The Astrophysical Journal, 954:151 (9pp), 2023 September 10 https://doi.org/10.3847/1538-4357/ace62d

Physics-driven Machine Learning for the Prediction of Coronal Mass Ejections' Travel Times

Sabrina Guastavino[1], Valentina Candiani[1], Alessandro Bemporad[2], Francesco Marchetti[3], Federico Benvenuto[1], Anna Maria Massone[1], Salvatore Mancuso[2], Roberto Susino[2], Daniele Telloni[2], Silvano Fineschi[2], and Michele Piana[1,2]

[1] MIDA, Dipartimento di Matematica, Università di Genova, via Dodecaneso 35 I-16146 Genova, Italy; guastavino@dima.unige.it

[2] Istituto Nazionale di Astrofisica (INAF), Osservatorio Astrofisico di Torino, Italy

[3] Dipartimento di Matematica "Tullio Levi-Civita", Università di Padova, Padova, Italy

Received 2023 May 12; revised 2023 June 30; accepted 2023 July 9; published 2023 September 1

Abstract

Coronal Mass Ejections (CMEs) correspond to dramatic expulsions of plasma and magnetic field from the solar corona into the heliosphere. CMEs are scientifically relevant because they are involved in the physical mechanisms characterizing the active Sun. However, more recently, CMEs have attracted attention for their impact on space weather, as they are correlated to geomagnetic storms and may induce the generation of solar energetic particle streams. In this space weather framework, the present paper introduces a physics-driven artificial intelligence (AI) approach to the prediction of CMEs' travel time, in which the deterministic drag-based model is exploited to improve the training phase of a cascade of two neural networks fed with both remote sensing and in situ data. This study shows that the use of physical information in the AI architecture significantly improves both the accuracy and the robustness of the travel time prediction.

Unified Astronomy Thesaurus concepts: Solar coronal mass ejections (310); Neural networks (1933)

1. Introduction

Coronal Mass Ejections (CMEs; Howard 2011) consist of large eruptions of plasma and magnetic field that are typically triggered by solar flares (Piana et al. 2022) and they can propagate from the solar corona into the heliosphere. From a phenomenological perspective, the rate of occurrence of CMEs is related to the solar cycle and typically ranges from one event per more than one week at the solar minimum to several CMEs per day at the maximum (Zhao & Dryer 2014). From an experimental perspective, the observations of CMEs are typically performed by means of remote-sensing instruments that can measure their most significant kinematic parameters, such as the initial propagation speed, the CME mass, and the initial cross section. Examples of telescopes appropriate for measuring remote sensing parameters are coronagraphs on board space clusters such as the Large Angle and Spectrometric Coronagraph (LASCO; Brueckner et al. 1995) on board the Solar and Heliophysics Observatory (SOHO; Domingo et al. 1995), the Sun Earth Connection Coronal and Heliospheric Investigation (SECCHI; Howard et al. 2008) on board STEREO-A/STEREO-B (Kaiser et al. 2008), and the recent Metis (Fineschi et al. 2012) on board Solar Orbiter (Cyr et al. 2020). Further, CMEs travel from the Sun to the Earth while embedded within the solar wind (Lazar 2012), which implies that some solar wind parameters play a significant role in the evolution of the CMEs' dynamics. In particular, the solar wind average density and speed can be inferred from measurements provided by in situ instruments such as the WIND Spacecraft (Wilson & Brosius 2021), the Advanced Composition Explorer (ACE; Stone et al. 1998), the Charge, Element, and Isotope Analysis System (CELIAS; Hovestadt et al. 1995) on board SOHO, and the Solar Wind Analyzer (SWA; Owen et al. 2020) on board Solar Orbiter.

Besides their relevance for the comprehension of the physical mechanisms involved in the active Sun, CMEs have also been attracting notable attention in the space weather context (Gopalswamy 2009; Howard 2014), and this is due to several reasons. First, fast CMEs may induce interplanetary shocks that may contribute to the generation of intense solar energetic particle (SEP) streams; second, when directed toward the Earth, CMEs are often correlated with the occurrence of geomagnetic storms when interacting with Earth's magnetosphere; finally, and most importantly, CMEs may impact the correct functioning of both space- and ground-based communication, navigation, and energy production systems.

The study of the space weather impact of CMEs implies the need for formulating, implementing, and validating forecasting approaches that must have the potential to easily become operational services for space weather monitoring and nowcasting. Specifically, the typical space weather end-user is primarily interested in rather practical issues, such as whether or not a CME that has been detected in the corona by a remote-sensing telescope will hit the Earth; and, if so, at which time (time of arrival, ToA) and speed (speed of arrival, SoA) the impact will occur. Focusing on the prediction of the ToA, these forecasting problems have been addressed in several fairly recent studies utilizing computational approaches that can be clustered into three families (see Zhao & Dryer 2014, and references therein). Empirical models (Gopalswamy et al. 2000) adopt simple equations to fit the relationship between the CME travel time (TT) and the corresponding observed parameters at the Sun; physics-based models (see Pomoell & Poedts 2018, and references therein) introduce physics to describe the CME propagation and, in the magnetohydrodynamic (MHD) versions, they can even account for the state of the background heliosphere; finally, machine/deep learning (ML/DL) approaches rely on artificial intelligence (AI) with purely data-driven algorithms to estimate the ToA given large sets of observations of the CME parameters at the Sun (Sudar et al. 2016; Liu et al. 2018; Shi et al. 2021) or CME images (Wang et al. 2019; Fu et al. 2021; Alobaid et al. 2022).

A reliable assessment of the prediction effectiveness of such methods is difficult for at least two reasons (Camporeale 2019; Vourlidas et al. 2019). First, these studies are performed using data acquired by means of different instruments and the lack of data standardization can impact the reliability of the prediction. Second, both the computational conditions under which the experiments are implemented and the ways the prediction effectiveness have been evaluated are often significantly heterogeneous. An effort to compare results obtained by different prediction models is being made by NASA's Community Coordinated Modeling Center (CCMC)[4] and the first extensive discussion of the obtained results is contained in Riley et al. (2018). Both the tables obtained in that study and the ones contained in more recent applications of AI-based techniques (see Camporeale 2019, and references therein) provide results that are significantly poor as far as both the ToA prediction accuracy and its robustness are concerned (on average, the prediction errors are almost systematically larger than 10 hr, with standard deviations that may exceed 20 hr).

