frontiers

Visualizing and Interpreting Unsupervised Solar Wind Classifications

Jorge Amaya [1,*], **Romain Dupuis** [1], **Maria-Elena Innocenti**[2] **and Giovanni Lapenta** [1]

[1] *Centre for mathematical Plasma-Astrophysics, CmPA, Mathematics Department, KU Leuven, University of Leuven, Belgium*
[2] *Jet Propulsion Laboratory, Interstellar and Heliospheric Physics Division, 4800 Oak Grove Dr, Pasadena, CA 91109, USA*

Correspondence*:
Jorge Amaya, Mathematics Department, Celestijnenlaan 200B, KU Leuven, 3001 Leuven, Belgium
jorge.amaya@kuleuven.be, jorgeluis.amaya@gmail.com

Word count: in text (10172), in headers (105), outside text (843). Number of floats/tables/figures: 14.

ABSTRACT

One of the goals of machine learning is to eliminate tedious and arduous repetitive work. The manual and semi-automatic classification of millions of hours of solar wind data from multiple missions can be replaced by automatic algorithms that can discover, in mountains of multi-dimensional data, the real differences in the solar wind properties. In this paper we present how unsupervised clustering techniques can be used to segregate different types of solar wind. We propose the use of advanced data reduction methods to pre-process the data, and we introduce the use of Self-Organizing Maps to visualize and interpret 14 years of ACE data. Finally, we show how these techniques can potentially be used to uncover hidden information, and how they compare with previous manual and automatic categorizations.

Keywords: solar wind, ACE, Self-Organizing Maps, clustering, autoencoder, PCA, unsupervised, machine learning

1 INTRODUCTION

The effects of solar activity on the magnetic environment of the Earth have been observed since the publication of Edward Sabine's work in 1852 (Sabine, 1852). During almost two hundred years we have learned about the intimate connection between our star and the plasma environment of the Earth. Three main physical processes connect the Sun to us: the transfer of electromagnetic radiation, the transport of energetic particles, and the flow of solar wind. The later is a continuous stream of charged particles that carries the solar magnetic field out of the corona and into the interplanetary space.

The name 'solar wind' was coined by Parker in 1958 because 'the gross dynamical properties of the outward streaming gas [from the Sun] are hydrodynamic in character'(Parker, 1958). Over time we have learned that the wind also has many more complex properties. Initially, it was natural to classify the solar wind by defining a boundary between 'fast' and 'slow' winds (Habbal et al., 1997). The former has been associated with mean speed values of 750 km/s, while the later shows a limit at 500 km/s, where the compositional ratio (Fe/O) shows a break (Feldman et al., 2005; Stakhiv et al., 2015). The solar wind also carries information about its origins on the Sun. At certain solar distances the ion composition of the solar wind is expected to be frozen-in, reflecting the electron temperature in the corona and its region of origin (Feldman et al., 2005; Zhao et al., 2009; Stakhiv et al., 2015). These particles have multiple energies and show a variety of kinetic properties, including non-Maxwellian velocity distributions (Pierrard and Lazar, 2010; Matteini et al., 2012).

The solar wind is also connected to the Sun by the Interplanetary Magnetic Field (IMF), thorough magnetic field lines directed towards the Sun, away from the Sun, or in the case of flux ropes connected at both ends (Owens, 2016; Gosling et al., 2010). The thin region where solar magnetic fields of opposite directions meet is called the Heliospherc Current Sheet (HCS). When a spacecraft crosses the HCS instruments onboard can detect the change in magnetic field direction as a 180° reversal. Changes in the flow properties are also observed around the HCS. This perturbed zone is called the Heliospheric Plasma Sheet (HPS), and the passage of the spacecraft from one side of the HPS to the other is known as a Sector Boundary Crossing (SBC) (Winterhalter et al., 1994). In spacecraft observations these are sometimes confused with Corotating Interaction Regions (CIR), which are zones of the solar wind where fast flows have caught up with slow downstream solar wind, compressing the plasma.

From the point of view of a spacecraft SBCs and CIRs can show similar sudden changes in the plasma properties. These two in turn are often grouped and mixed with other transient events, like Coronal Mass Ejections (CME) and Magnetic Clouds (MC). Since 1981 when (Burlaga et al., 1981) described the propagation of MC behind an interplanetary shock, it was suspected that CMEs and MC where coupled. However, more recent studies show that CMEs observed near the Sun do not necessarily become MC, but instead 'pressure pulses' (Gopalswamy et al., 1998; Wu et al., 2006).

Much more recently it has been revealed, by observations from Parker Solar Probe, that the properties of the solar wind can be drastically different closer to the Sun, were the plasma flow is more pristine and has not yet mixed with the interplanetary environment. Patches of large intermittent magnetic field reversals, associated with jets of plasma and enhanced Poynting flux, have been observed and named 'switchbacks' (Bale et al., 2019; Bandyopadhyay et al., 2020).

The solar wind is thus not only an hydrodynamic flow, but a compressible mix of different populations of charged particles and electromagnetic fields that carry information of their solar origin (helmet streamer, coronal holes, filaments, solar active regions, etc.) and is the dominion of complex plasma interactions (ICMEs, MC, CIRs, SBCs, switchbacks).

To identify and study each one of these phenomena we have relied in the past on a manual search, identification and classification of spacecraft data. Multiple authors have created empirical methods of wind type identification based on in-situ satellite observations and remote imaging of the solar corona. Over the years the number and types of solar wind classes has changed, following our understanding of the complexity of heliospheric physics.

Solar wind classification serves three main roles:

1. it is used for the characterization of its origins in the corona,
2. to identify the conditions where the solar wind is geoeffective,
3. to isolate different plasma populations in order to perform statistical analysis.

Among these classifications we can include the original review work by (Withbroe, 1986), the impressive continuous inventory by (Richardson et al., 2000; Richardson and Cane, 2010, 2012), and the detailed studies by (Zhao et al., 2009) and (Xu and Borovsky, 2015). These publications classify the solar wind based on its origins and on the transient events detected. Each system includes two, three or four classes, generally involving coronal-hole origins, CMEs, streamer belt origins and sector reversal regions.

We are moving now towards a new era of data analysis, where manual human intervention can be replaced by 'intelligent' software. The trend has already started, with the work by (Camporeale et al., 2017) who used the (Xu and Borovsky, 2015) classes to train a Gaussian Process algorithm that autonomously assigns the solar wind to the proper class, and more recently by (Roberts et al., 2020) who used unsupervised classification to perform a 4 and 8 class solar wind classification. Machine learning algorithms have been used in the past in other applications in solar physics (Lundstedt et al., 1996; Qahwaji et al., 2008; Ahmed et al., 2013; Bobra and Couvidat, 2015; Bobra and Ilonidis, 2016; Nishizuka et al., 2017; Camporeale et al., 2018), and astrophysics (VanderPlas et al., 2012; Ntampaka et al., 2015; Hajian et al., 2015; Süveges et al., 2017; Bai et al., 2018; Bonjean et al., 2019).

The most basic machine learning techniques learn using two methods: a) in supervised learning the algorithms are shown a group of inputs and outputs with the goal to find a non-linear relationship between them, b) in unsupervised learning the machine is presented with a cloud of multi-dimensional points that have to be autonomously categorized in different classes.

This means that we can program the computer to learn about the different types of solar wind using the existing empirical classifications, or by allowing it to independently detect patterns in the solar wind properties.

In the present work we show how the second method, unsupervised classification, can be used to segregate different types of solar wind. In addition, we show how to visualize and interpret such results. The goal of this paper is to introduce the method to the community, the best use practices and the opportunities that such system can bring. We implement one specific type of classification, called Self-Organizing Maps, and we compare it to simpler classification techniques, showing that it can reveal hidden information in years of solar wind data.

In the next sections we present in detail the techniques of data processing (section 2.1), data dimension reduction (sections 2.2.1 and 2.2.2) and data clustering (section 2.2.3) that we have used. We then present in detail the Self-Organizing Map technique and all its properties in section 2.2.4. We show how to connect all of these parts together in section 2.2.5, and finally we show how the full system can be used to study 14 years of solar wind data from the ACE spacecraft in section 3.

2 MATERIALS AND METHODS

2.1 Data and Processing

2.1.1 Data Set Used

The solar wind data used in this work was obtained by the Advanced Composition Explorer (ACE) spacecraft, during a period of 14 years, between 1998 and 2011. The data can be downloaded from the FTP servers of The ACE Science Center (ASC). The files in this repository correspond to a compilation of hourly average data from three instruments: MAG (Magnetometer), SWEPAM (Solar Wind Electron, Proton, and Alpha Monitor) and EPAM (Electron, Proton, and Alpha Monitor). A detailed description of the entries in this data set can be found in the ASC website listed in section 5.

A total of 122712 data points are available. However, routine maintenance operations, low statistics, instrument saturation and instrument degradation produce gaps and errors in the data. The SWICS data includes a flag assessing the quality of the calculated plasma moments. We retain only 'Good quality' entries. Our pre-processed data set contains a total of 72454 points.

2.1.2 Additional Derived Features

We created additional features for each entry, based on previous knowledge of the physical properties of the solar wind. Some are derived from the existing properties in the data set, others computed from statistical analysis of their evolution. We introduce here the additional 'engineered' features included in our data set.

Multiple techniques have been proposed in the literature to identify ejecta, Interplanetary Coronal Mass Ejections (ICME), and solar wind origins in the ACE data. (Zhao et al., 2009) suggest that, during solar cycle 23, three classes of solar wind can be identified using its speed, V_{sw}, and the oxygen ion charge state ratio, O^{7+}/O^{6+}. It has been shown that slow winds originating in coronal streamers correlate with high values of the charge state ratio and fast winds coming from coronal holes presents low values (D'Amicis and Bruno, 2015). Plasma formed in coronal loops associated with CMEs also show high values of the charge state ratio (Xu and Borovsky, 2015). The classification boundaries of the 'Zhao classification' are presented in Table 1.

(Xu and Borovsky, 2015) suggested an alternative four classes system based on the proton-specific entropy, $S_p = T_p/n_p^{2/3}$, the Alfvén speed, $V_A = B/(\mu_0 m_p n_p)^{1/2}$, and the expected proton temperature, $T_{\text{exp}} = (V_{sw}/258)^{3.113}$. The classification conditions based on these three parameters are also presented in Table 1. This solar wind discrimination system will be known in this manuscript as the Xu classification (Xu and Borovsky, 2015). For each entry in the data set we have included the values of S_p, V_A, T_{exp}, and the solar wind type given by the two classification methods. Two of the classes, i.e. ICME/ejecta and coronal hole, are common to the Xu and Zhao classifications. The number given to each class is arbitrary, but the two common classes share the same identification.

Auxiliary variables, like the Alfvén Mach number (M_A) and the temperature ratio (T_{exp}/T_p), have also been included in the data set. These features have been selected as they are the main components of the Xu classification and we want to compare the automatic classification methods against empirical models.

In addition to these instantaneous quantities, we can perform statistical operations over a window of time of six hours, including values of the maximum, minimum, mean, standard deviation, variance, auto-correlation, and range. We expect to capture with these quantities transient events in the immediate solar wind upstream and downstream. These are a complement to the stationary solar wind parameters mentioned above and add some information about the temporal evolution of the plasma. The selection of the statistical parameters and the window of time is arbitrary and will require a closer examination in a future publication.

Two additional terms, which have been successfully used in the study of solar wind turbulence and wave characterization (Zhao et al., 2018; Adhikari et al., 2020; Magyar et al., 2019; D'Amicis and Bruno, 2015), are included here to account for additional time correlations. These are: the normalized cross-helicity, σ_c eq. (1), and the normalized residual energy, σ_r eq. (2), where $\boldsymbol{b} = (\boldsymbol{B} - \langle \boldsymbol{B} \rangle) / (\mu_0 m_p n_p)^{1/2}$ is the fluctuating magnetic field in Alfvén units, $\boldsymbol{v} = \boldsymbol{V_{sw}} - \langle \boldsymbol{V_{sw}} \rangle$ is the fluctuating solar wind velocity, $\boldsymbol{z}^{\pm} = \boldsymbol{v} \pm \boldsymbol{b}$ are the Elsässer variables (Elsasser, 1950), and $\langle . \rangle$ denotes the averaging of quantities over the time window.

$$\sigma_c = 2 \langle \boldsymbol{b} \cdot \boldsymbol{v} \rangle / \left\langle \boldsymbol{b}^2 + \boldsymbol{v}^2 \right\rangle \tag{1}$$

$$\sigma_r = 2 \left\langle \boldsymbol{z}^+ \cdot \boldsymbol{z}^- \right\rangle / \left\langle \boldsymbol{z}^{-^2} + \boldsymbol{z}^{+^2} \right\rangle \tag{2}$$

Due to gaps in the data, some of the above quantities can not be obtained. We eliminate from the data set all entries for which the derived features presented in this section could not be calculated. This leaves a total of 51608 entries in the data set used in the present work.

To account for the differences in units and scale, each feature column $\boldsymbol{F}$ in the data set is normalized to values between 0 and 1, using: $\boldsymbol{f} = (\boldsymbol{F} - \min \boldsymbol{F}) / (\max \boldsymbol{F} - \min \boldsymbol{F})$.

Not all the features might be useful and some of them can be strongly correlated. We do not perform here a detailed evaluation of the inter dependencies of the different features, and we leave that task for a future work. The present manuscript focuses on the description of the methodology and on the visualization and interpretation capabilities of unsupervised machine learning classification. We limit our work here to test two models that use a different number of features. These are listed in table 2 and named Amaya-21, and Roberts-8, in honor of the work done by (Roberts et al., 2020). As the name suggests each model uses respectively 21, and 8 features of the data set.

2.1.3 Complementary Data Catalogs

We support the interpretation of our results using data from three solar wind event catalogs. The first is the well known Cane and Richardson catalog that contains information about ICMEs detected in the solar wind, ahead of the Earth (Cane and Richardson, 2003) (Richardson and Cane, 2010) [1]. We used the August 16, 2019 revision. As the authors state in their website, there is no spreadsheet or text version of this catalog and offline editing was necessary. We downloaded and re-formatted the catalog to use it in our application. The CSV file created has been made available in our repository. We call this, the Richardson and Cane catalog.

