Predicting coronal mass ejections' travel times by using physics-informed loss functions

Francesco Marchetti
francesco.marchetti@math.unipd.it
Dipartimento di Matematica "Tullio Levi-Civita"
PINN-PAD, Padova, 23/02/24

The collaborators

UniGe – MIDA group – HELCOMP lab

- Sabrina Guastavino
- Valentina Candiani
- Federico Benvenuto
- Anna Maria Massone
- Michele Piana

INAF – OATo

- Alessandro Bemporad
- Salvatore Mancuso
- Roberto Susino
- Daniele Telloni

Physics-driven Machine Learning for the Prediction of Coronal Mass Ejections' Travel Times, Sabrina Guastavino et al - 2023 ApJ 954 151

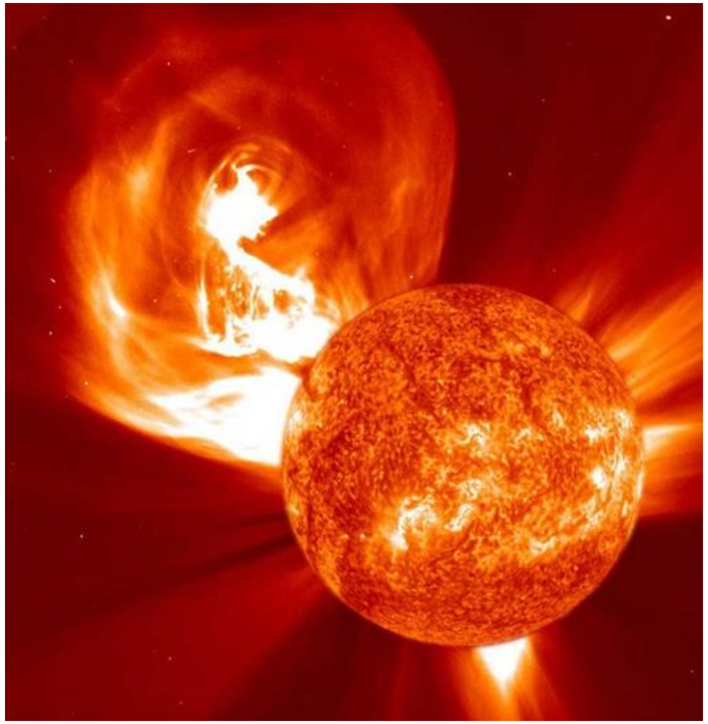
Outline

1. CMEs in the space weather context
2. Our physics-informed deep learning approach
3. Results
4. Comments and conclusions

Coronal Mass Ejections

Coronal Mass Ejections (CMEs) consist of large eruptions of plasma that are typically triggered by solar flares and they can propagate through the solar wind from the solar corona into the heliosphere.

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**) on board the Solar and Heliophysics Observatory (SOHO).



We are interested in predicting the travel time of interplanetary CMEs

The drag-based model

Drag-based model

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

$\ddot{r}$ = CME acceleration
$\dot{r}$ = CME speed
w = solar wind speed

Drag parameter

$$\gamma = C\frac{A\rho}{m}$$

ρ = solar wind density
A = CME impact area
m = CME mass
C = drag coefficient (unknown)

Drag Equation is completed to a Cauchy problem by including the two initial conditions

$$\mathrm{r}(t_0) = r_0$$

$$\dot{r}(t_0) = v_0$$

where r_0 is the height of the eruption ballistic propagation, and v_0 is the initial CME speed.

The drag-based model

Assuming that the solar wind speed and the drag parameter are constant and homogeneous, the drag equation leads to

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

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

This equation can be used to estimate the travel time as the solution of *r(t)* = 1 AU, if accurate estimates of the parameters are at disposal.

