Forecasting Geomagnetic Storm Disturbances and Their Uncertainties using Deep Learning

Carlos Escobar Ibáñez[1], Daniel Eduardo Conde Villatoro[1], Florencia Luciana Castillo[2], Carmen García García[1], Jose Enrique García Navarro[1], Verónica Sanz González[1], Bryan Zaldívar Montero[1], Juan José Curto[3], Santiago Marsal[4], and Joan Miquel Torta[5]

[1]Instituto de Física Corpuscular
[2]Laboratoire d'Annecy de Physique des Particules
[3]Observatori de l'Ebre-CSIC
[4]Ebre Observatory
[5]Observatori de l'Ebre

March 9, 2023

Abstract

Severe space weather produced by disturbed conditions on the Sun results in harmful effects both for humans in space and in high-latitude commercial flights, and for technological systems such as spacecraft or communications. Also, geomagnetically induced currents flowing on long ground-based conductors, such as power networks or pipelines, potentially threaten critical infrastructures on Earth. The first step in developing an alarm system against geomagnetically induced currents is to forecast them. This is a challenging task, though, given the highly non-linear dependencies of the response of the magnetosphere to these perturbations. In the last few years, modern machine-learning models have shown to be very good at predicting magnetic activity indices as the SYM-H. However, such complex models are on the one hand difficult to tune, and on the other hand they are known to bring along potentially large prediction uncertainties which are generally difficult to estimate. In this work we aim at predicting the SYM-H index characterising geomagnetic storms one hour in advance, using public interplanetary magnetic field data from the Sun–Earth L1 Lagrange point and SYM-H. We implement a type of machine-learning model called long short-term memory networks. Our scope is to estimate -for the first time to our knowledge- the prediction uncertainties coming from a deep-learning model in the context of space weather. The resulting uncertainties turn out to be sizeable at the critical stages of the geomagnetic storms. Our methodology includes as well an efficient optimisation of important hyper-parameters of the long short-term memory network and robustness tests.

Forecasting Geomagnetic Storm Disturbances and Their Uncertainties using Deep Learning

D. Conde[1], F. L. Castillo[2], C. Escobar[1], C. García[1], J. E. García[1], V. Sanz[1,3], B. Zaldívar[1], J. J. Curto[4], S. Marsal[4], J. M. Torta[4]

[1]Instituto de Física Corpuscular (IFIC), Centro mixto CSIC - Universitat de València, Valencia, Spain
[2]Laboratoire d'Annecy de Physique des Particules (LAPP), Université Grenoble Alpes, Université Savoie Mont Blanc, CNRS/IN2P3, Annecy, France
[3]Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, United Kingdom
[4]Observatori de l'Ebre (OE), CSIC - Universitat Ramon Llull, Roquetes, Spain

Key Points:

- An LSTM model is built to forecast the SYM-H index using interplanetary magnetic field measurements and past SYM-H values.
- The hyper-parameter optimisation and the robustness of the LSTM model is ensured by using dedicated algorithms and methods.
- Prediction uncertainties from the LSTM model are estimated and turn out to be considerable in the critical phases of geomagnetic storms.

Corresponding author: D. Conde, Daniel.Conde@ific.uv.es

Abstract

Severe space weather produced by disturbed conditions on the Sun results in harmful effects both for humans in space and in high-latitude commercial flights, and for technological systems such as spacecraft or communications. Also, geomagnetically induced currents flowing on long ground-based conductors, such as power networks or pipelines, potentially threaten critical infrastructures on Earth. The first step in developing an alarm system against geomagnetically induced currents is to forecast them. This is a challenging task, though, given the highly non-linear dependencies of the response of the magnetosphere to these perturbations. In the last few years, modern machine-learning models have shown to be very good at predicting magnetic activity indices as the SYM-H. However, such complex models are on the one hand difficult to tune, and on the other hand they are known to bring along potentially large prediction uncertainties which are generally difficult to estimate. In this work we aim at predicting the SYM-H index characterising geomagnetic storms one hour in advance, using public interplanetary magnetic field data from the Sun–Earth L1 Lagrange point and SYM-H. We implement a type of machine-learning model called long short-term memory networks. Our scope is to estimate -for the first time to our knowledge- the prediction uncertainties coming from a deep-learning model in the context of space weather. The resulting uncertainties turn out to be sizeable at the critical stages of the geomagnetic storms. Our methodology includes as well an efficient optimisation of important hyper-parameters of the long short-term memory network and robustness tests.