The second catalog corresponds to the ACE List of Disturbances and Transients[2] produced by the University of New Hampshire. As in the previous case, the catalog is only available as an html webpage, so we have manually edited the file and extracted the catalog data into a file also available in our repository. This is hereafter referred to as the UNH catalog.

Finally, we also included data from the Shock Database[3] maintained by Dr. Michael L. Stevens and Professor Justin C. Kasper at the Harvard-Smithsonian Center for Astrophysics. Once again we have gathered and edited multiple web-pages in a single file available in our repository. In this work this database will be known as the CfA catalog.

[1] Near-Earth Interplanetary Coronal Mass Ejections Since January 1996: http://www.srl.caltech.edu/ACE/ASC/DATA/level3/icmetable2.htm

[2] ACE Lists of Disturbances and Transients: http://www.ssg.sr.unh.edu/mag/ace/ACElists/obs_list.html

[3] Harvard-Smithsonian, Center for Astrophysics, Interplanetary Shock Database - ACE: https://www.cfa.harvard.edu/shocks/ac_master_data/

2.2 Dimension Reduction and Clustering

2.2.1 Dimension Reduction using PCA

Principal Component Analysis (PCA) is a mathematical tool used in data analysis to simplify and extract the most relevant features in a complex data set. This technique is used to create entries composed of linearly independent 'principal components'. These are the eigenvectors of the covariance matrix Σ applied to the centered data, eq.(4), ordered from the largest to the smallest eigenvalue, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$, where $\overline{\boldsymbol{X}}$ is the mean value of each original feature, eq.(3). The projection of the data onto the principal component space ensures a maximal variance on the direction of the first component. Each subsequent principal component is orthogonal to the previous ones and points in the direction of maximal variance in the residual sub-space (Shlens et al., 2014).

$$\overline{\boldsymbol{X}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{X}_i \tag{3}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m} \left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right)\left(\boldsymbol{X}_i - \overline{\boldsymbol{X}}\right) \tag{4}$$

The PCA transformation creates as many components in the transformed space, $\tilde{\boldsymbol{X}}$, as features in the original data space $\boldsymbol{X}$. However, components with small eigenvalues belong to a dimension where the variance is so small that it is impossible to separate points in the data. It is a general practice in data reduction to keep only the first k components that explain at least a significant portion of the total variance of the data, $\lambda_{i=1..k}/\mathrm{Tr}(\Sigma) > \epsilon$. This allows for a selection of information that will effectively differentiate data points, and for a reduction of the amount of data to process during analysis. Many techniques have been suggested for the selection of the values of k and the cut-off ϵ (Rea and Rea, 2016). We use the value of $k = 3$ to simplify the comparison among the different models, but for a detailed study of the solar wind, if a PCA transformation is applied, it is important to use a fixed criteria for the selection of the cut-offs.

Fig. 1 (A) is a 3D scatter plot of all the data points, colored by the Xu classification, projected on the first three PCA components. The features used to create this figure are presented in section 2.2.5.2. Panel (B) contains the same data colored by the Zhao classification. These projections show that the Xu and Zhao classification are defined by hyper-planes separating the points, even if the data has been linearly transformed by the PCA. Class 2, ICMEs-ejecta, is restricted to a small domain in this coordinate system (for both the Xu and Zhao classification). The lateral plots on panel (A) are 2D histograms of the point distribution on the three main PCA planes. They show that the concentration of points is not homogeneous and different zones can be isolated using unsupervised classification techniques. There is a clear segregation of points in the (1st,2nd)-component plane: as we will see in subsequent sections, one of the features of the solar wind presents a strong bimodal distribution that is prioritized by the PCA.

2.2.2 Dimension Reduction Using Autoencoders

PCA has a limitation: the principal components are a linear combination of the original properties of the solar wind. An alternative to data reduction is the use of autoencoders (AE). These are machine learning techniques that can create non-linear combinations of the original features projected on a latent space with less dimensions (Hinton and Salakhutdinov, 2006). This is accomplished by creating a system where an encoding function, ϕ, maps the original data $\boldsymbol{X}$ to a latent space, $\boldsymbol{\mathcal{F}}$, eq.(5). A decoder function, ψ, then maps the latent space back to the original input space, eq.(6). The objective of the autoencoder is to minimize the error between the original data and the data produced by the compression-decompression procedure as shown in eq.(7).

$$\phi : \boldsymbol{X} \rightarrow \boldsymbol{\mathcal{F}} \tag{5}$$

$$\psi : \boldsymbol{\mathcal{F}} \rightarrow \boldsymbol{X} \tag{6}$$

$$\phi, \psi = \underset{\phi,\psi}{\arg\min} \left\| \boldsymbol{X} - (\phi \circ \psi)\boldsymbol{X} \right\|^2 \tag{7}$$

Autoencoders can be represented as feed-forward neural networks, where fully connected layers lead to a central bottleneck layer with few nodes and then expands to reach again the input layer size. An encoded element, $\boldsymbol{z} \in \boldsymbol{\mathcal{F}}$, can be obtained from a data entry, $\boldsymbol{x} \in \boldsymbol{X}$, following the standard neural network function, eq.(8), where $\boldsymbol{W}$ is the weights matrix, $\boldsymbol{b}$ is the bias, and σ is the non-linear activation function.

$$\boldsymbol{z} = \sigma\left(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}\right) \tag{8}$$

$$\hat{\boldsymbol{x}} = \sigma'\left(\boldsymbol{W'}\boldsymbol{z} + \boldsymbol{b'}\right) \tag{9}$$

$$\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \left\| \boldsymbol{x} - \hat{\boldsymbol{x}} \right\|^2 \tag{10}$$

The decoding procedure, shown in eq.(9), transforms $\boldsymbol{z} \rightarrow \hat{\boldsymbol{x}}$, where the prime quantities are associated with the decoder. The loss function, $\mathcal{L}(\boldsymbol{x}, \hat{\boldsymbol{x}})$, is the objective to be minimized by the training of the neural network using gradient descent. Once training is completed, the vector $\boldsymbol{z}$ is a projection of the input vector $\boldsymbol{x}$ onto the lower dimensional space $\boldsymbol{\mathcal{F}}$.

Additional enhancements and variations of this simple autoencoder setup exist in the literature, including multiple regularization techniques to minimize over-fitting (Liang and Liu, 2015), Variational Autoencoders (VAE) that produce encoded Gaussian distribution functions (Kingma and Welling, 2013), and Generative Adversarial Networks (GAN) that produce new (unseen) data (Goodfellow et al., 2014). In this work we use the most basic form of autoencoders, presented above.

The second column of Fig.1, panels (C) and (D), contains the same information as the first column, but with the data set encoded in the three dimensional latent space $\boldsymbol{\mathcal{F}}$. Panel (C) shows that all the classes in the Xu and Zhao classification are easy to distinguish, including ICMEs-ejecta (class 2) that is difficult to discern in the PCA. This projection also shows that class 4 from the Zhao classification in the bottom panels, overlaps with class 3 (sector reversal origin), and partially with class 2 (ejecta) in the Xu classification on the top panels. Panel (C) shows, on the side planes, 2D histograms of the density of points. These can be seen as the volume integral of the point density in each direction. Here again it is possible to observe multiple zones of high concentration, suggesting that multiple types of solar wind are present in the data and that they can be differentiated using an unsupervised classification technique.

2.2.3 Clustering Techniques

The goal of unsupervised machine learning is to group data points in a limited number of clusters in the N-dimensional space $\Omega \in \mathbb{R}^N$, where N is the number of features (components or properties) in the data set. Multiple techniques can be used to perform multi-dimensional clustering. We present in Fig. 2 the three clustering techniques used to classify our 3D reduced data. The panels in the first column show the data projected in the PCA reduced space, $\tilde{\boldsymbol{X}}$, while the second column shows the data in the latent AE space, $\boldsymbol{\mathcal{F}}$. Each row corresponds to a different clustering method. The colors in the top panels (A) and (D) were obtained using the k-means method (Lloyd, 1982), the colors in the middle panels (B) and (E) were obtained using the Gaussian Mixture Model (GMM) (Bishop, 2006). The bottom panels are colored by the classes from the Self-Organizing Maps described later in section 2.2.4.

The k-means technique has already been used in a recent publication for the determination of solar wind states (Roberts et al., 2020). To our knowledge other clustering methods have never been used in the literature to classify the solar wind, but (Dupuis

et al., 2020) has used the GMM to characterize magnetic reconnection regions in simulations using their velocity distribution information.

The colors used in Fig.2 are assigned randomly by each clustering technique. The most glaring issue with them is that different methods can lead to different clusters of points. The GMM and the k-means agree on their classification in the PCA space, but show dissimilar results in the AE space. Moreover, for a single method, e.g. k-means, slight modifications of the clustering parameters, e.g. using a different seed for the random number generator, can lead to very different results. We address this last issue using an algorithm that launches the k-means and GMM algorithms 500 times until the methods converge to a quasi-steady set of clusters. But we warn that the results are implementation dependent.

In the present data set, the cloud of points is convex and well distributed in all three components. This raises one additional issue, observed more clearly in the first column of Fig.2: when classical clustering methods are applied to relatively homogeneously dense data, it divides the feature space in Voronoï regions with linear hyper-plane boundaries. This is an issue with all clustering techniques based on discrimination of groups using their relative distances (to a centroid or to the mean of the distribution). To avoid this problem density-based techniques, such as DBSCAN (Ester et al., 1996), and agglomeration clustering methods, use a different approach. However, we can not apply them here because in such homogeneous cloud of points these techniques lead to a trivial solution where all data points are assigned to a single class.

There is no guarantee that a single classification method, with a particular set of parameters will converge to a physically meaningful classification of the data if the points in the data do not have some level of separability, or have multiple zones of high density. This is also true for other classification methods based on 'supervised learning'. In those applications same issues will be observed if the training data uses target classes derived from dense data clouds using simple hyper-plane boundaries, as done for the Zhao and Xu classes. An example of such application was published by (Camporeale et al., 2017). The authors used the Xu classification to train a Gaussian Process classifier.

2.2.4 Self-Organizing Maps

2.2.4.1 Classical SOM

Following the definitions and notations by (Villmann and Claussen, 2006), a class can be defined as $C_i \stackrel{\text{def}}{=} \{x \in \Omega | \Phi(x) = \boldsymbol{w}_i\}$, where Φ is a function from Ω to a finite subset of k points $\{\boldsymbol{w}_i \in \mathbb{R}^N\}_{i=1..k}$. A cluster C_i is then a partition of Ω, and $\{\boldsymbol{w}_i\}$ are the code words (also known as nodes, weights or centroids) associated. The mapping from the data space to the code word set, $\Phi : \Omega \to \mathcal{W}$, is obtained by finding the closest neighbor between the points $\boldsymbol{x}$ and the code words $\boldsymbol{w}$, eq.(11). The code word $\boldsymbol{w_s}$, the closest node to the input $\boldsymbol{x_s}$, is called the 'winning element'. The class C_i corresponds to a Voronoï region of Ω with center in $\boldsymbol{w}_i$.

$$\Phi : x \to \arg\min_{i \in \mathcal{N}} (\|x - \boldsymbol{w}_i\|) \tag{11}$$

A Self-Organizing Map (SOM) also composed of structured nodes arranged in a lattice, and assigned to a fixed position $\boldsymbol{p}_i$ in $\mathbb{R}^q$, where q is the dimension of the lattice (generally $q = 2$). The map nodes are characterized by their associated code words. The SOM learns by adjusting the code words $\boldsymbol{w}_i$ as input data x is presented.

The SOM is the ensemble of code words and nodes $A_i = \{\boldsymbol{w}_i, \boldsymbol{p}_i\} \in (\Omega \times \mathbb{R}^q)$. For a particular entry $\boldsymbol{x_s}$, the code word $s \in \mathcal{N}$ is associated to the winning node $\boldsymbol{p_s}$ if the closest word to $\boldsymbol{x_s}$ is $\boldsymbol{w_s}$. At every iteration of the method, all code words of the SOM are shifted towards x following the rule:

$$\Delta \boldsymbol{w}_i = \epsilon(t) h_\sigma(t, i, s)(x - \boldsymbol{w}_i) \tag{12}$$

with $h_\sigma(t, i, j)$ defined as the lattice neighbor function:

$$h_\sigma(t,i,j) = e^{-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2}{2\sigma^2(t)}} \tag{13}$$

where $\epsilon(t)$ is the time dependent learning rate, eq.(14), and $\sigma(t)$ is the time dependent lattice neighbor width, eq.(15). The training of the SOM is an iterative process where each data point in the data set is presented to the algorithm multiple times $t = 0, 1, .., t_f$. In these equations the subscript 0 refers to initial values at $t = 0$ and the subscript f to values at $t = t_f$.

$$\epsilon(t) = \epsilon_0 \left(\frac{\epsilon_f}{\epsilon_0}\right)^{t/t_f} \tag{14}$$

$$\sigma(t) = \sigma_0 \left(\frac{\sigma_f}{\sigma_0}\right)^{t/t_f} \tag{15}$$

This procedure places the code words in the data space Ω in such a way that neighboring nodes in the lattice are also neighbors in the data space. The lattice can be presented as a q-dimensional image, called map, where nodes sharing similar properties are organized in close proximity.

The main metric for the evaluation of the SOM performance is called the quantization error:

$$Q_E = \frac{1}{M}\sum_{i=1}^{M} \|x_i - w_{x_i}\| \tag{16}$$

where M, is the total number of entries in the data set.

Once the training of the SOM is finished, the code words $\boldsymbol{w}_i$ can be grouped together using any clustering technique, e.g. k-means. The nodes of the SOM with close properties will be made part of the same class. The classes thus created are an ensemble of Voronoï subspaces, allowing a complex non-linear partitioning of the data space Ω.

The final number of clusters is an input of the algorithm, but can also be calculated autonomously. The Within Cluster Sum of Squares (WCSS) can be used as a metric of the compactness of the clustered nodes. As its name implies the WCSS is the sum of the squared distances from each node to their cluster point. If only one class is selected, the large spread of the nodes would produce a high WCSS. The lowest possible value of the WCSS is obtained for a very high number of classes, when the number of classes is equal to the number of nodes. But such extreme solution is also unpractical. The optimal number of clusters can be obtained using the Kneedle class number determination (Satopaa et al., 2011). In the present work we do not use this technique: we will perform comparisons with previous publications that propose a fixed number of solar wind types. We will explore the use of an automatic class number selection in a future publication.

