The Astrophysical Journal, 855:109 (10pp), 2018 March 10 https://doi.org/10.3847/1538-4357/aaae69

A New Tool for CME Arrival Time Prediction using Machine Learning Algorithms: CAT-PUMA

Jiajia Liu[1], Yudong Ye[2,3], Chenglong Shen[4,5], Yuming Wang[4,5], and Robert Erdélyi[1,6]
[1] Solar Physics and Space Plasma Research Center (SP2RC), School of Mathematics and Statistics,
The University of Sheffield, Sheffield S3 7RH, UK; jj.liu@sheffield.ac.uk
[2] SIGMA Weather Group, State Key Laboratory of Space Weather, National Space Science Center,
Chinese Academy of Sciences, Beijing 100190, People's Republic of China
[3] University of Chinese Academy of Sciences, Beijing 100049, People's Republic of China
[4] CAS Key Laboratory of Geospace Environment, Department of Geophysics and Planetary Sciences, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China
[5] Synergetic Innovation Center of Quantum Information & Quantum Physics, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China
[6] Department of Astronomy, Eötvös Loránd University, Budapest, Pázmány P. sétány 1/A, H-1117, Hungary
Received 2018 January 2; revised 2018 February 7; accepted 2018 February 7; published 2018 March 13

Abstract

Coronal mass ejections (CMEs) are arguably the most violent eruptions in the solar system. CMEs can cause severe disturbances in interplanetary space and can even affect human activities in many aspects, causing damage to infrastructure and loss of revenue. Fast and accurate prediction of CME arrival time is vital to minimize the disruption that CMEs may cause when interacting with geospace. In this paper, we propose a new approach for partial-/full halo CME Arrival Time Prediction Using Machine learning Algorithms (CAT-PUMA). Via detailed analysis of the CME features and solar-wind parameters, we build a prediction engine taking advantage of 182 previously observed geo-effective partial-/full halo CMEs and using algorithms of the Support Vector Machine. We demonstrate that CAT-PUMA is accurate and fast. In particular, predictions made after applying CAT-PUMA to a test set unknown to the engine show a mean absolute prediction error of ∼5.9 hr within the CME arrival time, with 54% of the predictions having absolute errors less than 5.9 hr. Comparisons with other models reveal that CAT-PUMA has a more accurate prediction for 77% of the events investigated that can be carried out very quickly, i.e., within minutes of providing the necessary input parameters of a CME. A practical guide containing the CAT-PUMA engine and the source code of two examples are available in the Appendix, allowing the community to perform their own applications for prediction using CAT-PUMA.

Key words: solar–terrestrial relations – Sun: coronal mass ejections (CMEs)

1. Introduction

Coronal mass ejections (CMEs) are one of the two major eruptive phenomena (the other being flares) occurring within the solar atmosphere that affect the heliosphere. CMEs leave the Sun at average speeds of 500 km s^{-1}, carrying a large amount of magnetized plasma with an average mass of 10^{15} g into interplanetary space and also carry a huge amount of kinetic energy, often on the order of 10^{30} erg (for reviews, see, e.g., Low 2001; Chen 2011; Webb & Howard 2012; Gopalswamy 2016, and references therein). The following observational facts highlight some of the most important aspects as to why enormous attention has been paid to CMEs in the past several decades since their first discovery (Hansen et al. 1971; Tousey 1973). (1) CMEs are usually accompanied by some other dynamic, large-scale phenomena including, e.g., filament eruptions (e.g., Jing et al. 2004; Wang et al. 2006; Liu et al. 2010a), flares (e.g., Harrison 1995; Qiu et al. 2004; Zhang et al. 2012), magneto-hydrodynamic (MHD) waves (e.g., Biesecker et al. 2002; Chen et al. 2005; Liu et al. 2010b), radio bursts (e.g., Jackson et al. 1978; Lantos et al. 1981; Shen et al. 2013a; Chen et al. 2014), and solar jets (e.g., Shen et al. 2012; Liu et al. 2015; Zheng et al. 2016). Combined studies of CMEs and their accompanying phenomena could improve our understanding of the physical processes taking place in various regimes of the Sun. (2) MHD shocks caused by CMEs could be employed to gain insight into the characteristic properties of the plasma state in the interplanetary space (for reviews, see, e.g., Vršnak & Cliver 2008). (3) CMEs occur with a range of rate of abundance both during solar minimum and maximum (e.g., Gopalswamy et al. 2003; Robbrecht et al. 2009), the study of which may help us explore the solar cycle and dynamo. (4) Shocks and an often large amount of magnetic fluxes carried by CMEs could cause severe disturbances in the Earth's magnetosphere (e.g., Wang et al. 2003, 2007; Zhang et al. 2007; Sharma et al. 2013; Chi et al. 2016) and further affect the operation of high-tech facilities like spacecraft and can cause disruption in the functioning of modern communication systems (including radio, TV, and mobile signals), navigation systems, and can affect the function of pipelines and high-voltage power grids.

Besides intensive efforts made toward a better understanding of how CMEs are triggered (e.g., Gibson & Low 1998; Antiochos et al. 1999; Forbes 2000; Lin & Forbes 2000), many studies have focused on predicting the arrival (or transit) times of CMEs at the Earth, having considered their potentials in largely affecting the Earth's magnetosphere and outer atmosphere. This has become one of the most important contents of the so-called space weather forecasting efforts. However, despite of the lack of in-situ observations of the ambient solar wind and CME plasma in the inner heliosphere at CMEs' eruption, there are several further effects that make it more complex and rather challenging to predict CMEs' arrival time, including, e.g., the fact that CMEs may experience significant deflection while traveling in interplanetary space (e.g., Wang et al. 2004; Gui et al. 2011; Isavnin et al. 2014; Kay et al. 2015;

Zhuang et al. 2017) and that CMEs may interact with each other causing mering or acceleration/deceleration (e.g., Wang et al. 2002a; Shen et al. 2012, 2013c; Mishra et al. 2016; Lugaz et al. 2017).

Current models on the prediction of CME arrival time may be classified into three types: empirical, drag-based, and physics-based (MHD) models (for a review, see, e.g., Zhao & Dryer 2014). Most empirical models use a set of observed CMEs to fit a simple relation (linear or parabolic) between observed CME speeds (and/or accelerations) and their transit times in the interplanetary space (e.g., Vandas et al. 1996; Wang et al. 2002b; Xie et al. 2004; Schwenn 2005; Manoharan 2006). Vršnak & Žic (2007) took the ambient solar-wind speed into account in their empirical model, but still utilized linear least-square fitting. The drag-based models (DBMs) have an advantage over the empirical models in that DBMs take into account the speed difference between CMEs and their ambient solar wind, which may cause considerable acceleration or deceleration of CMEs (e.g., Vršnak 2001; Subramanian et al. 2012). On the other hand, DBMs are based on a hydrodynamic (HD) approach and ignore the potentially important role of the magnetic field in the interaction between CMEs and solar wind. Finally physics-based (MHD) models (e.g., Smith & Dryer 1990; Dryer et al. 2001; Moon et al. 2002; Tóth et al. 2005; Detman et al. 2006; Feng & Zhao 2006; Feng et al. 2007; Riley et al. 2012, 2013) mostly utilize (M)HD simulations employing observations as boundary/initial conditions in the models to perform prediction of the transit times of CMEs. Though, considering the complexity and fewer prediction errors of physics-based (MHD) models, there are a few drawbacks, e.g., they are still highly idealized and may require extensive computational resources in terms of hardware and CPU time (e.g., Tóth et al. 2005). Complex or not, previous predictions give, on average, around 10 hr mean absolute errors on CME arrival times (see review by Zhao & Dryer 2014). Employing 3D observations from the *STEREO* spacecraft, Mays et al. (2013) reduced the mean absolute error to ∼8.2 hr, predicting the arrival time of 15 CMEs. Again using *STEREO* observations, but allowing only very short lead times (∼1 day), Möstl et al. (2014) further enhanced the performance for the arrival times to ∼6.1 hr after applying empirical corrections to their models. A fast and accurate prediction with large lead time, using only one spacecraft, is therefore still much needed.