Plain Language Summary

Geomagnetic storms are disturbances of the geomagnetic field caused by interactions between the solar wind and particle populations mainly in the Earth's magnetosphere. These time-varying magnetic fields induce electrical currents on long ground-based conductors that can damage power transmission grids and other critical infrastructures on Earth. As a first step to forecast the ground magnetic perturbations caused by geomagnetic storms at specific mid-latitude locations, the objective of this work is to predict the SYM-H activity index, which is generated from ground observations of the geomagnetic field at low and mid-latitudes, and which provides a measure of the strength and duration of geomagnetic storms. We use the interplanetary magnetic field data measured by the ACE spacecraft at the L1 Lagrangian point and past SYM-H values to forecast the behavior and severity of geomagnetic storms one hour in advance. This forecasting is done using a type of artificial neural network model called long short-term memory. We also propose ways to estimate the uncertainties of these predictions, which help us to better understand machine-learning models in space weather prediction and could lead to more accurate and reliable forecasting of geomagnetic storms and their ground effects in the near future.

1 Introduction

In the last decades, our society has become more interdependent and complex than ever before. Local impacts can cause global issues, as the COVID-19 pandemic clearly showed, affecting the health of millions of human beings. Our society is highly dependent on relevant technological structures, such as communications, transport, or power transmission networks, which can be very vulnerable to the effects of space weather (SW). The latter has its origin in the solar activity and their associated events, such as coronal mass ejections and co-rotating interaction regions. Among other effects, these phenomena have an impact on the electrical current systems surrounding the Earth, enhancing them and thus causing large magnetic field fluctuations that propagate down to the ground. The electric field associated with these fluctuations, which is influenced by the interaction with the conductive earth, induces telluric currents in the uppermost solid layers and geomagnetically induced currents (GICs) in long conductors running on the surface. These GICs may cause disturbances, interruptions, and even long-term damage to critical infrastructures such as railways, oil and gas pipelines and power grids, with drastic social, economic and even political consequences. The intensity of the GICs is determined by the strength of the geoelectric field, but the latter measurements are rarely available. Because GICs are driven by temporal changes in the magnetic field, if we have an estimate of the resistivity structure below a specific location, variations in the magnetic field measured by ground magnetometers can in principle be used as the input parameter for deriving the GICs built up locally in a power grid (e.g. Torta et al. (2017)). However, because of the three-dimensional lithospheric resistivity structure, the behaviour of the time derivative of the geomagnetic field to which the ground electric fields are associated is complex and, consequently, has proven to be very difficult to predict (Kellinsalmi et al., 2022). Predicting geomagnetic indices, which attempt to condense a rich set of information about the status of the magnetosphere in a single number, is simpler and has always been a very attractive area for machine-learning (ML) applications (Camporeale, 2019). Although attempts to forecast geomagnetic indices started several decades ago (e.g. Burton et al. (1975)), they feature highly non-linear dependencies which are not yet well understood, and their forecasting is still an open and intensive area of research. Perhaps not surprisingly, recent efforts have been exploiting the large expressiveness offered by modern ML models, and their ability to characterise complicated multidimensional datasets. The present work follows such a trend by investigating advanced ML techniques to predict the behaviour of geomagnetic storms.

More specifically, our scope is to predict, at a given time in advance, the SYM-H index, which describes the geomagnetic disturbances at low and middle latitudes in terms of longitudinally symmetric disturbances of the horizontal component of the geomagnetic field (Iyemori, 1990). The SYM-H index is known to track very well the evolution in time, the topology and intensity of geomagnetic storms and their relation with solar source phenomena (Wanliss, 2005; Wanliss & Uritsky, 2010). We use time-series data from the Sun–Earth L1 Lagrange point tracking several covariates describing the interplanetary magnetic field (IMF) and its different components in addition to the SYM-H index. For this purpose, we predict the SYM-H index with a type of artificial neural network model called long short-term memory (LSTM) neural network (Hochreiter & Schmidhuber, 1997) especially conceived for describing, among others, non-linear time-series data.