2.2.4.2 Dynamic SOM

The time dependence of the SOM training allows the code words $\boldsymbol{w}_i$ to reach steady coordinates by slowing down their movement over the iterations. Due to the minimization of the distance in eq.(11) code words tend to agglomerate around high density zones of the feature space. The Dynamic Self-Organizing Map (DSOM), introduced by (Rougier and Boniface, 2011), eliminate the time dependence and allows to cover larger zones of the space outside of the high density regions.

The DSOM is a variation of the SOM where the learning function (12) and the neighbor function (13) are replaced by eqs. (17) and (18) respectively:

$$\Delta \boldsymbol{w}_i = \epsilon \, \|x - \boldsymbol{w}_i\|_\Omega \, h_\eta(i,s,x)(x - \boldsymbol{w}_i) \tag{17}$$

$$h_\eta(i,s,x) = e^{-\frac{1}{\eta^2}\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|^2}{\|x - \boldsymbol{w}_s\|^2_\Omega}} \tag{18}$$

where ϵ is a constant learning rate, $h_\eta(i, s, x)$ is defined as the new lattice neighbor function, and η is the 'elasticity' parameter. In their work (Rougier and Boniface, 2011) show that DSOM can be used to better sample the feature space Ω, reducing the agglomeration of code words around high density zones. The DSOM does not converge to a steady solution, due to the lack of a temporal damping factor.

2.2.4.3 Visualization of SOM and DSOM

Clustering techniques do not necessarily converge to a steady immutable solution. Differences in the training parameters or slight changes in the data can have an important impact on the final classification. These tools can be used for statistical analysis, comparisons, data visualization and training of supervised methods. But it will be practically impossible to claim the existence of a general objective set of states discovered only by the use of these basic clustering techniques.

However, SOMs and DSOMs provide an important tool for the study of the solar wind: the maps are composed of nodes that share similar properties with its immediate neighbors. This allows for visual identification of patterns and targeted statistical analysis.

We used the python package MiniSom (Vettigli, 2013) as the starting point of our developments. Multiple methods of the MiniSom have been overloaded to implement the DSOM, and to use a lattice with hexagonal nodes. All auxiliary procedures used to calculate inter-nodal distances, node clustering, data-to-node mapping, and class boundary detection have been implemented by us. All visualization routines are original and have been developed using the python library Matplotlib (Hunter, 2007).

Fig.3 shows the basic types of plots and maps that can be generated using the SOM/DSOM techniques. This figure uses data from the model Amaya-21 which has been encoded, using AE, into a set of entries, $\boldsymbol{z}_i$, each one composed of three components. Panel (A) shows a histogram of the first two components of the feature space Ω, with dots marking the position of the code words $\boldsymbol{w}_i$. The colors of the dots represent their SOM classification. The red lines connect a single code word $\boldsymbol{w}_s$ with its six closest neighbors. The panel (B) shows the 'hit map' of the SOM. It contains the lattice nodes $\boldsymbol{p}_i$ associated to the code words $\boldsymbol{w}_i$. They are depicted as hexagons with sizes representing the number of data points connected to each node and colored by their SOM class. The thickness of the lines between lattice nodes represent the relative distance to its neighbors in the feature space Ω. Red lines connect the node $\boldsymbol{p}_s$, associated to the code word $\boldsymbol{w}_s$ in panel (A), to its closest neighbors.

Panel (C) of Fig.3 corresponds to the value of a single feature associated to each node; as an example we use the ionized oxygen ratio O^{7+}/O^{7+} ('O7to6'). To improve visualization all hexagon sizes have been set to their maximum and the inter-node distance line has been colored white. In order to obtain the correct value for each node, we must first perform a decoding of the data from the latent space, $\Omega = \boldsymbol{\mathcal{F}}$, to the original data set space, $\boldsymbol{X}$.

Panel (D) of Fig.3 shows that the nodes of the lattice can also be used to present data that has not been used in the training of the SOM. The method keeps track of the data set points associated to each lattice node, it is then possible to perform independent statistical operations on those points alone. Moreover, it is possible to activate the SOMs with just a subset of the data, i.e. with points that feature a specific solar wind type. In this case, as an example, we have colored the map using the average oxygen charge state $\langle Q_O \rangle$ ('avqO'), and we have set the size of the nodes to represent the frequency of points with solar wind type Xu=2 (ejecta). The dark line between the lattice nodes designate the boundaries between different SOM classes.

These four representations are only a few examples of the variety of data that can be represented using SOMs. The most important aspect of the SOMs is that data is represented in simple 2D lattices where the nodes share properties with their neighbors. Here we also decided to use hexagonal nodes, connecting 6 equidistant nodes, but other types of representations are also valid, e.g. square or triangular nodes.

The bottom row of Fig.3 displays all three components of the code words $\boldsymbol{w}_i$ associated with each one of the $\boldsymbol{p}_i$ nodes. In the first panel they have been mapped to the basic colors Red, Green and Blue (RGB). The remaining panels have been colored using each individual component. The first panel is then the RGB composition of the three remaining ones where the boundaries between the SOM classes have been highlighted.

2.2.5 The Full Architecture

The previous sections introduced all the individual pieces that we use for the present work. Here we give a global view of the full model. Fig.4 shows how all the components are interconnected. The data set is composed of clean and processed entries. We tested the PCA transformation in cases Amaya-21 and Roberts-8, keeping only the first three principal components. This possible setup is presented on the left of the figure. It is also possible to perform an unsupervised clustering directly on the un-processed data, as shown on the top of the figure, but it is not recommended. For the remaining of this manuscript we present only the cases where the non-linear AE encoding, shown at the right of Fig.4. The bottleneck of the AE network is three nodes, i.e. the data is encoded in three components. The transformed data is then used to train the SOM.

After training, the code words of the SOM are then clustered to group together nodes that share similar properties. This second level classification is done using the k-means++ algorithm with 500 re-initializations (it is in general recommended to use between 100 and 1000 iterations). The total number of classes selected is an input of the model and has been set to 8. This arbitrary choice was made following the results presented by (Roberts et al., 2020). All the software was implemented in Python using as main libraries PyTroch (Paszke et al., 2019), Scikit-learn (Pedregosa et al., 2011), Matplotlib (Hunter, 2007), MiniSom (Vettigli, 2013), Pandas (McKinney, 2010) and NumPy (Oliphant, 2006).

2.2.5.1 Autoencoder architecture

We use a basic, fully connected feed-forward neural network for the encoding-decoding process. The bottleneck of the network has been fixed to three neurons in order to simplify the visualization. This arbitrary choice is another parameter of the models that need further investigation. The neural network is symmetric in size but the weights of the encoder, $\boldsymbol{W}$, and the decoder, $\boldsymbol{W'}$, are not synchronized (see eqs.(8), (9)). We use multiple fully connected hidden layers, where the central layer is the size of the bottleneck. Each layer is composed of a linear regressor, followed by batch normalization and a ReLU activation function. The output layer of the network contains a linear regressor followed by a hyperbolic tangent activation function. The autoencoder has been coded in python using the PyTorch framework (Paszke et al., 2019).

We use an Adam optimizer (Kingma and Ba, 2014) for the gradient descent with a learning rate of 0.001 and a weight decay of 0.0001 for regularization. The loss function is the Mean Squared Error (MSE). We train the network for 30 epochs, after which we see no additional improvement in the loss function. The full data set was randomly divided 50%/50% between training and testing sets.

2.2.5.2 Two Models of Solar Wind Classification

We have tested the two models presented in Table 2. The models are inspired by the work of (Roberts et al., 2020). We call these cases Amaya-21 and Roberts-8. The table lists all the features used in each model. A detailed description of each feature can be found in the ACE Level 2 documentation. To spread the data over a larger range of values in each component, we have used the logarithm of all the quantities, except of those marked with an asterisk in the table.

Features 15 to 20 contain an additional suffix, corresponding to a statistical operation performed on the corresponding feature. The operations include the mean, the range, the standard deviation and the auto-correlation of quantities over a window of time of 6 hours. This window allows to capture temporal (spatial) fluctuations in some of the solar wind parameters.

On the lower part of Table 2 we present the range of dates used for each model. For Amaya-21 we use the full data set, while for the model Roberts-8 we try to replicate as much as possible the choices made in (Roberts et al., 2020). The same table also contains the hyper-parameters selected to run the two models. The number of neurons per layer in the encoding half of the neural network is listed in the table and was manually selected to minimize the final loss value of the AE.

All the figures presented until now correspond to the processing of data from model Amaya-21. The amount of data and figures produced in this work is very large and is not possible to include all of them in the present document. We will present in the next section some highlights, but more detailed analysis of each one of the cases will be presented in future publications.

2.2.5.3 Budget

Machine learning models require fine tuning of different parameters, from the selection and testing of multiple methods, to the parameterization of the final architecture. (Dodge et al., 2019) suggests that every publication in machine learning should

include a section on the budget used for the development and training of the method. The budget is the amount of resources used in the data processing, the selection of the model hyper-parameters (HP), and its training.

The most time-consuming task in the present work has been the data preparation, the model setup and debugging and the writing of the SOM visualization routines. All the techniques described in the previous sections have been coded in python and are freely accessible in the repositories listed in section 5. We estimate the effort to bring this work from scratch to a total of 2 persons month. Of these, one person week was dedicated to the manual testing an selection of different model HPs (autoencoder architecture, feature selection, learning rates, initialization methods, number of epochs for training, selection of data compression method, size of the time windows, etc.).

All classic clustering techniques presented in section 2.2.3 require only a few lines of code and can be trained in minutes on a mid-range workstation (e.g. Dell Precission T5600, featuring two Intel(R) Xeon(R) CPU E5-2643 0 @ 3.30GHz with four cores and eight threads each). The most time consuming tasks of our models are the training of the autoencoder (5% of the total run time), the multiple passages of the clustering algorithms (15% of the run time), and the optimization of the SOM hyper-parameters (80% of the run time). The training of the SOM is performed in less than a minute.

For reference, the total run-time for each one of the models used in this work are: 60 minutes for the Amaya-21 model and 20 minutes for the Roberts-8 model.

2.2.5.4 Hyper-Parameter Optimization

Our main goal in this manuscript is to introduce the use of the SOMs for the classification of solar wind data. SOMs require the selection of four main Hyper-Parameters (HPs): the size of the lattice, $(m \times n)$, the initial learning rate, ϵ_0, and the initial neighbor radius, σ_0. In the case of the DSOM algorithm, these two last HPs are replaced by the constant learning rate, ϵ, and the elasticity, η. The automatic selection of the best HP for machine learning model is called Hyper-Parameter Optimization (HPO).

We use the library 'optuna' (Akiba et al., 2019) to perform an automatic optimization of the four HPs. The optimization is based on a technique called Tree-structured Parzen Estimator (TPE) (Bergstra et al., 2013), which uses Bayesian Optimization to minimize a target function provided by the user. We propose the use of the objective function, $\mathcal{H}$, described in eq.(19):

$$\mathcal{H}\left(\sigma, \eta, m, n\right) = Q_E(\sigma, \eta, m, n) + \alpha \frac{m}{m_{max}} + \beta \frac{n}{n_{max}} + \gamma mn \tag{19}$$

where Q_E is the SOM quantization error, m and n are the number of lattice nodes in each dimension, and m_{max} and n_{max} are the given maximum number of possible nodes. The weight factors α, β and γ are used to impose restictions on each term. We have fixed their value to $\alpha = \beta = 0.4Q_E^0$ and $\gamma = 0.08Q_E^0$, where Q_E^0 is the quantization error at iteration zero of the first optimization trial. When a large number of nodes is available smaller values of Q_E are automatically obtained because the mean distance from the data set entries to the code words is reduced. The second and third terms on the RHS of $\mathcal{H}$ leads the optimizer to reduce the number of nodes in the SOM. The squaring term γmn forces the map to be as squared as possible.

After a total of 100 trial runs of the model using different HPs, the optimizer selected the parameters presented in the lower section of table 2. The total run time for the HPO of the case Amaya-21 is about 40 minutes. HPO is, understandably, one of the most expensive procedures in all our setup.

3 RESULTS AND COMPARISONS

3.1 Interpretation of ACE data using SOMs

3.1.1 General overview

Fig.3 (A) shows the distribution of code words in the latent space for the model Amaya-21. The feature and components maps in the bottom row show in addition that lattice nodes share common attributes with their neighbors. The regularity in the colors of the feature map confirms that the SOM keeps its most important feature: organization. This is clear by the proximity of

neighboring code words in the latent space, marked with red lines. We expect then to find common patterns in all the following maps in this section.

In the same figure the panels (B) and (C), and the component maps in the bottom row, show black and white lines indicating the relative distances between the lattice nodes: thicker lines represent larger inter-node distances. This shows that there are sections of the map (groups of code words in the latent space) that can form separate groups. This group separation has been highlighted in the 'Feature map': the k-means clustering of the code words divides the space following the inter-nodal divisions. The classified points in the 3D latent space are shown in the third row of Fig.2.

The 'Hit map' in Fig.3 presents a good distribution of points among all the lattice nodes, except in regions isolated from the rest. These zones represents solar wind types that have atypical properties. One of the goals of the DSOM is to cover those isolated zones where rare events can be classified in separate nodes. In contrast, the classic SOM method tends to cluster the code words in regions of high density, troubling the categorization of rare events, like ICMEs, ejecta or magnetic clouds, in the solar wind data.

Fig.5 shows the distribution of the SOM training data in the normalized range $[0..1]$. This is a violin plot superposed by a box plot, showing the data distribution for each one of the features listed in table 2. Normalization of the data is performed using the maximum and minimum values of each feature, but outliers (extremely large or small values) can hinder the use of particular features. In any classification problem, outlier detection and elimination is extremely important. The figure shows that all our data points are well represented in the data. Even in some cases, like for feature 13 (Alfvén Mach number), where the distribution has a small width, it still covers a significant part of the total range. This figure also shows that feature 6 (cross-helicity, σ_c) has a bimodal distribution, with two peaks close to the limits. A significant part of the data lays close to the $\sigma_c = \pm 1$ limit. The cross-helicity is a measure of the Alfvénicity of the solar wind, representing the direction and intensity of the propagation of Alfvénic fluctuations. At the Earth orbit this is an indication of the origins of the solar wind in the north or south hemispheres of the Sun.