In this paper, we propose a new approach to modeling the partial-/full halo CME Arrival Time Prediction Using Machine learning Algorithms (CAT-PUMA). We will divide 182 geo-effective CMEs observed in the past two decades, i.e., from 1996 to 2015, into two sets, for training and testing purposes. All inputs will be only observables. Without a priori assumptions or an underlying physical theory, our method provides a mean absolute prediction error around as little as 6 hr. Details on data mining are in Section 2. An overview of the employed machine learning algorithms and the implemented training process are described in Section 3. Results and comparison with previous prediction models are discussed in Section 4. We summarize in Section 5. A practical guide on how to perform predictions with CAT-PUMA is presented in the Appendix.

2. Data Mining

To build a suitable input set for the machine learning algorithms, our first step in data mining was to construct a list of CMEs that eventually arrived at Earth and caused disturbances to the terrestrial magnetic field, usually called geo-effective CMEs. We defined four different Python crawlers to automatically gather the onset time, which is usually defined as the first appearance in the field of view (FOV) of *SOHO* LASCO C2 (Brueckner et al. 1995), and the arrival time of the CMEs, which represents the arrival time of interplanetary shocks driven by CMEs hereafter, using the following lists.

1. The Richardson and Cane List (Richardson & Cane 2010). Available at http://www.srl.caltech.edu/ACE/ASC/DATA/level3/icmetable2.htm, the list contains various parameters, including the average speed, magnetic field, and the associated DST index of more than 500 Interplanetary CMEs (ICMEs) from 1996 to 2006 with the onset time of their associated CMEs, if observed. We discarded events with no or ambiguously associated CMEs and obtained the onset and arrival time of 186 geo-effective CMEs from this list.
2. List of Full Halo CMEs provide by the Research Group on Solar-TErrestrial Physics (STEP) at University of Science and Technology of China (USTC; Shen et al. 2013b). A full halo CME is defined when its angular width observed by *SOHO* LASCO is 360°. Available at http://space.ustc.edu.cn/dreams/fhcmes/index.php, this list provides the 3D direction, angular width, the real and projected velocities of 49 CMEs from 2009 to 2012, and the arrival time of their associated shocks, if observed. Events without observation of the associated interplanetary shocks are removed. The onset and arrival times of 24 geo-effective CMEs were obtained from this list.
3. The George Mason University (GMU) CME/ICME List (Hess & Zhang 2017). This list contains information similar to that of the Richardson and Cane list of 73 geo-effective CMEs and corresponding ICMEs from 2007 to 2017. It is available at http://solar.gmu.edu/heliophysics/index.php/GMU_CME/ICME_List. We only selected ICME events satisfying the following criterion: (i) presence of associated shocks and (ii) multiple CMEs are not involved. After implementing the selection criteria, 38 events were obtained from this list.
4. The CME Scoreboard developed at the Community Coordinated Modeling Center (CCMC), NASA. This website allows the community to submit and view the actual and predicted arrival times of CMEs from 2013 to the present (https://kauai.ccmc.gsfc.nasa.gov/CMEscoreboard/). For our analysis, we removed events that did not interact with the Earth and those that had a "note." Some events were labeled with a "note" because, e.g., the target CME did not arrive at Earth, there was some uncertainty in measuring the shock arrival time, or there were multiple CME events. Here, we obtained 134 CME events from this list.

Combining all four lists, we eventually obtained 382 geo-effective CME events via data mining. However, there are overlaps between these lists. To prevent duplicates, we removed one of such pairs if two CMEs had onset times with a difference of less than 1 hr, resulting in 90 events removed.

The *SOHO LASCO CME Catalog* (https://cdaw.gsfc.nasa.gov/CME_list/) provides a database of all CMEs observed by *SOHO* LASCO from 1996 to 2016 (Gopalswamy et al. 2009). By matching the onset time of CMEs in our list with the onset time of CMEs recorded in the *SOHO LASCO CME Catalog*, we obtained various parameters of the CMEs, including angular width, average speed, acceleration, final speed in the FOV of LASCO, estimated mass, and the main position angle (MPA, corresponding to the position angle of the fastest moving part of the CME's leading edge). The location of the source region of full halo CMEs can be obtained from the *SOHO/LASCO Halo CME Catalog* (https://cdaw.gsfc.nasa.gov/CME_list/halo/halo.html). CMEs that have no source-region information in the above catalog were further investigated manually, one-by-one, to determine their source-region location. Further, events from our compiled list were removed if they had: (i) an angular width less than 90°, (ii) no available mass estimation, or (iii) an ambiguous source-region location. Finally, two CMEs at 2003 October 29 20:54 UT and 2011 October 27 12:12 UT were also removed because the first has an incorrect velocity and acceleration estimation and the second erupted with more than a dozen CMEs during that day.

Eventually, after applying all of the above selection criteria, we obtained a list of 182 events containing geo-effective CMEs from 1996 to 2015, of which 56 are partial halo CMEs and 126 are halo CMEs. The average speed of these CMEs FOV ranges from 400 to 1500 km s^{-1} in the LASCO FOV.

3. Optimization

One of the most popular machine learning algorithms is the Support Vector Machine algorithm (SVM). It is a set of supervised learning methods for classification, regression, and outlier detection. The original SVMs were linear (see the review by Smola & Schölkopf 2004), though SVMs are also suitable for conducting nonlinear analysis via mapping input parameters into higher dimensional spaces with different kernel functions. An implementation of the SVM has been integrated into the Python *scikit-learn* library (Pedregosa et al. 2011), with open-source access and well-established documentation (http://scikit-learn.org/stable/). According to the *scikit-learn* documentation, major advantages of the SVM are that it is (1) effective in high-dimensional spaces, (2) still effective even if the number of dimensions is greater than the number of samples, and (3) memory efficient. Besides, it is particularly well suited for small- or medium-sized data sets (Géron 2017).

Recent works utilizing machine learning algorithms have been mainly focused on solar flare prediction, CME productivity, and solar feature identification using classification methods (e.g., Li et al. 2007; Qahwaji & Colak 2007; Ahmed et al. 2013; Bobra & Couvidat 2015; Bobra & Ilonidis 2016; Nishizuka et al. 2017) or multi-labeling algorithms (e.g., Yang et al. 2017). However, to the best of our knowledge, the SVM regression algorithm, which is suitable for a wide range of solar/space physics research, such as solar cycle prediction, DST index prediction, and active region occurrence prediction, has not yet been widely used by the solar/space physics community. Further, no previous study has attempted to employ the SVM regression algorithm in the context of applying it to the prediction of CME arrival time.

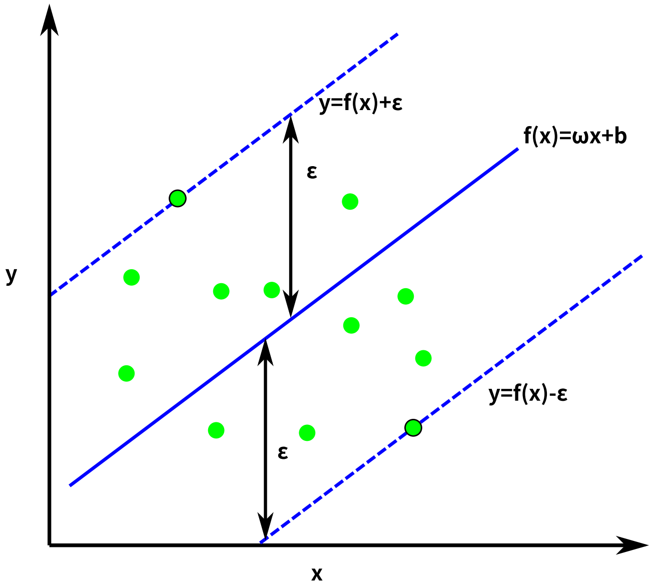


Figure 1. SVM regression in a simple one-dimensional linear and hard-margin problem. Adopted from Figures 5–10 in Géron (2017).