Highly-parameterised neural networks as the ones we use in this work (as well as in recent literature) carry an important amount of intrinsic prediction uncertainty, called among statisticians "epistemic" uncertainty. This should be taken into account, where possible, in any scientific application, even more in those with direct impact on society as the present study. Yet another issue with those models is the presence of parameters (named "hyper-parameters" in the ML community) which are not directly optimised during the fitting processes, but whose impact on the predictions are potentially very large. Consequently, some sort of extra optimisation should be performed, which is typically computationally costly.

Furthermore, robustness in non-linear time-series predictions obtained from ML models (including LSTM) can be challenging due to their complex and often unpredictable nature. However, several techniques exist and can be used to test and improve the robustness of the models.

While studies on the prediction of geomagnetic indices with ML techniques have been conducted recently (see section 2), the novelty of our work is two-fold:

- For the chosen ML model (LSTM in this case), we report our predictions for the SYM-H index with associated uncertainties.
- We optimise the hyper-parameters of our model, in particular, following an efficient Bayesian optimisation strategy.

- The robustness of our LSTM model is evaluated not only with the standard hold-out method but also by reshuffling the list of geomagnetic storms.

2 Related Work

Efforts to forecast geomagnetic indices date back to the 1980's (Mayaud, 1980), which started using linear prediction models which were unable to capture well enough the complexity of the response of the magnetosphere to SW. For this reason, the community started to rely on the arbitrarily high expressiveness of neural network models (e.g. Lundstedt and Wintoft (1994); Gleisner and Wintoft (1996)).

Among these works, the one developed by Siciliano et al. (2021) constitutes a valuable reference from which we have started our study. They forecast the SYM-H index, for which one can have a priori finer time granularity with respect to other indices, thus being advantageous from the point of view of an alert system. Siciliano et al. (2021) compared the SYM-H predictions using two different neural network models: the LSTM and the convolutional neural network (CNN), the latter being typically used for image recognition tasks (Zhang, 1988; Zhang et al., 1990). While they have obtained good performances with the CNN compared to the LSTM (in some cases even slightly better), in our study we concentrate on the LSTM only, which for us delivered similar performances as the CNN. However, as commented above, we address the important issue of the uncertainty estimation, along with a detailed and explicit hyper-parameter optimisation together with an additional robustness test.

Posterior work by Collado-Villaverde et al. (2021) revolves on the same idea, but using a neural network architecture which actually combines CNN and LSTM transformations to predict not only the SYM-H index but also the complementary ASY-H index. With respect to our work, their architecture is different in that we use a standard LSTM model. Bhaskar and Vichare (2019) also predict both indices using a non-linear autoregressive exogenous (NARX) model (Leontaritis & Billings, 1985). On the other hand, Bailey et al. (2022) aim at forecasting the geoelectric field with LSTMs as well. While Pinto et al. (2022) forecast the ground magnetic field time derivative with LSTMs, Madsen et al. (2022) forecast both the ground magnetic field and its time derivative with LSTM networks and hybrid CNN-LSTMs. None of the works mentioned above, nor others less related to our study but in the same context, estimate the prediction uncertainties, nor have they thoroughly optimised their hyper-parameters. The only exception we were able to find was the very recent work by Iong et al. (2022), which studied the SYM-H index by using not neural networks, but another ML model belonging to ensemble methods (in particular, using a regularising gradient boosting framework; the eXtreme Gradient Boosting (XGBoost) library (Chen & Guestrin, 2016)), obtaining very good performance as well. In their case, while not estimating their prediction uncertainties, the hyper-parameter optimisation was actually performed, using a gradient-free "black box" optimisation method. The latter is a generic algorithm most convenient for situations where little or no information is known about the structure of the function to optimise. In our study, on the other hand, we use an optimisation algorithm particularly suitable for the type of objective function we have, so it is arguably more efficient.

