, during the observation windows of these N-class instances, an M-class flare was occurring. Consequently, their magnetic parameters closely resemble those of the instances immediately labeled as M class based on the labeling mechanism in SWAN-SF. This statistical similarity to the strong-flare class sets them

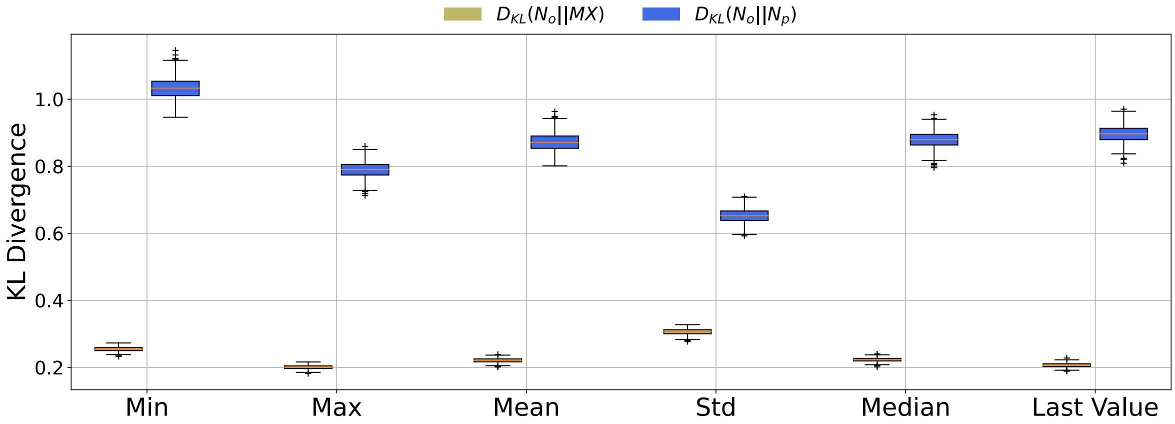

Figure 8. This plot shows the box plots of the KL divergence values obtained from 500 random selection processes for six common statistics extracted from TOTUSJH.

apart from the majority of N-class instances, leading to their detection as outliers.

5.3. Quantifying Similarity with Kullback–Leibler Divergence

To quantitatively assess the distinction in similarity between N_o and MX, as well as between N_o and N_p, we compute the Kullback–Leibler (KL) divergence (S. Kullback & R. A. Leibler 1951) for various statistical characteristics across different data groups.

The KL divergence measures how much one probability distribution P diverges from a reference probability distribution Q while ensuring P and Q contain the same number of data points, I. The KL divergence is always nonnegative, indicated as $D_{\mathrm{KL}}(P||Q) \geqslant 0$, with smaller values signifying greater similarity between the two distributions. A number of zero implies that the two distributions are identical.

$$D_{\mathrm{KL}}(P||Q) = \sum_{i\in I} P(i) \cdot \log\left(\frac{P(i)}{Q(i)}\right). \tag{4}$$

As the KL divergence necessitates an equal number of data points in both data sets, we employ a random sampling method. For instance, when computing $D_{\mathrm{KL}}(\mathrm{N_o}||\mathrm{MX})$, we randomly select 1254 samples from the N_o data set, which originally contains 1444 data points. To ensure the robustness of our results, we repeat this random selection process 500 times and subsequently create a box plot based on these 500 calculated values.

As depicted in Figure 8, the KL divergence between N_o and MX is consistently small and significantly lower than the divergence between N_o and N_p for each statistical characteristic. This observation suggests that the outliers bear greater resemblance to strong-flare data. This alignment may provide additional evidence for the observed performance decline when including those outliers in the training data.

5.4. Enhancing Model Performance by Treating Outliers as Strong-flare Data

After confirming the similarity between outliers and strong-flare data, we conduct another experiment. We further modify the N-NBC-MX experiment by treating the outliers as if they are strong-flare data and incorporating them into the training data set, referred to N-N_pBC-N_oMX. Our hypothesis is that treating these outliers as strong-flare data will lead to improved model performance for two main reasons: (1) the classifier would be trained on a larger data set, expanding from 2508 to 5396 instances, and (2) the introduction of more representative positive samples would enable the SVM classifier to establish a more effective decision boundary.