3.1. Brief Re-cap of SVM Regression

To make it simple and clear, we first briefly explain the SVM regression algorithm by demonstrating its capabilities with a simple one-dimensional linear and hard-margin problem. Let us suppose that there is an input set $x = (x_1, x_2, x_3 \ldots x_l)$ and a corresponding known result $y = (y_1, y_2, y_3 \ldots y_l)$ where l is the number of data points. The basic idea of SVM regression is to find a function

$$f(x) = \omega x + b, \tag{1}$$

where $f(x)$ has at most ϵ (>0) deviation from the actual result y_i for all x_i (as shown in Figure 1). Points at the margins (green dots with a black edge) are then called the "support vectors." A new observation x_{l+1} can therefore be taken into Equation (1) to yield a prediction for its unknown result y_{l+1}.

The solution for the above one-dimensional linear and hard-margin problem can be extended into multi-dimensional, linear, and soft-margin problems. In this case, the target for the SVM regression is to

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}||\omega||^2 + C\sum_1^l(\xi_i + \xi_i^*),\\ &\text{subject to} \quad \begin{cases} y_i - \langle\omega, x_i\rangle - b \leqslant \epsilon + \xi_i,\\ \langle\omega, x_i\rangle + b - y_i \leqslant \epsilon + \xi_i,\\ \xi_i, \xi_i^* \geqslant 0, i = 1, 2, 3 \ldots l, \end{cases} \end{aligned} \tag{2}$$

where $x_i = (x_i^1, x_i^2 \ldots x_i^n)$ is an n-dimensional vector with n the number of features, $i \in [1, l]$, $||\omega||$ is the norm of ω, $\langle\omega, x_i\rangle$ is the dot product between ω and x_i, and ξ_i, ξ_i^* are the introduced slack variables to perform the feasible constrains for the soft margins (Smola & Schölkopf 2004; Vapnik 2013). The regularization factor $C > 0$ is introduced to trade off the amount up to which deviations larger than ϵ are tolerated. A larger value of C indicates a lower tolerance on errors.

To extend the solution to be suitable for nonlinear problems, we map the original nonlinear n-dimensional input x into a higher dimensional space $\phi(x)$, in which the problem might be

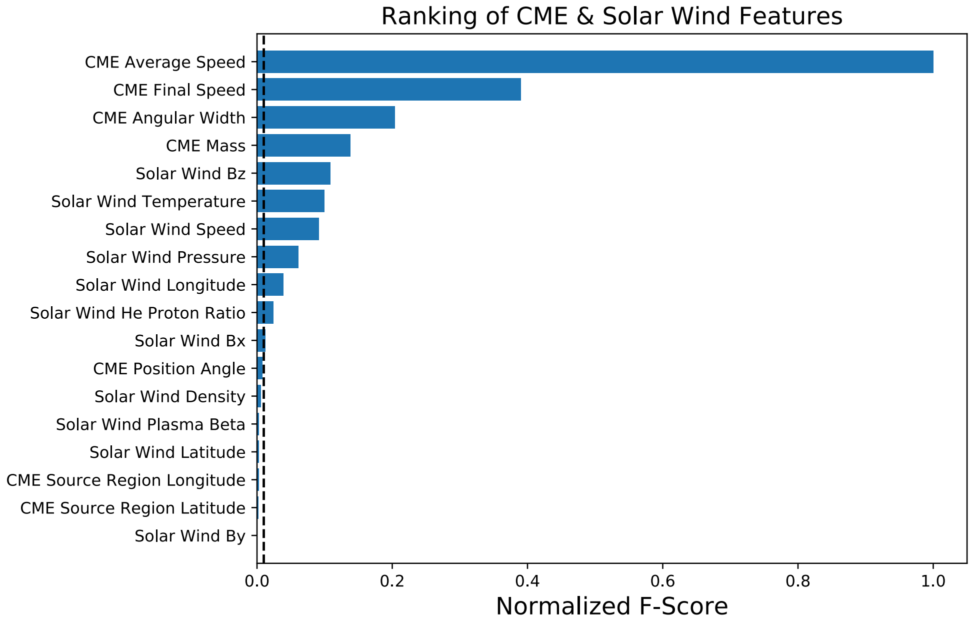


Figure 2. Normalized F-scores of all 18 CME and solar-wind features with $m = 6$ hr. The vertical dashed line indicates a normalized F-score of 0.01.

linear. $\phi(x)$ then replaces x in Equation (2). The most common way to map x into $\phi(x)$ is using kernel functions. One of the most frequently used kernels is the Radial Basis Function kernel,

$$K(x_i, x_j) = \exp(-\gamma\, ||x_i - x_j||^2), \tag{3}$$

where $||x_i{-}x_j||^2$ is the squared Euclidean distance between the two data points. Here, $\gamma > 0$ defines the area of a single point can influence. A larger γ indicates less influence of a point on its neighbors. The description on the SVM regression algorithm above is highly abbreviated. More details can be found in, e.g., Smola & Schölkopf (2004) and Vapnik (2013).

Besides C and γ, another important variable m will be introduced in the rest of this section. The definition of m is given at the beginning of Section 3.2. Processes determining the value of m employed in building the prediction engine are detailed in Section 3.3. Optimization on the selection of parameters C and γ are presented in Section 3.4.

3.2. Feature Selection

Employing the SVM regression algorithms to make predictions of CME arrival time, we take the 182 vectors, each of which contains n parameters of the CME and corresponding solar-wind plasma, as x and their actual transit times as y. Because it is not currently feasible to determine the actual background solar-wind plasma where a CME is immersed, we use averaged in-situ solar-wind parameters at Earth detected from the onset of the CME to m hours later to approximate the actual solar-wind parameters at the CME location. In-situ solar-wind observations at the Earth, including solar-wind B_x, B_y, B_z, plasma density, alpha to proton ratio, flow latitude (north/south direction), flow longitude (east/west direction), plasma beta, pressure, speed, and proton temperature are downloaded from the *OMNIWeb Plus* (https://omniweb.gsfc.nasa.gov/). Together with suitable CME parameters including CME average speed, acceleration, angular width, final speed, mass, MPA, source-region latitude, and source-region longitude, described in Section 2, we have in total 19 ($n = 19$) features in the input x space.

However, some of the above features might be important in determining the CME transit time, while some might be irrelevant and unnecessary. First, the CME acceleration is removed from the feature space because it is not independent and is basically determined by the CME average speed and final speed. To determine the importance of the rest of the features, following Bobra & Ilonidis (2016) but for regression in this case, we use a univariate feature-selection tool (*sklearn.feature_selection.SelectKBest*) implemented in the Python *scikit-learn* library to test the F-score of each individual feature. For $k \in [1, n]$, x^k is a vector with length of l. The correlation between x^k and y of and the F-score of feature k is then defined as

$$\mathrm{Corr} = \frac{(x^k - \overline{x^k}) \cdot (y - \bar{y})}{\sigma_{x^k}\sigma_y},$$
$$F = \frac{\mathrm{Corr}^2}{1 - \mathrm{Corr}^2}(l - 2), \tag{4}$$

where l is the number of data points as defined in Section 3.1, σ_{x^k} and σ_y are the standard deviations of x^k and y, respectively. A higher F-score indicates a higher linear correlation between the kth feature and the CME transit time y in this case.

Table 1 lists the rankings of all 18 features (excluding CME acceleration) with m from 1 to $m_{\max}$ hours. Again, m represents the number of hours after the onset of the CME. $m_{\max}$, the upper limit of m, is set as 12 hr after considering the prediction purpose of CAT-PUMA, because an extremely fast CME (with speed over 3000 km s^{-1}) could reach the Earth within around 13 hr (Gopalswamy et al. 2010). Features with higher F-scores have lower ranking numbers in the table. It turns out that the rankings of all features keep relatively stable. The changes are minor with increasing m, especially for the first 12 features in the table. Figure 2 depicts the normalized F-scores of all features when $m = 6$ hr with the largest F score as 1.

