Feature Selection on a Flare Forecasting Testbed: A Comparative Study of 24 Methods

Atharv Yeolekar[‖‡*], Sagar Patel[‖‡*], Shreejaa Talla[‖‡*], Krishna Rukmini Puthucode[‖‡*],
Azim Ahmadzadeh[‖§], Viacheslav M. Sadykov[†§], and Rafal A. Angryk[‖§]
[‖]Department of Compute Science, Georgia State University, USA
[†]Physics & Astronomy Department, Georgia State University, USA
Email: [‡]{ayeolekar1,spatel389,stalla1,kputhucode1}@student.gsu.edu,
[§]{aahmadzadeh1,vsadykov,rangryk}@gsu.edu
[*]*These authors contributed equally to this work.*

***Abstract*—The Space-Weather ANalytics for Solar Flares (SWAN-SF) is a multivariate time series benchmark dataset recently created to serve the heliophysics community as a testbed for solar flare forecasting models. SWAN-SF contains 54 unique features, with 24 quantitative features computed from the photospheric magnetic field maps of active regions, describing their precedent flare activity. In this study, for the first time, we systematically attacked the problem of quantifying the relevance of these features to the ambitious task of flare forecasting. We implemented an end-to-end pipeline for preprocessing, feature selection, and evaluation phases. We incorporated 24 Feature Subset Selection (FSS) algorithms, including multivariate and univariate, supervised and unsupervised, wrappers and filters. We methodologically compared the results of different FSS algorithms, both on the multivariate time series and vectorized formats, and tested their correlation and reliability, to the extent possible, by using the selected features for flare forecasting on unseen data, in univariate and multivariate fashions. We concluded our investigation with a report of the best FSS methods in terms of their top-k features, and the analysis of the findings. We wish the reproducibility of our study and the availability of the data allow the future attempts be comparable with our findings and themselves.**

***Index Terms*—feature selection, feature ranking, multivariate time series, supervised, unsupervised**

I. Introduction

With rapid advances in machine learning and its unprecedented success across various domains, in recent years, interdisciplinary researchers have shown growing interest in using such tools and techniques for space-weather analytics and forecast. Extreme space-weather events are somewhat similar to extreme terrestrial events. The difference is that the main creating source of all space-weather events is the Sun. The solar flares triggered in the solar active regions can initiate Coronal Mass Ejections (CMEs) and enhance the flux of Solar Energetic Particles (SEP), both directly affecting the terrestrial environment.

Extreme space-weather events can have severe economic and collateral impact on us and our interconnected technologies here on Earth, as well as in space, due to the growing number of satellites [1]. This includes Earth climate, aviation and space radiation environment, electric power grid, GPS systems, HF radio communications, satellite communications, and everything that depends on these systems to function properly. In 2008, the direct economic impact of an extreme space-weather event was estimated (by the National Research Council) to be $\$1-2$ trillion for the United States, during the first year alone [2]. In realization of such a natural threat, several agencies have been directly studying or funding research in this area, to name a few in the United States alone, the National Science Foundation (NSF), the National Aeronautical and Space Administration (NASA), the National Oceanic and Atmospheric Administration (NOAA), the Department of Defense (DoD), and the Federal Aviation Administration (FAA).

Solar flares are one of such events. They are sudden brightness increases on the Sun visible in the wide wavelength range, from radio to gamma, representing the release of the free magnetic energy in the active region in the form of radiation, heat, kinetic energy of plasma, and accelerated particles. Since 1974, NOAA GOES satellites have been detecting X-ray flares and labeling them based on their peak 1-8 Å soft X-ray flux. From weakest to strongest flares are classified logarithmically as A (10^{-8} to $10^{-7}W/m^2$), B (10^{-7} to $10^{-6}W/m^2$), C (10^{-6} to $10^{-5}W/m^2$), M(10^{-5} to $10^{-4}W/m^2$), and X ($> 10^{-4}W/m^2$), meaning that an X-class flare is 10 times stronger than an M-class flare, and 100 times stronger than a C-class flare. Often in flare forecasting/prediction studies, the magnitude or the probability of occurrence of different classes of flares in an h-hour prediction window is of interest, e.g., [3]–[5].

The vast majority of the implemented approaches in forecasting the occurrence of a strong flare (typically, an M-class flare or larger) utilize properties computed from the photospheric vector or line-of-sight magnetic field maps of solar active regions [4]–[6]. The SWAN-SF data set [7] described in Section IV contains these characteristics carefully computed for May, 2010 — December, 2018 time period. Utilization of SWAN-SF for the considered problem of feature ranking in flare forecasting is appropriate and allows to mostly avoid an exhausting data preparation phase.

The main contribution of this paper is the initiation of a series of comparable feature selection strategies for the ambitious task of flare forecasting. Because the existing attempts were carried out on different datasets with different collection

strategies, imbalance ratios, and post-processing steps, a fair comparison between the discovered ranks is not possible. Using the flare forecasting benchmark dataset, SWAN-SF, we initiate this attempt that allows future feature selection, ranking, and extraction investigations to be comparable with our findings and with themselves. To help achieve this goal, we do our best to present a reproducible study by extensively explaining the details of data preparation, feature selection, and evaluation strategies and releasing our code [1] for further investigation and reusability. This highlights the importance and novelty of the current work.

This paper is structured as follows: Section II describes the work related to feature ranking, including the attempts for ranking in flare forecasting. Section III summarizes the feature ranking methods utilized in this work. The data and prediction metrics are described in Section IV. Sections V and VI highlight the feature ranking and evaluation methodology. The results of the investigation are described in Section VII and are followed by conclusions in Section VIII.

II. Related Work

Feature subset selection (FSS) methods are often categorized by their selection strategy from different perspectives. An FSS method is called a *filter* if it searches for an optimal subset based on the general patterns in data regardless of a particular learning model's performance. In contrast, *wrapper* methods utilize a greedy technique to search the space of all possible subset features by using a machine learning algorithm at its core [8]. When an FSS method combines these two strategies, it is then called *embedded* [9]. The FSS methods which rely on learners can be further divided into two sub-categories: supervised and unsupervised. The former takes advantage of the class labels of the data to find the most relevant features, while the latter group primarily employs clustering algorithms to do so [10]. From a different angle, an FSS method may consider multiple features at once to determine their collective relevance, in contrast to single-feature assessment. With this criterion, FSS methods can be classified into *univariate* or *multivariate* groups. Another category of distinction between the FSS methods is the *vectorized* and *MTS-based* algorithms. The former utilizes the descriptive statistics from the multivariate time series data as its input, whereas the latter employs the MTS data in its original format.