Feature 20 (Pearson auto-correlation of the magnetic field magnitude) also shows a large distribution function, with a marked peak near one. This quantity was calculated for a window of time of six hours and a time lag of one hour. It shows the extent to which the values of the magnetic field have changed in one hour. High autocorrelation values represent situations in which the magnetic field does not change during the window, i.e. values in the window at time t are the same as values shifted by one hour, $t - 1$. Completely uncorrelated signals, produced by random changes in time, will produce autocorrelation values close to 0 (0.5 after normalization), i.e. the data in the window at time t is different from the data in the shifted window. Positive (negative) values represent a signal with a periodicity of one hour. Additional time lags could be used to create extra features, but here we use only one to test its effectiveness.

The features selected for this model have not followed a meticulous vetting process. We included features inspired from previous publications and new interesting additions. Our goal in this work was to test if the data transformation into an encoded latent space can account for redundant or un-interesting features. This is a very useful property for data sets where expert knowledge is not available. It also shows how the SOM can point to features that don't have added value. As we will see in the next sections, the method converges to meaningful classes, even when some of the features used turn out to be not very relevant. We will perform in a future publication a more detailed selection of the features, based on the experience of human experts.

3.1.2 Interpretation of the Automatic Classification

Lattice nodes are characterized by their weights (code). Applying a reverse transformation, followed by a re-scaling, we obtain their values in the original N-dimensional space. Fig.6 shows the DSOM clustering and the 21 solar wind properties associated to each lattice node. We have clustered the nodes in eight classes. This is a subjective selection inspired by other works in the literature. In our case the clustering leads to contiguous groups of nodes.

The maps show the properties that differentiate each of the eight classes. We can try to attribute a physical significance to the classes by analyzing, together, the characteristic features of each class in Fig.6 and examples of the recorded solar wind data in Fig.7. This figure presents a window of time of four months, from the beginning of May 2003 until the end of September 2003.

We have plotted entries in the Richardson and Cane, UNH and CfA catalogs on top of time series of the solar wind speed in panels (A), (B) and (C). These plots have been colored by the class number of the k-means, GMM and SOM classes.

The polarity of the solar wind can be observed in panel (D). Changes from red to green are associated with crossings of the HCS. The z-component of the magnetic field is plotted in blue (positive) and red (negative) in panel (E).

Fig.7 (F) shows the evolution of the O^{7+}/O^{6+} ratio using a dotted black line in logarithmic scale. The Zhao classification boundaries (see table 1) are plotted using a black continuous line. The red area corresponds to 'non-coronal hole origin' solar wind, and data points receive this classification if the dotted line enters the red zone. If it stays above it, the solar wind is considered an ICME. If the curve drifts bellow the red zone, the wind is considered to have origins in a coronal-hole.

The three time series in panels (A), (B) and (C) show that multiple techniques can be used for the clustering of solar wind properties. But SOMs allow fast visualization and interpretation not available in other clustering methods. All time series have a strong tendency to group the solar wind in two groups, depending on the heliospheric sector. This is due to the importance of the cross-helicity in the data set and its bimodal distribution.

Class 5 can be mapped to transient events, CMEs and ejecta. It presents very high values of the oxygen ion charge state ('O7to6'), a feature that (Zhao et al., 2009; Stakhiv et al., 2015; Xu and Borovsky, 2015) associate to CME plasma. It is also characterized by high solar wind velocity, cross-helicity $\sigma_c \sim 0$, high values of magnetic field magnitude and Alfvén speed. These features are usually associated to explosive transient activity (Roberts et al., 2020; Xu and Borovsky, 2015). Fig. 7, panel (C) shows that class 5 maps well to the Richardson and Cane, UNH and CfA CME catalogs.

We can then use the cross-helicity, σ_c, to identify two groups of classes: the ones with mostly positive (1 and 3) and mostly negative cross-helicity (0, 2, 4, 6, 7). As already done in (Roberts et al., 2020), we associate them to solar wind plasma originating from areas with different magnetic polarity, respectively northern and southern sector. Inspection of Fig. 7 confirms this association.

Among the classes with negative σ_c, class 7 can be quite confidently associated with coronal hole plasma. It is characterized by the very low values of the O^{7+}/O^{6+} ratio that (Zhao et al., 2009; Stakhiv et al., 2015; Xu and Borovsky, 2015) associate with plasma originating from open magnetic field lines. It also exhibits high wind speed, low proton density and proton density variability, high absolute values of cross-helicity and equipartition levels of the residual energy σ_r (telltale signs of Alfvénicity), high proton entropy and moderately high values of Alfvén speed, high proton temperature and temperature variability. These are characteristics widely associated to fast coronal holes solar wind plasma. Inspection of Fig. 7 again supports this identification.

Class 4 and 0 are both possibly composed of a mix of slow Alfvénic wind (D'Amicis and Bruno, 2015) and 'conventional' slow and intermediate speed wind. Slow Alfvénic wind shares coronal hole origin and a number of characteristics related to the origin (O^{7+}/O^{6+} ratio, low density values, high $|\sigma_c|$ and low σ_r values) with the fast Alfvénic wind. The main difference, apart, of course, from the speed, is the proton temperature, which tends to be lower in Alfvénic and 'conventional' slow wind with respect to the Alfvénic fast wind. The parameters that are usually used to distinguish between slow and fast wind (speed, density, proton entropy, proton temperature) span a quite large interval in both classes 0 and 4, and in fact point to the presence of a mix of slow and fast wind in both. The main difference between class 0 and 4 is given by the very high values of the residual energy σ_r in class 4, which point to kinetically dominated structures.

Class 6 is shows features generally associated with intermediate and slow wind of streamer belt origin: intermediate values of O^{7+}/O^{6+} ratio, intermediate and slow speed, high proton density, low absolute value of cross-helicity, low values of magnetic field magnitude, proton entropy, Alfvén speed, proton temperature.

Class 2 is characterized mainly by very high values of the Alfvénic Mach number. It is a class with a very low number of hits (see the hit map), and rarely spotted in Fig. 7.

Among the classes associated with positive cross-helicity, class 1 and 3, we associate class 3 to coronal hole origin and class 1 to streamer belt origin. Our class 3, maps quite closely to the 'red' class in Fig. 1 and 2 of (Roberts et al., 2020), associated there to coronal hole plasma from sectors of positive polarity. Class 3 indeed shows high absolute values of cross-helicity and near-zero values of residual energy, as expected form Alfvénic wind from coronal holes. We notice, however, that its O^{7+}/O^{6+} ratio, velocity, density, proton entropy and proton temperature values are somehow less 'coronal hole-like' than the

ones observed in class 7 for the opposite polarity. A qualitative difference between the wind from sectors of opposite magnetic polarity can be seen in Fig. 7, and is already remarked upon in (Roberts et al., 2020). While we are quite surprised that it extended to the large time interval that we used to generate the SOM, we plan to conduct further analysis on the topic in the future.

We will perform further refinements of the model and its interpretation in a future work. These preliminary results show the great potential of the techniques introduced in this paper. SOMs show the variability of solar wind and how it can be visually characterized. The SOM is a helpful guide in the study of the different types of solar wind, but is not necessarily an objective, unbiased and final classification method. SOMs open the possibility for a fast visual characterization of large and complex data sets.

The short analysis of the different SOM classes performed above, was also informed by the data presented in Fig.8. Each row corresponds to a single solar wind class, represented by its 21 features using box plots. The colors correspond to the class number (0 to 7 from top to bottom). The first column has been built from a classification using k-means, the second with GMM and the third with DSOM. Here we can find again the properties described in the previous section. We call these plots class 'fingerprints'.

Different classification methods lead to different classes with different fingerprints. A visual inspection of the fingerprints is much more difficult to interpret than the SOMs. (Roberts et al., 2020) and (Xu and Borovsky, 2015) performed detailed descriptions of particular solar wind classes based on the mean values observed in each subset of points. But Fig.8 shows that some features can have a very large distributions. For example the values of the solar wind speed, feature 2, have a very large spread on all the classes and all classifications, except for class 7 in the GMM classification. Other features with large fingerprint spread includes the cross-helicity (6) and the residual energy (7). These are a consequence of the bimodal nature of the two features, as shown in the violin plots of Fig.5. This bimodal distribution plays a very critical role in the separation of classes, as shown in our results and the results presented by (Roberts et al., 2020). Depending on the expected use of the solar wind classification, using the raw values of σ_c might not be recommended.

It is noticeable that in the k-means classification multiple classes have very similar fingerprints, at the exception of a single feature. For example classes 6 and 7 have similar characteristics except for features 6 (cross-helicity). The same is true for classes 0 and 2 in the GMM classification, where the largest difference is the spread of feature 20. SOM classes tend to present fingerprints that are more variable. This is a consequence of the use of the dynamic version of the SOM that does not agglomerate nodes in zones with high density of points. On the contrary, k-means and GMM will tend to put more points in high density zones, creating very similar class fingerprints.

3.1.2.1 Maps of Empirical Classifications

SOM allows visual analysis of previously published results. In this section we show how the Xu and Zhao classifications activate different nodes of the SOM. We use two properties of the SOM simultaneously: the size of the lattice nodes will represent the number of hits for a particular class, and the color will represent one property of the solar wind.

To perform this analysis, instead of using the full data set, we extract 3 subsets corresponding to the entries categorized as CHW, ICME and NCHW in the Zhao catalog. Each one of these three subsets is passed through the Amaya-21 model and we observe how each one activates the SOMs. All properties are normalized between zero and one, using the maximum and minimum values for each feature in the full data set, so we can perform comparisons among all the subsets.

Fig.9 shows the SOMs of the three Zhao classes produced by the Amaya-21 model. CHW, ICME and NCHW classes have different number of hits in the SOM. If these solar wind types were clustered in different SOM classes, they would activate different nodes in the lattice. However, class CHW and NCHW activate very similar regions in this model's map. The colors in each map are homogeneous, demonstrating that all points in the class share similar solar wind properties. The values of oxygen ion state ratio and the solar wind speed do not seem to play an important role in the automatic classification of our model and both classes activate similar nodes with similar solar wind properties. The goal of our team for our next publication is to work in collaboration with solar experts to design a more sophisticated model that can accurately reproduce the Zhao and Xu classes in our maps.

In these maps it is clear that the ICME class is mainly contained in the zone corresponding to class 5, which has been previously identified as such in the map and time series analysis above. Here the total number of hits is only 445, which explains why it is so difficult to observe in the time series. This is an additional benefit of using SOMs: we are able to detect important data points that can easily be overlooked with other methods.

In a similar way, Fig.10 shows the SOMs of the Xu classification. This time we used four subsets of the data set, each one corresponding to a different Xu class. Once again 'ejecta' is confined to the region of SOM class 5, and 'sector reversal origin' solar wind activates some of the nodes corresponding to the non-coronal hole wind in the Zhao classification. This same zone is also overlapping with 'streamer belt origin' zones in the Xu classification. This class seems to be included in the SOM class 2, and in part in class 3. It is possible to isolate singular nodes and study more in detail all their characteristics, but this is out of the scope of the present work and will be presented in a future publication.

The separation of the Xu classes is also not perfect in this model. We have tested other models in which the separation is more clear. Those models were based on different number of features and time ranges. An example of such model is presented in the next section.

3.1.3 Model Roberts-8

The same techniques used for model Amaya-21 were applied to this model. The only difference between these two is the amount of data used and the selected initial features. We can see that these two modifications can have an important impact in the final results. Fig.11 (A) shows the volume integrated point density and the distribution of the code words. This is a projection in the latent space after transformation using an autoencoder. The 'hit map' and the 'feature map' shows a clear segregation of points, allowing for a proper splitting of the data set.

The time series in Fig.11 shows that the model can differentiate zones of high and low speed, as well as zones of polarity inversion and some of the shocks and ICMEs. The use of less features gives more dominance to properties with larger distribution spread, like the cross-helicity. It is clear from the time series that class 4 and 5 are dominant northwards and southwards of the HCS respectively. Class 3 in this model is associated with transient events, including ICMEs and sector reversals.

For simplicity we will not present a detailed analysis of the Roberts-8 model. We would like only to point an important difference with model Amaya-21: Fig.12 shows how the solar wind classifications by Xu and Zhao are interpreted by the model. First, it is important to notice that all classes activate nodes predominantly in different SOM classes. Second, the small variations in colors inside each map demonstrates that the classes are well represented by the main properties proposed by Xu and Zhao. One exception is the class 'ejecta' that shows uneven values of proton specific entropy, S_p, and temperature ratio T_{exp}/T_p.

Coronal hole solar wind classes by Xu and Zhao activate exactly the same nodes in the map, corresponding to class 4 in the Roberts-8 model. ICME from Zhao is considered as a subset of the ejecta class from Xu, while the NCHW class from Zhao contains the sector reversal class from Xu.

A careful selection of features and the data range of the models can produce a particularly powerful tool for the analysis of solar wind information. We are currently working towards the creation of an accurate solar wind classification system based on the developments presented here.

4 DISCUSSION

In this paper we show how the categorization of solar wind can be informed by classic unsupervised clustering methods and Self-Organizing Maps (SOM). We demonstrate that a single technique used in isolation can be misleading for the interpretation of automatic classifications. We show that it is important to examine the SOM lattices, in conjunction with the solar wind fingerprints and the time series. Thanks to these tools we can differentiate classes associated with known heliospheric events.

We are convinced that basic unsupervised clustering techniques will have difficulties in finding characteristic solar wind classes when they are applied to unprocessed data. A combination of feature engineering, non-linear autoencoding and SOM training leads to a more appropriate segmentation of the data points.

The classification of the solar wind also depends on the objectives that want to be attained: if the goal is to classify the solar wind to study its origin on the Sun, features related to solar activity must be included in the model; however, if the goal is to identify geoeffectiveness, other parameters should be added to the list of features, including geomagnetic indices.