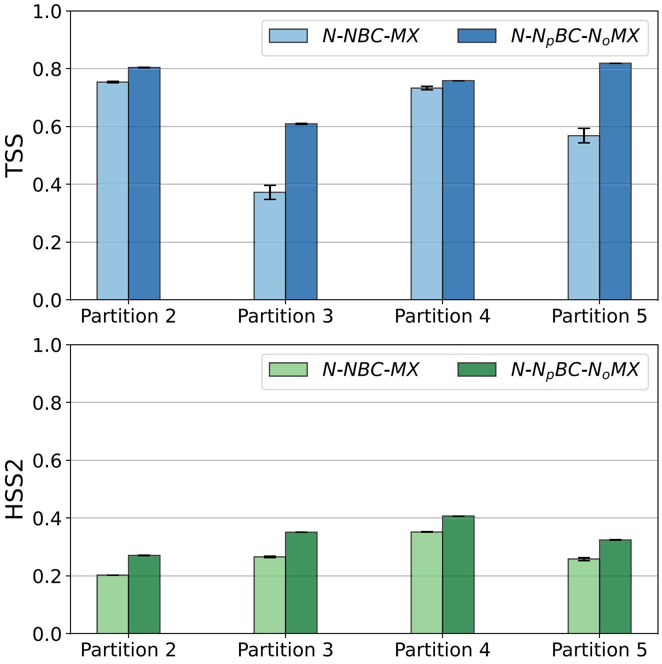

Figure 9. This plot shows the comparison between the model performance in N-NBC-MX at 2.4% contamination rate, where the outliers are removed, and the model performance in N-N_pBC-N_oMX, where those outliers are treated as strong-flare data.

In Figure 9, we present a comparison of model performance between scenarios where the outliers are removed and where they are treated as strong-flare data, both at a 2.4% contamination rate. As depicted, treating the outliers as strong-flare data results in further performance enhancement. For instance, in Partition 3, TSS exhibits an average increase from 0.372 to 0.609, representing a 64% improvement, while HSS2 sees a 32% improvement, rising from 0.266 to 0.351. In Partition 5, TSS shows a substantial improvement from 0.568 to 0.819, and HSS2 also experiences an increase from 0.258 to 0.324.

Furthermore, we provide the results for all the contamination rates in Figure 10. As shown in this figure, further enhancements were also observed at several other contamination rates, particularly in Partition 3 and Partition 5.