Data driven and physics-based losses

To use *r(t)* in the construction of a loss function, we adopt the approximation

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

Then, we can consider a loss function of the form

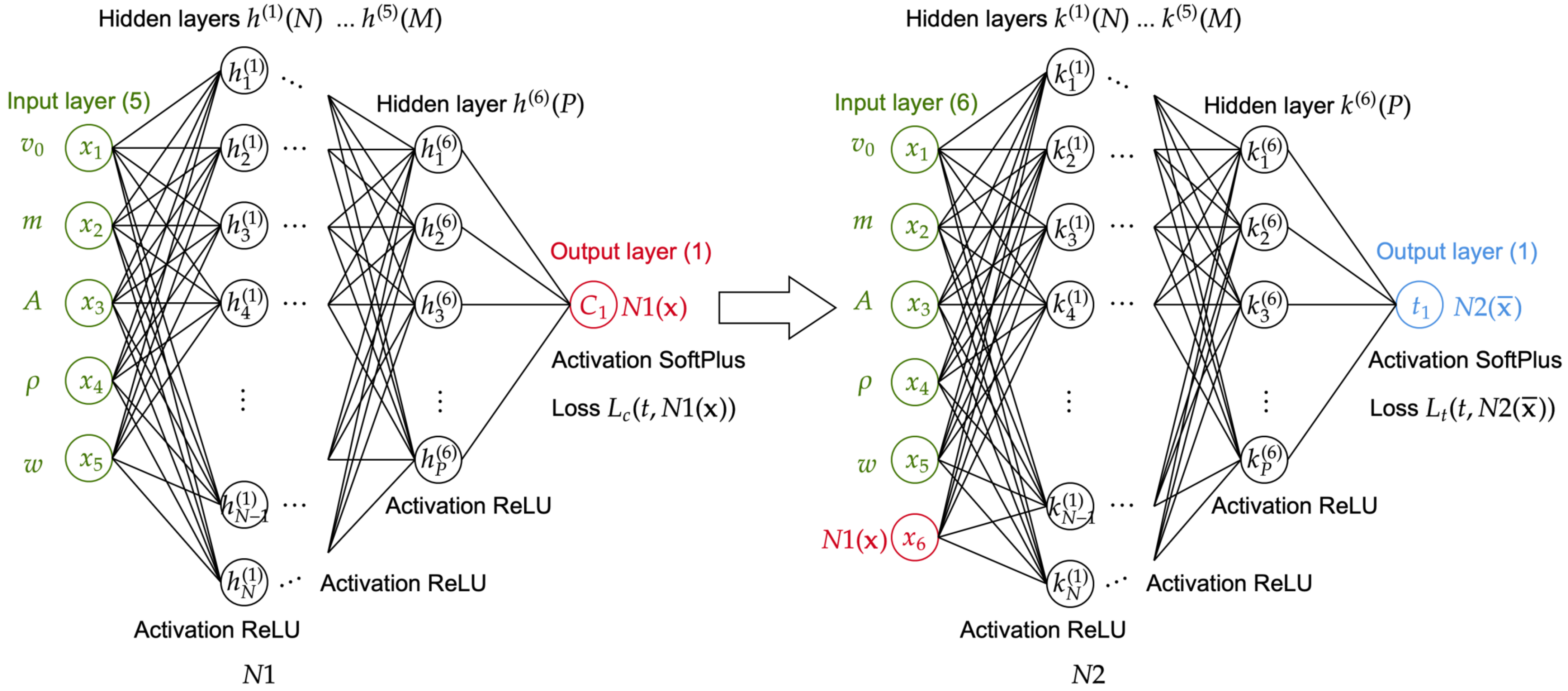
Data-driven term — Physics-driven term

$$L_t(t, f(w,x)) = \lambda \boxed{(t - f(w,x))^2} + (1-\lambda) \boxed{(1 - r(f(w,x), C))^2}$$

λ=1 → only data-driven term → Fully data-driven
λ=0 → only physics-driven term → Fully physics-driven
λ ∈ (0,1) (e.g. λ =0.5) → both terms → Mix

However... we need to estimate C !