In the present paper, we aim to improve these prediction estimates using an approach that combines the computational effectiveness of AI with physical information contained in deterministic models. Conceptually, such a combination can be done in two ways: either data-driven AI can be used to constrain the parameters contained in the MHD equations, or physics-based models can be exploited to better realize the training phase in supervised ML/DL processes. Following this latter path, here we used the well-established drag-based model (Cargill 2004; Vršnak et al. 2010, 2013) to design the loss functions of an architecture made of two neural networks, each one characterized by six hidden layers. Among the well-established approaches based on kinematic models, which rely on simplified assumptions on the solar wind propagation mechanisms and on the interaction between solar wind and CMEs, the drag-based model is mostly used thanks to its notable computational effectiveness and the limited number of input parameters. Specifically, the only input parameter of the model that is not provided by experimental measurements and, therefore, must be a priori estimate is the free drag parameter. The first neural network of our AI architecture is applied to estimate this parameter and is characterized by a fully model-driven loss function. Then, the second network, which has a loss function that is a weighted combination of a data-driven and a model-driven component, is used to predict the CME TT. The results we obtained by means of this physics-supported AI approach showed that (1) the accuracy with which the free drag parameter is estimated has a notable impact on the accuracy of the TT forecast; and, (2) an AI architecture that incorporates the deterministic physical model in the training process performs better than a purely data-driven algorithm in terms of both the accuracy and the robustness of the prediction.

The plan of the paper is as follows. Section 2 provides a quick overview of the drag-based model. Section 3 illustrates the neural network architecture designed for the analysis. Section 4 shows the results of our approach when applied to a limited set of LASCO observations. Our conclusions are offered in Section 5.

2. The Drag-based Model

According to drag-based models (Vršnak et al. 2010; Žic et al. 2015; Dumbović et al. 2021), the kinematics of CMEs is mainly determined by their interaction with the solar wind in which they are immersed. Specifically, the origin of the name of such approaches is that the drag acceleration (or deceleration) must follow a fluid dynamic analogy, i.e., it must have a quadratic dependence on the relative speed between the CME and the background solar wind. As a consequence, the standard drag-based model equation reads as

$$\ddot{r}(t) = -\gamma|\dot{r}(t) - w|(\dot{r}(t) - w), \tag{1}$$

where $r(t)$, $\dot{r}(t)$, and $\ddot{r}(t)$ are the position, speed, and acceleration of the CME as a function of time t, respectively; $w(r, t)$ is the solar wind speed and γ is the drag parameter, which measures the interaction effectiveness between the CME and the solar wind and it can be expressed as

$$\gamma = C\frac{A\rho}{m}, \tag{2}$$

where A and m are the CME impact area and mass, respectively; $\rho(r, t)$ is the solar wind density; C is the dimensionless drag coefficient. Equation (1) is completed to a Cauchy problem by including the two initial conditions $r(t_0) = r_0$ and $\dot{r}(t_0) = v_0$, where r_0 is the height of the eruption ballistic propagation, and v_0 is the initial CME speed. Both r_0 and v_0 should be considered known and provided as experimental measurements. The physical limitations of the drag-based model have been fully described in Vršnak et al. (2013). In particular, the model considers a simplified structure for the background solar wind, i.e., it is assumed that all parts of the CME are embedded in an isotropic flow, where the flow speed does not change with distance.

A typical application of the drag-based model is the estimate of the CME ToA given estimates and measurements of the model parameters. In fact, assuming that the solar wind speed and the drag parameter are constant and homogeneous, Equation (1) leads to

$$\dot{r}(t) = \frac{v_0 - w}{1 + \gamma\ \mathrm{sign}(v_0 - w)(v_0 - w)t} + w, \tag{3}$$

and

$$r(t) = \frac{1}{\gamma}\mathrm{sign}(v_0 - w)\log(1 + \gamma\mathrm{sign}(v_0 - w)(v_0 - w)t) + wt + r_0. \tag{4}$$

Equation (4) can be used to estimate the TT as the solution of $r(t) = 1$ au. Once the TT is estimated, it can be included in Equation (3) to obtain an estimate of the SoA. If we substitute γ with Equation (2), this approach is reliable just if accurate estimates of the parameters A, m, C, ρ, w, r_0, and v_0 are at disposal. Specifically, measurements of A, m, r_0, and v_0 can be provided by coronagraphic instruments such as LASCO or, more recently, Metis on board Solar Orbiter. In situ instruments such as WIND, ACE, and CELIAS can provide estimates of ρ and w at the CME onset, and these same values can be used in the equations as approximations of average values of the two parameters. The determination of C is particularly critical. A