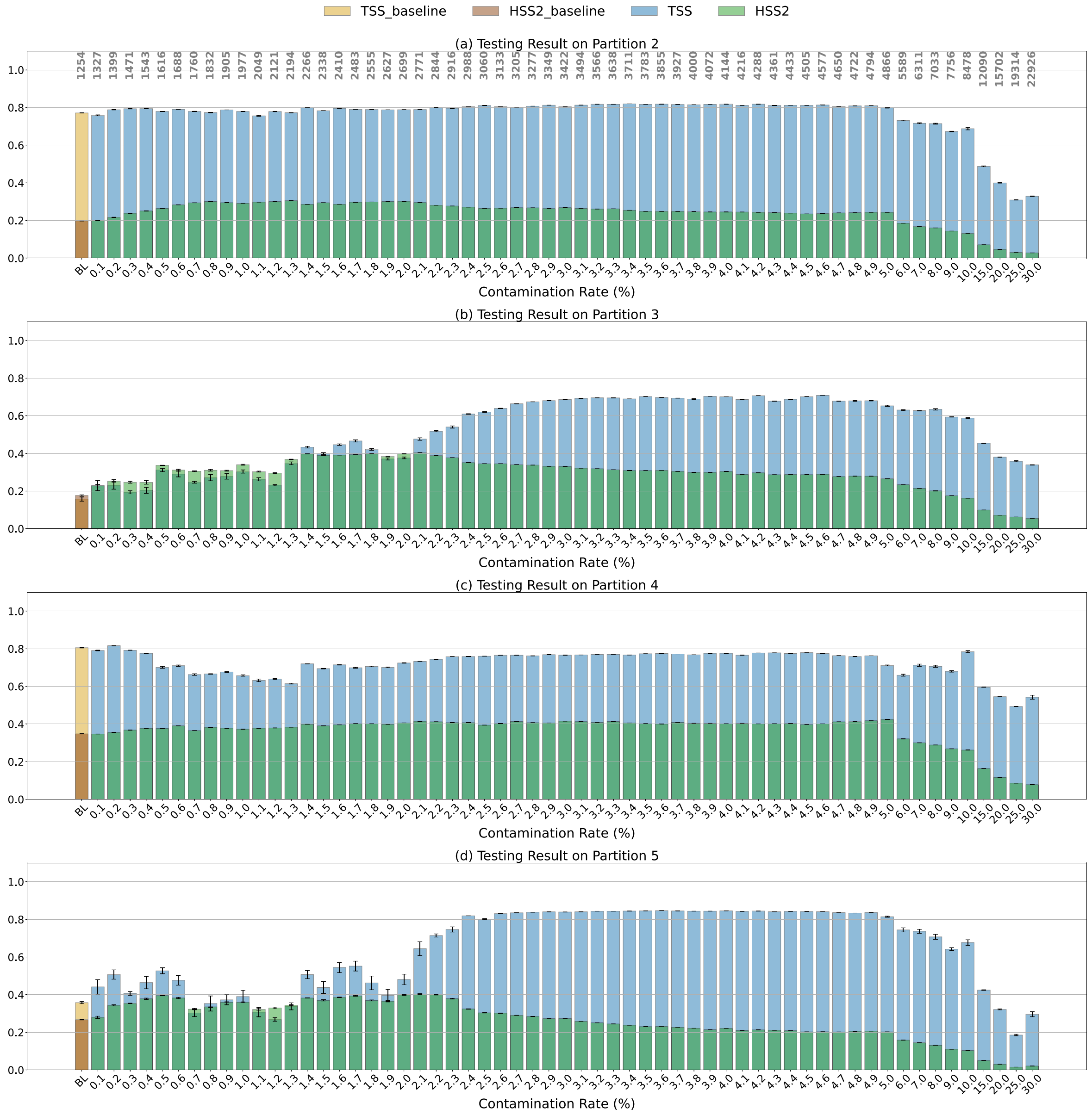

Figure 10. The result of N-N$_p$BC-N$_o$MX. The size of training data for each contamination is shown on the top of panel (a).

6. Summary of Experiments

In this section, we summarize the experiments conducted in this study and discuss the aggregated observations from the experimental results and outlier analysis. To clearly illustrate our points, we selected three contamination rates from the full list to represent the “base case” (0.0%), “our recommendation” (2.4%), and “going overboard” (30.0%). Those three experiments, along with the number of detected outliers and the size of the training data at each contamination rate, are summarized in Table 4.

The results of those experiments are shown in Figure 11. To intuitively compare model performance, we plot the TSS and the updated HSS2 against each other, following the approach used in J. Wen & R. A. Angryk (2024). Any points that lie on the same curve are equidistant from the top right corner (1,1) (i.e., the perfect model) and are considered to have the same model performance.

As shown in Figure 11(a), when only N and X classes are involved, outlier removal significantly improves model performance. Since the training data sizes are unaffected by different contamination rates (as outliers are only removed in N class), the improvement suggests the existence of outliers in the N class that hinder the separation between N and X classes. However, excessive outlier removal (i.e., 30% contamination rate) is detrimental, as it removes too many normal instances, making class N too sparse and decreasing model performance.

In Figure 11(b), while our recommended contamination rate still outperforms the baseline on Partitions 3 and 5, it shows the difficulty of real solar flare prediction. When all classes are included, even though outlier removal does not hurt the model performance, due to the overlap between C and M classes along with many detected outliers being closer to M- and C-class samples (as shown in Figure 5), the disproportions in the data after outlier removal remain, thus not aiding the separation between NBC and MX classes. Therefore, the model