Our approach: architectures



$$L_C(t, N1(x)) = (1 - r(t, N1(x)))^2$$

$$L_t(t, N2(\bar{x})) = \lambda\,(t - N2(\bar{x}))^2 + (1-\lambda)(1 - r(N2(\bar{x}), N1(x)))^2$$

The dataset of CMEs

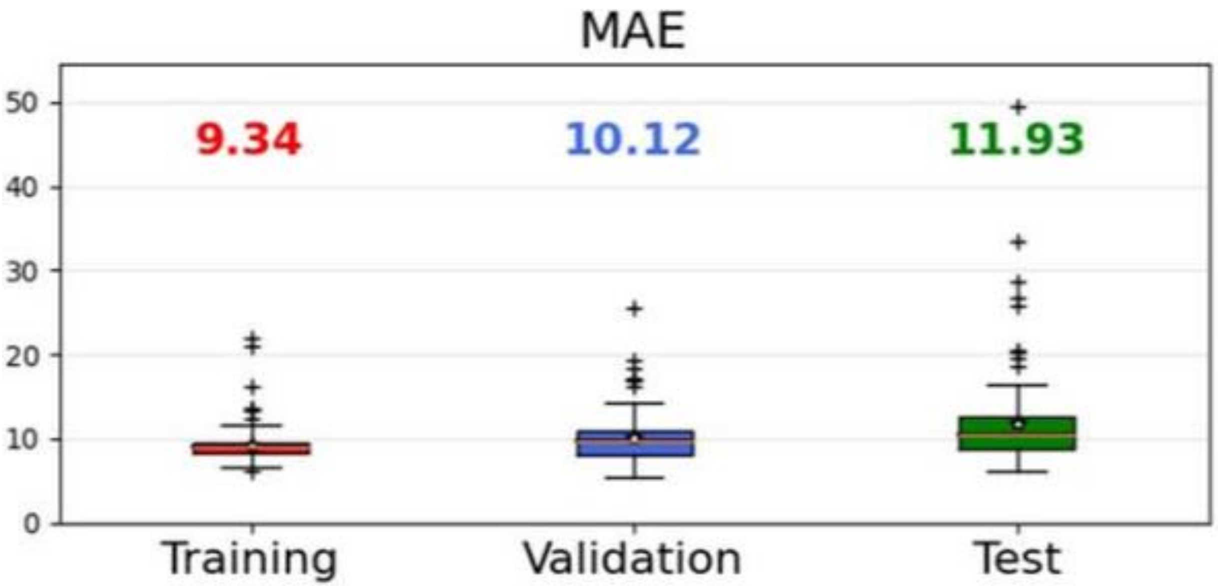
We considered 123 CME events occurred in the time range between 1997 and 2018 (that comply with the DB model)