[4] https://kauai.ccmc.gsfc.nasa.gov/CMEscoreboard/

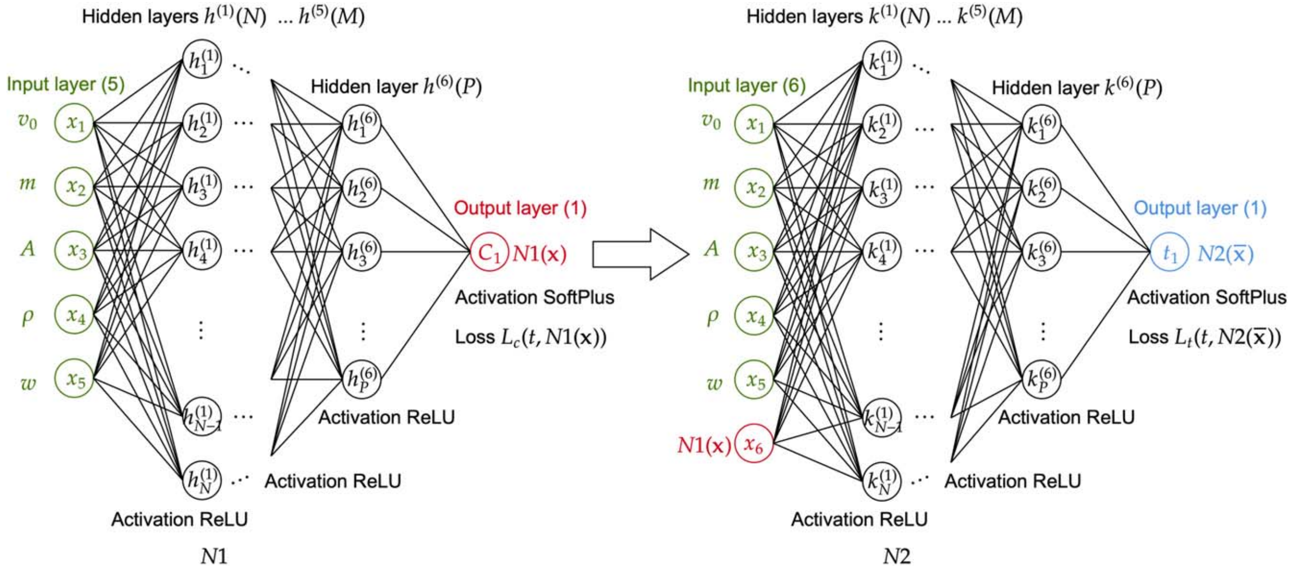

Hidden layers $h^{(1)}(N)$ ... $h^{(5)}(M)$
Input layer (5)
v_0 x_1
m x_2
A x_3
ρ x_4
w x_5
$h_1^{(1)}$ $h_2^{(1)}$ $h_3^{(1)}$ $h_4^{(1)}$ $h_{N-1}^{(1)}$ $h_N^{(1)}$
Hidden layer $h^{(6)}(P)$
$h_1^{(6)}$ $h_2^{(6)}$ $h_3^{(6)}$ $h_P^{(6)}$
Output layer (1)
C_1 $N1(\mathbf{x})$
Activation SoftPlus
Loss $L_c(t, N1(\mathbf{x}))$
Activation ReLU
Activation ReLU
Activation ReLU
N1
Hidden layers $k^{(1)}(N)$ ... $k^{(5)}(M)$
Input layer (6)
v_0 x_1
m x_2
A x_3
ρ x_4
w x_5
$N1(\mathbf{x})$ x_6
$k_1^{(1)}$ $k_2^{(1)}$ $k_3^{(1)}$ $k_4^{(1)}$ $k_{N-1}^{(1)}$ $k_N^{(1)}$
Hidden layer $k^{(6)}(P)$
$k_1^{(6)}$ $k_2^{(6)}$ $k_3^{(6)}$ $k_P^{(6)}$
Output layer (1)
t_1 $N2(\overline{\mathbf{x}})$
Activation SoftPlus
Loss $L_t(t, N2(\overline{\mathbf{x}}))$
Activation ReLU
Activation ReLU
Activation ReLU
N2

Figure 1. Neural network architectures in cascade: The first network $N1$ can be used to estimate the drag parameter C from CME data. The second network $N2$ is then employed to estimate the ToA of the considered CME at 1 au. Here, $N = 200$, $M = 25$, and $P = 10$ are the number of neurons in the first, fifth, and sixth hidden layers, respectively.

Table 1
Network Settings for Both Neural Network Architectures

Layer	Input	Hidden 1	Hidden 2	Hidden 3	Hidden 4	Hidden 5	Hidden 6	Output
Nodes	5/6	200	100	50	30	25	10	1
Activation function	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	SoftPlus

Notes. The only difference lies in the number of values in the input layer: five for the first network N1 and five or six for the second network $N2$, depending on the considered configuration (see Table 2). The activation function is the rectified linear unit (ReLU) $f(x) = \max(0, x)$ for the input and the hidden layers, and the SoftPlus function $f(x) = \log(1 + \exp(x))$ for the output.

possible procedure is given in Napoletano et al. (2018) and another approach is described below, together with a discussion of the impact of the accuracy of this estimate on the determination of the TT.

3. Neural Networks Architecture

We propose an AI-based, physics-supported approach to the estimate of the CME TT that exploits the use of two neural networks in cascade (see Figure 1). The two algorithms have the same design, whose parameters are contained in Table 1. The first network ($N1$) can take as input measurements of the initial CME speed (v_0), the CME mass (m) and impact area (A), together with estimates of the solar wind density (ρ) and speed (w). The output of this first network is an estimate of the drag parameter C. Therefore, the second neural network ($N2$) can take as input the same parameters as the first network plus this estimate of C and forecasts the corresponding TT (we point out that the CME ToA is easily determined from the corresponding TT since the time of occurrence of the CME at onset is known from observations).

3.1. Loss Functions

The training phase for the two networks relies on choices of the loss functions that, programmatically, want to incorporate physical information encoded into the drag-based model. To accomplish this objective, the loss function must be differentiable. Therefore, the sign function at the denominator of Equation (3) is approximated as

$$\mathrm{sign}(v_0 - w) \approx \frac{(v_0 - w)}{\sqrt{(v_0 - w)^2 + \delta}}, \tag{5}$$

where δ is a small positive value. By incorporating approximation (5) into (3), by analytically integrating the resulting form for $\dot{r}(t)$, and by substituting γ with Equation (2), we obtained

$$r(t, C) = \frac{1}{\frac{A}{m}C\rho\sigma} \log\left(1 + \frac{A}{m}C\rho\sigma(v_0 - w)t\right) + wt + r_0, \tag{6}$$

where

$$\sigma = \frac{(v_0 - w)}{\sqrt{(v_0 - w)^2 + \delta}}. \tag{7}$$

The first network $N1$ estimates the drag parameter C by utilizing the fully model-inspired quadratic loss function:

$$\begin{aligned} L_c(t, N1(\mathbf{x})) &= (r(t, N1(\mathbf{x})) - 1)^2 \\ &= \left(\frac{m\sigma}{A\rho N1(\mathbf{x})} \log\left(1 + \sigma\frac{A\rho N1(\mathbf{x})}{m}(v - w)t + wt\right)\right. \\ &\quad \left. + r_0 - 1\right)^2, \end{aligned} \tag{8}$$

where $\mathbf{x} = (v_0, m, A, \rho, w)$ is the input vector and the CME position is measured in astronomical units. We point out that,

Table 2
In the First Three Configurations the Drag Parameter Is Not Considered as an Input of the Second Neural Network and It Is Estimated by the First Network Only When Needed in the Loss Function

Configuration	Training Phase			Testing Phase
	$N1$	$N2$	λ	Drag Parameter as Input of $N2$
C1	off	on	1	off
C2	on	on	0.5	off
C3	on	on	0	off
C4	on	on	1	on
C5	on	on	0.5	on
C6	on	on	0	on

Note. The choice of the parameter $\lambda = 1$ corresponds to the fully data-driven case, whereas $\lambda = 0.5$ and $\lambda = 0$ represent the mixed and the fully model-driven case, respectively.

in this first case, observational values are used for t, but the estimate of the drag parameter provided by the network $N1$ is not intended as explicitly depending on this time value ($N1$ depends just on the five input parameters). In this way, once the network is trained, it is able to forecast C without the need for any knowledge of t, which is the actual unknown quantity to predict at the end of the whole neural network cascade.