Supervised FSS methods employ various supervised machine learning algorithms (also called *estimators* or *base learners* in this context) to identify an optimal subset of the original features by considering the class labels and removing the redundant features (with a minimal or negative impact on the discrimination task) with the a priori knowledge. For example, Support Vector Channel Selection for brain-computer interface (BCI) dataset [11] combines Recursive Feature Elimination (RFE) [12] and Fisher Criterion (FC) [13] along with Support Vector Machines (SVMs). This supervised wrapper approach established a reliable and intuitive ranking method and quickly became the state-of-the-art FSS method for the proceeding traditional FSS methods. This method however, loses the correlation information as it utilizes a univariate selection. Understanding this shortcoming gave birth to a new family of wrapper methods that let classifiers process some statistical interpretation of data instead of the original data. Corona [14] is an example of such a family. It preserves the correlation information by employing correlation coefficients as features. Due to the general limits of supervised FSS methods [15], unsupervised methods are equally useful, or perhaps even more, given that most of the available data are unlabeled. Unsupervised FSS methods, in their simplest form, compute the similarity between the features and then reduce the redundancy in data by dropping the similar features. For example, CLeVer [16] selects an optimal subset of features by means of clustering on the loadings obtained by running PCA on the original data.

Univariate FSS methods, as mentioned before, take into account only one feature at a time to identify their relevance to the class label. The Relief algorithm [17] is a popular example of this group. Relief is a wrapper, supervised method that determines the relevance of a feature by examining each instance's values with respect to the values of its neighboring instances of the same class. Derived from Relief, a large family of FSS methods has been introduced that particularly attracted the attention in bioinformatics [18]. Equally popular, but in the multivariate category of FSS methods, is the Minimum-Redundancy Maximum-Relevance (mRMR) algorithm that was originally applied to microarray data [19]. The name is inspired by the core idea behind the algorithm, searching for a subset of features by maximizing their correlation with the class labels and minimizing their correlation with one another. Similar to the Relief algorithm, mRMR also inspired numerous FSS methods [20], especially in gene selection studies [21], even mixed with Relief [22].

Among the wrapper methods, the Support Vector Channel Selection algorithm (that we mentioned earlier) is perhaps the most popular one. This FSS method ranks features by training an SVM on the labeled data and assigning scores to features based on the concept of margin maximization [12], and was originally applied for the selection of electroencephalogram (EEG) channels. Among the filter FSS methods, the F-score [23], mutual information [24], and information gain [25] methods laid out the basic selection ideas that feed many complex filter methods. The Fast Correlation-Based Filter (FCBF) method [26] is another popular method in this category.

Although not widely used, the application of feature ranking methods for flare forecasting is definitely not a new idea. One of the most frequently used feature ranking techniques is the univariate F-score [4], [5], [27]. Another popular FSS methodology mentioned in the research papers employing a Random Forest algorithm [28] for flare forecasting are a selection of features based on Gini importance or the Mean Decrease Gini [29]. Other examples are the use of unsupervised feature extraction/selection from the magnetic Polarity Inversion Line

[1] https://bitbucket.org/gsudmlab/featuresubsetselection_on_swan-sf/

(PIL) masks based on kernel PCA [30], and the employment of CBF- and mRMR-like FSS methods [31]. Nevertheless, to the best of our knowledge, the effectiveness of a variety of other FSS methods on flare forecasting datasets was never examined. Our objective is to study the efficacy of a larger number of methods, but more importantly, to introduce a framework of comparability for future attempts.

It is worth noting that in the literature, some times the terms *selection*, *ranking*, and *extraction* are used loosely and even interchangeably. To avoid confusion, we clarify our understanding of these terms here before we move on to the next section. *Feature extraction* is specific to the algorithms which create (extract) new features by combining the original ones. The objective in this family of methods is to find a new feature space such that it has a lower dimension than the original space, and at the same time, allows a better separability of classes. PCA [32] is the typical example of a feature extraction method. Each new feature extracted by PCA is a linear combination of all the original features (possibly with close-to-zero coefficients for some of them). While this approach provides more flexibility as it is not limited to the original feature space, for problems where the importance of the existing features is of interest, it is not useful. This is where *feature selection* methods come into play. These methods aim at finding an optimal subset of the original features in such a way that the selected features carry most of the information about the distribution of data while excluding the redundant features, i.e., highly correlated or noisy features. The term *feature subset selection* (FSS) is sometimes used to emphasize this difference between selection and extraction. A sub-family of this group of methods additionally assign a score to each feature and find the optimal subset by means of thresholding. They are known as the *feature ranking* methods.

III. Background

A summary of the methods we employed in this study, in order to gain insight into the relevance of the SWAN-SF's features, is given in Table. I. In our list, there are filters and wrappers, supervised and unsupervised methods, univariate and multivariate approaches, and vectorized-based and Multivariate Time Series-based (MTS-based) strategies. In this section, we briefly review their implementations.

Recursive Feature Elimination (RFE) [12] is a feature selection method that fits data using a base learner such as Random Forest or Logistic Regression, and removes the weakest feature(s) recursively until the stipulated number of features is reached. Either the model's coefficients or the feature importance attributes are used to rank the features. Recursively eliminating features, RFE attempts to eliminate dependencies and collinearity in the model (if any).

Correlation as Features (Corona) [14] operates by computing the correlation coefficients matrix C for each MTS instance. The feature selection takes place on the upper triangular of C. Using RFE with SVM as the estimator, each feature is assigned a weight. The weights are then used to rank the features in ascending order to generate a rank vector RV. A symmetric matrix SM is generated by shaping RV as the lower and upper triangular matrix with diagonals as zero. To obtain ranks of original features, the column-wise summation is applied to SM.

Relief [17] gauges the quality of each feature by estimating the separability it provides for distinguishing the neighboring instances of different classes. The algorithm randomly selects an instance R and finds its nearest neighbors H from the same class (i.e., hit) and M from the opposite class (i.e., miss). The feature score for an arbitrary feature f is initialized with zero and updated by $W[f] := W[f] - \frac{d_f(R_i,H)}{m} + \frac{d_f(R_i,M)}{m}$, where d_f is the Manhattan distance between the feature f of two instances, and m is the longest distance in the space and is used for the normalization purpose. It can be observed that the feature score is (a) penalized for long distances between instances of similar class, and (b) incentivized for long distances between opposite class. This approach claims to have contextual awareness and performs well in problems with strong dependencies between the features.

Minimum Redundancy Maximum Relevance (mRMR) [19] finds an optimal subset of feature by iteratively adding features with maximum relevance with the class labels while having minimum redundancy with other already-selected features. The relevance is determined using ANOVA test with respect to the class labels, and the redundancy is determined by mean Pearson correlation in accordance with the selected features. For every i-th iteration, mRMR selects the feature having the highest score determined by $\text{score}_i(f) = \frac{F(f,\text{class})}{\sum_{s\in\mathcal{S}} |r(f,s)|/(i-1)}$. In this formula, f is an arbitrary feature, F is the F-statistic, r denotes the Pearson correlation, and $\mathcal{S}$ is the set of all features selected prior to f_i. This approach is known to be computationally efficient and robust to different downstream classification models [34].

Select k-Best [35] is a simple univariate feature selection approach that utilizes a scoring function passed as a parameter. Various statistical tests such as ANOVA test and Chi-Square test, and mutual information can be used to quantify the scoring function. The scoring function must return an array of scores corresponding to all features. Select k-Best then retains the top k features with the highest scores.

Select From Model [35] determines the feature importance based on the estimators' optimization weights, e.g., the *coefficient*s in Logistic Regression or *mean reduction* in the Gini index. The ones with scores lesser than a pre-set threshold parameter are excluded and considered unimportant. The top k features are those which obtained the highest scores.