Not surprisingly, the average and final CME speeds have the highest F-scores, suggesting their importance in determining

Table 1
Ranking of All 18 Features with m from 1 to 12 hr

Feature	m (hours)											
	1	2	3	4	5	6	7	8	9	10	11	12
CME Average Speed	1	1	1	1	1	1	1	1	1	1	1	1
CME Final Speed	2	2	2	2	2	2	2	2	2	2	2	2
CME Angular Width	3	3	3	3	3	3	3	3	3	3	3	3
CME Mass	4	4	4	4	4	4	4	5	5	5	4	4
Solar-wind B_z	5	5	5	5	5	5	5	4	4	4	5	6
Solar-wind Temperature	7	7	7	7	7	6	6	6	6	6	6	5
Solar-wind Speed	6	6	6	6	6	7	7	7	7	7	7	7
Solar-wind Pressure	8	8	8	8	8	8	8	8	8	8	8	8
Solar-wind Longitude	11	9	9	9	9	9	9	9	9	9	9	9
CME Acceleration	10	10	10	10	10	10	10	10	10	10	10	10
Solar-wind He Proton Ratio	12	12	11	11	11	11	11	11	11	11	11	11
Solar-wind B_x	9	11	12	12	12	12	12	12	12	13	15	15
CME Position Angle	13	13	13	13	13	13	13	13	15	14	13	12
Solar-wind Density	16	17	15	14	14	14	14	14	14	15	14	13
Solar-wind Plasma Beta	19	18	17	15	18	15	15	15	13	12	12	14
Solar-wind Latitude	18	19	19	19	16	16	18	18	17	17	17	16
CME Source-region Longitude	15	15	14	16	15	17	16	16	16	16	16	17
CME Source-region Latitude	17	16	16	18	17	18	17	17	18	18	18	18
Solar-wind B_y	14	14	18	17	19	19	19	19	19	19	19	19

Note. The column in bold denotes the ranking of all features at $m = 6$ hr, which is the most favorable value in building the prediction engine (Section 3.3).

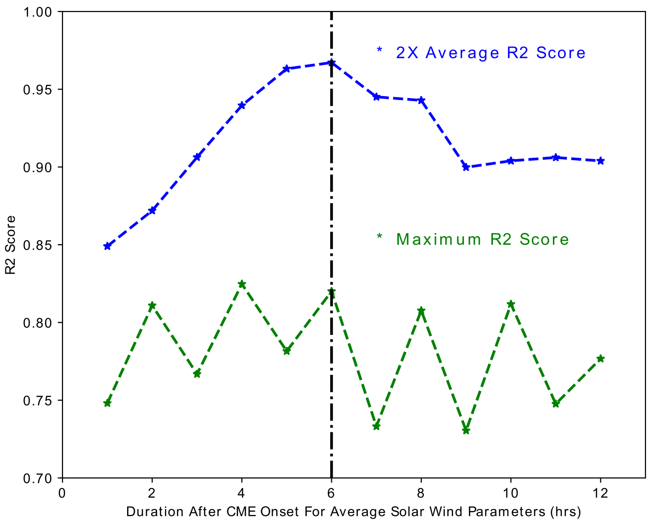


Figure 3. Variation of the average (blue curve) and maximum (green curve) R^2 scores during a 100,000 times training with changing values of m for calculating average solar-wind parameters after CME onset.

the CME transit time. CME angular width and mass rank third and fourth, respectively, which might be due to the fact that the angular width contains information on CME propagating direction and the CME angular width and mass together imply CMEs' plasma density, which could play an important role in the interaction between the CME and the ambient solar wind. Solar-wind features, including magnetic fields B_z and B_x (strength and poloidal direction of the solar-wind magnetic field), proton temperature, plasma pressure, plasma speed, and flow longitude (toroidal direction of the solar-wind plasma flow) also play important roles with relatively high normalized F-scores. The alpha-particle to proton number density ratio in solar wind also ranks high in all of the features, which may be caused the fact that the ratio is usually high in CMEs and co-rotating interaction regions (CIRs; e.g., Prise et al. 2015). CMEs/CIRs in front of a CME could potentially influence its transit time. However, this needs to be further examined via analyzing the in-situ observations preceding all the CMEs. Finally, we select 12 features with normalized F-scores over 0.01 from high to low as the input of the SVM. CME MPA is also included because it has a normalized Fisher score of 0.008, very close to 0.01.

3.3. Determining Solar-wind Parameters

In the previous sub-section, we showed the result of feature selection using solar-wind parameters averaged between the onset time of CMEs and m hours later, where m ranges from 1 to 12. To determine the most favorable value of m in building the prediction engine, (1) we find the optimal C and γ for the data set, followed by (2) training the SVM for 100,000 times, then (3) we re-calculate the optimal C and γ for the best training result. Finally, we repeat the above three steps for m ranging from 1 to 12 hr. Details on the first three steps will be given in Section 3.4. To evaluate how good the models using solar-wind parameters with different values of m are, we use the R^2 score, defined as

$$R^2 = 1 - \frac{\sum_1^l (y_i - f(x_i))^2}{\sum_1^l (y_i - \bar{y})^2}, \qquad (5)$$

where $y_i, f(x_i)$, l are the same as defined in Section 3.1 and $\bar{y}$ is the average value of y. The variation of the maximum and average R^2 scores with increasing m is shown in Figure 3. The average R^2 score peaks at $m = 6$ hr, indicating that the best fitting result is revealed with 6 hr averaged solar-wind parameters after CME onset. The maximum R^2 score varies "periodically" within the range of 0.7–0.85 without an overall peak. This "periodicity" might have been caused by the combined effect that (1) 100,000 is only a fraction of all C_{182}^{37} ($\sim 6 \times 10^{38}$) possibilities (for further details, see Section 3.4),

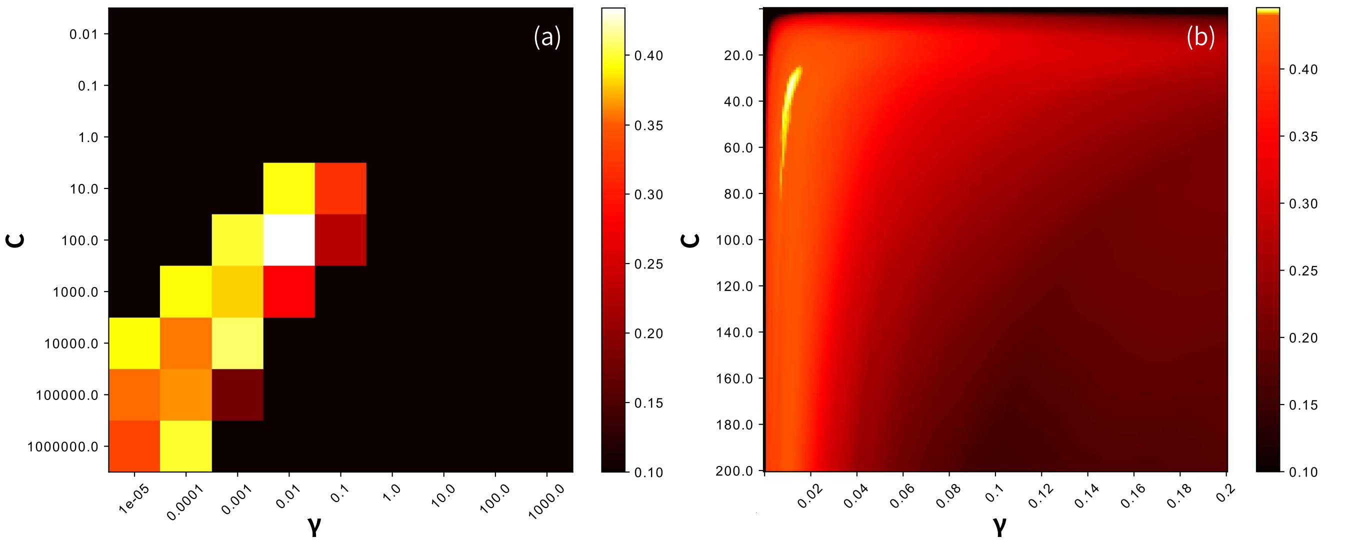


Figure 4. Distribution of the average correlation coefficient between the predicted and actual CME transit times of test sets during three-fold cross-validations repeated for different pairs of C and γ. In panel (a), C ranges in $[10^{-2}, 10^{6}]$ and γ ranges in $[10^{-5}, 10^{3}]$. In panel (b), C ranges from (0, 200] and γ ranges from (0, 0.2].

