Draft version May 29, 2022
Typeset using LaTeX **modern** style in AASTeX631

Machine Learning for Space Weather Forecasting

Ekaterina S. Ivshina[1]

[1]*Department of Mathematics, Princeton University, Princeton, NJ 08544, USA*

ABSTRACT

We study the problem of forecasting B_z magnetic field component from upstream in situ observations of solar coronal mass ejections. The ability to forecast B_z is key to predicting intense geoeffective events on Earth, and existing forecasting tools do not predict geomagnetic storms accurately. We propose a feature extraction procedure based on persistent homology and use machine learning algorithms to forecast $\min(B_z)$ value in the magnetic obstacle. This work is a step towards building an operational tool for reliable forecasts of geomagnetic storms on Earth.

1. INTRODUCTION

The space between the Sun and the Earth is filled with solar wind – streams of plasma that travel nearly radially out from the Sun. A Coronal Mass Ejection (CME) is a significant release of plasma and magnetic field from the Sun into interplanetary space, and an Interplanetary Coronal Mass Ejection (ICME) is the interval of the resulting disturbed solar wind conditions (Rouillard 2011). The capacity of ICMEs to produce extreme geomagnetic storms depends on their internal plasma structure and their B_z magnetic field (Reiss et al. 2021). Extended periods of large southward B_z, which points opposite to the Earth's magnetic field, tend to cause intense geoeffective events (Gonzalez & Tsurutani 1987): the amount of energy and momentum transferred from the solar wind into the Earth's magnetosphere during magnetic reconnection at the dayside magnetopause depends primarily on the B_z component of the interplanetary magnetic field (Dungey 1961). The largest southward B_z disturbances are associated with coronal mass ejections (CMEs). Hence, predicting the B_z magnetic field embedded within interplanetary coronal mass ejections (ICMEs), also known as the B_z problem, is a key challenge in space weather forecasting. If a CME hits Earth, it could damage power grids, GPS satellites, and other infrastructure. Thus, having the ability to forecast geomagnetic storms is important for preventing the possible damage of the electrical and communication systems on Earth. Nowadays, we can not predict the B_z magnetic field component – a proxy for ICME geoeffectiveness – with sufficient warning time before the ICME arrival at Earth.

Currently, there are several observational limitations to forecasting B_z. Available observational data do not allow to accurately deduce the magnetic properties of CMEs such as magnetic field magnitude, topology, and helicity at the time of formation. We also do not have observational tools for tracking the evolution of a CME, in particular its rotation, compression, deflection, and reconnection in interplanetary space with the ambient solar wind. Moreover, it is challenging to predict the magnetic structure of a CME due to our limited understanding of the coronal conditions within the Alfvén surface below approximately 20 solar radii, through which the CME evolves (Reiss et al. 2021).

Therefore, simulating the propagation of ICMEs using MHD codes (see, e.g., Shiota & Kataoka (2016); Jin et al. (2017); Török et al. (2018); Poedts et al. (2020)) is challenging: the boundary conditions for all MHD codes are based on solar magnetic field measurements, but these are known to have large inherent uncertainties. Möstl et al. (2018) have shown that small uncertainties in the initial conditions of CMEs grow to large uncertainties in the predictions of the magnetic field at 1AU. Moreover, physics-based models of CME propagation are limited because the solar corona is not an MHD environment. While an MHD simulation can describe the large-scale coronal conditions within the Alfven surface, the evolution of CMEs occurs over a broad range of spatial scales. Small-scale processes that may trigger instabilities must therefore be taken into account for accurate modeling.

Machine learning presents an alternative to physics-based predictive models. To date, there has been little focus in the literature on using upstream in situ solar wind measurements at the Sun-Earth L_1 point for predicting the B_z component of the *magnetic obstacle* embedded within ICMEs. A *magnetic obstacle* is defined as the magnetic structure embedded in an ICME, which can deviate from in situ signatures of an idealized magnetic flux rope introduced in Burlaga et al. (1981). A *sheath* is the region of compressed solar wind between the ICME shock front and the leading edge of the magnetic obstacle (Owens et al. 2005). Recently, Reiss et al. (2021) has explored the hypothesis that upstream in situ measurements of the sheath region and the first few hours of the magnetic obstacle are sufficient for predicting the minimum value of the B_z component in solar coronal mass ejections. The authors have identified ~350 ICME events in the ICMECAT catalog (Möstl et al. 2017, 2020) and used the corresponding time-series data from the WIND, STEREO-A, and STEREO-B satellites to create a dataset used to train a number of machine learning algorithms. For each event, Reiss et al. (2021) computed six statistical measures for the following physical parameters: B_t, v_t, the magnetic field components (B_x, B_y, B_z), the proton temperature and density. The statistical measures that were computed are the mean value, the standard deviation, the minimum and maximum values, the ratio between the maximum and the minimum values, and the ratio between the mean value and the standard deviation. The authors then used these $7 \times 6 = 42$ features to characterize

each event and train the algorithms. The algorithms that have been considered include the linear regressor, the random forest regressor, and the gradient boosting regressor. While Reiss et al. (2021) showed reasonable results, improvements in the prediction accuracy are desirable. In this paper, we expand on the work of Reiss et al. (2021) and use machine learning techniques to predict the min(B_z) component in solar coronal mass ejections from upstream in situ solar wind measurements.