CLeVer [16] is a family of unsupervised FSS methods for MTS data based on descriptive common principal component analysis (DCPC). DCPCs are obtained by bisecting the angles between their principal components after each MTS instance undergoes PCA. The correlation matrix of each MTS instance is passed as an input to obtain its correlation information. The CLeVer algorithm comprises three phases: (1) Computing the Principal Components for each MTS instance by applying Singular Value Decomposition on the correlation matrix; (2) Computing the DCPDs which depict the common direction of

	Abbreviation	FSS Algorithm	Estimator/Method	Supervised	Unsupervised	Multivariate	Univariate	Vectorized	MTS	Embedded	Wrapper	Filter
1	SFM_Logistic	Select From Model	Logistic Regression	✓		✓		✓		✓		
2	SFM_SVC		SVM	✓		✓		✓		✓		
3	SFM_RF		Random Forest	✓		✓		✓		✓		
4	SFM_ADA		Ada Boost	✓		✓		✓		✓		
5	SFM_Gb		Gradient Boosting	✓		✓		✓		✓		
6	SFM_BAG		Bagging Trees	✓		✓		✓		✓		
7	SFM_XT		Extremely Randomized Trees	✓		✓		✓		✓		
8	RFE_RF	Recursive Feature Elimination	Random Forest	✓		✓		✓			✓	
9	RFE_ADA		Ada Boost	✓		✓		✓			✓	
10	RGE_Gb		Gradient Boosting	✓		✓		✓			✓	
11	RFE_BAG		Bagging Trees	✓		✓		✓			✓	
12	RFE_XT		Extremely Randomized Trees	✓		✓		✓			✓	
13	RFE_SVC		SVM	✓		✓		✓			✓	
14	RFE_logistic		Logistic Regression	✓		✓		✓			✓	
15	SKB_MI	Select k-Best	Mutual Information	✓			✓	✓				✓
16	SKB_FVAL		F-statistic	✓			✓	✓				✓
17	SKB_CHI		Chi Square	✓			✓	✓				✓
18	mRMR	Min Redundancy Max Relevance	F-statistic / Pearson Correlation	✓		✓		✓				✓
19	Relief	Relief	F-statistic / Pearson Correlation	✓			✓	✓				✓
20	Corona	Correlation as Features	SVM	✓		✓		✓			✓	
21	Clever	CLeVer	k-means / PCA		✓	✓			✓	✓		
22	PIE	PIE	Normalized Mutual Information	✓		✓			✓			✓
23	CSFS	CSFS	Pairwise Mutual Information	✓		✓			✓			✓
24	FCBF	FCBF	Mutual Information	✓			✓		✓			✓

TABLE I
LIST OF ALL FSS METHODS USED IN THIS STUDY, WITH THEIR SPECIFICATIONS FOR REFERENCE

the maximum variance of all MTS instances; (3) Ranking of the features by their contribution to the DCPC model features. The last step is carried out by applying the ℓ^2-norm on the DCPCs (CLeVer-Rank), which is followed by running the k-means clustering algorithm to eliminate the redundant features (CLeVer-Cluster).

Fast Correlation-Based Filter (FCBF) [26], instead of a traditional linear correlation, employs Symmetric Uncertainty (SU), which involves entropy-based information gain (aka mutual information). To recap, the entropy of a random variable in information theory is the average level of uncertainty information inherent in the variable's possible outcomes. As a result, the amount by which a feature's entropy decreases reflects additional information about features provided by the label and is referred to as information gain. Since this information gain is biased in favor of features with more values, the data must be normalized beforehand. In FCBF, this normalization is accomplished by the Symmetrical Uncertainty, by first determining the probability distribution of each feature, and then its entropy corresponding to its class label. FCBF compiles a list of relevant features based on a given threshold and sorts them by their Symmetric Uncertainty values in descending order.

Power Iteration Embedding (PIE) [36] ranks the features to compute the similarity graph for all features (time series) across all time segments. Dynamic Time Warping (DTW) is used to calculate the distance between time series [37]. The largest eigenvector of the normalized adjacency matrix of the graph is computed to determine the cluster structure of these segments. Features' relevance score is evaluated using Normalized Mutual Information [38] between the eigenvector and the ground truth labels. Since the ranks of features with high correlation tend to be similar, to avoid redundancy, a subset selection algorithm is employed. The first step in the subset selection algorithm is to calculate the adjacency matrix of each sensor and then find a linear combination of these matrices that approximates the similarity matrix of the labels while using a small number of minimally redundant sensors.

Class Separability Feature Selection (CSFS) [39] relies on Mutual Information (MI) which measures both linear and non-linear correlations between the features. Since the pairwise MI matrix is symmetric, an MTS instance is vectorized to build new features using the upper triangular matrix. To rank these features, the ratio of the between-class scatter matrix to the within-class scatter matrix is considered, which is computed over the MI matrix. To avoid redundancy, the between-feature scatter matrix, which reflects the relation between features in terms of class labels, is produced. The larger the matrix value, the lesser the redundancy between its features.

IV. DATA AND METRICS

A. Data

The Space-Weather ANalytics for Solar Flares (SWAN-SF) benchmark dataset [7] comprises multivariate time series

data in five classes, namely X, M, C, B, and a non-flaring class denoted by FQ (flare quiet). The dataset is broken down into five temporally non-overlapping partitions such that each partition has the same number of X- and M-class flares. The various time series parameters of the dataset are derived from the solar photospheric magnetograms and NOAA's flare history. Magnetograms and their metadata are provided by the Solar Dynamics Observatory's HMI Active Region Patches (HARP) data product. To explore the details of collection, integration, and curation of the SWAN-SF benchmark dataset, as well as the definitions of the physical parameters we aim to rank in this study, we encourage the reader to see the corresponding publication. The dataset is also publicly available on the Harvard Dataverse repository [40].

B. Metrics

In this paper, the metrics utilized as performance measures are the True Skill Statistic (TSS) [41] and the updated Heidke Skill Score (HSS) [42] (sometimes denoted as HSS2). These are the two metrics that domain experts found appropriate for the task of flare forecasting to provide a meaningful evaluation of models given the scarcity of strong flares [43], [44]. TSS measures the difference between the true positive rate (detection) and the false positive rate (false alarm). In other words, TSS = $\frac{tp}{p} - \frac{fp}{n}$; where $p = tp + fn$ and $n = fp + tn$. It ranges from -1 to +1, with -1 indicating that the model makes all the wrong predictions/classifications, 0 implying that the model possesses no skill (random-guess), and +1 representing a model that assigns correct class labels to all the instances. TSS is not sensitive to class imbalance [4], [45]. This allows comparison of models which are trained on subsets of data with different imbalance ratios. HSS measures the performance of a model by comparing it to a random-guess model. It is formulated as $\frac{2((tp \cdot tn)-(fn \cdot fp))}{p(fn+tn)+n(tp+fp)}$. HSS2 ranges from -1 to +1, where 0 intimates no difference between the model's performance and random-guess. The magnitude of the negative value is directly proportional to the number of misclassified samples. The positive values quantify how much the model performs better than the random-guess model.