The second network $N2$ predicts the TT and, in this case, the loss function is a weighted sum of a fully data-driven quadratic function and of a fully model-driven component, i.e.,

$$\begin{aligned} L_t(t, N2(\bar{\mathbf{x}})) &= \lambda(t - N2(\bar{\mathbf{x}}))^2 + (1 - \lambda)(r(N2(\bar{\mathbf{x}}), N1(\mathbf{x})) - 1)^2 \\ &= \lambda(t - N2(\bar{\mathbf{x}}))^2 + (1 - \lambda)\left(\frac{m\sigma}{A\rho N1(\mathbf{x})}\right. \\ &\quad \left. \times \log\left(1 + \sigma\frac{A\rho N1(\mathbf{x})}{m}\right)(v - w)N2(\bar{\mathbf{x}}) + vN2(\bar{\mathbf{x}}) + r_0 - 1\right)^2, \end{aligned} \tag{9}$$

where $N1(\mathbf{x})$ is the value predicted by the first neural network, $N2(\bar{\mathbf{x}})$ is the predicted TT, and $\bar{\mathbf{x}}$ is the input vector for $N2$, which may or may not contain the estimate of C provided by $N1$.

3.2. Cascade Configurations

The design in Figure 1 is flexible enough to allow the TT prediction according to the six possible configurations described in Table 2. The differences characterizing these configurations depend on the role played by the drag-based model in the training phase of $N2$, and by the fact that the drag parameter C either may or may not be used as an input feature. Specifically, in Configuration 1 (C1), Configuration 2 (C2), and Configuration 3 (C3), the input of the second network is $\bar{\mathbf{x}} = \mathbf{x}$, i.e., the drag parameter is not used as an input feature. In C1, $N1$ is switched off, while the loss function in $N2$ is fully data-driven (i.e., $\lambda = 1$). In configurations C2 and C3, the first network is switched on and estimates of C are used in the loss function of $N2$: indeed, in these two latter configurations, the loss function is represented by the weighted sum of a fully data-driven and a fully physics-driven component (i.e., $\lambda = 0.5$), and is fully physics-driven (i.e., $\lambda = 0$), respectively. However, in C2 and C3, the second network does not use C as an input feature. In the remaining three configurations, the first network is always switched on and the drag parameter C is always utilized as a feature of the second network, with values provided as predictions by the first network. In particular, in Configuration 4 (C4), Configuration 5 (C5), and Configuration 6 (C6), the loss function of $N2$ is fully data-driven (i.e., $\lambda = 1$ in $N2$), has both the data- and physics-driven components (i.e., $\lambda = 0.5$), and is fully physics-driven, respectively.

3.3. Optimization

The two networks have been trained over 10,000 epochs using the Adam optimizer and a learning rate equal to 10^{-3} (Kingma & Ba 2015). The validation of the networks has been performed by using an early stopping rule based on the monitoring of the loss function in the validation phase (Caruana et al. 2000). Specifically, we stop the iterations when the loss function during validation does not improve after 2000 epochs. The two networks are characterized by six hidden layers, and we initialized the network weights with a random uniform distribution in the range (0, 0.01). Details of the two networks are provided in Table 2. The design of the architecture and the choice of the hyperparameters are the result of an empirical trial-and-error optimization process carried out in several experiments.

4. Applications

4.1. Data Set

We have performed the validation of this approach to TT forecasting by means of a set of 123 events observed by LASCO (Napoletano et al. 2022) and CELIAS. The events occurred in the time range between 1997 and 2018 (Richardson & Cane 2010, hereafter dubbed R and C). Specifically, LASCO provided all of the CME parameters at the onset, while CELIAS provided the solar wind density and speed at the onset (see Table 3). These observed values have been used as ρ and w in the loss functions, i.e., we assumed the homogeneity and the stationarity of these solar wind parameters. A preprocessing step was necessary to filter out the events that could not be explained by the drag-based model. More specifically, we excluded from the initial data set of 160 events 37 events for which the speed condition was not satisfied, i.e., the mean CME speed does not lie in the range between the initial CME speed and the average solar wind speed and vice versa; i.e., we considered just events such that

$$v_0 \leqslant \bar{v} \leqslant w \quad \text{or} \quad w \leqslant \bar{v} \leqslant v_0, \tag{10}$$

where $\bar{v} = 1\ \text{au}/\text{TT}$ is the mean CME speed. Moreover, prior to the network training, the data were normalized so that the considered quantities would be comparable, and this has been done separately each time on the training, validation, and test set.

4.2. Estimate of the Drag Parameter

An accurate estimate of the drag parameter C is crucial for an effective prediction of the TT. Such an estimate can be performed according to the following three approaches:

1. Following Napoletano et al. (2022), an analytic inversion of Equation (4) for each observed event can be computed.
2. One can exploit the first neural network in the cascade described in the previous section by using a subset of the whole data archive as a training set.

Table 3
CMEs Data Set Used in This Work, Where r_0 Is a Fixed Parameter and the Travel Time Is the Final Network Output and Quantity of Interest

Name	Notation	Unity	Description	Source
CME height of eruption	r_0	km	$r_0 = 20\ R_\odot$, $R_\odot = 6.957 \cdot 10^5$ km	⋯
CME time of eruption	t_0	s	eruption time on the Sun at r_0	(Napoletano et al. 2022)
CME time of arrival	ToA	s	estimated arrival time at 1 au	R and C
CME travel time	TT	s	estimated time between t_0 and ToA	R and C, (Napoletano et al. 2022)
CME initial speed	v_0	km s^{-1}	initial propagation speed from eruption	LASCO
CME mass	m	g	estimated CME mass	LASCO
CME impact area	A	km^2	CME impact area, constant angular width	LASCO
Solar wind density	ρ	g km^{-3}	mean over one hour after t_0	CELIAS
Solar wind speed	w	km s^{-1}	mean over one hour after t_0	CELIAS
Drag parameter	C	dimensionless	parameter of the drag-based model	This work

Note. The last six quantities are the network's possible input features.

Table 4
Minimum (min), Mean, Median, and Maximum (max) Values of the MAEs on the 100 Realizations of the Test Sets and Corresponding Relative Absolute Errors with Respect to the Observed TT

Loss Function	Drag Parameter as Input	Configuration	MAE (h)				Relative Absolute Error			
			min	median	mean	max	min	median	mean	max
Fully data-driven	off	C1	6.1	10.43	11.93	49.5	0.1	0.16	0.2	1.32
	on	C4	**4.8**	9.96	10.48	36.09	**0.07**	0.15	0.17	0.9
Mix	off	C2	5.89	10.03	10.23	25.29	0.08	0.15	0.16	0.6
	on	C5	5.74	**9.46**	**9.64**	**13.75**	0.08	**0.14**	**0.15**	**0.24**
Fully physics-driven	off	C3	5.76	10.28	10.67	29.63	0.08	0.15	0.17	0.66
	on	C6	5.27	9.59	10.04	28.45	0.08	0.14	0.16	0.72