This paper is organized as follows. In Section 2, we discuss the dataset of ICMEs. In Section 3, we present our feature engineering approach. In Section 4, we provide details on the machine learning algorithms considered in this work. Section 5 states the metrics used to assess the algorithm performance. Section 6 summarizes the results. Section 7 discusses the problem of predicting CMEs on the Sun, which is relevant to building an operational space weather foreasting tool. In Section 8 we summarize our work.

2. DATASET

To allow for comparison of our work with that of Reiss et al. (2021), we use the same ICMECATv2.0 dataset (Möstl et al. 2017, 2020) that Reiss et al. (2021) had used. The catalog includes upstream in situ measurements of 558 ICMEs observed close to 1 AU, at the Sun-Earth L_1 point, by WIND, STEREO-A, and STEREO-B satellites, between January 1, 2007 – April 1, 2021. The dataset contains the following in situ magnetic field and bulk plasma measurements: B_t, v_t, the magnetic field components (B_x, B_y, B_z), the proton temperature T_p and density N_p. The cadence of time-series is one to two minutes. An example solar wind time-series are shown in Figure 1. From the 558 ICMEs, we follow Reiss et al. (2021) and focus on 362 ICMEs that have either sheath region signatures or a density pileup in front of a magnetic obstacle. Figure 2 shows the probability distribution of the physical properties of the 362 ICMEs. Left

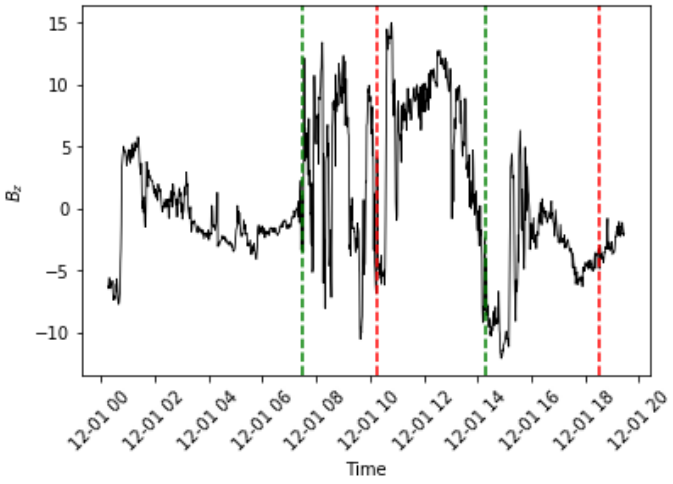

Figure 1. Solar wind time-series. The vertical red dashed lines mark the time interval in the magnetic obstacle where we predict B_z. The green dashed lines indicate the beginning and end of the time range from which we compute input features.

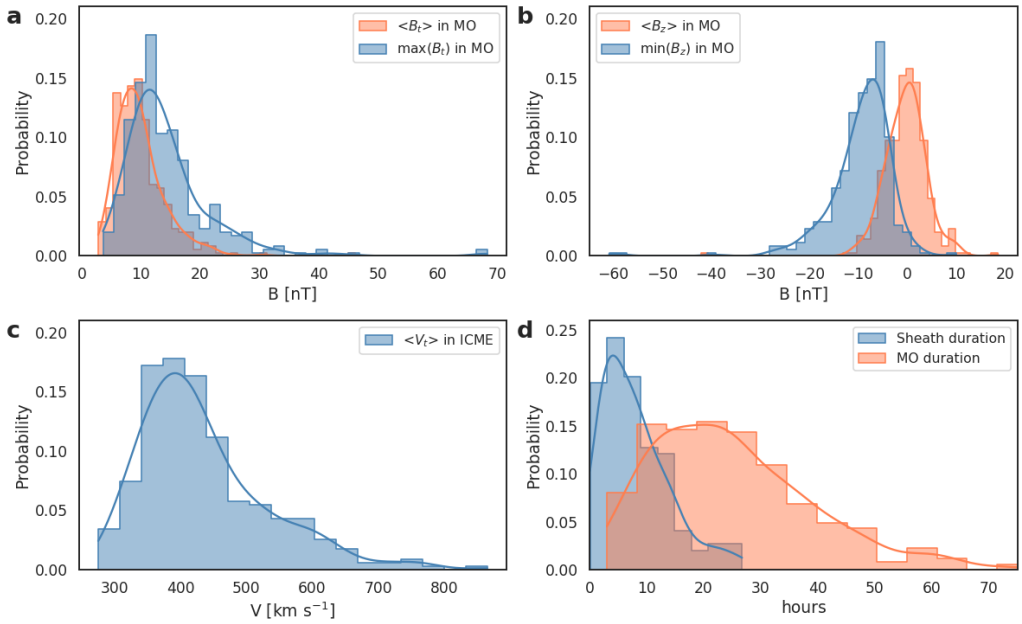

Figure 2. The probability distribution of the physical properties of 362 ICMEs in the dataset. (a) The mean and maximum total magnetic field in the magnetic obstacle; (b) the minimum and mean B_z in the magnetic obstacle; (c) the mean bulk plasma speed in the ICME including the sheath and the magnetic obstacle; and (d) the sheath and magnetic obstacle durations.

subplot of Figure 3 shows the heliocentric distance as a function of time for the 362 ICMEs. Right subplot of Figure 3 shows the minimum value of the B_z component in the magnetic obstacles.