V. Feature Selection Methodology

The list of all FSS methods we utilize in this study, along with their specifications, is given in Table I. We set up two feature selection pipelines, one for vectorized-based FSS methods and the other for MTS-based methods. In this section, we provide a general overview of the pre-processing steps, as well as the strategies adopted in our feature ranking methodology. Fig. 1 is provided to aid in the following of these steps.

A. Data Preparation

The first 24 physical parameters of SWAN-SF are quantitative features describing the magnetic fields in active regions, and we limit our study only to them. To prepare our data for both the ranking and evaluation, first, we impute the missing values, which account for $\leq 0.01\%$ of the entire data. We linearly interpolate the missing values using their neighboring values. For the boundary values, we use backward and forward fill, i.e., propagating the last valid values forward or backward.

As majority of the FSS methods we utilize are designed for tabular data (as opposed to MTS data), we need to create a vectorized version of SWAN-SF as well. To do so, we compute 7 descriptive statistics for each physical parameter, yielding 168 statistical features. These statistics are *min*, *max*, *median*, *standard deviation*, *skewness*, *kurtosis*, and *last value* which is simply the last value of each time series. These basic statistics have been used before in several studies working on SWAN-SF to simplify the problem [45]–[47].

As mentioned in Section IV, SWAN-SF is a collection of 5 different flare types. For simplification of the problem, we group the stronger flares (X- and M-class flares) and weaker flares (C- and B-class flares, plus FQ class) and create a dichotomized dataset.

B. Rank Aggregation for Vectorized Data

Since the features in the vectorized data are the descriptive statistics and not the original SWAN-SF features, in order to be able to transfer the obtained ranks back to the original feature space, we assign each feature the average scores obtained across all 7 statistics corresponding to that features. This way, we obtain a single score per (original) feature. We use this score, similar to the scores MTS-based FSS methods return, to rank the features.

For the preparation of data, extraction of statistical features, and model fitting we used the pandas and scikit-learn [35], MVTS Data Toolkit [48], and tslearn [49] packages, respectively.

VI. Evaluation Methodology

A common way to validate the selected features is through domain knowledge and expertise. However, such knowledge does not always exist, and more often than not, it happens that experts have contrasting opinions. In this section, we investigate the reliability of the obtained rankings from three different angles; using an independently trained and tested classifier in a univariate and multivariate fashion, and by looking at the correlations of the rankings across all the 24 FSS methods we employed.

A. Pre-Evaluation: Sampling and Balancing

A dataset is class imbalanced when the frequency of samples of one or more data classes are significantly smaller than those of the majority classes. This is known as the class-imbalance issue [50] and results in superficial performance if not treated properly. The SWAN-SF benchmark dataset [7] exhibits a severe case of class imbalance across all the partitions. To remedy this issue, we employ the *climatology-preserving undersampling* method as suggested in [45]. This is a stratified undersampling (of all 5 flare classes) where the obtained samples constitute a 1:1 ratio between the strong and weak flares, i.e., the total number of X and M instances is the same as the total number of C, B, and FQ instances. Using

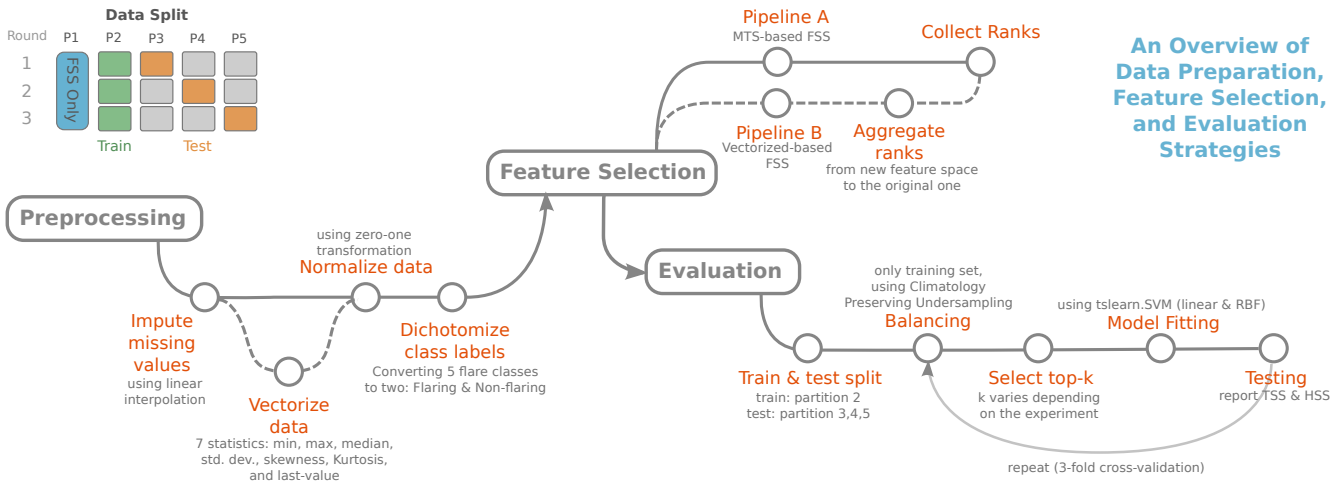

Fig. 1. The schematic diagram illustrating the 3 main components of our study: data preprocessing, feature selection, and evaluation. Each part is explained in details in their corresponding sections. The dashed lines indicate the process that is only applicable to vectorized data. The usage of SWAN-SF's 5 partitions is also shown on the top-left area.

this strategy, the subclass ratios will be preserved, hence the name. Note that this strategy is only used for the evaluation phase and only during the model training, and not for testing, i.e., the test set is not balanced. Also, the FSS methods had access to the entire Partition 1 and not subsets of it.

Training and testing on the same partition of SWAN-SF results in unreliable performance because of the auto-correlation of the temporally neighboring data points within the partition, and hence it breaks the underlying assumption that random variables need to be independent for models. [45] studied this notion on SWAN-SF and defined it as *temporal coherence* of data. Following their suggestion, as shown in Fig. 1, we avoid mixing partitions for different purposes. Partition 1 is used only to compute the FSS ranks; Partition 2 is reserved for training SVM on the ranked features; a 3-fold cross-validation, each fold being repeated 5 times to generate different undersampled versions of the training partition. The uncertainties are thus computed using the 15 scores. Partitions 3, 4, and 5 are assigned to testing the efficacy of the obtained ranked using SVM.

B. Univariate Ranking Evaluation

Our univariate evaluation pipeline is designed to assess the quality of ranking of features individually, regardless of the ranking methodology (univariate or multivariate). This pipeline functions as follows: it create a subset of SWAN-SF, that contains only one feature and the class labels. It uses Partition 2 of this subset as the training set, and Partitions 3, 4, and 5 as the test set. Note that for obtaining the ranked features, only Partition 1 was used. Next, as shown in Fig. 1 (the evaluation branch), it carries out a 3-fold cross-validation (repeating each trial 5 times) during which we train SVM on subsets of Partition 2 and test it on the test set. This subset is obtained at each iteration, using the climatology-preserving undersampling method discussed in Section VI-A. The model performance is reported in terms of the average TSS and HSS, with the variability being reported by their standard deviation. The pipeline repeats this procedure for the remaining features and reports the results similarly.