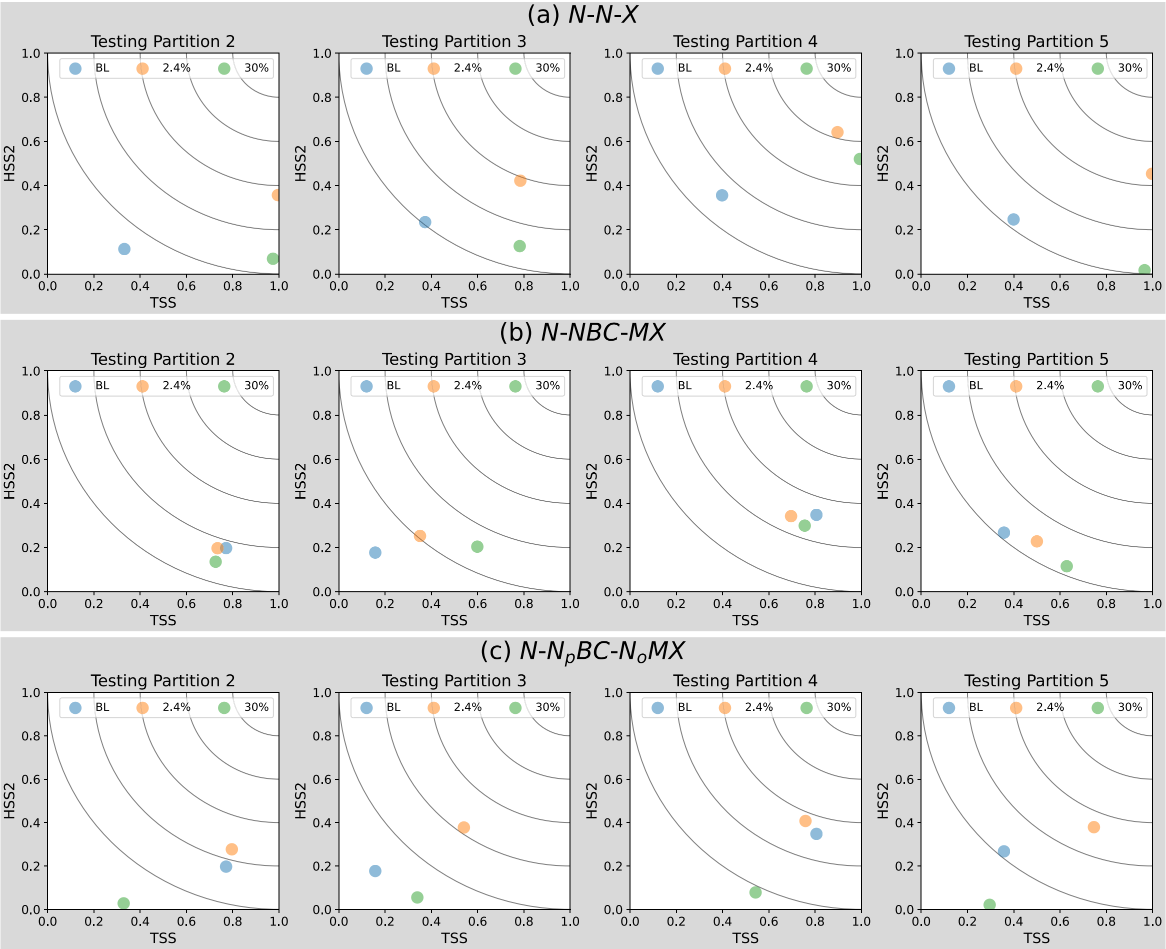

Figure 11. Results of three experiments with three selected contamination rates for outlier detection. Each dot represents the mean values of TSS and HSS2 of the 10 runs of random undersampling for the corresponding contamination rate.

Table 4
Summary of Experiments

Experiment	Description	Conta. Rate	Outliers Count	Training Size
N-N-X	Detect and remove outliers from the N class and perform binary classification between the N class and the X class.	0.0%	0	330
		2.4%	1444	330
		30.0%	18,039	330
N-NBC-MX	Detect and remove outliers from the N class and perform binary classification between the nonflaring class (N, B, and C) and the flaring class (M and X).	0.0%	0	2508
		2.4%	1444	2508
		30.0%	18,039	2508
N-N_pBC-N_oMX	Detect outliers from the N class and treat the detected outliers N_o as MX, then perform binary classification between the nonflaring class and the flaring class.	0.0%	0	2508
		2.4%	1444	5396
		30.0%	18,039	38,586

Note. A contamination rate of 0.0% is the baseline, indicating no outlier removal for class N. The 2.4% rate is our recommended setting, while the 30.0% rate represents an extreme condition for testing purposes. *Training Size* refers to the size of the training data used to train the classifier, which is balanced after undersampling.

performance in N-NBC-MX does not improve significantly. Additionally, excessive outlier removal does not degrade the model performance as much as it does in the N-N-X experiment, and even yields better performance in Partition 3.

In N-N_pBC-N_oMX (Figure 11(c)), detected outliers are treated as the flaring class (MX), resulting in different training data sizes for each contamination rate. Treating detected outliers at 2.4% as the flaring class significantly improves model performance compared to (b), likely due to the increased training data size (from 2508 to 5396). This approach not only enriches the N class (NBC) to be trained but also suggests that the detected outliers are more similar to the flaring class, thereby aiding flare prediction. Conversely, a contamination rate of 30% increases the training data size dramatically to 38,586 but the model performance decreases and is even worse than the baseline. This is because such high outlier detection makes the N class less representative and adds too many normal nonflaring samples to the flaring class, complicating the separation between the flaring and N classes.