In this work we have presented a first test of the capabilities of the SOMs for the analysis of data from a full solar cycle. Due to the extent of the work done, in this paper we introduce all the methods and techniques developed, but we leave for a future publication a detailed selection of all the model parameters, and the corresponding detailed physical interpretation of the solar wind classification.

All the tools and the techniques presented here can be applied to any other data set consisting of large amounts of points with a fixed number of properties. SOMs have already been used in astrophysics(Süveges et al., 2017) and magnetospheric physics(Camporeale et al., 2018), and we are currently working on their deployment for the study of active regions on the Sun.

All the software and the data used in this work are freely available for reproduction and improvement of the results presented above.

5 ADDITIONAL REQUIREMENTS

CONFLICT OF INTEREST STATEMENT

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

AUTHOR CONTRIBUTIONS

JA created the software used in this work, built the data sets and wrote the manuscript. RD provided important insights into the use of the machine learning techniques, and performed revisions of the different drafts. MEI gathered information from external collaborators, provided insights into the data usage, and proofread different drafts. GL supervised the work. All authors contributed to manuscript revision, read and approved the submitted version.

FUNDING

The work presented in this manuscript has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 754304 (DEEP-EST, www.deep-projects.eu), and from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776262 (AIDA, www.aida-space.eu).

ACKNOWLEDGMENTS

The authors would like to acknowledge the helpful advice and suggestions by Olga Panasenco (UCLA), Raffaella D'Amicis (INAF-IAPS), and Aaron Roberts (NASA-GSFC).

SUPPLEMENTAL DATA

All the software and the data used in this work can be found in the following git repository: github.com/murci3lag0.

All original source code is distributed under MIT license.

DATA AVAILABILITY STATEMENT

All processed data, including edited catalogs and enhanced data sets are available in our git repository: github.com/murci3lag0.

The original ACE data sets were downloaded from ftp://mussel.srl.caltech.edu/pub/ace/level2/multi.

A detailed description of all ACE multi-instrument data set entires can be found here: http://www.srl.caltech.edu/cgi-bin/dib/rundibviewmultil2/ACE/ASC/DATA/level2/multi

REFERENCES

Adhikari, L., Zank, G. P., Zhao, L.-L., Kasper, J. C., Korreck, K. E., Stevens, M., et al. (2020). Turbulence Transport Modeling and First Orbit Parker Solar Probe (PSP) Observations . *The Astrophysical Journal Supplement Series* 246, 38. doi:10.3847/1538-4365/ab5852

Ahmed, O. W., Qahwaji, R., Colak, T., Higgins, P. A., Gallagher, P. T., and Bloomfield, D. S. (2013). Solar Flare Prediction Using Advanced Feature Extraction, Machine Learning, and Feature Selection. *Solar Physics* 283, 157–175. doi:10.1007/s11207-011-9896-1

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 2623–2631doi:10.1145/3292500.3330701

Bai, Y., Liu, J., Wang, S., and Yang, F. (2018). Machine Learning Applied to Star–Galaxy–QSO Classification and Stellar Effective Temperature Regression. *The Astronomical Journal* 157, 9. doi:10.3847/1538-3881/aaf009

Bale, S. D., Badman, S. T., Bonnell, J. W., Bowen, T. A., Burgess, D., Case, A. W., et al. (2019). Highly structured slow solar wind emerging from an equatorial coronal hole. *Nature* 576, 237–242. doi:10.1038/s41586-019-1818-7

Bandyopadhyay, R., Goldstein, M. L., Maruca, B. A., Matthaeus, W. H., Parashar, T. N., Ruffolo, D., et al. (2020). Enhanced Energy Transfer Rate in Solar Wind Turbulence Observed near the Sun from Parker Solar Probe . *The Astrophysical Journal Supplement Series* 246, 48. doi:10.3847/1538-4365/ab5dae

Bergstra, J., Yamins, D., and Cox, D. (2013). Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In *Proceedings of the 30th International Conference on Machine Learning*, eds. S. Dasgupta and D. McAllester (Atlanta, Georgia, USA: PMLR), vol. 28 of *Proceedings of Machine Learning Research*, 115–123

Bishop, C. M. (2006). Machine learning and pattern recognition. *Information science and statistics. Springer, Heidelberg*

Bobra, M. G. and Couvidat, S. (2015). Solar flare prediction using SDO/HMI vector magnetic field data with a machine-learning algorithm. *Astrophysical Journal* 798, 135. doi:10.1088/0004-637X/798/2/135

Bobra, M. G. and Ilonidis, S. (2016). Predicting Coronal Mass Ejections Using Machine Learning Methods. *The Astrophysical Journal* 821, 127. doi:10.3847/0004-637X/821/2/127

Bonjean, V., Aghanim, N., Salome, P., Beelen, A., Douspis, M., and Soubrie, E. (2019). Star formation rates and stellar masses from machine learning. *Astronomy and Astrophysics* 622, 1–12. doi:10.1051/0004-6361/201833972

Burlaga, L., Sittler, E., Mariani, F., and Schwenn, R. (1981). Magnetic loop behind an interplanetary shock: Voyager, Helios, and IMP 8 observations. *Journal of Geophysical Research: Space Physics* 86, 6673–6684. doi:10.1029/JA086iA08p06673

Camporeale, E., Carè, A., and Borovsky, J. E. (2017). Classification of Solar Wind with Machine Learning. *Journal of Geophysical Research: Space Physics* doi:10.1002/2017JA024383

Camporeale, E., Wing, S., and Johnson, J. R. (2018). *Machine learning techniques for space weather* (Elsevier)

Cane, H. V. and Richardson, I. G. (2003). Interplanetary coronal mass ejections in the near-Earth solar wind during 1996-2002. *Journal of Geophysical Research: Space Physics* 108. doi:10.1029/2002JA009817

D'Amicis, R. and Bruno, R. (2015). On The Origin of Higly Alfvénic Slow Solar Wind. *The Astrophysical Journal* 805, 84. doi:10.1088/0004-637X/805/1/84

Dodge, J., Gururangan, S., Card, D., Schwartz, R., and Smith, N. A. (2019). Show Your Work: Improved Reporting of Experimental Results

Dupuis, R., Goldman, M. V., Newman, D. L., Amaya, J., and Lapenta, G. (2020). Characterizing Magnetic Reconnection Regions Using Gaussian Mixture Models on Particle Velocity Distributions. *The Astrophysical Journal* 889, 22. doi:10.3847/1538-4357/ab5524

Elsasser, W. M. (1950). The Hydromagnetic Equations. *Phys. Rev.* 79, 183. doi:10.1103/PhysRev.79.183

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., and Others (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*. vol. 96, 226–231

Feldman, U., Landi, E., and Schwadron, N. A. (2005). On the sources of fast and slow solar wind. *Journal of Geophysical Research: Space Physics* 110. doi:10.1029/2004JA010918

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative Adversarial Networks

Gopalswamy, N., Hanaoka, Y., Kosugi, T., Lepping, R. P., Steinberg, J. T., Plunkett, S., et al. (1998). On the relationship between coronal mass ejections and magnetic clouds. *Geophysical Research Letters* 25, 2485–2488. doi:10.1029/98GL50757

Gosling, J. T., Teh, W. L., and Eriksson, S. (2010). A torsional Alfvén wave embedded within a small magnetic flux rope in the solar wind. *Astrophysical Journal Letters* 719, 36–40. doi:10.1088/2041-8205/719/1/L36

Habbal, S. R., Woo, R., Fineschi, S., O'Neal, R., Kohl, J., Noci, G., et al. (1997). Origins of the Slow and the Ubiquitous Fast Solar Wind. *The Astrophysical Journal* 489, L103–L106. doi:10.1086/310970

Hajian, A., Alvarez, M. A., and Bond, J. R. (2015). Machine learning etudes in astrophysics: selection functions for mock cluster catalogs. *Journal of Cosmology and Astroparticle Physics* 2015, 038–038. doi:10.1088/1475-7516/2015/01/038

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science* 313, 504 LP – 507. doi:10.1126/science.1127647

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering* 9, 90–95

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes

Liang, J. and Liu, R. (2015). Stacked denoising autoencoder and dropout together to prevent overfitting in deep neural network. In *2015 8th International Congress on Image and Signal Processing (CISP)*. 697–701. doi:10.1109/CISP.2015.7407967

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory* 28, 129–137. doi:10.1109/TIT.1982.1056489

Lundstedt, H., Wintoft, P., Wu, J. G., Gleisner, H., and Dovheden, V. (1996). Space Environment Modelling with the Use of Artificial Intelligence Methods. In *Environment Modeling for Space-Based Applications*. vol. 392, 269

Magyar, N., Van Doorsselaere, T., and Goossens, M. (2019). The Nature of Elsässer Variables in Compressible MHD. *The Astrophysical Journal* 873, 56. doi:10.3847/1538-4357/ab04a7

Matteini, L., Hellinger, P., Landi, S., Trávníček, P. M., and Velli, M. (2012). Ion kinetics in the solar wind: Coupling global expansion to local microphysics. *Space Science Reviews* 172, 373–396. doi:10.1007/s11214-011-9774-z

McKinney, W. (2010). {D}ata {S}tructures for {S}tatistical {C}omputing in {P}ython. In *{P}roceedings of the 9th {P}ython in {S}cience {C}onference*, eds. S. van der Walt and J. Millman. 56–61. doi:10.25080/Majora-92bf1922-00a

Nishizuka, N., Sugiura, K., Kubo, Y., Den, M., Watari, S., and Ishii, M. (2017). Solar Flare Prediction Model with Three Machine-learning Algorithms using Ultraviolet Brightening and Vector Magnetograms. *The Astrophysical Journal* 835, 156. doi:10.3847/1538-4357/835/2/156

Ntampaka, M., Trac, H., Sutherland, D. J., Battaglia, N., Póczos, B., and Schneider, J. (2015). A Machine Learning Approach for Dynamical Mass Measurements of Galaxy Clusters. *The Astrophysical Journal* 803, 50. doi:10.1088/0004-637X/803/2/50

Oliphant, T. E. (2006). *A guide to NumPy*, vol. 1 (Trelgol Publishing USA)

Owens, M. J. (2016). Do the Legs of Magnetic Clouds Contain Twisted Flux-Rope Magnetic Fields? *The Astrophysical Journal* 818, 197. doi:10.3847/0004-637x/818/2/197

Parker, E. N. (1958). Interaction of the Solar Wind with the Geomagnetic Field. *The Physics of Fluids* 1, 171–187. doi:10.1063/1.1724339

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research* 12, 2825–2830

Pierrard, V. and Lazar, M. (2010). Kappa Distributions: Theory and Applications in Space Plasmas. *Solar Physics* 267, 153–174. doi:10.1007/s11207-010-9640-2

Qahwaji, R., Colak, T., Al-Omari, M., and Ipson, S. (2008). Automated Prediction of CMEs Using Machine Learning of CME-Flare Associations. *Solar Physics* 248, 471–483. doi:10.1007/s11207-007-9108-1

Rea, A. and Rea, W. (2016). How Many Components should be Retained from a Multivariate Time Series PCA?

Richardson, I. G. and Cane, H. V. (2010). Near-earth interplanetary coronal mass ejections during solar cycle 23 (1996 - 2009): Catalog and summary of properties. *Solar Physics* 264, 189–237. doi:10.1007/s11207-010-9568-6

Richardson, I. G. and Cane, H. V. (2012). Near-earth solar wind flows and related geomagnetic activity during more than four solar cycles (1963-2011). *Journal of Space Weather and Space Climate* 2. doi:10.1051/swsc/2012003

Richardson, I. G., Cliver, E. W., and Cane, H. V. (2000). Sources of geomagnetic activity over the solar cycle: Relative importance of coronal mass ejections, high-speed streams, and slow solar wind. *Journal of Geophysical Research: Space Physics* 105, 18203–18213. doi:10.1029/1999JA000400

Roberts, D. A., Karimabadi, H., Sipes, T., Ko, Y.-K., and Lepri, S. (2020). Objectively Determining States of the Solar Wind Using Machine Learning. *The Astrophysical Journal* 889, 153. doi:10.3847/1538-4357/ab5a7a

Rougier, N. and Boniface, Y. (2011). Dynamic self-organising map. *Neurocomputing* 74, 1840–1847. doi:10.1016/J.NEUCOM.2010.06.034

Sabine, E. (1852). VIII. On periodical laws discoverable in the mean effects of the larger magnetic disturbance.—No. II. *Philosophical Transactions of the Royal Society of London* 142, 103–124. doi:10.1098/rstl.1852.0009

Satopaa, V., Albrecht, J., Irwin, D., and Raghavan, B. (2011). Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*. 166–171. doi:10.1109/ICDCSW.2011.20

Shlens, J., View, M., and Introduction, I. (2014). A Tutorial on Principal Component Analysis

Stakhiv, M., Landi, E., Lepri, S. T., Oran, R., and Zurbuchen, T. H. (2015). On the Origin of Mid-Latitude Fast Wind: Challenging the Two-State Solar Wind Paradigm. *The Astrophysical Journal* 801, 100. doi:10.1088/0004-637X/801/2/100

Süveges, M., Barblan, F., Lecoeur-Taïbi, I., Prša, A., Holl, B., Eyer, L., et al. (2017). Gaia eclipsing binary and multiple systems. Supervised classification and self-organizing maps. *Astronomy & Astrophysics* 603, A117. doi:10.1051/0004-6361/201629710

VanderPlas, J., Connolly, A. J., Ivezić, Ž., and Gray, A. (2012). Introduction to astroML: Machine learning for astrophysics. In *2012 Conference on Intelligent Data Understanding*. 47–54. doi:10.1109/CIDU.2012.6382200

Vettigli, G. (2013). MiniSom: minimalistic and numpy based implementation of the self organizing maps

Villmann, T. and Claussen, J. C. (2006). Magnification Control in Self-Organizing Maps and Neural Gas. *Neural Computation* 18, 446–469. doi:10.1162/089976606775093918

Winterhalter, D., Smith, E. J., Burton, M. E., Murphy, N., and McComas, D. J. (1994). The heliospheric plasma sheet. *Journal of Geophysical Research: Space Physics* 99, 6667–6680. doi:10.1029/93JA03481

Withbroe, G. L. (1986). Origins of the Solar Wind in the Corona BT - The Sun and the Heliosphere in Three Dimensions (Dordrecht: Springer Netherlands), 19–32