3 Dataset Selection and Processing

The dataset used in this work corresponds to a sample of geomagnetic storms that occurred between 1998 and 2018, which were recorded at ground-based geomagnetic observatories, and were preceded by changes in the magnetic field and plasma parameters of the interplanetary medium, which were measured at the L1 Lagrange point by NASA's Advanced Composition Explorer (ACE) spacecraft. The geomagnetic storms have been selected following the same criteria as in Siciliano et al. (2021), in order to make a direct comparison with this previous work. The sample contains 42 of the most intense geomagnetic storms, distributed in two solar cycles. The intensity of the storms is defined by the SYM-H index. This index can be considered as a proxy of the response of the Earth's magnetosphere (especially the ring current) to solar activity and it is computed from data of a network of six magnetic observatories distributed in longitude across the low and middle-latitude region, with a time resolution of 1 min and precision of 1 nT. All the geomagnetic storms selected have a minimum SYM-H index lower than -100 nT, so they can be considered as either severe or extreme (Patowary et al., 2013). This ensures a high signal-to-noise ratio. Indeed, 55% of all these geomagnetic storms (23 out of 42) have a minimum SYM-H value between -200 nT and -100 nT, while the rest (i.e. 19 geomagnetic storms) have a minimum SYM-H value below -200 nT.

As Siciliano et al. (2021), we follow the commonly used hold-out method for training a ML model which is the process of dividing the full dataset into different splits and then using one split for training the model and other splits to validate and test it. Table 1 lists the geomagnetic storms classified in three sub-datasets containing data from different storms. These three sub-datasets are uniformly populated in terms of geomagnetic storm intensity and complexity. The training sub-dataset is used to train the LSTM model, the validation sub-dataset stops the network training and prevent over-fitting, while the test (also known as hold-out) sub-dataset is used as a proxy to evaluate the performance of the model on unseen data.

The length of the time interval of the considered geomagnetic storms range from 6 to 25 days, with an average of 10 days. This choice allows us to consider not only the main phase periods but also the initial and recovery phases, as well as previous and later quiet periods. The three sub-sets are uniformly populated in terms of geomagnetic storm intensity and complexity, the latter measured by the presence of multiple depressions of the magnetic field.

As already mentioned by Siciliano et al. (2021), a larger number of geomagnetic storms can be considered, as done for instance by Cai et al. (2010) and Bhaskar and Vichare (2019), though the additional ones are just either weak or moderate geomagnetic storms, adding no further predictive power to our LSTM model. This is due to the fact that all storm phases including quiet periods are already considered in all three sub-datasets.

The independent variables (commonly named "features" in ML) used for training the LSTM model are the squared value of the IMF magnitude B, the squared value of the IMF B_y component, the IMF B_z component (all these in GSM coordinates recorded at L1 Lagrange point by the ACE satellite) and the SYM-H index. The forecasting variable is the SYM-H index as mentioned above. All these variables are shown in table 2. All data are extracted from the NASA's OMNIWeb page (https://omniweb.gsfc.nasa.gov) with time resolution of 5 min (Papitashvili & King, 2023b). Although the data are available with a resolution of 1 min (Papitashvili & King, 2023a), the election of a lower resolution allows to reduce the computation time without reducing predictive power, allowing also a direct comparison with the results of Siciliano et al. (2021). The 5 min sample is computed by averaging the 1 min samples, so that the data at minute 0 corresponds to the average from minutes 0 to 4.

The IMF variables are propagated to the nominal magnetospheric bow shock following the method described in the OMNIWeb site (King & Papitashvili, 2005). It is important to note in this context that in this study (and also in that of Siciliano et al. (2021)) the information available on SYM-H at the Earth's surface is assumed to be simultaneous with that of the IMF projected at the bow shock. However, the spacecraft measuring the IMF is located upstream of the solar wind at the L1 Lagrange point, which allows these data to be known some time in advance (typically between 15 and 60 min). This advantageous position, which is therefore not exploited here, is expected to have a significant role in the efficiency of the SYM-H predictions.