7. Conclusion and Future Work

In this study, we explored the presence of outliers within the N class of the benchmark multivariate time-series data set used for solar flare prediction, SWAN-SF. Utilizing iForest, a tree-based algorithm, we successfully identified and characterized outliers within the N-class data.

Our investigation encompassed two distinct experiments: N-N-X and N-NBC-MX, representing extreme and realistic scenarios, respectively. These experiments were designed to assess the influence of outliers on machine learning models. In both experiments, the elimination of iForest-detected outliers consistently led to noteworthy improvements in model performance, as measured by two prevalent evaluation metrics: the TSS and the updated HSS2.

While readers experienced with machine learning for flare forecasting may notice low HSS2 scores in some of our experiments, it is important to clarify that our focus is not on obtaining the highest prediction scores for operational flare forecasting. Instead, our investigation centers around a singular and challenging scenario. We aim to understand how cleaning data in Partition 1 of SWAN-SF through gradual outlier removal impacts predictive models when deployed on other partitions (Partitions 2–5), many of which span distant periods in the Sun's overall activity (solar cycle). This approach is more academic/theoretical, as operational communities would not typically restrict themselves to learning from a limited data set collected during the Sun's quiet period (Partition 1 covers 2010 May–2012 March) while optimizing for solar flare prediction in later periods (e.g., 2015 March–2018 August in Partition 5). Therefore, in each of our experiments, we present base case (yellowish bars in Figures 2, 3, and 10) against which we compare effectiveness of our outlier analysis for the sake of machine learning.

Our findings were enriched by an examination of the spatial and temporal co-occurrence between these outliers and intense solar flares. Intriguingly, we observed a higher degree of similarity between outliers and strong-flare data compared to the similarity between outliers and the remaining N-class data. Subsequently, we quantified this observation using the KL divergence. This analysis reveals an affinity between outliers and strong-flare behavior.

Additionally, we introduced a novel experimental approach by treating the outliers as if they were instances of strong-flare data. This experimental setup yielded further enhancements in model performance, offering support for our underlying hypothesis: that outliers in nonflaring instances exhibit behaviors akin to those of strong-flare instances.

This study is the beginning of our exploration into outlier detection for solar flare prediction. There are several avenues for our future investigations, including the detection of outliers within other weak-flare classes (B or C), exploring alternative outlier detection algorithms tailored for time-series data, and integrating outlier detection with complementary strategies, such as data augmentation, to address challenges like class imbalance in the SWAN-SF data set. These potentially promising directions will form the foundation of our ongoing efforts to advance solar flare prediction, ultimately making it more accurate.

Acknowledgments

This project is supported in part by funding from CISE, MPS, and GEO Directorates under NSF award No. 1931555, and by funding from NASA, under award Nos. 80NSSC23 K1026 and 80NSSC24K0238.

ORCID iDs

Junzhi Wen https://orcid.org/0000-0002-9176-5273
Azim Ahmadzadeh https://orcid.org/0000-0002-1631-5336
Manolis K. Georgoulis https://orcid.org/0000-0001-6913-1330
Viacheslav M. Sadykov https://orcid.org/0000-0002-4001-1295
Rafal A. Angryk https://orcid.org/0000-0001-9598-8207