Wu, C. C., Lepping, R. P., and Gopalswamy, N. (2006). Relationships Among Magnetic Clouds, CMES, and Geomagnetic Storms. *Solar Physics* 239, 449–460. doi:10.1007/s11207-006-0037-1

Xu, F. and Borovsky, J. E. (2015). A new four-plasma categorization scheme for the solar wind. *Journal of Geophysical Research: Space Physics* 120, 70–100. doi:10.1002/2014JA020412

Zhao, L., Zurbuchen, T. H., and Fisk, L. A. (2009). Global distribution of the solar wind during solar cycle 23: ACE observations. *Geophysical Research Letters* 36, 1–4. doi:10.1029/2009GL039181

Zhao, L. L., Adhikari, L., Zank, G. P., Hu, Q., and Feng, X. S. (2018). Analytical investigation of turbulence quantities and cosmic ray mean free paths from 1995-2017. *Journal of Physics: Conference Series* 1100. doi:10.1088/1742-6596/1100/1/012029

#	Solar wind type name	Condition		Reference
1	Coronal hole wind	$O^{7+}/O^{6+} \leq$	0.145	(Zhao et al., 2009)
2	ICMEs	$O^{7+}/O^{6+} >$	$6.008e^{(-0.00578V_{sw})}$	
4	Non-coronal hole wind	$0.145 <$	$O^{7+}/O^{6+} < 6.008e^{(-0.00578V_{sw})}$	
0	Streamer belt origin	Not type 1, 2, or 3		(Xu and Borovsky, 2015)
1	Coronal hole origin	Not type 2, and $\log_{10}(S_p) >$	$-0.525\log_{10}(T_{\text{exp}}/T_p)$ $-0.676\log_{10}(V_A)$ $+1.74$	
2	Ejecta	$\log_{10}(V_A) >$	$0.055\log_{10}(T_{\text{exp}}/T_p)$ $+0.277\log_{10}(S_p)$ $+1.83$	
3	Sector reversal origin	Not type 2, and $\log_{10}(S_p) <$	$-0.125\log_{10}(T_{\text{exp}}/T_p)$ $-0.658\log_{10}(V_A)$ $+1.04$	

Table 1. The four solar wind types in the Zhao and Xu classification. ID numbers are arbitrary. Only two types overlap in both classifications: classes 1 and 2.

TABLES

FIGURE CAPTIONS

ID	Name	Amaya-21	Roberts-8
1	O7to6	✓	✓
2	proton_speed	✓	✓
3	proton_density	✓	✓
4	FetoO	✓	✓
5	avqFe	✓	✓
6	sigmac$^{(*)}$	✓	✓
7	sigmar$^{(*)}$	✓	✓
8	Bmag	✓	✓
9	Sp	✓	
10	Va	✓	
11	Tratio	✓	
12	proton_temp	✓	
13	Ma	✓	
14	C6to5	✓	
15	proton_density_range	✓	
16	proton_temp_range	✓	
17	Bx_range	✓	
18	Bz_range	✓	
19	Bmag_range	✓	
20	Bmag_acor$^{(*)}$	✓	
21	Delta$^{(*)}$	✓	
	Initial year	1998	2002
	Final year	2011	2004
	Neurons / encoding layer	[21, 17, 9, 3]	[8, 13, 7, 3]
	Lattice nodes	8x8	7x7
	ϵ	0.39	0.73
	η	4.07	4.14

Table 2. List of features used for each model. The logarithm of all quantities was used, except for the features marked with an asterisk (*). Bottom: data range and hyper-parameters of the SOMs.

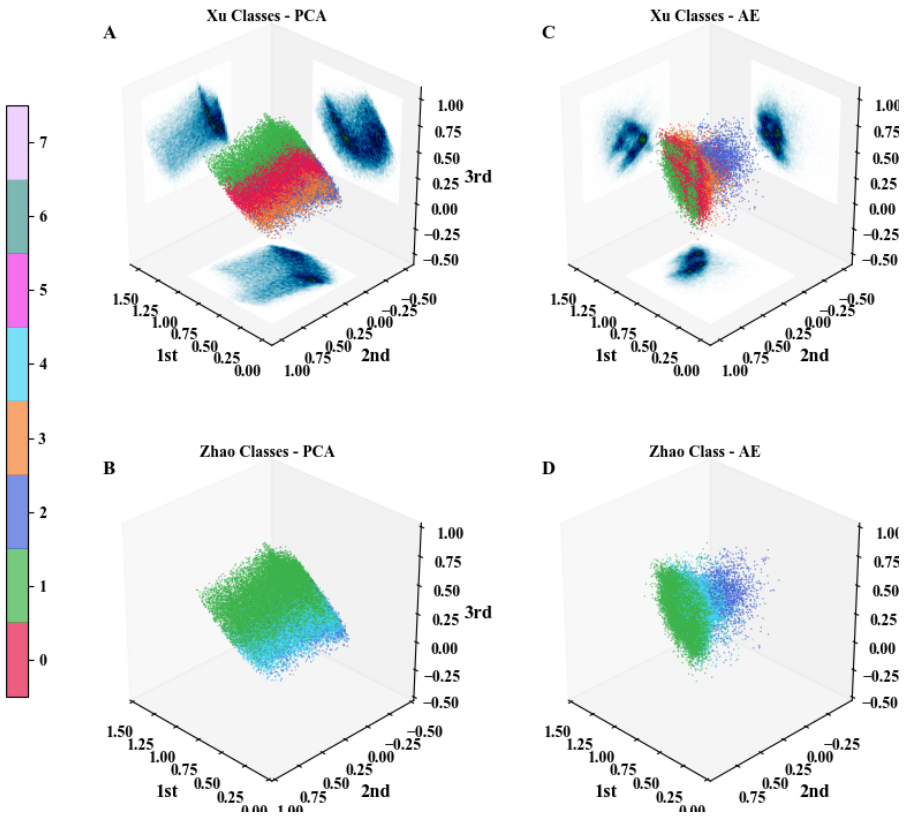

Figure 1. Dimension reduction scatter plot in 3D. (A): data set projected on the three coordinates of the reduced PCA space. The density of points is projected in 2D histograms on the lateral planes. Colors correspond to the Xu classification. (B): same plot as (A), but without the density histograms. Colors correspond to the Zhao classification. (C): data set projected on the three coordinates of the latent AE space. The density of points is projected in 2D histograms on the lateral planes. Colors correspond to the Xu classification. (D): same as (C), but without the density histograms. Colors correspond to the Zhao classification.

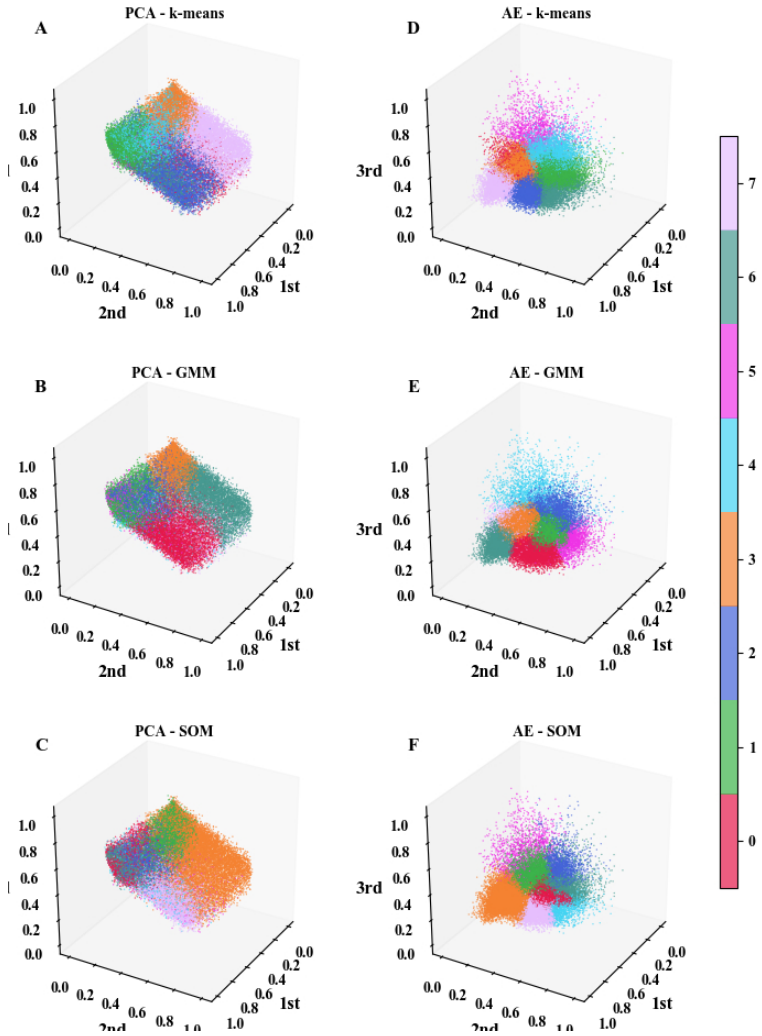

Figure 2. 3D Scatter plot of all data points projected on the transformed spaces. The left (right) column correspond to the PCA (autoencoder) projection. First row: k-means classification. Second row: Gaussian Mixture Model classification. Third row: Self-Organizing Maps classification. Colors correspond to the respective transformation and classification.

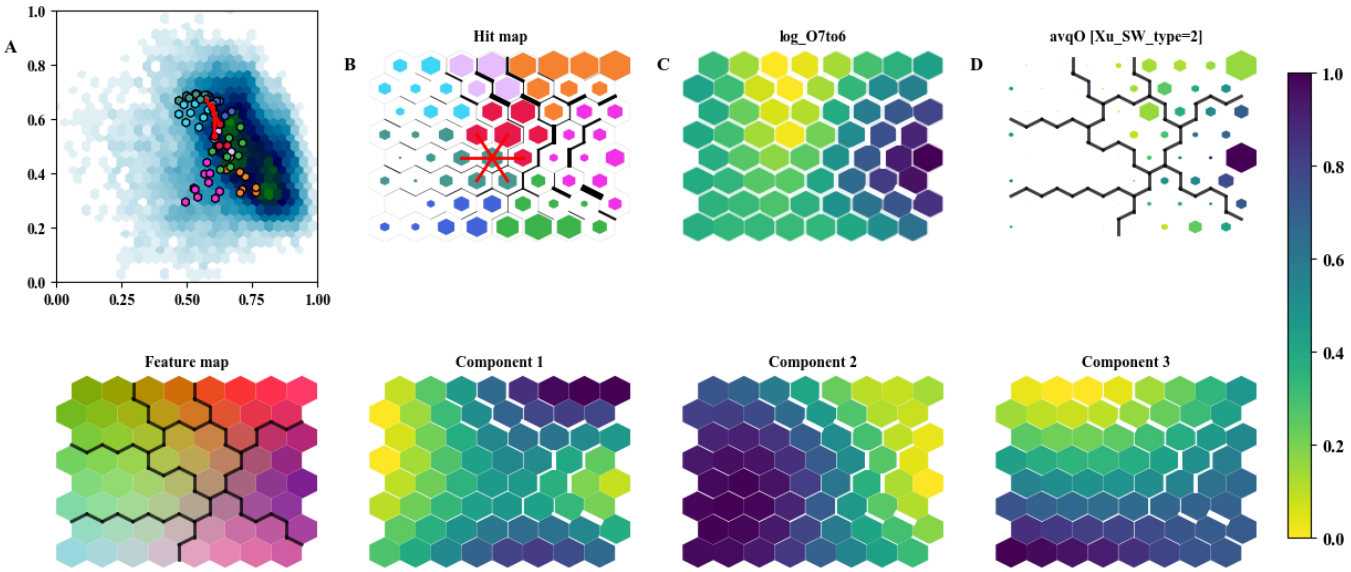

Figure 3. Visualization of the Self-Organizing maps. All quantities in panels (C), (D), and components 1 to 3, have been normalized between 0 and 1. Panel (A): histogram with the normalized density of data points superposed by the code words of the SOM, projected on the first two components of the latent space. A single node is connected to its closest neighbors by red lines. Panel (B) Hit map: the size of the hexagon corresponds to the number of data points associated to the node, and the color is their SOM class. Black lines between nodes correspond to their relative distance. Red lines connect the nodes similarly highlighted in panel (A). Panel (C): Value of one of the features of the model, the oxygen ion charge state ratio. Panel (D): example of the statistics applied to the points associated to map nodes, in this case mean value of 'avqO' in color, with solar wind type hits as hexagon size. Panels in the bottom row: the three components of the map nodes, where the first panel is a Red Green Blue (RGB) combination of the other three.

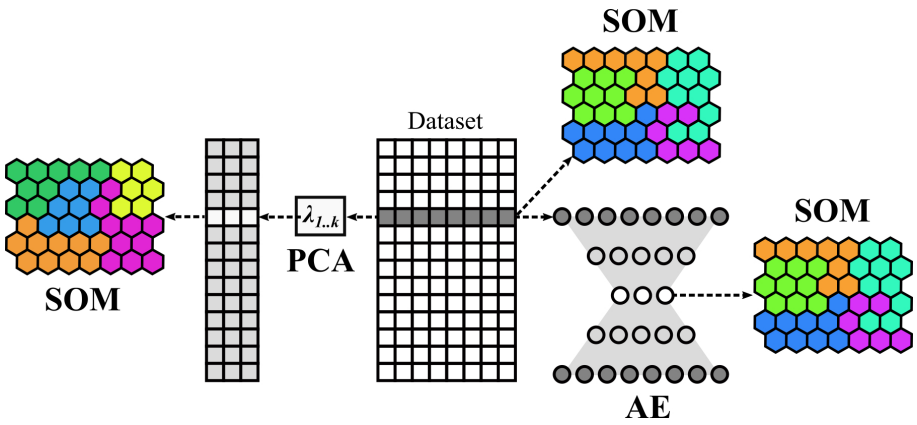

Figure 4. General architecture used in this project. Three types of processed data are presented to the SOM: a) the database is transformed using PCA, keeping only the most significant components, b) the data set entries are encoded into a latent space with a smaller number of dimensions, c) the data is directly presented to the SOM. Clustering techniques are used to group together nodes of the SOM lattice.