The IMF data overflows are removed from the full sample of geomagnetic storms and the remaining empty gaps are filled using a linear interpolation method. Geomagnetic storms generally have short periods with IMF data overflows but in few cases (e.g. training storm TR13 and validation storm V3) the overflows are large and occur near the peak of the storm activity. While linear interpolation is one possible way to address the problem of gaps, it is clearly not optimal when these are located in periods of high activity. We are aware that more sophisticated approaches could be exploited (e.g. the interpolation schemes proposed by Qin et al. (2007) or Marsal and Curto (2009) or even a ML-based method we are currently developing), but we used the same approach as Siciliano et al. (2021) to perform a direct comparison between their results and ours. Figure 1 illustrates the removal of overflows in the IMF variables (B^2, B_y^2, B_z) followed by a linear interpolation to fill the resulting gaps for a particular storm, TR13, of the training sub-dataset. It was decided to keep the storms with overflows near maximum of activity in the study, both to train and to stop training the network, since they represent a real possibility when part of the data is lost due to measurement errors or overflows or even detector failures.

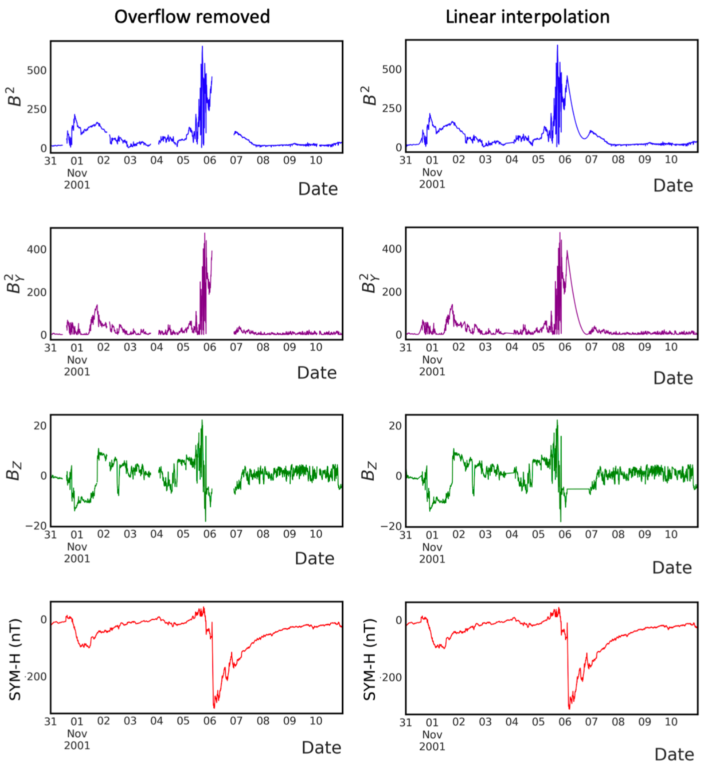


Figure 1. Training variables (B^2, B_y^2, B_z, SYM-H) for storm TR13 of the training sub-dataset, after overflow removal (left) and after linear interpolation to fill the gaps (right). The SYM-H index is also shown for completeness as it is the fourth variable used in the training.

Each geomagnetic storm time series is standardised using the associated mean and standard deviation, similar to Siciliano et al. (2021). However, in our case, the mean and standard deviation are just estimated from the training sub-dataset, given that this is the only data previously known for the study. With the standardisation the values are centred around the mean and given a unit standard deviation using the equation:

$$X_S = \frac{X - \mu_{\mathrm{TR}}}{\sigma_{\mathrm{TR}}}\,, \tag{1}$$

where X_S is the standardised time-series data, X is the original time-series data, μ_{TR} is the mean of the training series and σ_{TR} its standard deviation. Making each time series similar to a normal distribution, it is less sensitive to the scale of features and more consistent with each other, thus allowing the model to predict outputs more accurately. In addition, this scaling method is more resilient to outliers than the more common normalisation between $[-1, 1]$ or $[0, 1]$, which only considers the minimum and maximum values instead of the overall statistics of the data.