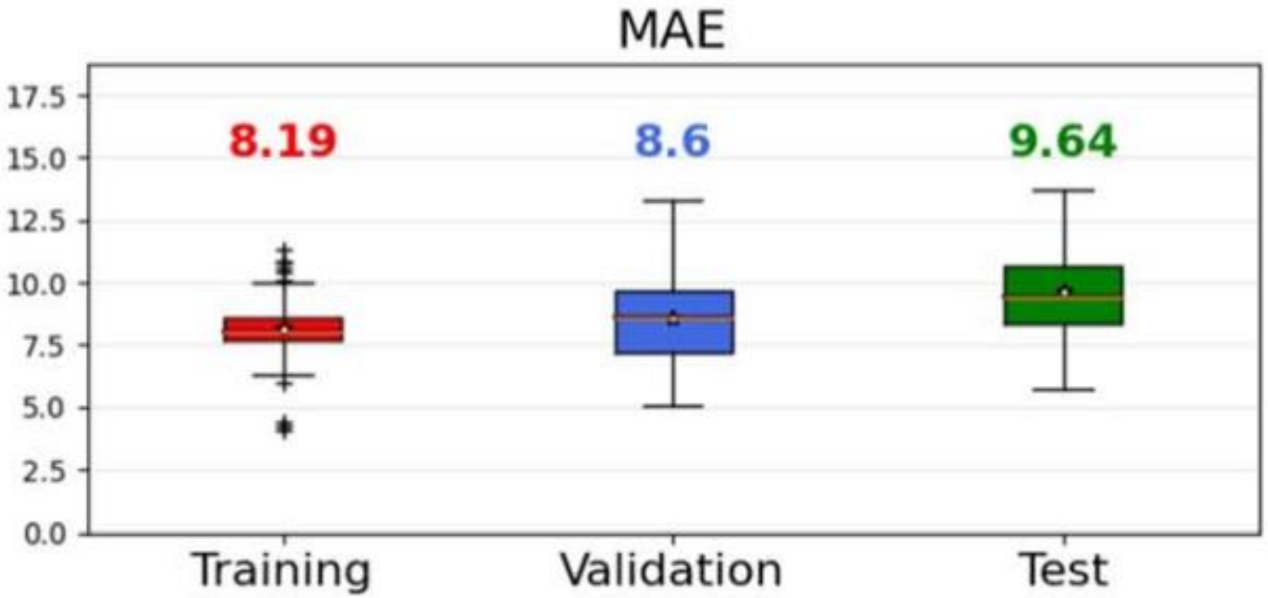
Name	Notation	Unity	Description	Source
CME height of eruption	r_0	km	$r_0 = 20\ R_\odot$, $R_\odot = 6.957 \cdot 10^5$ km	-
CME time of eruption	t_0	s	eruption time on the Sun at r_0	(Napoletano et al. 2022)
CME Time of Arrival	ToA	s	estimated arrival time at 1 AU	R & C
CME Travel time	TT	s	estimated time between t_0 and ToA	R & C, (Napoletano et al. 2022)
CME initial speed	v_0	km/s	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	mean over one hour after t_0	CELIAS
Drag parameter	C	dimensionless	parameter of the drag based model	this work

In order to perform a statistical assessment of the physics-driven machine learning approach to travel time prediction, we realized 100 random realizations of the training, validation, and test (70-15-15)

Results at a glance

Comparison between completely data-driven approach versus the new mix physics-driven approach





Completely data-driven
N1 is switched off
C is not an input of the second N2
λ = 1

$$L_T(t, N2(x)) = (t - N2(x))^2$$

Mix physics-driven approach
N1 is switched on
C is an input of the second N2
λ = 0.5

$$L_T(t, N2(\bar{x})) = \frac{1}{2}(t - N2(\bar{x}))^2 + \frac{1}{2}(1 - r(N1(x), N2(\bar{x})))^2$$

More results

Loss function	drag-parameter as input	Configuration	MAE (h)			
			min	median	mean	max
Fully data-driven	off	C1	6.1	10.43	11.93	49.5
	on	C4	4.8	9.96	10.48	36.09
Mix	off	C2	5.89	10.03	10.23	25.29
	on	C5	**5.74**	**9.46**	**9.64**	**13.75**
Fully physics-driven	off	C3	5.76	10.28	10.67	29.63
	on	C6	5.27	9.59	10.04	28.45

Legend of configurations

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

Future work, work in progress

Tuning the C parameter seems to be important:

- In our experiments, for some events the estimated value of C leads to $r(t)<0.95$ or $r(t)>1.05$ when t is the true travel time (we should have $r(t)\approx 1$!)
- Using other strategies proposed in literature for tuning C does not solve the problem.
- A better understanding of the CMEs in the dataset might lead to a better training process (with an improved splitting strategy)

The drag-based model is simple:

- Investigating possible modifications of the drag-based model to be included in the loss functions
- Study of analytical solutions
- The new model can include events that cannot be physically explained by drag-based model

The dataset is not that large:

- Use simulated data (… and transfer learning?)

Thanks for the attention!