C. Multivariate Ranking Evaluation

Our multivariate evaluation methodology is implemented to examine the collective contribution of the selected features, regardless of the ranking methodology (univariate or multivariate). The pipeline is very similar to that of the univariate, with one difference that instead of a single feature, a collection of the features is examined at each iteration, starting with one single feature (the best one according to a given FSS method), and iteratively appending the next best feature (according to the same FSS method) to the collection. Since there are 2^{24} possible subsets, unlike the univariate pipeline, this has to be repeated for each FSS method's returned ranks separately.

D. Cross Comparison of Ranked Features

To study the degree of which all the employed FSS methods agree with each other's rankings, we compare all returned rankings and measure their Pearson correlation coefficient. Higher positive correlations indicate that the features are similarly ranked by the compared methods, and conversely, the higher negative correlations indicate the opposite order of relevance. Values closer to zero should be interpreted as no (linear) correlation. While some degree of positive correlation is expected, it is important to note that such a comparative analysis on its own is not indicative of the efficacy of the obtained rankings.

Note that, although we have two groups of FSS methods, namely vectorized-based and MTS-based, in our evaluation pipeline, we only use MTS data (without vectorization). This decision is made for two reasons: (1) to guarantee the comparability of the algorithms and their rankings, and (2) to test rankings' efficacy on the raw time series, since the statistics chosen for vectorization of the MTS data can change from one study to another. Also, for the sake of completeness, we run all of our training experiments once with the *rbf* kernel and

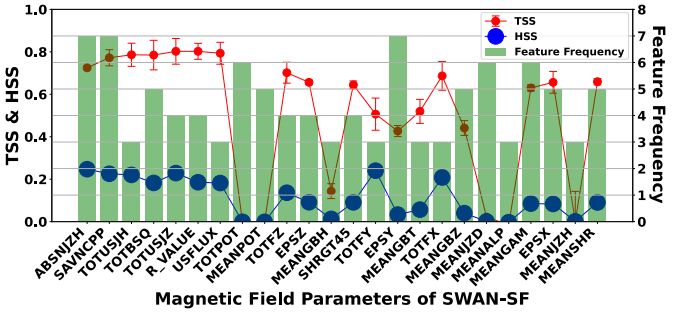

Fig. 2. Univariate comparison of the ranked features: On the x-axis the 24 SWAN-SF features are listed and sorted by their aggregated ranks across all FSS methods. The green bars represent features' frequency of occurrence, and their position shows their overall ranks across all 24 FSS methods. TSS and HSS are reported as models' performance.

then with the *linear* kernel. We do not use *gak* kernel however, because (1) we did not observe any significant improvement (compared to the other two) in our studies on subsets of the data, despite the theoretical support that the *gak* might be more appropriate for the high-dimensional data such as ours. Also, (2) it is extremely resource hungry due to its reliance on the optimization needed for DTW. We suspect that the ineffectiveness of the *gak* kernel on SWAN-SF might be rooted in the noisiness of the data. Smoothing the SWAN-SF's time series requires a rigorous and comprehensive analysis that we consider out of the scope of this work.

VII. Results

In this section, we analyze the output of the pipelines illustrated in Fig. 1, through multiple lenses. For description of the SWAN-SF feature names and their physical meaning, please see Table 1 in [7].

The result of our univariate evaluation is illustrated in Fig. 2. The bar at the i-th position represents how often the corresponding feature was ranked as the i-th feature. For example, *TOTBSQ* is the 4th most relevant feature as it was overall ranked 5 times (highest) as the 4th feature. While there seems to be a decreasing trend in terms of TSS and HSS, the high fluctuations prevent us from drawing any conclusions in terms of the reliability of the ranking. The first seven features (*ABSNJZH*, *SAVNCPP*, *TOTUSJH*, *TOTBSQ*, *TOTUSJZ*, *R-VALUE*, and *USFLUX*) correspond to the highest individual TSS values, with HSS values being among the highest. On the other hand, even features from the bottom of the list, e.g., *MEANSHR* and *EPSX*, achieve very high TSS values, although accompanied by low HSS values. We know that a good flare forecast performance corresponds to comparable TSS and HSS values [45]. Also, it is interesting to note that features like *TOTPOT* and *MEANPOT* which correspond to near-zero TSS and HSS, are ranked individually higher than other features such as *TOTFX* which corresponds to a much higher TSS ($\sim$ 0.7) value. This suggests that while some features (e.g., *TOTPOT*) may not be found relevant individually, when combined with others they might help better discriminate the strong and weak flares. This is why multivariate selection strategies are often preferred over the

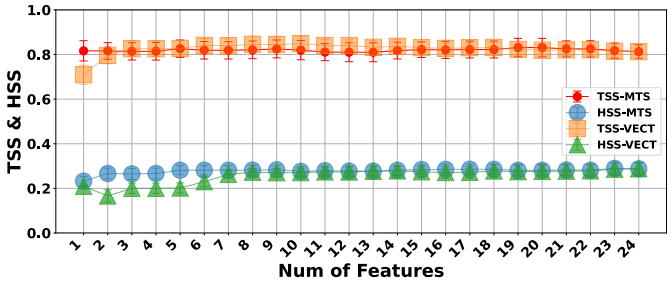

Fig. 3. Multivariate comparison of the ranked features: The x-axis represents k in the top-k features of SWAN-SF ranked by the mean rank of different MTS- and vectorized-based methods. The y-axis represents the mean TSS and mean HSS across all 24 methods. The *linear* kernel is used for this set of experiments.

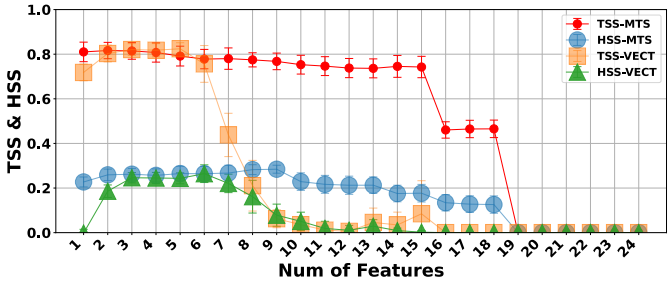

Fig. 4. Multivariate comparison of the ranked features: This is similar to Fig. 3, except that the *rbf* kernel is used (instead of *linear*) for this set of experiments.

univariate ones, given their limitations. Interestingly, the top-10 SWAN-SF features according to their aggregated ranks across all FSS methods correspond to the top-11 features for flare prediction mentioned by [4] according to the F-score (the authors also had the *AREA_ACR* within the top-11 features which is not a part of SWAN-SF).