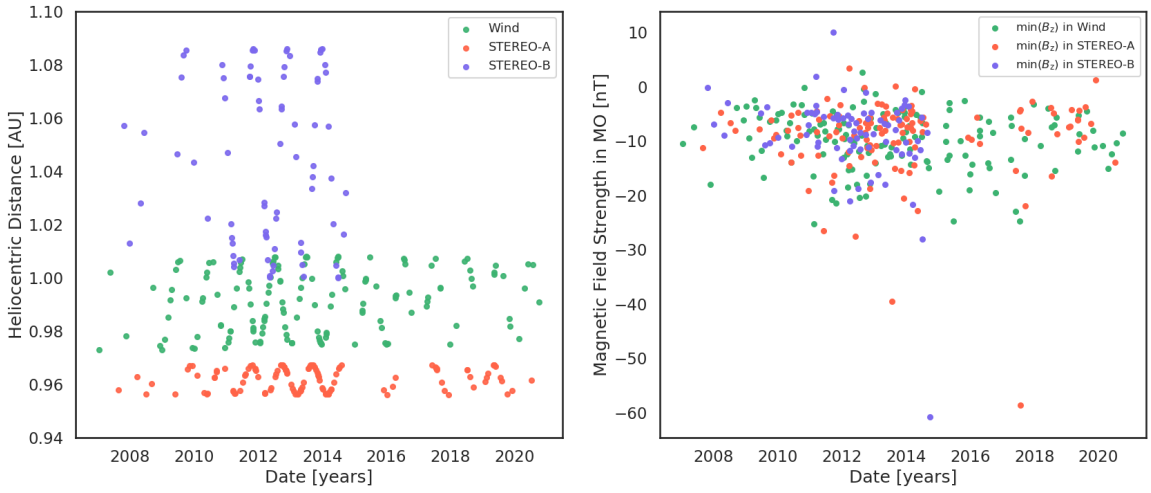

Figure 3. Properties of the 362 ICMEs in the dataset. (a) Heliocentric distance for each ICME event as a function of time, observed with Wind (green), STEREO-A (red), and STEREO-B (blue); (b) minimum value of the B_z component in the magnetic obstacle of the ICME for each event as a function of time.

3. FEATURE ENGINEERING AND VISUALIZATION

In this section we describe how we may visualize the raw time-series provided in the ICME catalog and extract features. Engineering new features from the data may allow to better characterize the dataset and improve algorithm performance. Properly defining and characterizing the "shape" of data is one way to extract new features. Topological Data Analysis (TDA) is a set of tools for studying the shape of data using the methods from algebraic topology (Carlsson 2014). Persistent homology is one of the main tools in TDA. It has been successfully applied in different research fields, helping to study questions ranging from protein folding (Xia & Wei 2014) to econometric analysis of financial time series (Gidea & Katz 2018). Below we discuss the basic notions from algebraic topology and introduce persistent homology.

A *simplicial complex* K is a set of simplices satisfying the following conditions:

- if a simplex is in K, then all of its faces must be in K.
- if $\sigma_1, \sigma_2 \in K$ and $\sigma_1 \cap \sigma_2 \neq \varnothing$, then $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2.

The underlying topological space of K is the union of the geometric realization of its simplices: points for 0-simplices, line segments for 1-simplices, filled triangles for 2-simplices, et cetera. In this paper, we only consider finite simplicial complexes with finite dimension.

Vietoris-Rips complex is one example of a simplicial complex widely used in TDA applications. We start with V, a (finite) point cloud in $\mathbb{R}^d$, and form the Vietoris-Rips complex $VR_V(r)$ as follows. For each subset S of points in V, construct a closed ball of a fixed radius r around each point in S, and include S as a simplex if all the balls have pairwise intersections. See Figure 4 for a visualization of the formation of r-balls.

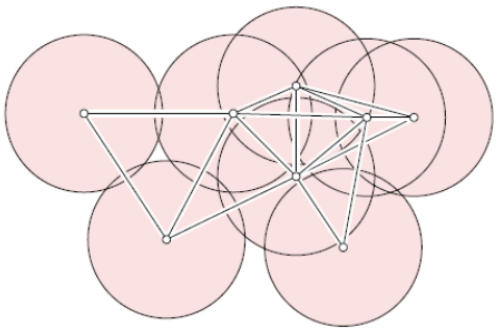

Figure 4. Nine points with pairwise intersections among the disks indicated by straight edges connecting their centers. Source: Edelsbrunner & Harer (2009).

Given a simplicial complex K, we may form a chain complex:

$$\ldots \xrightarrow{\partial_{n+1}} C_n(K) \xrightarrow{\partial_n} C_{n-1}(K) \xrightarrow{\partial_{n-1}} \ldots \xrightarrow{\partial_1} C_0(K) \xrightarrow{\partial_0} 0$$

where $C_n(K)$ is the free abelian group of n-chains generated by all the n-simplices in K and $\partial_n : C_n(K) \to C_{n-1}(K)$ are homomorphisms, called *boundary operators*, that satisfy $\partial_{n-1} \circ \partial_n = 0$. The n-th homology group is defined as the quotient group $H_n(K) = \ker \partial_n / Im(\partial_{n+1})$. For a more detailed discussion of homology theory, see Hatcher (2000).

Persistent homology is a method for computing the evolution of k-dimensional "holes" of a topological space with spatial resolutions. It is defined via the following steps.