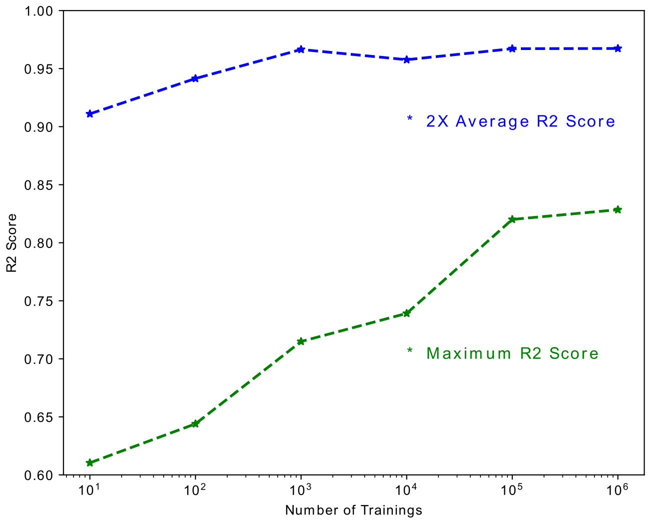


Figure 5. Variation of the average (blue curve) and maximum (green curve) R^2 scores with increasing number of trainings.

thus the best R^2 score out of all possibilities cannot always be found during every training, and (2) the imperfect stochastic process of the computer in shuffling the data set (see Paragraph 2, Section 3.4). Even though the exact causes of the above "periodicity" need further investigation, the variation of the average R^2 scores suggests that 100,000 is large enough to reflect the overall distribution of the R^2 scores.

To summarize the above, we found that using 6 hr averaged solar-wind parameters after the CME onset can result in the best output.

3.4. Training the SVM

One major concern of the SVM regression is the choice of parameters C and γ. In Section 3.1, it was demonstrated that the regularization factor C trades off the tolerance on errors. A larger (smaller) C indicates that the SVM will attempt to incorporate more (fewer) data points. An ill-posed C or γ could result in over-fitting (the SVM attempts to fit all data points, which may result in bad prediction for new inputs) or under-fitting (the SVM fits too few data points—it cannot represent the trend of variation of the data). To find the optimal parameters, we utilize the *sklearn.model_selection.GridSearchCV* function to perform exhaustive searches over specified values. First, we build a logarithmic grid with a basis of 10, in which C ranges from $[10^{-2}, 10^{6}]$ and γ ranges from $[10^{-5}, 10^{3}]$, as the input of the *GridSearchCV* function. It turns out that the R^2 score peaks when C is on the order of 10^2 and γ of 10^{-2} (Figure 4(a)). Then, we perform the above exhaustive search again but with C from (0, 200] with a step of 1 and γ from (0, 0.2] with a step of 10^{-3}. A more accurate pair of C and γ is then found, $C = 32$ and $\gamma = 0.012$ (Figure 4(b)).

For the purpose of cross-validation, we split the entire data set into two subsets: the training set and the test set. Amari et al. (1997) found the optimal number of the test set as $l/\sqrt{2n}$, where l and n are the number of data points and features, respectively. Taking $l = 187$ and $n = 13$ in our case, we found that the partition of the entire data set between the training set and the test set should be 80%:20% (145:37). Using the optimal pair of parameters C and γ found above, we fed the training set into the SVM regression algorithm to build a prediction engine. Next, we made a prediction of the CME transit times using the test set and calculating the R^2 score between the predicted and actual transit times. To find the best result with the highest R^2 score, we randomly shuffled the entire data set (the order of events in the data set is shuffled, which is a general practice to avoid bias, see, e.g., Géron 2017) and repeated the above steps (i.e., split the shuffled data set into the training and test sets, build an engine using the training set, and calculated the R^2 score of the test set). Theoretically, there are C_{182}^{37} ($\sim 6 \times 10^{38}$) possible combinations of the training set and test set. This is a huge number and it is impossible to exhaustively test all of the possibilities given the available computer power.

Figure 5 shows the variation of the average (blue curve) and maximum (green curve) R^2 scores among all of the test sets with the increasing number of trainings. The average R^2 score increases continuously before the number of trainings reaches

1000, and remains almost unchanged after that. This suggests that when training is performed over 1000 times, the result can reflect the basic distribution of the R^2 scores for all C_{182}^{37} possibilities. The maximum R^2 score increases steeply when the number of performed trainings is less than 100,000 and yields a similar value when it is increased by a factor of 10. This indicates that it becomes more feasible to find the best engine with an increasing number of trainings.

Considering the above results and reasonable CPU time consumption, we repeated 100,000 times of trainings to find the best training set, which results in a highest R^2 score of its corresponding test set, to construct the engine. This could be rather costly. However, via paralleling the process employing the open-source Message Passing Interface (Open MPI, http://www.open-mpi.org/), a 100,000-time training only takes ∼25 minutes on an *Intel(R) Core(TM) i7-7770K* desktop with 8 threads. However, we should note that training the SVM regression 100,000 times cannot always reveal the best result (as shown by the green dashed line in Figure 3), because 100,000 is only a fraction of all of the possibilities (C_{182}^{37}). Multiple runs are sometimes needed to repeat the 100,000 times of trainings.

4. Results and Comparison

Let us now use the shuffled data set that yields the highest R^2 score of the test set among all the training instances as the input to the engine. The optimal $C = 71$ and $\gamma = 0.012$ are obtained, again, based on the selected shuffled data set. We then split this data set into a training set and a test set. CAT-PUMA is then built based on the training set and optimal parameters.

Figure 6(a) shows the relation between the actual transit time and the predicted transit time given by CAT-PUMA of the test set. Different blue dots represent different CME events. The black dashed line represents a perfect prediction when the predicted transit time has the same value as the actual transit time. From the distribution of the dots, one sees that they scatter close to the dashed line. The R^2 score is ∼0.82, the mean absolute error of the prediction is 5.9 ± 4.3 hr, and the root-mean-square error is 7.3 hr. The probability of detection (POD) is defined as

$$\mathrm{POD} = \frac{\mathrm{Hits}}{\mathrm{Hits} + \mathrm{Misses}}, \tag{6}$$

where events with absolute prediction errors less and more than 5.9 hr are defined as "hits" and "misses," respectively. There are 20 events in the test set having absolute prediction errors less than 5.9 hr (Table 2), giving a POD of 54%.

There are currently more than a dozen different methods submitted to the *NASA CME Scoreboard* by a number of teams to present their predictions of CME arrival times. These methods include empirical, drag-based, and physics-based models. More details on the utilized models can be found in the *NASA CME Scoreboard* website (https://kauai.ccmc.gsfc.nasa.gov/CMEscoreboard/) and references therein. Let us now compare the absolute prediction error of CAT-PUMA and the average absolute errors of all other methods available from the *NASA CME Scoreboard* and determine how much progress we have made over the average level of current predictions. Figure 6(b) shows the comparison for CMEs included in both the test set and the *NASA CME Scoreboard*, with Figure 6(c)