The results of our multivariate evaluation are illustrated in Figs. 3 and 4 for SVM with the *linear* and *rbf* kernels, respectively. Looking at Fig. 3, it is evident that utilizing the *linear* kernel results in consistent performance in terms of TSS and HSS, with any k in the top-k features. The insignificant fluctuations and the non-descending pattern in TSS and HSS demonstrate that the *linear* kernel does not distinguish between different features' effectiveness, whether using the vectorized-based nor MTS-based FSS methods. Using the *rbf* kernel (Fig. 4), on the other hand, the performances are very distinct. The performance of the vectorized-based FSS methods drops drastically by adding the 7th feature. MTS-based methods, however, demonstrate higher tolerance for a larger number of top features. This observation might be of interest from the domain experts' point of view, as it allows a larger number of features to be picked for further investigations. An observation can also be made in favor of the vectorized-based methods; using those methods, with very few features (3 to 5) one can get similar or marginally better performance than using MTS-based methods with any (large or small) combination of the features. This is particularly important for operational forecast modules as they must take into account the factor of computation time. A model can perform faster on data with

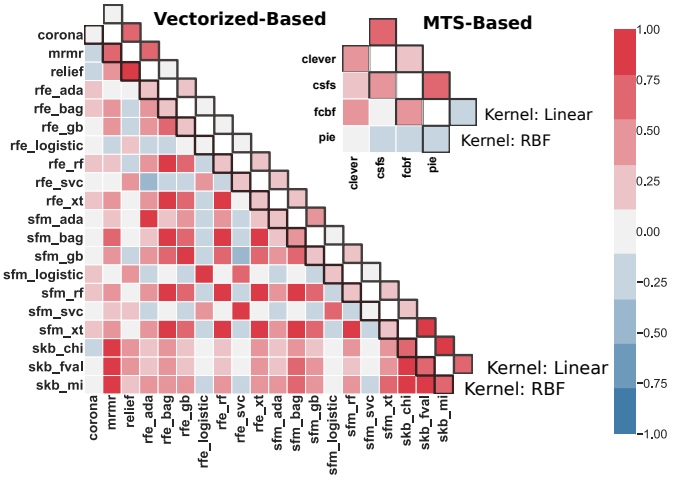

Fig. 5. Cross comparison of the 24 FSS methods: each cell of the heatmaps (except those on the diagonal) represents the Pearson correlation of two rankings. The diagonal cells compares a ranking with that of SVM (univariate) using a *linear* or *rbf* kernel. The rankings of vectorized-based and MTS-based FSS methods are separated for a better visibility.

lower dimensionality.

The result of our cross comparison evaluation, is illustrated through the heatmaps in Fig. 5. Looking at the heatmaps, it is not difficult to see that the distribution of the correlations is negatively skewed (i.e., in favor of the positive values). Using the Kolmogorov-Smirnov test [51], we assess whether the distribution of correlations is different from the uniform noise or not. With the test statistic of ~ 0.48 associated with the very small p-value of $4.1\text{e}{-122}$, we confidently reject the null hypothesis that the two distributions are similar. Therefore, the FSS methods, with the overall average correlation (using *rbf*) of ~ 0.60 for vectorized-based and ~ 0.39 for MTS-based methods (non-diagonal cells), agree with each other much more than they disagree. On the degree of this agreement, the theoretical expectation might be to have a near-perfect average (linear) correlation.

In a closer look, for vectorized-based FSS methods (the larger heatmap) a strong correlation ($\sim 90\%$) is observed between methods with the same estimator irrespective of their different selection strategies. For example, *rfe_logistic* and *sfm_logistic* both use Logistic Regression as their estimator, but the former uses Recursive Feature Elimination (RFE) as its selection strategy while the latter uses Select From Model (SFM). While the computational complexity of RFE is significantly higher than that of SFM, on the SWAN-SF dataset they return similar feature rankings. This observation is not unique to those two methods only. A good correlation (60%-80%) between strategies having tree-based estimators is also observed, e.g., Random Forest (RF), Extra Trees (XT), Gradient Boosting (GB), Ada Boosting (ADA), and Bagging Trees (bag). We believe the main reason behind this similarity is their similar heuristics, and this should not be confused with the reliability of their results.

For MTS-Based methods none of the utilized FSS methods demonstrate strong correlation within themselves. This can also be attributed to the inherent differences in the algorithmic approaches across all the MTS-based methods. That said, such FSS methods might be good candidates for making an ensemble feature selection model. This, however, is just a hypothesis yet to be investigated.

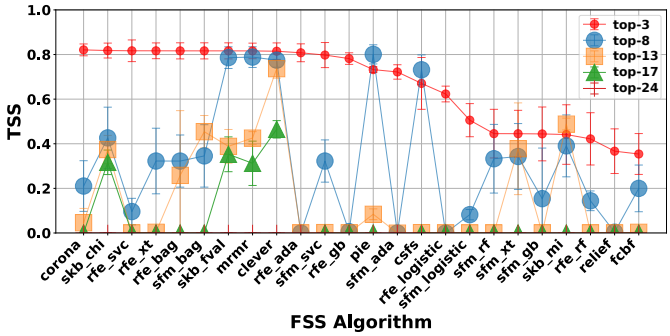

Fig. 6. Performance of the top-k features per FSS method, reported in terms of TSS, using SVM with *rbf* kernel. The x-axis represents the 24 FSS methods and their corresponding top-k features based on TSS score. The FSS methods on the x-axis are sorted by the TSS report of their top-3 features.

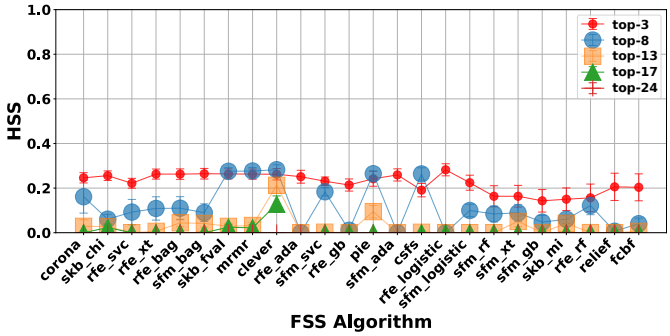

Fig. 7. Performance of the top-k features per FSS method, reported in terms of HSS using SVM with *rbf* kernel. The x-axis represents the 24 FSS methods and their corresponding top-k features based on HSS score. The FSS methods on the x-axis are sorted by the TSS (not HSS) report of their top-3 features.

Looking at the diagonals of the heatmaps, the average correlations with the SVM's rankings (using *rbf*) are 0.31 and 0.30 for the vectorized-based and MTS-based methods, respectively. Interestingly, most of the FSS methods whose rankings are much more similar to those obtained by SVM are in fact filter methods, such as *mRMR*, Relief, Select k-Best (with different estimators), and *FBCF*. CLeVer which is an embedded FSS method also correlates very strongly with the SVM's ranking.

Another way to assess the collective relevance of the features is by looking at the performance SVM can achieve using the top-k features. As illustrated in Figs. 6 and 7, most of the methods' top-3 features (first 11 ones) perform relatively well, in terms of TSS and HSS. But when more features are used together, i.e., top-17, most of the RFE-based methods (e.g., *rfe_ada*, *rfe_gb*, *rfe_logistic*) perform significantly worse. In contrast, (yet again) the filter methods seem to have selected their large set of features very well, as manifested by the TSS and HSS obtained by the top-17 features selected by *skb_chi*, *skb_fval*, and *mRMR*. The top-17 features of the CLeVer method achieve the highest TSS and HSS compared to

other methods in this group. CLeVer utilizes an unsupervised estimator (k-means) to drop the redundant features, which might be the reason for its selection to maintain such a high performance. The outperforming FSS methods with their top-8 features are also mainly filters, e.g., *skb_fval*, *mRMR*, *CSFS*, and *PIE*.