- We represent space as a simplicial complex K and define a metric on the space.
- Then, we compute a filtration F of K, which is a collection of simplicial complexes $F = \{K_t \mid t \in \mathbb{R}\}$ of K such that $K_t \subset K_s$ for $t < s$ and there exists $t_{\max} \in \mathbb{R}$ such that $K_{t_{\max}} = K$. The filtration time of a simplex $\sigma \in K$ is the smallest t such that $\sigma \in K_t$. Clearly, $VR_V(r) \subset VR_V(r')$ if $r \leq r'$ via the inclusion map. The simplicial complexes $VR_V(r)$ together with the inclusion maps define a filtered simplicial complex VR_V called *Vietoris-Rips filtration.*
- Persistent homology then describes how the homology of the simplicial complex K changes with filtration F. If the same k-dimensional hole is detected along a large number of subsets in the filtration, then it may be a true feature of the underlying space, rather than noise. A bar in the k-dimensional persistence barcode, with endpoints $[t_{start}, t_{end})$ corresponds to a k-dimensional hole that appears at filtration time t_{start} and remains until filtration time t_{end}. The set of bars $[t_{start}, t_{end})$ representing birth and death times of homology classes is called the *persistence barcode* $B(F)$ of the filtration F. The set of points $(t_{start}, t_{end}) \in \mathbb{R}^2$ is called the *persistence diagram* $dgm(F)$ of the filtration F.

Recently, Rucco et al. (2016) introduced a new entropy measure, called *persistent entropy*, for quantifying how much the construction of a filtration is ordered. This measure is defined as the Shannon entropy of the persistence barcode of a given filtration:

Definition Given a filtration $F = \{K_t \mid t \in \mathbb{R}\}$ and the corresponding persistence diagram $dgm(F) = \{(x_i, y_i) | 1 \leq i \leq n\}$, let $L = \{l_i = y_i - x_i | 1 \leq i \leq n\}$. The persistent entropy $E(F)$ is defined as $E(F) = -\sum_{i=1}^{n} p_i \log(p_i)$ where $p_i = \frac{l_i}{S_L}$ and $S_L = \sum_{i=1}^{n} l_i$.

Persistent entropy is a stable topological feature, as was shown in Atienza et al. (2017). See Chazal & Michel (2017) for a comprehensive introduction to other aspects of topological data analysis.

Having discussed the basics of Topological Data Analysis, we now turn our attention to defining our approach for visualizing the data and applying TDA tools to extract features. In Figure 5, we place the six time series characterizing one particular event in our ICME dataset on T^2, deforming the radial component of each small circle

according to the normalized value of a given physical parameter at that point. In mathematical terms, let R, r be the big and small radii used to parametrize the torus in the Toroidal/Poloidal coordinate system. Denote a given time series by S_j and suppose that we have m time series in total, each containing n points. We denote by $S_{i,j}$ the i-th value of the j-th time series. If we denote by (θ, ϕ) the poloidal and toroidal angles, respectively, then a point $(x_{i,j}, y_{i,j}, z_{i,j}) \in \mathbb{R}^3$ representing the i-th value in the j-th time series is defined as follows:

$$x_{i,j} = \left(R + \left(r + S_{i,j}\right)\cos\theta_j\right)\cos\phi_i, \tag{1}$$

$$y_{i,j} = \left(R + \left(r + S_{i,j}\right)\cos\theta_j\right)\sin\phi_i, \tag{2}$$

$$z_{i,j} = \left(r + S_{i,j}\right)\sin\theta_j, \tag{3}$$

where $i \in \{1, ..., n\}$, $j \in \{1, ..., m\}$.

From the deformed torus shown in Figure 5, we then compute its persistent diagram using Vietoris-Rips persistence (see Figure 6). From the persistent diagram, we can in principle compute the persistent entropy for the zeroth, first, and second homology groups. In this section, we have proposed a general procedure for visualizing a dataset and extracting topological features from it. Due to time constraints of this project, we will consider incorporating persistent entropy and other TDA-based features as input to our machine learning algorithms in the future.

For now, we construct the dataset used to train and test our machine learning algorithms in this work as follows. The features are calculated for the sheath region and the first 4 hours of the magnetic obstacle. See Figure 1 for an example of the solar wind time-series. Each input time-series for a given event span ten hours. We split each time series into ten one-hour bins and compute the six statistical features introduced in Reiss et al. (2021) in each bin. The statistical features are the mean value, the standard deviation, the minimum and maximum values, the ratio between the maximum and the minimum values, and the ratio between the mean value and the standard deviation. Having created the dataset, we randomly split it into the training and testing sets (the testing set is 30% of the whole dataset). The testing set is not used during algorithm training and instead used to validate the performance.

4. MACHINE LEARNING

To predict the B_z component in the magnetic obstacle embedded within an ICME, we train machine learning algorithms on the properties of 362 ICMEs. We consider the following algorithms: Linear Regressor (LR), support-vector regressor (SVR, Smola & Schölkopf (2004)), Gradinet Boosting Regressor (GBR Friedman (2001)), Random

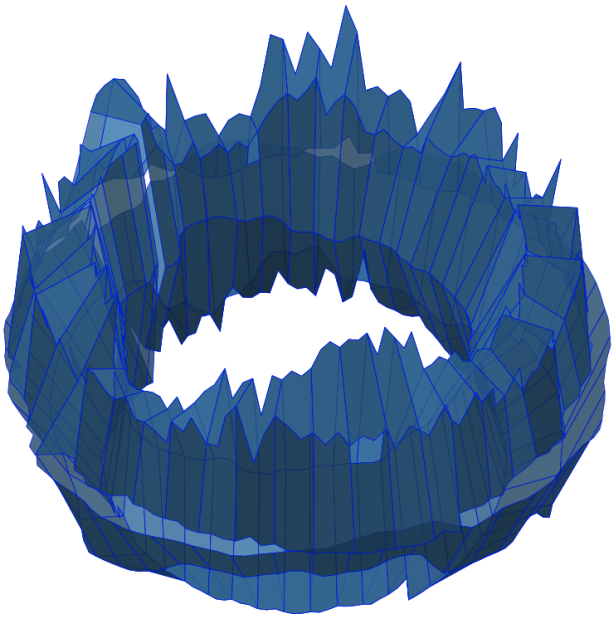

Figure 5. T^2 deformed by the solar wind.