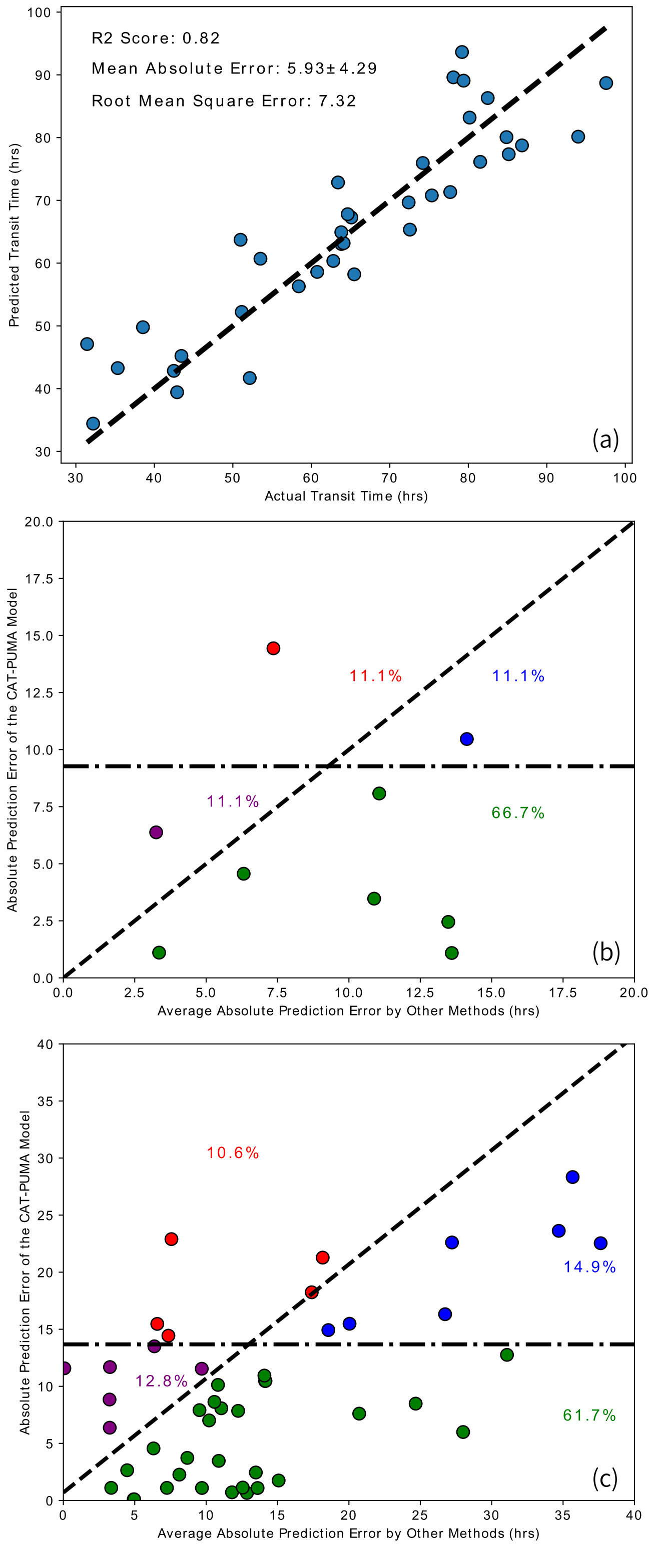


Figure 6. (a) Predicted transit time by CAT-PUMA vs. actual transit time for CMEs in the test set. The black dashed line denotes the same values of the predicted and actual transit time. (b) Comparison between absolute prediction errors by CAT-PUMA and average absolute errors of other methods in the *NASA CME Scoreboard*. Only data points included in both the *NASA CME Scoreboard* and the test set are shown in this panel. (c) Similar to panel (b) but for all CMEs included in the *NASA CME Scoreboard*. The black dashed lines represent that CAT-PUMA has the same prediction errors with the average of other methods. Black dash–dotted lines indicate an absolute error of 9.3 (panel b) and 13.7 (panel c) hours, respectively.

for all CMEs included in the *NASA CME Scoreboard*. The dashed lines in both panels indicate when CAT-PUMA has the same prediction error as the average of other models. The dash–dotted lines represent a prediction error level of 9.3 (panel b) and 13.7 (panel c) hours, which are the mean values of the average absolute errors of other methods. Both panels show very similar results. Considering there are only 9 data points in panel (b), we focus on results revealed by panel (c). Green dots (61.7%) are events where CAT-PUMA performs better and has errors less than 13.7 hr, blue dots (14.9%) are where CAT-PUMA performs better but has errors larger than 13.7 hr, and purple dots (12.8%) are where CAT-PUMA performs worse but has errors less than 13.7 hr. Finally, red dots (10.6%) are events where CAT-PUMA performs worse and has errors larger than 13.7 hr. In total, CAT-PUMA gives a better prediction for 77% of the events and has an error less than 13.7 hr for 74% of the events.

5. Summary

In this paper, we proposed a new tool for partial-/full halo CME Arrival Time Prediction Using Machine learning Algorithms (CAT-PUMA). During building the prediction engine, we investigated which observed features may be important in determining the CME arrival time via a feature-selection process. CME properties including the average speed, final speed, angular width, and mass were found to play the most relevant roles in determining the transit time in the interplanetary space. Solar-wind parameters including magnetic fields B_z and B_x, proton temperature, flow speed, flow pressure, flow longitude, and alpha-particle to proton-number-density ratio were also found important.

The average values of solar-wind parameters between the onset time of the CME and 6 hr later were found to be the most favorable in building the engine. Considering an average speed of 400 km s^{-1} of the solar wind, it typically takes a 104 hr traveling time from the Sun to Earth. Our results indicate that properties of solar wind detected at Earth might have a periodicity of $(104 + 6)/24 = 4.6$ days. However, this needs to be further examined very carefully by future works.

After obtaining the optimal pair of input parameters C and γ, the CAT-PUMA engine was then constructed based on the training set that yields a highest R^2 of the test set during trainings carried out 100,000 times. The constructed engine turns out to have a mean absolute error of about 5.9 hr in predicting the arrival time of CMEs for the test set, with 54% of the predictions having absolute errors less than 5.9 hr. Comparing with the average performance of other models available in the literature, CAT-PUMA has better predictions in 77% events and prediction errors less than the mean value of average absolute errors of other models in 74% events.

To summarize, the main advantages of CAT-PUMA are that it provides accurate predictions with a mean absolute error less than 6 hr; it does not rely on a priori assumptions or theories; due to the underlying principles of machine learning, CAT-PUMA can evolve and promisingly improve with more input events in the future; and finally, CAT-PUMA is a very fast open-source tool allowing all interested users to give their own predictions within several minutes after providing necessary inputs. The shortcoming of CAT-PUMA is that it cannot give a prediction whether a CME will hit the Earth or not.

Table 2
Number and Percentage of Hits and Misses in the Test Set

	Hits	Misses
Number	20	17
Percentage	54%	46%

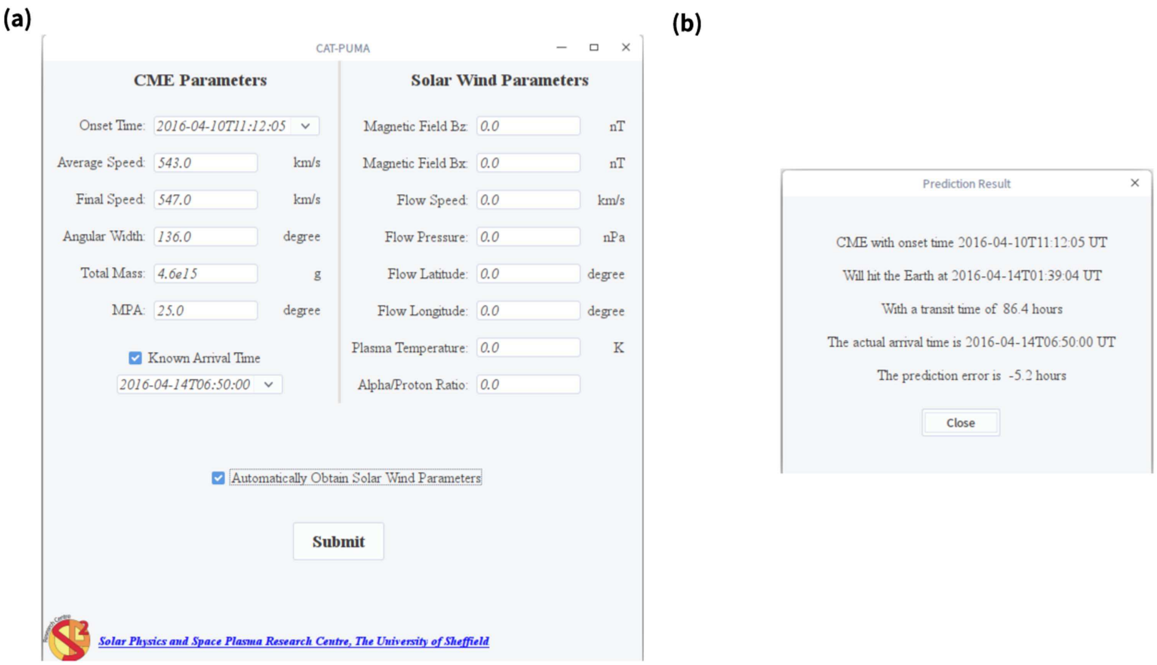
CAT-PUMA has not included information on the 3D propagating direction of CMEs. We propose that future efforts toward including the 3D propagation direction and 3D de-projected speed, employing either the graduated cylindrical shell (GCS) model with multi-instrument observations (Thernisien et al. 2006) or the integrated CME-arrival forecasting (iCAF) system (Zhuang et al. 2017), together with more observed geo-effective CME events, will further improve the prediction accuracy of CAT-PUMA.