It is also interesting to note that although previously Corona did not show a high correlation with any of the other vectorized-based FSS methods (see Fig. 5), neither had it any correlation with the rankings obtained by SVM, its top-3 selection outperforms (in terms of TSS) all other top-k features of all FSS methods. This does not hold true however, when HSS is used for evaluation.

VIII. Conclusion

In summary, our FSS investigation on SWAN-SF dataset brings us to the following conclusions:

- Although different FSS methods rank SWAN-SF's features differently, we statistically showed that overall these methods agree with each other's rankings much more than they disagree. To avoid presenting an unreliable ranking of the features, we presented the FSS methods whose top-k features performed better than others (see Figs. 6 and 7).
- The high correlations between FSS methods' returned rankings are often rooted in the similarities in their strategies and not necessarily the methods' confidence in their rankings. These similarities can be looked for in (1) the estimators (for wrapper and embedded methods), the statistical inference (for filter methods), and the embedded selection algorithms.
- In the absence of domain knowledge, the reliability of the ranked features cannot be trusted on its own, and it should be further tested by (1) cross-verification with the rankings obtained by other methods made up of non-similar components; (2) being verified by the same verification metrics that will be used for evaluation of the main problem. In this study, we used SVM as our control classifier, and TSS and HSS to keep track of its performance.
- Ambiguities in correlations between different FSS methods indicate that the performance of a solar flare forecast model (trained on SWAN-SF) may depend on the choice of the FSS method. Therefore, further investigation on a more appropriate FSS method is warranted.
- We observed that the top-k features returned by FSS methods may perform differently, depending on the size of k. Those which work better with smaller values of k are useful particularly for situations where the time and space complexity matters, whereas others can find a larger number of features without adding too much of redundancy. These are often more useful when the objective is to obtain domain insight into the features and their importance.
- All FSS methods (except Clever) utilized in this study are supervised methods, partially because there is a large emphasis on this group of methods in the literature. It was interesting, however, to observer that Clever outperformed most of the supervised methods. This hints at the importance of such methods and invites further investigation in that direction on SWAN-SF features.

Acknowledgment

The authors would like to thank Heramb C. Lonkar and Egill Gunnarsson for kindly sharing their implementations of two FSS methods with us.

This work was supported in part by two NASA Grant Awards [No. NNH14ZDA001N, 80NSSC20K1352], and two NSF Grant Awards [No. 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. VMS acknowledges the NSF FDSS grant 1936361 and NASA grant 80NSSC20K0302.

References

[1] J. Eastwood, E. Biffis, M. Hapgood, L. Green, M. Bisi, R. Bentley, R. Wicks, L.-A. McKinnell, M. Gibbs, and C. Burnett, "The economic impact of space weather: Where do we stand?" *Risk Analysis*, vol. 37, no. 2, pp. 206–218, 2017. [Online]. Available: https://doi.org/10.1111/risa.12765

[2] N. R. Council, *Severe Space Weather Events–Understanding Societal and Economic Impacts: A Workshop Report*. Washington, DC: The National Academies Press, 2008. [Online]. Available: https://doi.org/10.17226/12507

[3] G. Barnes, K. D. Leka, C. J. Schrijver, T. Colak, R. Qahwaji, O. W. Ashamari, Y. Yuan, J. Zhang, R. T. J. McAteer, D. S. Bloomfield, P. A. Higgins, P. T. Gallagher, D. A. Falconer, M. K. Georgoulis, M. S. Wheatland, C. Balch, T. Dunn, and E. L. Wagner, "A comparison of flare forecasting methods. i. results from the "all-clear" workshop," *apj*, vol. 829, no. 2, p. 89, Oct. 2016.

[4] M. G. Bobra and S. Couvidat, "Solar flare prediction using sdo/hmi vector magnetic field data with a machine-learning algorithm," *The Astrophysical Journal*, vol. 798, no. 2, p. 135, 2015. [Online]. Available: doi.org/10.1088/0004-637X/798/2/135

[5] V. M. Sadykov and A. G. Kosovichev, "Relationships between Characteristics of the Line-of-sight Magnetic Field and Solar Flare Forecasts," *The Astrophysical Journal*, vol. 849, no. 2, p. 148, Nov. 2017.

[6] N. Nishizuka, K. Sugiura, Y. Kubo, M. Den, S. Watari, and M. Ishii, "Solar Flare Prediction Model with Three Machine-learning Algorithms using Ultraviolet Brightening and Vector Magnetograms," *The Astrophysical Journal*, vol. 835, no. 2, p. 156, Feb. 2017.

[7] R. A. Angryk, P. C. Martens, B. Aydin, D. Kempton, S. S. Mahajan, S. Basodi, A. Ahmadzadeh, S. F. Boubrahimi, S. M. Hamdi, M. A. Schuh *et al.*, "Multivariate time series dataset for space weather data analytics," *Sci. Data*, 2019. [Online]. Available: https://doi.org/10.1038/s41597-020-0548-x

[8] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1-2, pp. 273–324, 1997. [Online]. Available: https://doi.org/10.1016/S0004-3702(97)00043-X

[9] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[10] J. Miao and L. Niu, "A survey on feature selection," *Procedia Computer Science*, vol. 91, pp. 919–926, 2016.

[11] T. N. Lal, M. Schröder, T. Hinterberger, J. Weston, M. Bogdan, N. Birbaumer, and B. Schölkopf, "Support vector channel selection in BCI," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1003–1010, 2004. [Online]. Available: https://doi.org/10.1109/TBME.2004.827827

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1, pp. 389–422, 2002.

[13] C. Bishop, "Neural networks for pattern recognition," 1995.

[14] K. Yang, H. Yoon, and C. Shahabi, "A supervised feature subset selection technique for multivariate time series," in *Proceedings of the workshop on feature selection for data mining: Interfacing machine learning with statistics*, 2005, pp. 92–101.

[15] P. Smialowski, D. Frishman, and S. Kramer, "Pitfalls of supervised feature selection," *Bioinform.*, vol. 26, no. 3, pp. 440–443, 2010. [Online]. Available: https://doi.org/10.1093/bioinformatics/btp621

[16] H. Yoon, K. Yang, and C. Shahabi, "Feature subset selection and feature ranking for multivariate time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1186–1198, 2005.

[17] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine learning*, vol. 53, no. 1, pp. 23–69, 2003.