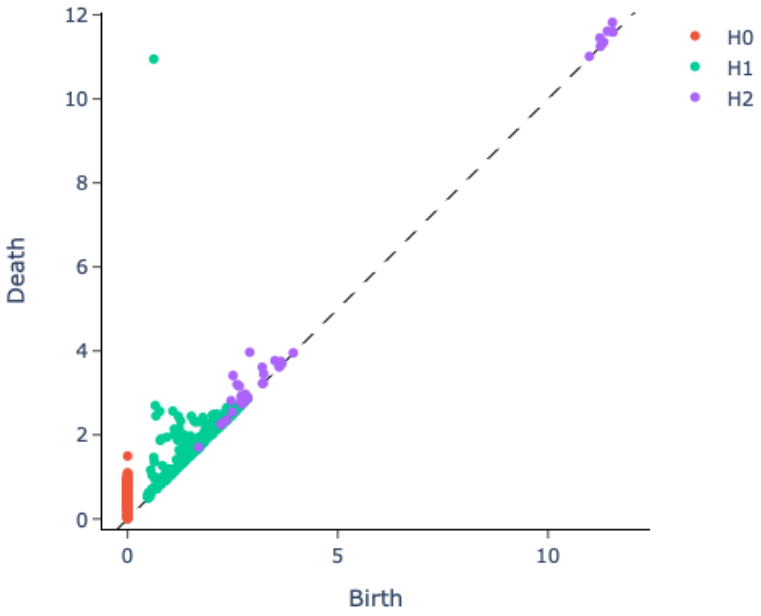

Figure 6. Persistence diagram of the T^2 deformed by the solar wind (Vietoris-Rips persistence). Different colors indicate different homology groups, as shown in the legend.

Forest Regressor (RFR, Biau & Scornet (2015)), xgboost (Chen & Guestrin 2016), lightgbm (Ke et al. 2017), and catboost (Prokhorenkova et al. 2017). To keep this paper manageable in scope, below we discuss the framework of Random Forest regressors, and we refer the reader to the literature for a comprehensive review of each of the other algorithms.

Random Forest is a high performance regression technique which is widely used in various industries. As discussed in Biau & Scornet (2015), the general goal is to predict the square integrable random response $Y \in \mathbb{R}$ by estimating the regression function $m(x) = \mathbb{E}[Y|X = x]$ using an observed input random vector $X \in \chi \subset \mathbb{R}^p$. In practice, we construct an estimate $m_n : X \to \mathbb{R}$ of the function m from a training sample $D_n = ((X_1, Y_1), ..., (X_n, Y_n))$ of independent random variables distributed as the pair (X, Y). A random forest consists of a collection of M randomized regression trees. We denote by $m_n(x; \theta_j, D_n)$ the predicted value at the query point x of the j-th tree, where $\theta_1, ..., \theta_M$ are independent random variables, distributed the same as a generic random variable θ and independent of D_n. In practice, θ is used to resample the training set before the construction of each tree and to select the directions for splitting. The j-th tree estimate is given by

$$m_n(x; \theta_j, D_n) = \sum_{i \in D_n^*(\theta_j)} \frac{\mathbb{1}_{X_i \in A_n(x;\theta_j,D_n)} Y_i}{N_n(x; \theta_j, D_n)} \tag{4}$$

where $D_n^*(\theta_j)$ is the set of data points selected before to the tree construction, $A_n(x; \theta_j, D_n)$ is the cell containing x, and $N_n(x; \theta_j, D_n)$ is the number of preselected points that fall into $A_n(x; \theta_j, D_n)$. Then, the random forest regression estimate is obtained by averaging the estimates from the individual trees:

$$m_{M,n}(x; \theta_1, ..., \theta_M, D_n) = \frac{1}{M} \sum_{j=1}^{M} m_n(x; \theta_j, D_n) \tag{5}$$

The root of a given tree is χ itself and, at each step of the tree construction, a node ("leave") is split into two parts. The terminal leaves give a partition of χ. To construct the random forest regressor, we build M different trees as follows. First, a_n observations are drawn at random with (or without) replacement from the dataset. These observations are then used as input to a given tree. Next, at each node of the given tree, we perform a split by maximizing the CART-criterion over *mtry* directions chosen uniformly at random among the p original ones. We call the resulting subset of the coordinates as M_{try}. We stop building the tree when each of its nodes contains less than *nodesize* points. For any point $x \in X$, our tree's prediction is the average of the Y_i (that were among the a_n points) for which the corresponding X_i falls into the cell of x.