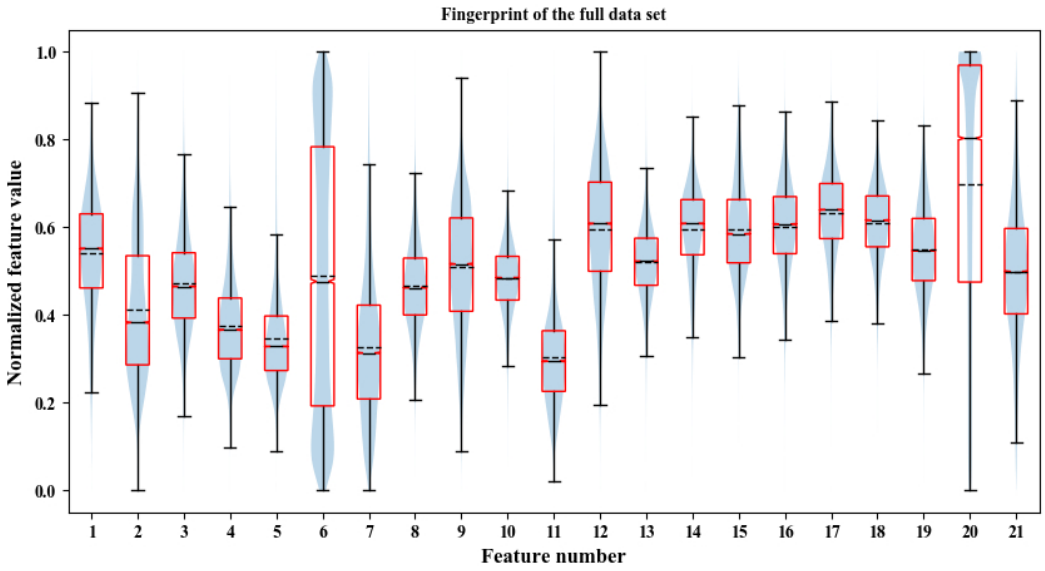

Figure 5. Composition of all the solar wind points. Violin plot superposed by a box plot of the solar wind features. The box height represents the Inter Quartile Range (IQR), the central continuous line represents the median and the dashed line the mean. The central notch represents the confidence of the median. The upper (lower) whiskers represent the lesser 25th (greater 75th) percentile of the data.

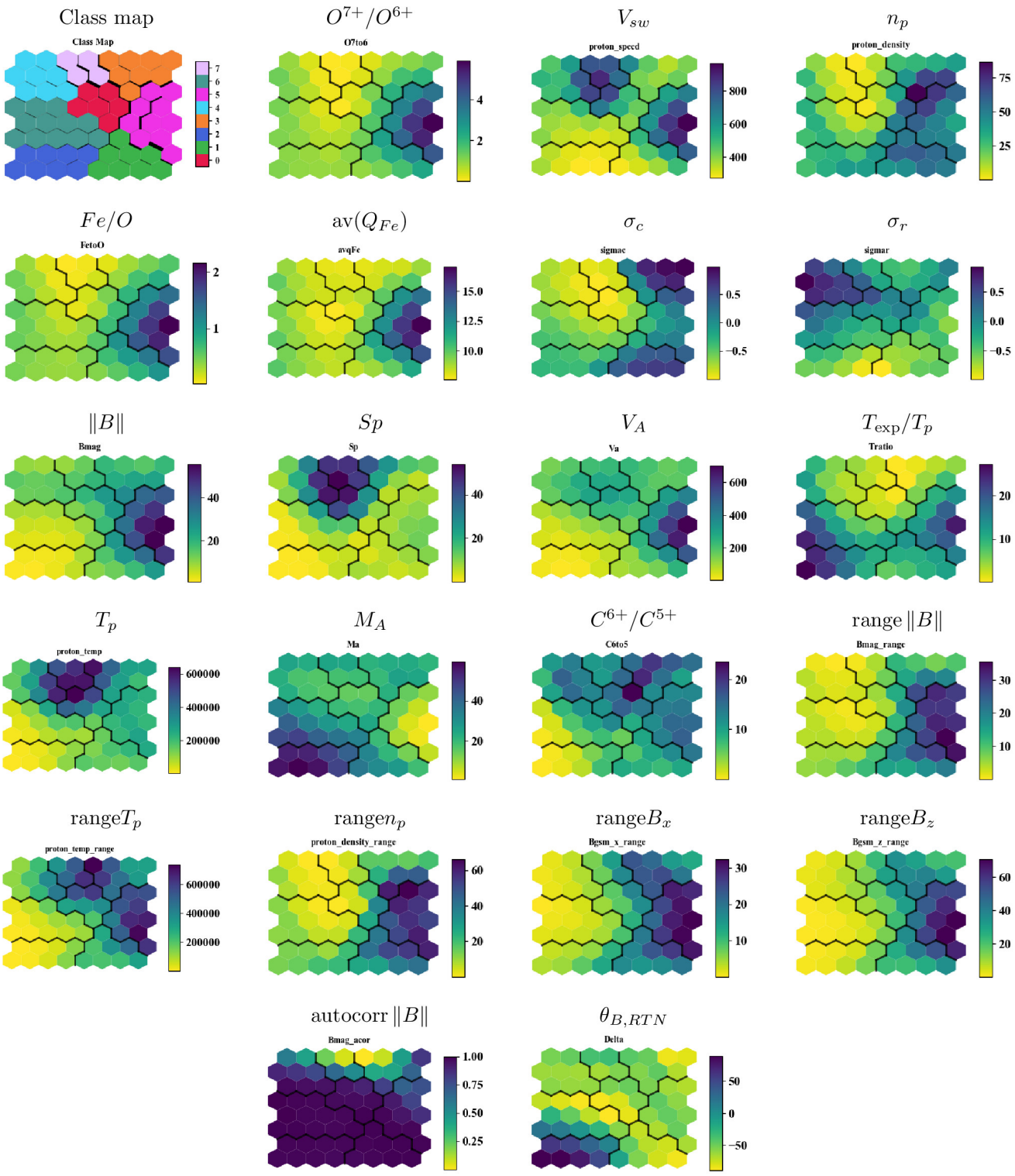

Figure 6. Map of the clustering of the SOM nodes, and map of the features used in model Amaya-21.

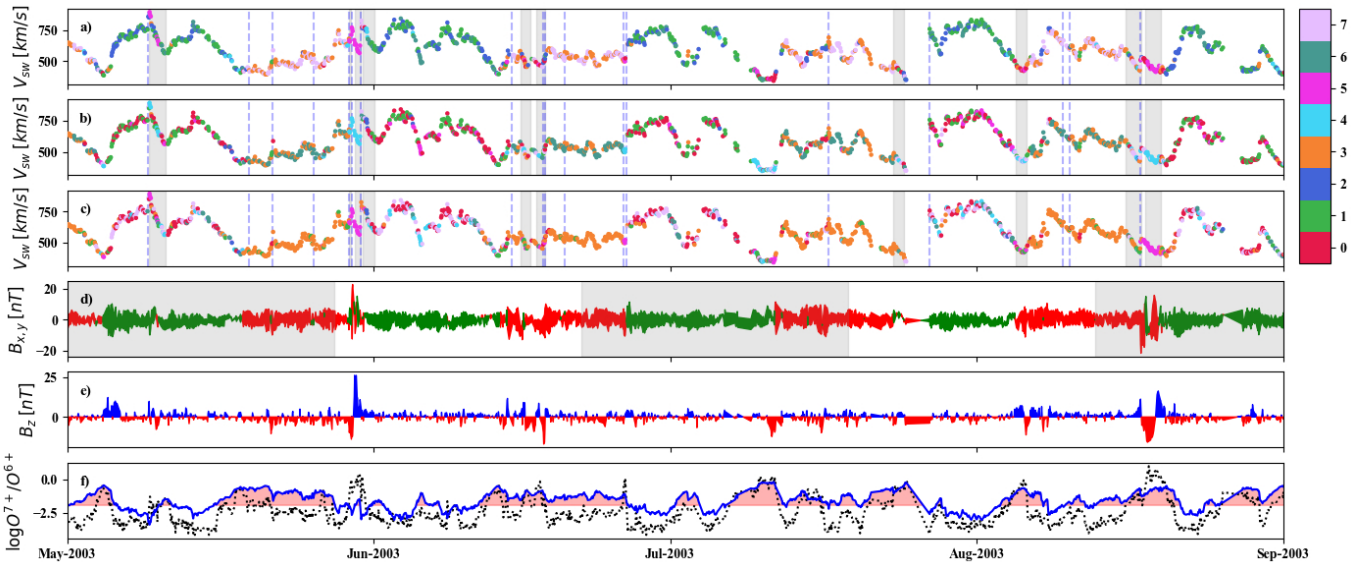

Figure 7. Four months of solar wind information. This is the same period used by (Roberts et al., 2020). Top three panels: solar wind speed colored by a) k-means classification, b) Gaussian Mixture Model classification, and c) DSOM classification. The colors correspond to the 'fingerprints' in Fig.8. Vertical gray zones correspond to Richardson and Cane ICME catalog entries, and vertical dashed lines to entries in the UNH and CfA catalogs. d) Magnetic field polarity color representation using $B_{x,y}$ as the top and bottom limits: red $B_x > B_y$, green $B_x < B_y$. The vertical gray zones correspond periods of 27 days. e) B_z component of the magnetic field: blue positive, red negative. f) plot of the ionized oxygen (dotted line), and the solar wind type limits from (Zhao et al., 2009) in table 1

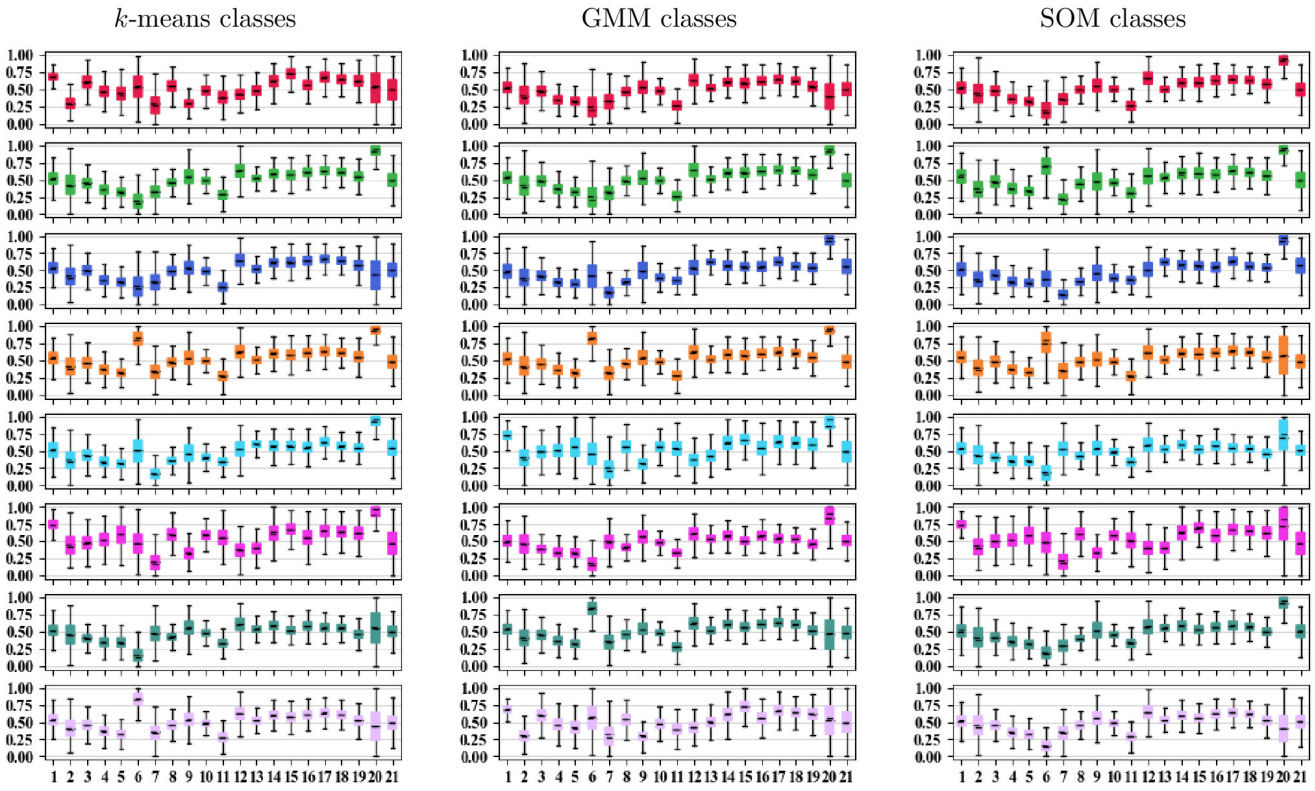

Figure 8. Solar wind 'fingerprint': composition of each class (one color per class), obtained with the k-means classification (left), the GMM classification (center), and the DSOM classification (right).