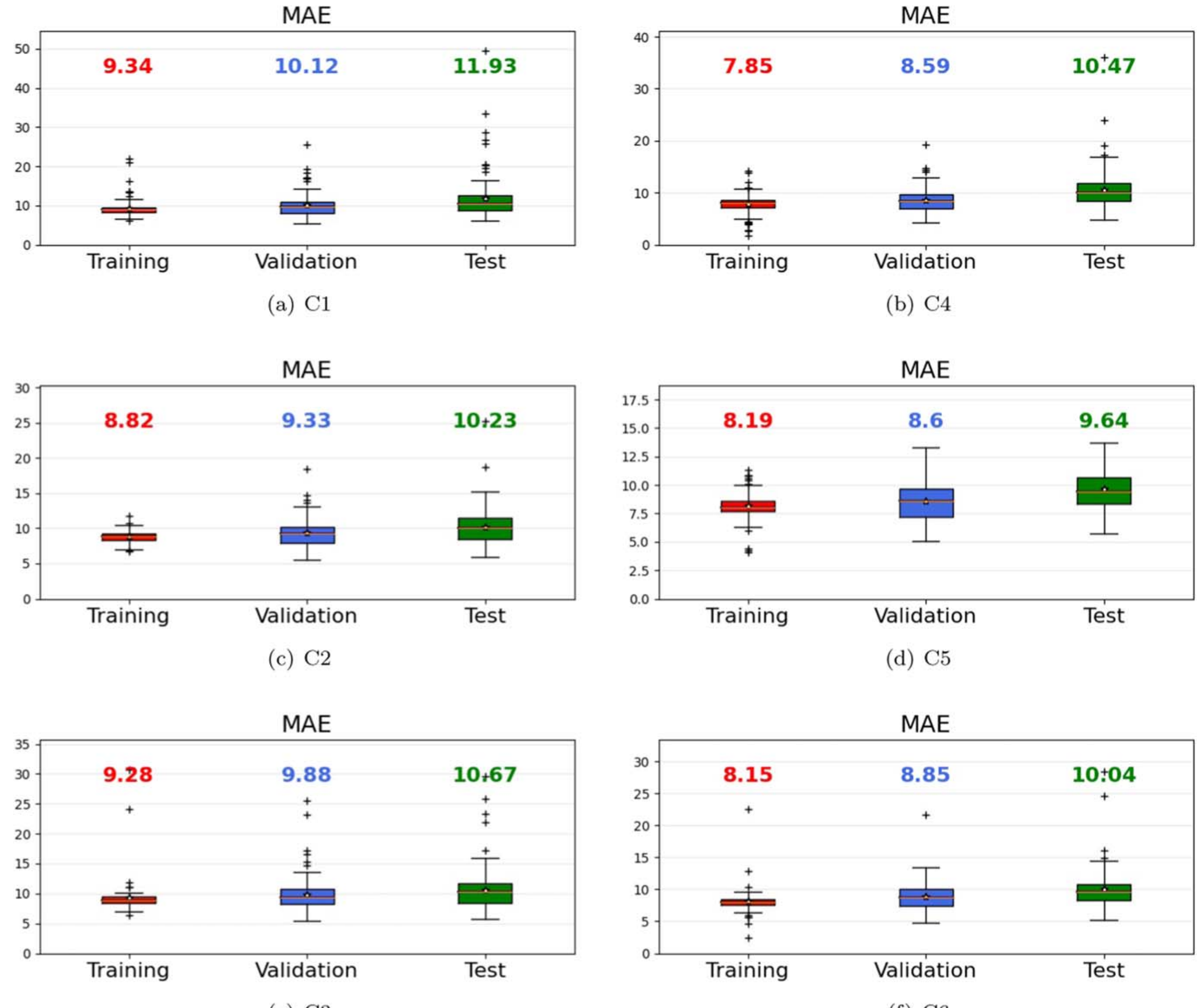
Notes. The overall best results are in boldface. Keeping the loss function fixed, we underline the best results between the two related configurations.

3. As in the previous item, one can apply $N1$ but, this time, the training phase is performed using the whole data set at one's disposal.

In order to validate the accuracy of the outcome of these three approaches, for each estimated C value we computed $r(C)$ a posteriori, as in Equation (4), and compared the result with respect to 1 au (specifically, we required $0.95 < r(C) < 1.05$ au). We found that, using approach 1, i.e., taking the values from Napoletano et al. (2022), the success rate for this condition is 19.78%; using approach 2, it is between 70% and 80% (by varying the subset of the whole data archive as a training set); and, finally, using approach 3, it is above 90%. This result is not surprising since the training process explicitly minimizes the discrepancy between $r(t,\ C)$ and 1 au. Understanding the reason why the network is not able to generalize on the test set still represents an open issue. This problem, which should be addressed in a separate study, is most likely related to the size of the archive at one's disposal and to the correct balance of the training set (Guastavino et al. 2022a).

4.3. TT Prediction

In order to perform a statistical assessment of this physics-driven AI approach to ToA prediction, we realized 100 random realizations of the training, validation, and test sets using the 123 events in our archive made of LASCO and CELIAS observations. Once the network cascade has been trained, we assessed the prediction performances of the six cascade configurations for each element in the training, validation, and test sets by comparing the prediction outcomes with the experimental TT values. As a result, Figure 2 contains the box plots corresponding to the 100 computed absolute errors and, for each box plot, the corresponding mean absolute error (MAE). Furthermore, for each realization of the training set, we have computed the impact on predictions of each feature according to permutation importance. Permutation importance is computed after a model has been fitted, and it is commonly used to assess how the accuracy of predictions is affected if a single column of the validation data is randomly shuffled (Breiman 2001). Model accuracy is more negatively affected if one shuffles a column that the model heavily relied on for predictions. The results are described in the bar plots shown in Figure 3: for each realization, we made a ranking of feature importance and we computed how many times each feature is present in the top three ranking. Then, in Table 4 we reported the minimum (min), mean, median, and maximum (max) values of the MAEs on the 100 realizations of the test sets. Figure 4 contains the distributions of the absolute errors between the predicted TT and the experimental TT values on all of the test samples. The first three bins are 4 hr wide, the fourth bin is between 12 hr and 15 hr, the fifth and sixth bins are 15 hr–20 hr and 20 hr–25 hr, and the other bins are 10 hr wide. In the same panels, we report the standard deviation (STD) values of such distribution errors. Figure 5 contains scatterplots of the actual and predicted TT values corresponding to the realization of the test set characterized by the minimum MAE. Blue dots represent the different CME events in the selected test set and the black dashed line represents a



(a) C1

(b) C4

(c) C2

(d) C5

(e) C3

(f) C6

Figure 2. Distributions of mean absolute errors for the prediction of TT on 100 realizations of training (red box plot), validation (blue box plot), and the test (green box plot) sets. The numbers in figures represent the mean value of the distributions of MAEs. First row: results with configurations C1 and C4. Second row: results with configurations C2 and C5. Third row: results with configurations C2 and C5.

perfect prediction, i.e., when the predicted TT matches the actual TT values.