We now give a defintion of the CART-split criterion. For simplicity, consider a tree that uses the entire dataset D_n. Let A be a generic leave and $N_n(A)$ be the number of data points falling in A. We define a cut in A as a pair (j, z), where $j \in \{1, ..., p\}$ and z the position of the cut along the j-th coordinate, within the limits of A. Define C_A to be the set of all such possible cuts in A. Let $X_i = (X_i^{(1)}, ..., X_i^{(p)})$. Then for any $(j, z) \in C_A$, we define the CART-split criterion as follows:

$$L_{reg,n}(j,z) = \frac{1}{N_n(A)} \sum_{i=1}^{n} (Y_i - \overline{Y}_A)^2 \mathbb{1}_{X_i \in A} - \frac{1}{N_n(A)} \sum_{i=1}^{n} (Y_i - \overline{Y}_{A_L} \mathbb{1}_{X_i^{(j)} < z} - \overline{Y}_{AR} \mathbb{1}_{X_i^{(j)} \geq z})^2 \mathbb{1}_{X_i \in A} \tag{6}$$

where $A_L = \{x \in A : x^{(j)} < z\}$, $A_R = \{x \in A : x^{(j)} \geq z\}$ and $\overline{Y}_A$ (respectively, $\overline{Y}_{A_L}, \overline{Y}_{A_R}$) is the average of the Y_i belonging to A (respectively, A_L, A_R). For each node A, the best cut $(j_n^\star, z_n^\star)$ is selected by maximizing $L_{reg,n}(j, z)$ over M_{try} and C_A, i.e.

$$(j_n^\star, z_n^\star) \in \arg \max_{j \in M_{try}, (j,z) \in C_A} L_{reg,n}(j, z). \tag{7}$$

In other words, for a given node of a particular tree, the algorithm finds the best cut by choosing uniformly at random *mtry* coordinates in $\{1, ..., p\}$ and evaluating function (6) over all possible cuts along the directions in M_{try}.

The following hyperparameters can be tuned to enhance the performance of the random forest regressor: the number of sampled data points in each tree, the number of trees that the algorithm builds before averaging the predictions; the maximum number of features the random forest considers splitting a node; the minimum number of leaves required to split an internal node. The random forest regressor has the following advantages. Since not all features are considered while making an individual tree, each tree is different, which leads to the reduction of the feature space. Moreover, since the output is based on majority voting or averaging, it helps address the problem of overfitting. From the practical point of view, random forest can be implemented efficiently via parallelization since each decision tree is created independently of one another. The main disadvantage of the random forest algorithm is its complexity and training time (which depends on the number of trees).

In this work, we use GBR and RFR algorithms with 300 decision trees, where the maximum depth of each tree is 3.

5. METRICS

We evaluate the algorithm performance using point-to-point metrics summarized in Table 1. The metrics include the mean error (ME), mean absolute error (MAE), and the root mean square error (RMSE), the skill score (SS), and the Pearson correlation

Table 1. Point-to-point comparison metrics.

Metric	Definition
Mean error (ME)	$\frac{1}{n}\sum_{i=1}^{n}(f_{obs,i} - f_{pred,i})$
Mean square error (MSE)	$\frac{1}{n}\sum_{i=1}^{n}(f_{obs,i} - f_{pred,i})^2$
Mean absolute error (MAE)	$\frac{1}{n}\sum_{i=1}^{n}\lvert f_{obs,i} - f_{pred,i}\rvert$
Root mean square error (RMSE)	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_{obs,i} - f_{pred,i})^2}$
Skill score (SS)	1 - $\frac{MSE_{pred}}{MSE_{ref}}$

NOTE—$f_{obs,i}$ and $f_{pred,i}$ denote observed and predicted values, respectively. MSE_{pred} is the mean square error of the prediction, and MSE_{ref} is the mean square error of the baseline model.

coefficient (PCC). The skill score (SS) is a measure that quantifies the skill of a forecast in comparison to the baseline model. Table 1 gives the definition of the SS. A negative SS means the model is worse than the baseline model, a SS of 0 means the model is equal to the baseline model, whereas 1 indicates an ideal prediction. In our case, the baseline model is the model that returns the average of all the targets that we trained our machine learning algorithms on.

6. RESULTS

We used our training dataset to train the algorithms described in Section 4. We validate the algorithms using the test dataset, which consists of 105 ICMEs. The 105 ICMEs were randomly selected from January 1, 2007 to April 1, 2021 and were not used during the training. The algorithms' performance on the test dataset is summarized in Table 2. Our GBR regressor produces better results than the results obtained in Reiss et al. (2021). The GBR in Reiss et al. (2021) had ME of 0.52, MAE of 3.12, RMSE of 4.77, SS of 0.49, and PCC of 0.71, whereas in our case the ME is 0.51, MAE is 2.83, RMSE is 4.05, SS is 0.65, and PCC is 0.81.

7. PREDICTING CORONAL MASS EJECTIONS USING TOPOLOGICAL DATA ANALYSIS-BASED CLASSIFIER

Working towards an opearational space weather forecasting tool, in this section we consider another research problem related to forecasting geomagnetic storms. In general, a successful space weather forecasting tool should be capable of the following: it should be able to forecast whether a Coronal Mass Ejection will happen on the Sun in the first place and then predict the impact that it might have once it arrives to

Table 2. Performance summary.

Model	ME	MAE	RMSE	SS	PCC
LR	-6.87	18.15	89.59	-168.52	-0.61
RFR	0.67	2.90	4.13	0.64	0.81
GBR	0.51	**2.83**	**4.05**	**0.65**	0.81
SVR	-0.66	4.36	6.91	-0.01	0.07
xgboost	-3.26	4.11	6.27	0.17	0.70
lightgbm	**0.32**	3.14	4.68	0.54	**0.74**
catboost	0.79	3.84	5.09	0.45	0.69

NOTE—Performance summary of the algorithms (without parameter fine tuning). ME, MAE, RMSE are measured in nT.

the Earth and interacts with the Earth's magnetic field. While the previous sections are focused on addressing the latter point, here we are interested in the problem of predicting the emergence of Coronal Mass Ejections using solar flare time-series data. Despite the progress in numerical modeling, it is still unclear which conditions will produce a CME; however, it is known that CMEs and solar flares are associated as "a single magnetically driven event" (Webb & Howard 2012). In this section, we develop a topological data analysis-based classifier to predict whether an M- or X-class flaring active region will produce a CME.