References

Aggarwal, C. C. 2017, Outlier Analysis (Cham: Springer), 1
Ahmadzadeh, A., Aydin, B., Georgoulis, M. K., et al. 2021, ApJS, 254, 23
Ahmadzadeh, A., Aydin, B., Kempton, D. J., et al. 2019a, in 2019 18th IEEE Int. Conf. on Machine Learning and Applications (ICMLA), ed. M. A. Wani et al. (Piscataway, NJ: IEEE), 1814
Ahmadzadeh, A., Hostetter, M., Aydin, B., et al. 2019b, in 2019 IEEE Int. Conf. on Big Data (Big Data) (Piscataway, NJ: IEEE), 1423
Aleskerov, E., Freisleben, B., & Rao, B. 1997, in Proc. IEEE/IAFE 1997 Computational Intelligence for Financial Engineering (CIFEr) (Piscataway, NJ: IEEE), 220
Angryk, R. A., Martens, P. C., Aydin, B., et al. 2020, NatSD, 7, 227
Aschwanden, M. J., Caspi, A., Cohen, C. M. S., et al. 2017, ApJ, 836, 17
Bloomfield, D. S., Higgins, P. A., McAteer, R. J., & Gallagher, P. T. 2012, ApJL, 747, L41
Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135
Bobra, M. G., & Ilonidis, S. 2016, ApJ, 821, 127
Boucheron, L. E., Al-Ghraibah, A., & McAteer, R. J. 2015, ApJ, 812, 51
Camporeale, E., Carè, A., & Borovsky, J. E. 2017, JGRA, 122, 10
Chandola, V., Banerjee, A., & Kumar, V. 2007, ACM Comput. Surv., 14, 15
Chen, Y., Kempton, D. J., Ahmadzadeh, A., et al. 2022, Neural Comput. Appl., 34, 13339
Chen, Y., Manchester, W. B., Hero, A. O., et al. 2019, SpWea, 17, 1404
Colak, T., & Qahwaji, R. 2009, SpWea, 7, S06001
Cuturi, M. 2011, in Proc. 28th Int. Conf. on Machine Learning (ICML-11), ed. L. Getoor & T. Scheffer (Madison, WI: Omnipress), 929
Ding, Z., & Fei, M. 2013, IFAC Proc. Vol., 46, 12
Escalante, H. J. 2005, Proc. Int. Conf. on Communications in Computing, 228, https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=cf069b7460ce1b5a0434a6a19f420544a780f35d
Florios, K., Kontogiannis, I., Park, S.-H., et al. 2018, SoPh, 293, 28
Gao, Y., Calhoun, V. D., & Miller, R. L. 2022, in 44th Annual Int. Conf. of the IEEE Engineering in Medicine & Biology Society (EMBC) (Piscataway, NJ: IEEE), 4645
Georgoulis, M. K., Bloomfield, D. S., Piana, M., et al. 2021, JSWSC, 11, 39
Georgoulis, M. K., Yardley, S. L., Guerra, J. A., et al. 2024, AdSpR, in press
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 (Big Data) (Piscataway, NJ: IEEE), 2543
Harris, C. R., Millman, K. J., Van Der Walt, S. J., et al. 2020, Natur, 585, 357
Huang, X., Zhang, L., Wang, H., & Li, L. 2013, A&A, 549, A127
Ji, A., Wen, J., Angryk, R., & Aydin, B. 2022, in 26th Int. Conf. on Pattern Recognition (ICPR) (Piscataway, NJ: IEEE), 2907
Jiao, Z., Sun, H., Wang, X., et al. 2020, SpWea, 18, e2020SW002440

Kullback, S., & Leibler, R. A. 1951, Ann. Math. Stat., 22, 79
Kumar, V. 2005, IEEE Distrib. Syst. Online, 6
Li, R., Cui, Y., He, H., & Wang, H. 2008, AdSpR, 42, 1469
Liu, C., Deng, N., Wang, J. T., & Wang, H. 2017, ApJ, 843, 104
Liu, F. T., Ting, K. M., & Zhou, Z.-H. 2008, in 2008 8th IEEE Int. Conf. on Data Mining (Piscataway, NJ: IEEE), 413
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
Ma, R., Ahmadzadeh, A., Boubrahimi, S. F., Georgoulis, M. K., & Angryk, R. A. 2019, in 2019 IEEE Int. Conf. on Big Data (Big Data) (Piscataway, NJ: IEEE), 4967
Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., & Ishii, M. 2018, ApJ, 858, 113
Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, ApJ, 835, 156
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, JMLR, 12, 2825
Sadaf, K., & Sultana, J. 2020, IEEEA, 8, 167059
Sadykov, V. M., & Kosovichev, A. G. 2017, ApJ, 849, 148
Sadykov, V. M., Kosovichev, A. G., Oria, V., & Nita, G. M. 2017, ApJS, 231, 6
Sun, Z., Bobra, M. G., Wang, X., et al. 2022, ApJ, 931, 163
Tavenard, R., Faouzi, J., Vandewiele, G., et al. 2020, JMLR, 21, 4686
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, NatMe, 17, 261
Wang, X., Chen, Y., Toth, G., et al. 2020, ApJ, 895, 3
Wen, J., & Angryk, R. A. 2024, in Artificial Intelligence and Soft Computing. ICAISC 2024, ed. L. Rutkowski et al. (Cham: Springer), 362
Yeolekar, A., Patel, S., Talla, S., et al. 2021, in 2021 Int. Conf. on Data Mining Workshops (ICDMW) (Piscataway, NJ: IEEE), 1067
Yuan, Y., Shih, F. Y., Jing, J., & Wang, H.-M. 2010, RAA, 10, 785