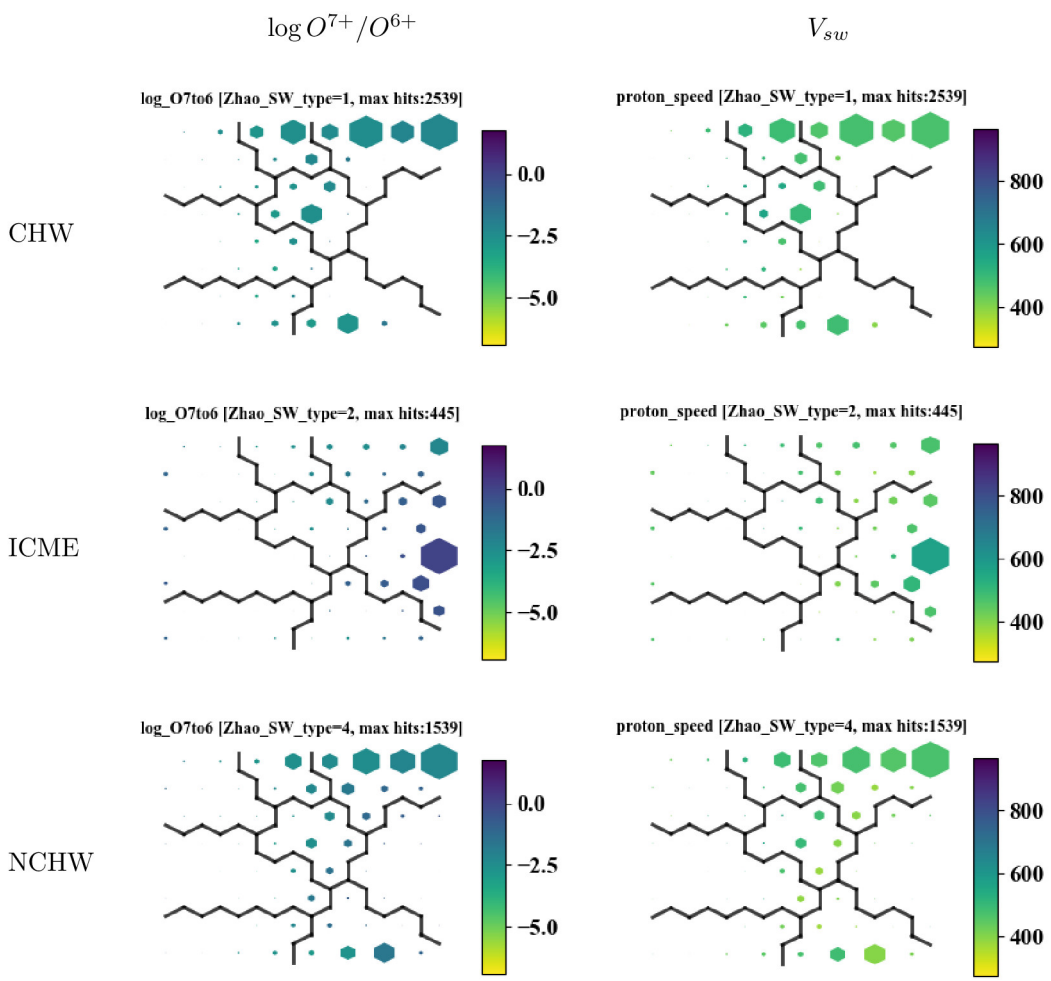

Figure 9. Maps for model Amaya-21 of the two features (one per column) used int the Zhao classification system. The size of the nodes represent the number of hits for each one of the three classes (rows). CHW: coronal hole wind, NCHW: non-coronal hole wind.

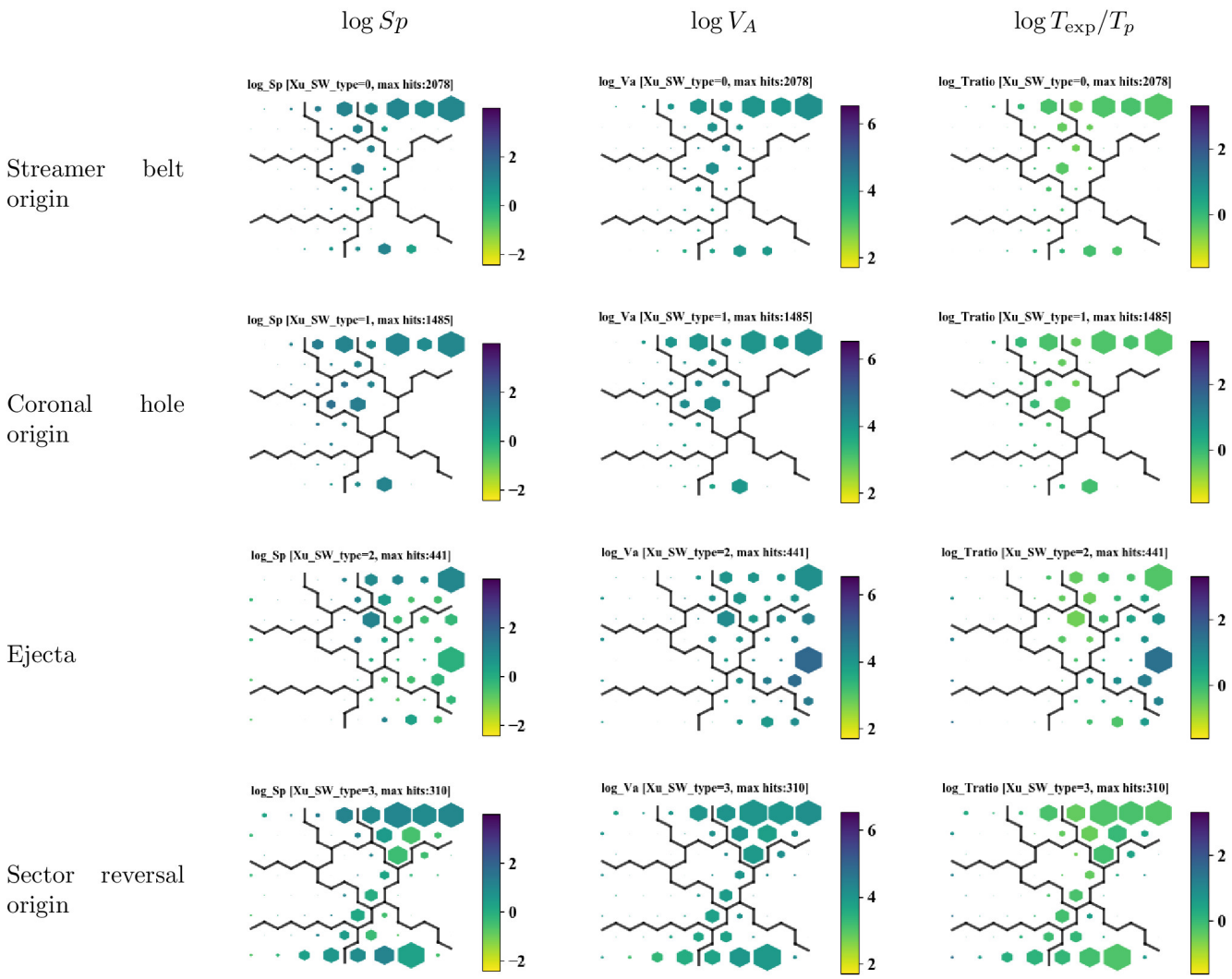

Figure 10. Maps for model Amaya-21 of the three features (one per column) used int the Xu classification system. The size of the nodes represent the number of hits for each one of the four classes (rows).

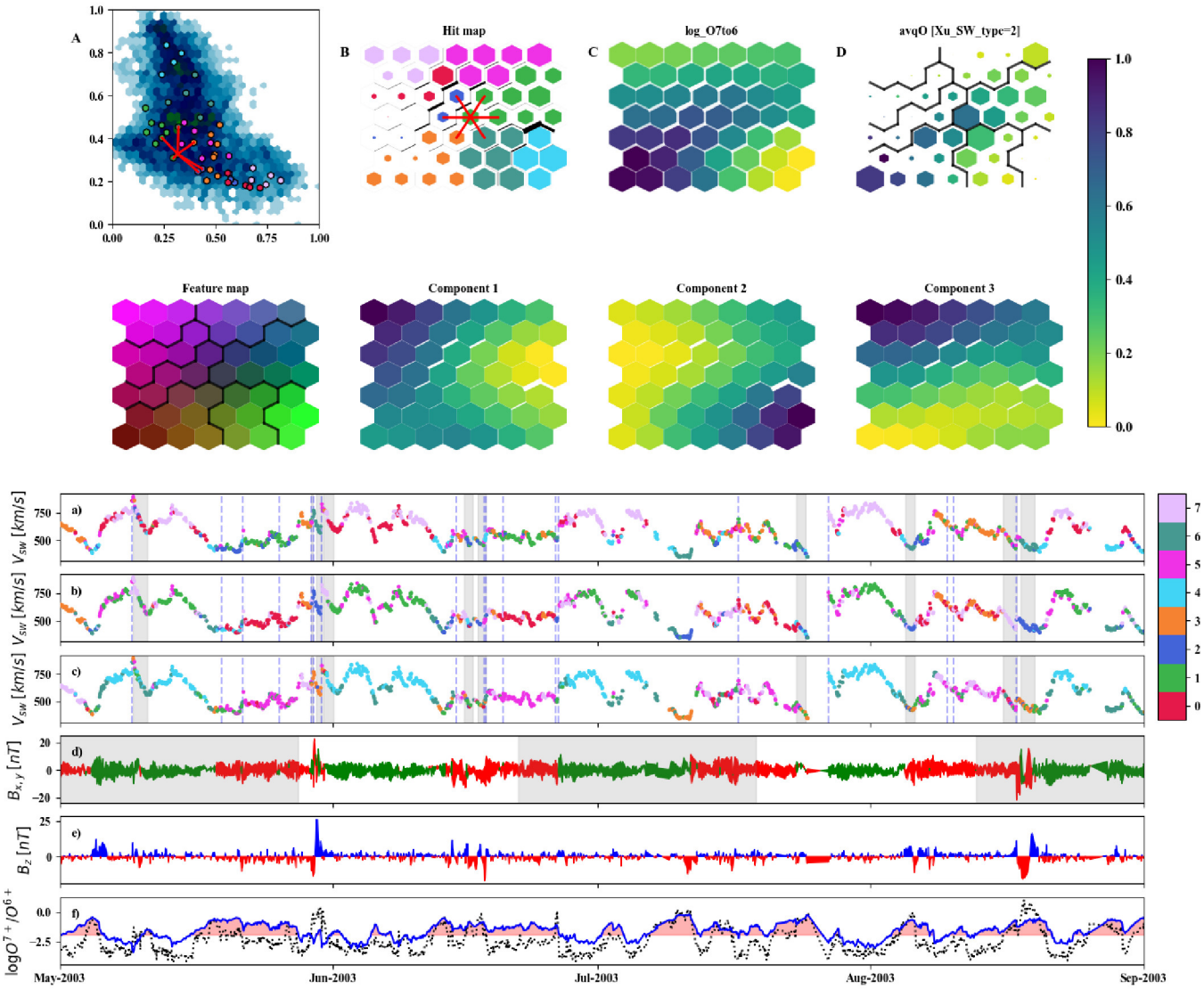

Figure 11. Visualization of the Self-Organizing map of model Roberts-8, and the corresponding time series example. See the captions of Fig.3 and 7 for a detailed description.

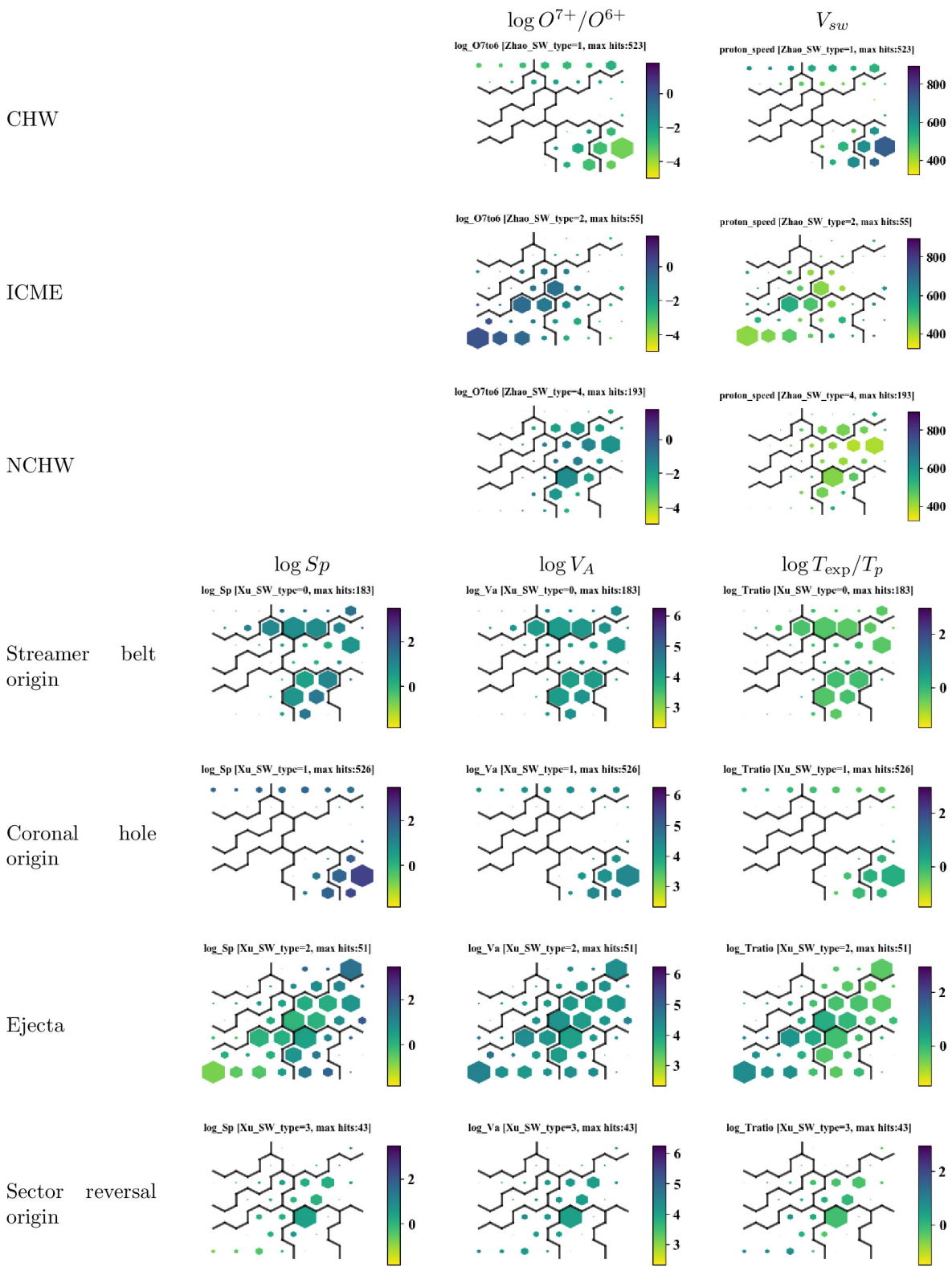

Figure 12. Maps for the Xu and Zhao classes in the Roberts-8 model. Node sizes indicate hits, colors indicate values of the corresponding solar wind property.