7.1. *Data*

We build a catalog of active regions (ARs) that either produced both a flare and a CME (the positive class) or only a flare (the negative class). To determine if an AR produced a CME, we retrieve data from the SOHO Large Angle and Spectrometric Coronagraph Experiment instrument and both coronagraphs on the Solar Terrestrial Relations Observatory Sun Earth Connection Coronal and Heliospheric Investigation instruments by querying the Space Weather Database Of Notification, Knowledge, Information[1] for events between 2010 May 1 and 2019 July 1. We reject flares unassociated with an AR and flares below the M1.0-class because they release a limited amount of energy. To determine whether or not an AR produced an X- or M-class flare, we retrieve a GOES flare list by querying the Heliophysics Events Knowledgebase[2] (Hurlburt et al. 2012) using a SunPy python library (SunPy Community et al. 2015). We reject flares in the flare list unassociated with an AR and that are not within ±70 degrees of central meridian during the GOES X-ray Flux peak time because the signal-to-noise ratio in the HMI vector magnetic field data decreases considerably after this longitude.

[1] http://kauai.ccmc.gsfc.nasa.gov/DONKI/
[2] http://www.lmsal.com/hek

We use Spaceweather HMI Active Region Patches (SHARPs) data[3] to characterize every event in our catalog at t hours before the GOES X-ray flare peak time, where t ranges from 6 to 24 hr in 6 hr intervals; SHARPs data contains 18 features that parameterize the vector magnetic field within ARs observed by the Solar Dynamic Observatory (SDO), such as the magnetic flux contained in AR and the current helicity (for details, see Bobra, M. G. and Sun, X. and Hoeksema, J. T. et al. (2014)).

We also restrict events to those where (1) the absolute value of the radial velocity of SDO is less than 3500 m s^{-1}(see Section 7.1.2 of Hoeksema, J. Todd and Liu, Yang and Hayashi et al. (2014)) and (2) HMI data are of high quality (see Section Appendix A of Hoeksema, J. Todd and Liu, Yang and Hayashi et al. (2014)) at this time. In the end, we are left with 62 events in the positive class and 338 events in the negative class as input to the topological data analysis-based classifier.

7.2. *Topological Data Analysis-based Classifier*

Later in the text, a "datapoint" refers to a single set of 18 SHARP features. Each "sample" is an event in the catalog, which is associated with four datapoints. The TDA-based classifier presented here begins by randomly sampling each sample using a number of datapoints. Then, it applies a one-dimensional filter function (the first and the second principle components in our study) to the data space. Next, the range of values created by this filter function is divided into non-overlapping intervals of some arbitrary length. Within these intervals, local clustering is conducted. Linkage method can be chosen; in our study, the metric is Euclidean distance, and the linkage is complete linkage. Since the created clusters contain a number of datapoints from each of the samples, an $n \times m$ matrix is constructed as input to our classifier (which is a Feed-forward Neural Network with one hidden layer) where n is the number of samples, m is the number of clusters, and entries are the number of datapoints in a given cluster.

7.3. *Results*

The algorithm's sampling rate can be chosen; since each event is represented with only 4 datapoints in our study, we use all 4 datapoints to characterize every event. 50 runs of the TDA-based classifier were conducted to obtain the average classification accuracy. The results are presented in Figure 7. The TDA-based classifier had 80.5% accuracy using 15 training examples, increasing to 86.2% with 285 training examples. In general, the classifier seems to become more stable (the classification accuracy deviates less among 50 runs) when increasing the number of training examples.

8. CONCLUSION

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

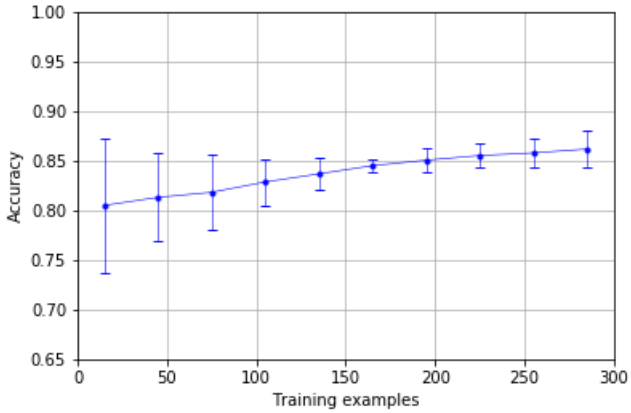

Figure 7. Prediction accuracy of the TDA-based classifier

The ability to predict $\min(B_z)$, a proxy for ICME geoeffectiveness, is essential for any space weather forecast. We have trained and tested a number of machine learning algorithms to predict $\min(B_z)$ component in solar coronal mass ejections from upstream in situ solar wind measurements. The best-performing algorithm improves upon the performance of Reiss et al. (2021): our GBR has the ME of 0.51, MAE of 2.83, RMSE of 4.05, SS of 0.65, and PCC of 0.81. Our machine learning tool can be used to improve space weather forecasting methods in the future. Working towards an operational forecasting tool, in this paper we have also discussed the problem of predicting the emergence of CMEs on the Sun. In the future, we would like to study other features that can be extracted from the solar wind time-series and provide physical insight/interpretation of the impact that the engineered features may have on the algorithm performance.