The *SOHO LASCO CME catalog* is generated and maintained at the CDAW Data Center by NASA and The Catholic University of America in cooperation with the Naval Research Laboratory. *SOHO* is a project of international cooperation between ESA and NASA. J.L. appreciates discussions with Dr. Xin Huang (National Astronomical Observatories, Chinese Academy of Sciences). We thank Dr. Manolis K. Georgoulis (Research Center for Astronomy and Applied Mathematics, Academy of Athens) for his useful advice in improving this paper. J.L. and R.E. acknowledge the support (grant number ST/M000826/1) received by the Science and Technology Facility Council (STFC), UK. R.E. is grateful for the support received from the Royal Society (UK). Y.W. is supported by grants 41574165 and 41774178 from NSFC.

Appendix
A Practical Guide of Using the CAT-PUMA to Predict CME Arrival Time

CAT-PUMA is designed to have a very easy user-friendly approach. Users can download the CAT-PUMA engine (“engine.obj”), the source code (“cat_puma.py”) of an example demonstrating how we performed the prediction, and the source code (“cat_puma_qt.py”) of a well-designed User Interface (UI) from the following link: https://github.com/PyDL/cat-puma. All codes are written in Python, and have been tested with Python 2.7 on two *Debian*-based x86-64 Linux systems (*Ubuntu* and *Deepin*) and the x86-64 *Windows* 10 system. Modifications of the code will be needed if one prefers to run CAT-PUMA with Python 3. Python libraries, including *datetime, numpy, pandas, pickle, and scikit-learn (v0.19.1)*, are needed for a proper run of “cat_puma.py.” In the following, we explain the example code “cat-puma.py” in details.

The first 134 lines in the code import necessary libraries and define functions that will be used in the main program. Lines 138–152 define features that we are going to use, the value of m (see Section 3.3), and the location of the engine file. Users are not suggested to revise these lines. Lines 155–163 are as follows.



CME Parameters

Onset Time: 2016-04-10T11:12:05

Average Speed: 543.0 km/s

Final Speed: 547.0 km/s

Angular Width: 136.0 degree

Total Mass: 4.6e15 g

MPA: 25.0 degree

☑ Known Arrival Time 2016-04-14T06:50:00

Solar Wind Parameters

Magnetic Field Bz: 0.0 nT

Magnetic Field Bx: 0.0 nT

Flow Speed: 0.0 km/s

Flow Pressure: 0.0 nPa

Flow Latitude: 0.0 degree

Flow Longitude: 0.0 degree

Plasma Temperature: 0.0 K

Alpha/Proton Ratio: 0.0

☑ Automatically Obtain Solar Wind Parameters

Solar Physics and Space Plasma Research Centre, The University of Sheffield

Prediction Result

CME with onset time 2016-04-10T11:12:05 UT

Will hit the Earth at 2016-04-14T01:39:04 UT

With a transit time of 86.4 hours

The actual arrival time is 2016-04-14T06:50:00 UT

The prediction error is -5.2 hours

Figure 7. User Interface of CAT-PUMA.

```
# CME Parameters
time = ‘2015-12-28T12:12:00’ # CME Onset time in LASCO C2
width = 360. # angular width, degree, set as 360 if it
  is halo
speed = 1212. # linear speed in LASCO FOV, km/s
final_speed = 1243. # second order final speed leaving
  LASCO FOV, km/s
mass = 1.9e16 # estimated mass using ‘cme_mass.pro’ in
  SSWIDL or
  # obtained from the SOHO LASCO CME Catalog
mpa = 163. # degree, position angle corresponding to the
  fasted front
actual = ‘2015-12-31T00:02:00’ # Actual arrival time,
  set to None if unknown
```

The above lines define the onset time, angular width, average speed, final speed, estimated mass, and MPA of the target CME. These parameters can easily be obtained from the *SOHO LASCO CME Catalog* (https://cdaw.gsfc.nasa.gov/CME_list/) if available or by analyzing LASCO fits files otherwise. Here, we employ a fast halo CME that erupted at 2015-12-28T12:12 UT as the first example. This event was not included in our input data set when constructing CAT-PUMA. Line 166 defines whether a user prefers to obtain the solar-wind parameters automatically. If yes, the code will download solar-wind parameters for the specified CME automatically from the *OMNIWeb Plus* website (https://omniweb.gsfc.nasa.gov/).

Next, one can then run the code, typically via typing in the command *python2 cat_puma.py*, after following the above instructions to setup the user's own target CME. The prediction will be given within minutes. The prediction result for the above CME is as follows (information in the last two lines will not be given if one has not specified the actual arrival time).

```
CME with onset time 2015-12-28T12:12:00 UT
will hit the Earth at 2015-12-30T18:29:33 UT
with a transit time of 54.3 hours
The actual arrival time is 2015-12-31T00:02:00 UT
The prediction error is −5.5 hours
```

Alternatively, one can use the well-designed UI via running the command *python2 cat_puma_qt.py*. A proper run needs additional Python library *PyQt5* installed. Let us illustrate how this UI can be used with another example CME that erupted at 2016 April 10T11:12 UT. Again, this event was not included in our input data set when constructing CAT-PUMA either. Figure 7(a) shows the UI and corresponding CME parameters for this event. Average speed (543 km s^{-1}), final speed (547 km s^{-1}), angular width (136°), and the MPA (25°) were obtained from the *SOHO LASCO CME Catalog*. The mass of the CME was estimated by the built-in function "cme_mass.pro" in the *SolarSoft IDL*, which turns out to be $\sim 4.6 \times 10^{15}$ g. By checking the option "Automatically Obtain Solar-wind Parameters," solar-wind parameters are obtained automatically from the *OMNIWeb Plus* website (https://omniweb.gsfc.nasa.gov/) after clicking the "Submit" button. Then, actual values of the solar-wind parameters are shown. Parameters that are not available from the *OMNIWeb Plus* website are set to 0.00001 (manually input of these parameters are then needed in this case, near real-time solar-wind data can be download from the *CDAWeb* website https://cdaweb.sci.gsfc.nasa.gov/istp_public/). Figure 7(b) shows the prediction result for the above CME, revealing an error of 5.2 hr.