5. Comments and Conclusions

The objective of this study was to understand whether and to what extent the use of deterministic information allows machine learning to improve its effectiveness in the forecasting of the CME TT given a very limited number of experimental features at our disposal. To this aim, we have exploited the drag-based model as the source of such deterministic information, essentially for the reason why its integrated form can be easily encoded as the loss function applied in the training phase of the AI data-driven approach. The need to optimize the drag parameter in the model inspired a complex architecture made of two neural networks, the first one used to estimate the parameter, and the second one used to predict the CME TT. The possibility to use (or not use) the drag parameter as an input feature for this second network allowed the definition of six configurations for the cascade and the results

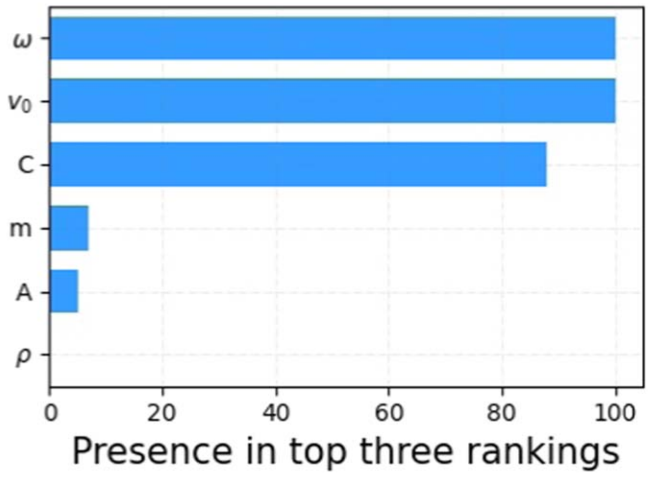
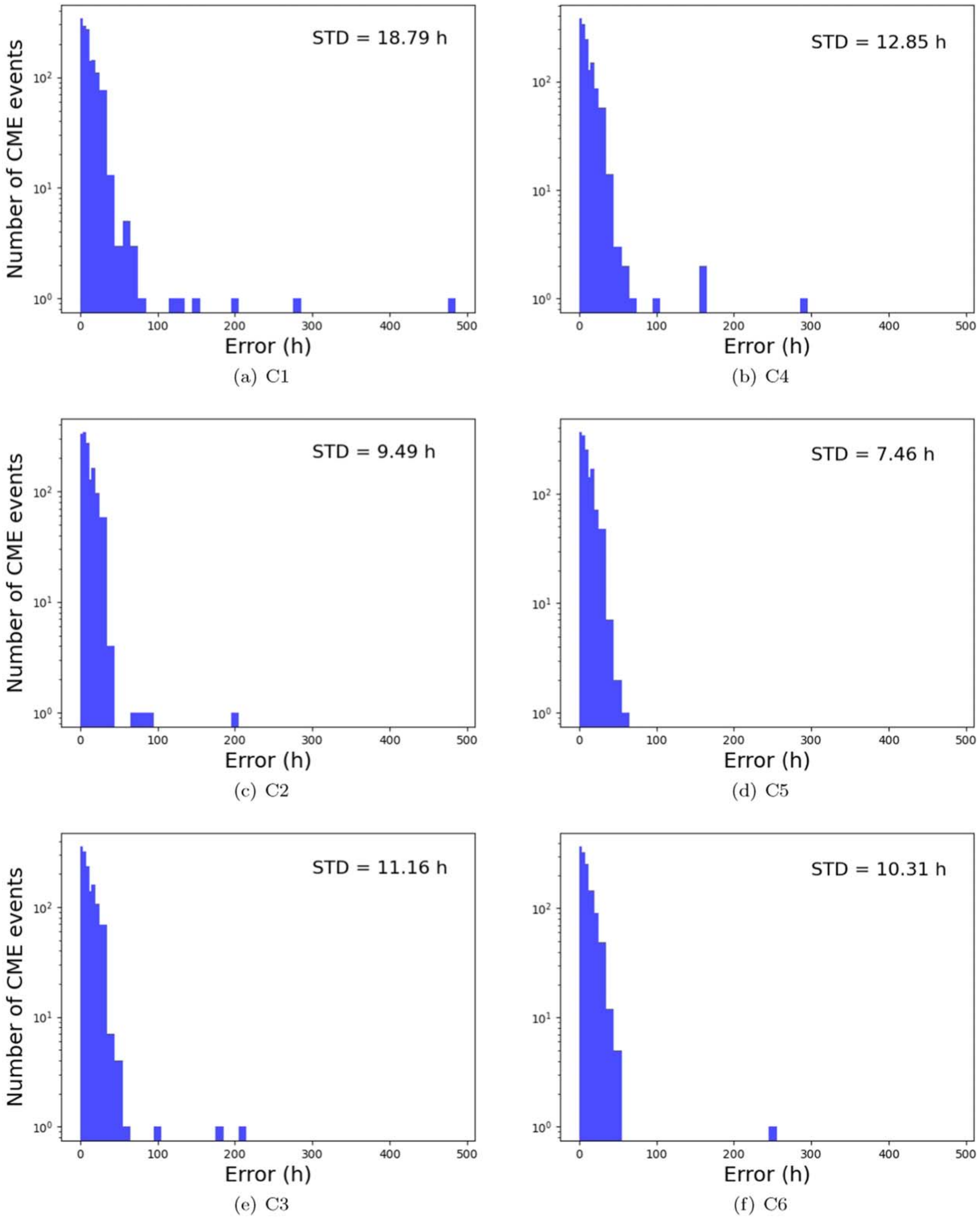


Figure 3. Impact of features on network predictions according to permutation importance. The most important features are the wind speed, the CME speed, and the drag parameter *C*.