9. ACKNOWLEDGEMENTS

I am grateful to Herman Verlinde and Zoltan Szabo for helpful discussions. I would also like to thank Christian Mostl and Martin Reiss for compiling the ICME catalog and for productive correspondence regarding the use of the dataset.

REFERENCES

Atienza, N., Gonzalez-Diaz, R., & Rucco, M. 2017, Persistent Entropy for Separating Topological Features from Noise in Vietoris-Rips Complexes, arXiv, doi: 10.48550/ARXIV.1701.07857

Biau, G., & Scornet, E. 2015, A Random Forest Guided Tour, arXiv, doi: 10.48550/ARXIV.1511.05741

Bobra, M. G. and Sun, X. and Hoeksema, J. T. et al. 2014, Solar Physics, 289, 3549, doi: 10.1007/s11207-014-0529-3

Burlaga, L., Sittler, E., Mariani, F., & Schwenn, R. 1981, Journal of Geophysical Research, 86, 6673, doi: 10.1029/ja086ia08p06673

Carlsson, G. 2014, Acta Numerica, 23, 289–368, doi: 10.1017/S0962492914000051

Chazal, F., & Michel, B. 2017, arXiv e-prints, arXiv:1710.04019. https://arxiv.org/abs/1710.04019

Chen, T., & Guestrin, C. 2016, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM), doi: 10.1145/2939672.2939785

Dungey, J. W. 1961, Physical Review Letters, 6, 47, doi: 10.1103/physrevlett.6.47

Edelsbrunner, H., & Harer, J. 2009, Computational Topology (American Mathematical Society), doi: 10.1090/mbk/069

Friedman, J. H. 2001, The Annals of Statistics, 29, 1189. http://www.jstor.org/stable/2699986

Gidea, M., & Katz, Y. 2018, Physica A: Statistical Mechanics and its Applications, 491, 820, doi: 10.1016/j.physa.2017.09.028

Gonzalez, W. D., & Tsurutani, B. T. 1987, Planetary and Space Science, 35, 1101, doi: 10.1016/0032-0633(87)90015-8

Hatcher, A. 2000, Algebraic topology (Cambridge: Cambridge Univ. Press). https://cds.cern.ch/record/478079

Hoeksema, J. Todd and Liu, Yang and Hayashi et al. 2014, Solar Physics, 289, 3483, doi: 10.1007/s11207-014-0516-8

Hurlburt, N., Cheung, M., & Schrijver, C. et al. 2012, Solar Physics, 275, 67, doi: 10.1007/s11207-010-9624-2

Jin, M., Manchester, W. B., van der Holst, B., et al. 2017, The Astrophysical Journal, 834, 173, doi: 10.3847/1538-4357/834/2/173

Ke, G., Meng, Q., Finley, T., et al. 2017, in Advances in Neural Information Processing Systems, ed. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett, Vol. 30 (Curran Associates, Inc.). https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

Möstl, C., Isavnin, A., Boakes, P. D., et al. 2017, Space Weather, 15, 955, doi: 10.1002/2017sw001614

Möstl, C., Amerstorfer, T., Palmerio, E., et al. 2018, Space Weather, 16, 216, doi: 10.1002/2017sw001735

Möstl, C., Weiss, A. J., Bailey, R. L., et al. 2020, The Astrophysical Journal, 903, 92, doi: 10.3847/1538-4357/abb9a1

Owens, M. J., Arge, C. N., Spence, H. E., & Pembroke, A. 2005, Journal of Geophysical Research, 110, doi: 10.1029/2005ja011343

Poedts, S., Lani, A., Scolini, C., et al. 2020, Journal of Space Weather and Space Climate, 10, 57, doi: 10.1051/swsc/2020055

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. 2017, CatBoost: unbiased boosting with categorical features, arXiv, doi: 10.48550/ARXIV.1706.09516

Reiss, M. A., Möstl, C., Bailey, R. L., et al. 2021, Space Weather, 19, doi: 10.1029/2021sw002859

Rouillard, A. 2011, Journal of Atmospheric and Solar-Terrestrial Physics, 73, 1201, doi: 10.1016/j.jastp.2010.08.015

Rucco, M., Castiglione, F., Merelli, E., & Pettini, M. 2016, in Proceedings of ECCS 2014, ed. S. Battiston, F. De Pellegrini, G. Caldarelli, & E. Merelli (Cham: Springer International Publishing), 117–128

Shiota, D., & Kataoka, R. 2016, Space Weather, 14, 56, doi: 10.1002/2015sw001308

Smola, A. J., & Schölkopf, B. 2004, Statistics and Computing, 14, 199, doi: 10.1023/B:STCO.0000035301.49549.88

SunPy Community, Mumford, T., & Christe, S. J. et al. 2015, Computational Science and Discovery, 8, 014009, doi: 10.1088/1749-4699/8/1/014009

Török, T., Downs, C., Linker, J. A., et al. 2018, The Astrophysical Journal, 856, 75, doi: 10.3847/1538-4357/aab36d

Webb, D. F., & Howard, T. A. 2012, Living Reviews in Solar Physics, 9, 3, doi: 10.12942/lrsp-2012-3

Xia, K., & Wei, G.-W. 2014, International Journal for Numerical Methods in Biomedical Engineering, 30, 814, doi: 10.1002/cnm.2655