[18] R. J. Urbanowicz, M. Meeker, W. G. L. Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Informatics*, vol. 85, pp. 189–203, 2018. [Online]. Available: https://doi.org/10.1016/j.jbi.2018.07.014

[19] H. Peng, F. Long, and C. H. Q. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005. [Online]. Available: https://doi.org/10.1109/TPAMI.2005.159

[20] S. Ramírez-Gallego, I. Lastra, D. Martínez-Rego, V. Bolón-Canedo, J. M. Benítez, F. Herrera, and A. Alonso-Betanzos, "Fast-mrmr: Fast minimum redundancy maximum relevance algorithm for high-dimensional big data," *Int. J. Intell. Syst.*, vol. 32, no. 2, pp. 134–152, 2017. [Online]. Available: https://doi.org/10.1002/int.21833

[21] A. E. Akadi, A. Amine, A. E. Ouardighi, and D. Aboutajdine, "A two-stage gene selection scheme utilizing MRMR filter and GA wrapper," *Knowl. Inf. Syst.*, vol. 26, no. 3, pp. 487–500, 2011. [Online]. Available: https://doi.org/10.1007/s10115-010-0288-x

[22] Y. Zhang, C. H. Q. Ding, and T. Li, "A two-stage gene selection algorithm by combining relieff and mrmr," in *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering, BIBE 2007, October 14-17, 2007, Harvard Medical School, Boston, MA, USA*. IEEE Computer Society, 2007, pp. 164–171. [Online]. Available: https://doi.org/10.1109/BIBE.2007.4375560

[23] S. Ding, "Feature selection based f-score and aco algorithm in support vector machine," in *2009 Second International Symposium on Knowledge Acquisition and Modeling*, vol. 1, 2009, pp. 19–23.

[24] L. T. Vinh, S. Lee, Y. Park, and B. J. d'Auriol, "A novel feature selection method based on normalized mutual information," *Appl. Intell.*, vol. 37, no. 1, pp. 100–120, 2012. [Online]. Available: https://doi.org/10.1007/s10489-011-0315-y

[25] B. Azhagusundari, A. S. Thanamani *et al.*, "Feature selection based on information gain," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 2, no. 2, pp. 18–21, 2013.

[26] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, T. Fawcett and N. Mishra, Eds. AAAI Press, 2003, pp. 856–863. [Online]. Available: http://www.aaai.org/Library/ICML/2003/icml03-111.php

[27] T. Cinto, A. L. S. Gradvohl, G. P. Coelho, and A. E. A. da Silva, "Solar Flare Forecasting Using Time Series and Extreme Gradient Boosting Ensembles," *Solar Physics*, vol. 295, no. 7, p. 93, Jul. 2020.

[28] L. Breiman, "Random Forests." *Machine Learning*, vol. 45, pp. 5–32, Jan. 2001.

[29] C. Liu, N. Deng, J. T. L. Wang, and H. Wang, "Predicting Solar Flares Using SDO/HMI Vector Magnetic Data Products and the Random Forest Algorithm," *The Astrophysical Journal*, vol. 843, no. 2, p. 104, Jul. 2017.

[30] J. Wang, Y. Zhang, S. A. Hess Webber, S. Liu, X. Meng, and T. Wang, "Solar Flare Predictive Features Derived from Polarity Inversion Line Masks in Active Regions Using an Unsupervised Machine Learning Algorithm," *The Astrophysical Journal*, vol. 892, no. 2, p. 140, Apr. 2020.

[31] O. W. Ahmed, R. Qahwaji, T. Colak, P. A. Higgins, P. T. Gallagher, and D. S. Bloomfield, "Solar Flare Prediction Using Advanced Feature Extraction, Machine Learning, and Feature Selection," *Solar Physics*, vol. 283, no. 1, pp. 157–175, Mar. 2013.

[32] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, vol. 2, no. 11, pp. 559–572, 1901.

[33] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[34] Z. Zhao, R. Anand, and M. Wang, "Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform," in *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2019, pp. 442–452.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[36] S. Han and A. Niculescu-Mizil, "Supervised feature subset selection and feature ranking for multivariate time series without feature extraction," *CoRR*, vol. abs/2005.00259, 2020. [Online]. Available: https://arxiv.org/abs/2005.00259

[37] J. Serra and J. L. Arcos, "An empirical evaluation of similarity measures for time series classification," *Knowledge-Based Systems*, vol. 67, pp. 305–314, 2014.

[38] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[39] M. Han and X. Liu, "Feature selection techniques with class separability for multivariate time series," *Neurocomputing*, vol. 110, pp. 29–34, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231212009009

[40] R. Angryk, P. Martens, B. Aydin, D. Kempton, S. Mahajan, S. Basodi, A. Ahmadzadeh, X. Cai, S. Filali Boubrahimi, S. M. Hamdi, M. Schuh, and M. Georgoulis, "SWAN-SF," 2020. [Online]. Available: https://doi.org/10.7910/DVN/EBCFKM

[41] A. Hanssen and W. Kuipers, *On the Relationship Between the Frequency of Rain and Various Meteorological Parameters (with Reference to the Problem of Objective Forecasting)*. Koninklijk Nederlands Meteorologisch Instituut, 1965, vol. 81.

[42] C. C. Balch, "Updated verification of the Space Weather Prediction Center's solar energetic particle prediction model," *Space Weather*, vol. 6, no. 1, p. S01001, Jan. 2008. [Online]. Available: doi.org/10.1029/2007SW000337

[43] G. Barnes and K. Leka, "Evaluating the performance of solar flare forecasting methods," *apjl*, vol. 688, no. 2, p. L107, 2008. [Online]. Available: https://doi.org/10.1086/595550

[44] D. S. Bloomfield, P. A. Higgins, R. T. J. McAteer, and P. T. Gallagher, "Toward Reliable Benchmarking of Solar Flare Forecasting Methods," *apjl*, vol. 747, no. 2, p. L41, Mar. 2012. [Online]. Available: https://iopscience.iop.org/article/10.1088/2041-8205/747/2/L41

[45] A. Ahmadzadeh, B. Aydin, M. K. Georgoulis, D. J. Kempton, S. S. Mahajan, and R. A. Angryk, "How to train your flare prediction model: Revisiting robust sampling of rare events," *The Astrophysical Journal Supplement Series*, vol. 254, no. 2, p. 23, May 2021. [Online]. Available: http://dx.doi.org/10.3847/1538-4365/abec88

[46] M. Hostetter, A. Ahmadzadeh, B. Aydin, M. K. Georgoulis, D. J. Kempton, and R. A. Angryk, "Understanding the impact of statistical time series features for flare prediction analysis," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 4960–4966.

[47] A. Ahmadzadeh, B. Aydin, D. J. Kempton, M. Hostetter, R. A. Angryk, M. K. Georgoulis, and S. S. Mahajan, "Rare-event time series prediction: A case study of solar flare forecasting," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1814–1820.

[48] A. Ahmadzadeh, K. Sinha, B. Aydin, and R. A. Angryk, "Mvts-data toolkit: A python package for preprocessing multivariate time series data," *SoftwareX*, vol. 12, p. 100518, 2020. [Online]. Available: https://doi.org/10.1016/j.softx.2020.100518

[49] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, and E. Woods, "Tslearn, A machine learning toolkit for time series data," *J. Mach. Learn. Res.*, vol. 21, pp. 118:1–118:6, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-091.html

[50] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997), Nashville, Tennessee, USA, July 8-12, 1997*, D. H. Fisher, Ed. Morgan Kaufmann, 1997, pp. 179–186.

[51] A. Kolmogorov-Smirnov, A. Kolmogorov, and M. Kolmogorov, “Sulla determinazione empírica di uma legge di distribuzione,” 1933.