ORCID iDs

Jiajia Liu https://orcid.org/0000-0003-2569-1840
Yudong Ye https://orcid.org/0000-0002-1854-8459
Chenglong Shen https://orcid.org/0000-0002-3577-5223
Yuming Wang https://orcid.org/0000-0002-8887-3919
Robert Erdélyi https://orcid.org/0000-0003-3439-4127

References

Ahmed, O. W., Qahwaji, R., Colak, T., et al. 2013, SoPh, 283, 157
Amari, S.-i., Murata, N., Muller, K.-R., Finke, M., & Yang, H. H. 1997, ITNN, 8, 985
Antiochos, S. K., DeVore, C. R., & Klimchuk, J. A. 1999, ApJ, 510, 485

Biesecker, D. A., Myers, D. C., Thompson, B. J., Hammer, D. M., & Vourlidas, A. 2002, ApJ, 569, 1009

Bobra, M. G., & Couvidat, S. 2015, ApJ, 798, 135

Bobra, M. G., & Ilonidis, S. 2016, ApJ, 821, 127

Brueckner, G. E., Howard, R. A., Koomen, M. J., et al. 1995, SoPh, 162, 357

Chen, P. F. 2011, LRSP, 8, 1

Chen, P. F., Fang, C., & Shibata, K. 2005, ApJ, 622, 1202

Chen, Y., Du, G., Feng, L., et al. 2014, ApJ, 787, 59

Chi, Y., Shen, C., Wang, Y., et al. 2016, SoPh, 291, 2419

Detman, T., Smith, Z., Dryer, M., et al. 2006, JGRA, 111, A07102

Dryer, M., Fry, C. D., Sun, W., et al. 2001, SoPh, 204, 265

Feng, X., & Zhao, X. 2006, SoPh, 238, 167

Feng, X., Zhou, Y., & Wu, S. T. 2007, ApJ, 655, 1110

Forbes, T. G. 2000, JGR, 105, 23153

Géron, A. 2017, Hands-on Machine Learning with Scikit-Learn and TensorFlow: Concepts Tools and Techniques to Build Intelligent Systems (Sebastopol, CA: O'Reilly Media)

Gibson, S. E., & Low, B. C. 1998, ApJ, 493, 460

Gopalswamy, N. 2016, GSL, 3, 8

Gopalswamy, N., Lara, A., Yashiro, S., Nunes, S., & Howard, R. A. 2003, in ESA Special Publication 535, Solar Variability as an Input to the Earth's Environment, ed. A. Wilson (Noordwijk: ESA), 403

Gopalswamy, N., Yashiro, S., Michalek, G., et al. 2009, EM&P, 104, 295

Gopalswamy, N., Yashiro, S., Michalek, G., et al. 2010, SunGe, 5, 7

Gui, B., Shen, C., Wang, Y., et al. 2011, SoPh, 271, 111

Hansen, R. T., Garcia, C. J., Grognard, R. J.-M., & Sheridan, K. V. 1971, PASAu, 2, 57

Harrison, R. A. 1995, A&A, 304, 585

Hess, P., & Zhang, J. 2017, SoPh, 292, 80

Isavnin, A., Vourlidas, A., & Kilpua, E. K. J. 2014, SoPh, 289, 2141

Jackson, B. V., Sheridan, K. V., Dulk, G. A., & McLean, D. J. 1978, PASAu, 3, 241

Jing, J., Yurchyshyn, V. B., Yang, G., Xu, Y., & Wang, H. 2004, ApJ, 614, 1054

Kay, C., Opher, M., & Evans, R. M. 2015, ApJ, 805, 168

Lantos, P., Kerdraon, A., Rapley, G. G., & Bentley, R. D. 1981, A&A, 101, 33

Li, R., Wang, H.-N., He, H., Cui, Y.-M., & Du, Z.-L. 2007, ChJAA, 7, 441

Lin, J., & Forbes, T. G. 2000, JGR, 105, 2375

Liu, J., Wang, Y., Shen, C., et al. 2015, ApJ, 813, 115

Liu, R., Liu, C., Wang, S., Deng, N., & Wang, H. 2010a, ApJL, 725, L84

Liu, W., Nitta, N. V., Schrijver, C. J., Title, A. M., & Tarbell, T. D. 2010b, ApJL, 723, L53

Low, B. C. 2001, JGR, 106, 25141

Lugaz, N., Temmer, M., Wang, Y., & Farrugia, C. J. 2017, SoPh, 292, 64

Manoharan, P. K. 2006, SoPh, 235, 345

Mays, M. L., Taktakishvili, A., Pulkkinen, A. A., et al. 2013, in American Geophysical Union Fall Meeting 2013 (Washington, DC: AGU), SH53A-2143

Mishra, W., Wang, Y., & Srivastava, N. 2016, ApJ, 831, 99

Moon, Y.-J., Dryer, M., Smith, Z., Park, Y. D., & Cho, K. S. 2002, GeoRL, 29, 1390

Möstl, C., Amla, K., Hall, J. R., et al. 2014, ApJ, 787, 119

Nishizuka, N., Sugiura, K., Kubo, Y., et al. 2017, ApJ, 835, 156

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, arXiv:1201.0490

Prise, A. J., Harra, L. K., Matthews, S. A., Arridge, C. S., & Achilleos, N. 2015, JGRA, 120, 1566

Qahwaji, R., & Colak, T. 2007, SoPh, 241, 195

Qiu, J., Wang, H., Cheng, C. Z., & Gary, D. E. 2004, ApJ, 604, 900

Richardson, I. G., & Cane, H. V. 2010, SoPh, 264, 189

Riley, P., Linker, J. A., Lionello, R., & Mikic, Z. 2012, JASTP, 83, 1

Riley, P., Linker, J. A., & Mikić, Z. 2013, JGRA, 118, 600

Robbrecht, E., Berghmans, D., & Van der Linden, R. A. M. 2009, ApJ, 691, 1222

Schwenn, R., dal Lago, A., Huttunen, E., & Gonzalez, W. D. 2005, AnGeo, 23, 1033

Sharma, R., Srivastava, N., Chakrabarty, D., Möstl, C., & Hu, Q. 2013, JGRA, 118, 3954

Shen, C., Liao, C., Wang, Y., Ye, P., & Wang, S. 2013a, SoPh, 282, 543

Shen, C., Wang, Y., Pan, Z., et al. 2013b, JGRA, 118, 6858

Shen, C., Wang, Y., Wang, S., et al. 2012, NatPh, 8, 923

Shen, F., Shen, C., Wang, Y., Feng, X., & Xiang, C. 2013c, GeoRL, 40, 1457

Shen, Y., Liu, Y., Su, J., & Deng, Y. 2012, ApJ, 745, 164

Smith, Z., & Dryer, M. 1990, SoPh, 129, 387

Smola, A. J., & Schölkopf, B. 2004, Statistics and Computing, 14, 199

Subramanian, P., Lara, A., & Borgazzi, A. 2012, GeoRL, 39, L19107

Thernisien, A. F. R., Howard, R. A., & Vourlidas, A. 2006, ApJ, 652, 763

Tóth, G., Sokolov, I. V., Gombosi, T. I., et al. 2005, JGRA, 110, A12226

Tousey, R. 1973, in Space Research Conference, The Solar Corona, ed. M. J. Rycroft & S. K. Runcorn (Berlin: Akademie-Verlag), 713

Vandas, M., Fischer, S., Dryer, M., Smith, Z., & Detman, T. 1996, JGR, 101, 15645

Vapnik, V. 2013, The Nature of Statistical Learning Theory (New York: Springer-Verlag)

Vršnak, B. 2001, SoPh, 202, 173

Vršnak, B., & Cliver, E. W. 2008, SoPh, 253, 215

Vršnak, B., & Žic, T. 2007, A&A, 472, 937

Wang, Y., Shen, C., Wang, S., & Ye, P. 2004, SoPh, 222, 329

Wang, Y., Ye, P., & Wang, S. 2007, SoPh, 240, 373

Wang, Y., Zhou, G., Ye, P., Wang, S., & Wang, J. 2006, ApJ, 651, 1245

Wang, Y. M., Wang, S., & Ye, P. Z. 2002a, SoPh, 211, 333

Wang, Y. M., Ye, P. Z., Wang, S., & Xue, X. H. 2003, GeoRL, 30, 1700

Wang, Y. M., Ye, P. Z., Wang, S., Zhou, G. P., & Wang, J. X. 2002b, JGRA, 107, 1340

Webb, D. F., & Howard, T. A. 2012, LRSP, 9, 3

Xie, H., Ofman, L., & Lawrence, G. 2004, JGRA, 109, A03109

Yang, Y. H., Tian, H. M., Peng, B., Li, T. R., & Xie, Z. X. 2017, SoPh, 292, 131

Zhang, J., Cheng, X., & Ding, M.-D. 2012, NatCo, 3, 747

Zhang, J., Richardson, I. G., Webb, D. F., et al. 2007, JGRA, 112, A10102

Zhao, X., & Dryer, M. 2014, SpWea, 12, 448

Zheng, R., Chen, Y., Du, G., & Li, C. 2016, ApJL, 819, L18

Zhuang, B., Wang, Y., Shen, C., et al. 2017, ApJ, 845, 117