(a) C1

(b) C4

(c) C2

(d) C5

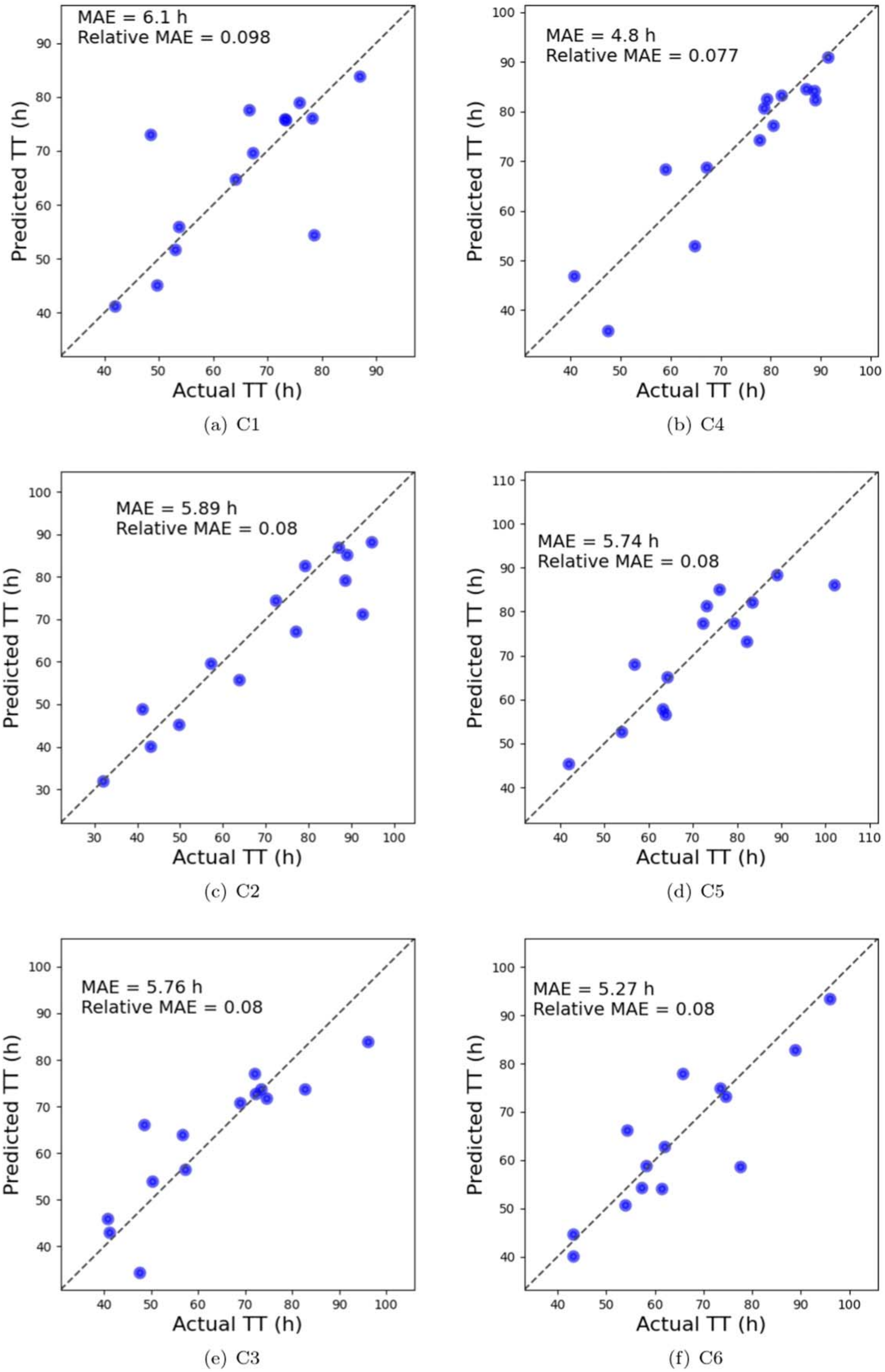
(e) C3

(f) C6

Figure 4. Distribution of absolute errors for the prediction of TT for all samples in test sets for each configuration. First row: results with configurations C1 and C4. Second row: results with configurations C2 and C5. Third row: results with configurations C2 and C5.

in the previous section now allow a comparison of their reliability and robustness.

We first point out that C1 is the only fully data-driven configuration, and it is the one characterized by the worst predicted TT values in the training, validation, and test sets. Further, Table 4 and Figure 4 show that this configuration is characterized by several significant outliers. The drag-based model plays an active role in all other configurations, and this increases both the prediction accuracy and its robustness. In particular, the best results are obtained when the loss function has both data- and physics-driven components and *C* is considered as an input feature for the second network (the best overall performance is highlighted in bold in Table 4). In general, for each configuration adding *C* as an input feature improves the predictions. Figure 5 shows this behavior in a very clear-cut fashion: the scatterplots in the second column of



(a) C1

(b) C4

(c) C2

(d) C5

(e) C3

(f) C6

Figure 5. Predicted transit time vs. actual transit time for CMEs in the test set in which the method reached the best performance. First row: results for configurations C1 and C4. Second row: results for configurations C2 and C5. Third row: results for configurations C3 and C6.

the figure, which correspond to configurations C4, C5, and C6, respectively, present MAE values that are among the smallest ones one can find in the scientific literature as far as our knowledge is concerned; the same holds for the absolute errors corresponding to the single predictions. This is confirmed also in Table 4, in which, keeping the loss function fixed, we underlined the lowest min, mean, median, and max values of MAE and relative absolute errors provided when using or not the drag parameter as input. Furthermore, Figure 4 highlights that the predictions are more robust when the drag parameter is used as an additional feature: each panel on the right-hand side displays a smaller number of outliers with respect to its corresponding panel on the left-hand side, as assessed also by a smaller standard deviation value. However, when the physics component is present in the loss function, good results are obtained also when C is not used as an additional input feature: from an operational viewpoint when a new input arrives, we can use the second trained network without the need to estimate C for that event.

As said, in configurations C4, C5, and C6, the drag parameter C is used as an input feature. The feature importance analysis in Figure 3 shows that, in these configurations, the drag parameter is among the features that mostly impact the prediction, coherently with the fact that an accurate estimate of C leads to accurate predictions. When C is estimated by $N1$ using the whole archive as a training set, the corresponding values of $\gamma = \frac{CA\rho}{m}$ are almost all in the range $10^{-8}\,\mathrm{km}^{-1} < \gamma < 10^{-6}\,\mathrm{km}^{-1}$ observed in Napoletano et al. (2022).

Our work-in-progress concerning CME prediction and characterization is currently in three directions. First, we want to generalize this approach to a multitarget version able to provide physics-supported forecasting of both the ToA and the SoA. Second, following the approach proposed in Guastavino et al. (2022a, 2023), we aim at generating balanced training, validation, and test sets in order to account for the data types present in the mission archives. Third, at a more technical level, we want to update the training phase of the cascade by using probabilistic loss functions designed to optimize specific skill scores (Guastavino et al. 2022b; Marchetti et al. 2022). Moreover, in the future developments of this work, we will investigate possible modifications of the drag-based model applied here to include other possible physical phenomena occurring during the ICME propagation, such as the effect of plasma pile-up due to the interaction of the CME flux rope with the surrounding solar wind plasma (an effect that can be taken into account by introducing the so-called "virtual mass"; see, e.g., Cargill 2004, and observed, e.g., by Telloni et al. 2021), and the occurrence of magnetic reconnections with the background interplanetary magnetic field (an effect leading to the so-called "magnetic erosion" process, see, e.g., Wang et al. 2018 and Telloni et al. 2020).

Acknowledgments

S.G. and F.M. were supported by the Programma Operativo Nazionale (PON) "Ricerca e Innovazione" 2014–2020; V.C. by the Università di Genova within the *Bando per l'incentivazione alla progettazione europea (BIPE)*—Mission 1 "Promoting Competitiveness" 2020. V.C. was also partially supported by INdAM-GNCS. A.M.M., R.S., and D.T. acknowledge the support of the Fondazione Compagnia di San Paolo within the framework of the Artificial Intelligence Call for Proposals, AIxtreme project (ID Rol: 71708). A.M.M. is also grateful to the HORIZON Europe ARCAFF Project, grant No. 101082164.

ORCID iDs

Sabrina Guastavino https://orcid.org/0000-0001-7047-1148
Alessandro Bemporad https://orcid.org/0000-0001-5796-5653
Federico Benvenuto https://orcid.org/0000-0002-4776-0256
Salvatore Mancuso https://orcid.org/0000-0002-9874-2234
Roberto Susino https://orcid.org/0000-0002-1017-7163
Daniele Telloni https://orcid.org/0000-0002-6710-8142
Michele Piana https://orcid.org/0000-0003-1700-991X

References

Alobaid, K. A., Abduallah, Y., Wang, J. T. L., et al. 2022, FrASS, 9, 1013345
Breiman, L. 2001, Mach. Learn., 45, 5
Brueckner, G. E., Howard, R. A., Koomen, M. J., et al. 1995, SoPh, 162, 357
Camporeale, E. 2019, SpWea, 17, 1166
Cargill, P. J. 2004, SoPh, 221, 135
Cyr, O. S., Marsden, R., Nieves-Chinchilla, T., et al. 2020, A&A, 642, A1
Caruana, R., Lawrence, S., & Giles, C. 2000, in Advances in Neural Information Processing Systems 13, ed. T. Leen, T. Dietterich, & V. Tresp (Cambridge, MA: MIT Press)
Domingo, V., Fleck, B., & Poland, A. I. 1995, SoPh, 162, 1
Dumbović, M., Čalogović, J., Martinić, K., et al. 2021, FrASS, 8, 58
Fineschi, S., Antonucci, E., Naletto, G., et al. 2012, Proc. SPIE, 8443, 84433H
Fu, H., Zheng, Y., Ye, Y., et al. 2021, RemS, 13, 1738
Gopalswamy, N. 2009, in Climate and Weather of the Sun-Earth System (CAWSES): Selected Papers from the 2007 Kyoto Symp., ed. T. Tsuda et al. (Tokyo: TERRAPUB), 77
Gopalswamy, N., Lara, A., Lepping, R., et al. 2000, GeoRL, 27, 145
Guastavino, S., Marchetti, F., Benvenuto, F., Campi, C., & Piana, M. 2022a, A&A, 662, A105
Guastavino, S., Marchetti, F., Benvenuto, F., Campi, C., & Piana, M. 2023, FrASS, 9, 399
Guastavino, S., Piana, M., & Benvenuto, F. 2022b, IEEE Trans. Neural Networks Learn. Systems (Piscataway, NJ: IEEE)
Hovestadt, D., Hilchenbach, M., Bürgi, A., et al. 1995, SoPh, 162, 441
Howard, R. A., Moses, J. D., Vourlidas, A., et al. 2008, SSRv, 136, 67
Howard, T. 2011, Coronal Mass Ejections: An Introduction, Vol. 376 (Berlin: Springer)
Howard, T. 2014, Space Weather and Coronal Mass Ejections (Berlin: Springer)
Kaiser, M. L., Kucera, T. A., Davila, J. M., et al. 2008, SSRv, 136, 5
Kingma, D. P., & Ba, J. 2015, in 3rd Int. Conf. on Learning Representations, ICLR 2015, ed. Y. Bengio & Y. LeCun (Trier: Computer Science Bibliography), https://dblp.org/db/conf/iclr/iclr2015.html
Lazar, M. 2012, Exploring the Solar Wind (Rijeka: InTech)
Liu, J., Ye, Y., Shen, C., Wang, Y., & Erdélyi, R. 2018, ApJ, 855, 109
Marchetti, F., Guastavino, S., Piana, M., & Campi, C. 2022, PatRe, 132, 108913
Napoletano, G., Foldes, R., Camporeale, E., et al. 2022, SpWea, 20, e2021SW002925
Napoletano, G., Forte, R., Del Moro, D., et al. 2018, JSWSC, 8, A11
Owen, C., Bruno, R., Livi, S., et al. 2020, A&A, 642, A16
Piana, M., Emslie, A. G., Massone, A. M., & Dennis, B. R. 2022, Hard X-Ray Imaging of Solar Flares, Vol. 164 (Berlin: Springer)
Pomoell, J., & Poedts, S. 2018, JSWSC, 8, A35
Richardson, I. G., & Cane, H. V. 2010, SoPh, 264, 189
Riley, P., Mays, M. L., Andries, J., et al. 2018, SpWea, 16, 1245
Shi, Y.-R., Chen, Y.-H., Liu, S.-Q., et al. 2021, RAA, 21, 190
Stone, E. C., Frandsen, A. M., Mewaldt, R. A., et al. 1998, SSRv, 86, 1
Sudar, D., Vršnak, B., & Dumbović, M. 2016, MNRAS, 456, 1542
Telloni, D., Scolini, C., Möstl, C., et al. 2021, A&A, 656, A5
Telloni, D., Zhao, L., Zank, G. P., et al. 2020, ApJL, 905, L12
Vourlidas, A., Patsourakos, S., & Savani, N. P. 2019, RSPTA, 377, 20180096
Vršnak, B., Žic, T., Falkenberg, T. V., et al. 2010, A&A, 512, A43
Vršnak, B., Žic, T., Vrbanec, D., et al. 2013, SoPh, 285, 295
Wang, Y., Liu, J., Jiang, Y., & Erdélyi, R. 2019, ApJ, 881, 15
Wang, Y., Shen, C., Liu, R., et al. 2018, JGRA, 123, 3238
Wilson, Lynn B. I., Brosius, A. L., & Gopalswamy, N. 2021, RvGeo, 59, e2020RG000714
Zhao, X., & Dryer, M. 2014, SpWea, 12, 448
Žic, T., Vršnak, B., & Temmer, M. 2015, ApJS, 218, 32