4 Deep-Learning Model

Humans, as intelligent beings, do not have to learn how to speak, walk or cook from scratch every time. Our previous experiences on these tasks endure, and the ability to do it improves after several repetitions. Traditional neural networks do not have this feature, and it is a major deficiency for some specific tasks such as time-series forecasting or natural language processing. Recurrent neural networks (RNNs) (Rumelhart & McClelland, 1987) address this problem. RNNs integrate cyclic connections, allowing information to persist, making them a more powerful tool to model sequential data than the traditional feed-forward neural networks. LSTM networks are a variety of RNNs, capable of learning long-term dependencies and also forgetting irrelevant information, especially in sequence prediction problems. RNNs read the data sequentially, and attribute higher weights to the recent information. The back-propagation training of RNNs suffers from the so-called "vanishing gradient" problem, which limits the learning capabilities of the network. LSTMs address this problem by recognising between long-term and short-term memory through a gating mechanism that regulates the flow of information. This allows the model to selectively retain or forget information based on its relevance, making it more robust and able to handle complex sequential patterns. In this work we use one such type of neural network with a typical architecture shown in Figure 2, which we briefly describe below.

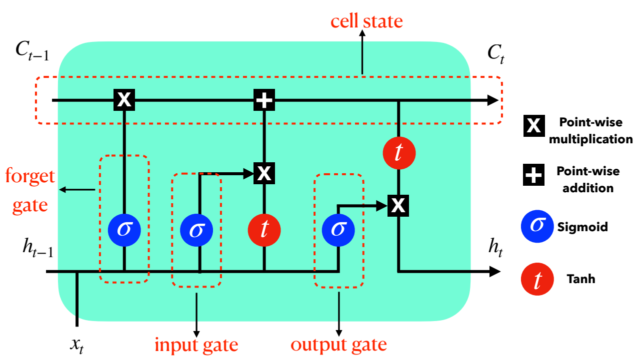


Figure 2. LSTM "cell" based on four interacting layers (cell state, input gate, forget gate and output gate). An LSTM network consists of repetitions of such a cell for every step t of the time series.

An LSTM network consists of a series of non-linear transformations for each time step with shared parameters (weights). The transformation for a generic input x_t at time step t is schematically represented in Figure 2. For each t, there are two outputs: the cell state c_t and the hidden output h_t, which are computed from four quantities. The first quantity is the result of the "forget gate" f_t, consisting typically of a sigmoid function applied to the linear function $W_f x_t + U_f h_{t-1} + b_f$. Here x_t is the input, h_{t-1} the hidden output of the previous time step, while W_f and U_f are the weight matrices and b_f is the bias vector. The motivation for the forget layer -whose value is a number in the range $[0, 1]$- is to decide how much to "forget" about the previous time step's cell state c_{t-1} (i.e. "0" forgets the previous data and "1" uses the previous data). The second quantity is the "cell input" $\tilde{c}_t$, similar in structure to the forget gate, but with its own set of weights W_c, U_c and b_c, and using a tanh function instead of a sigmoid. It represents the new information that would potentially be included in the new cell state c_t. How much of such information to be retained is determined by the third quantity: the result of the "input gate" i_t, defined as another sigmoid transformation, but with its corresponding weights W_i, U_i and b_i. With the above three quantities and their interpretations, the new cell state is defined in an intuitive way as: $c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t$, with $c_0 = 0$ by definition. Finally, the fourth quantity is the result of the "output gate" o_t, being yet another sigmoid with weights W_o, U_o and b_o. It has the role of a weight factor for the hidden output, computed as $h_t = o_t \times \tanh(c_t)$, with $h_0 = 0$.

The above series of transformations are consecutively applied to a finite number of time steps, from $t - l_b$ to t, with l_b being an optimisable hyper-parameter called the "look-back". The final aim is to predict an observable y_{t+l_f} at a future time $t+l_f$, with l_f being the "look-ahead", representing how much time in advance we want to make a prediction, which is fixed by the domain needs. In order to make such a prediction, right after the last LSTM cell applied on t and resulting in the hidden output h_t, a number of dense network layers are applied to transform the vector h_t into the predicted value of y_{t+l_f}, while potentially adding more expressiveness to the model. The number of dense layers is also another hyper-parameter to be optimised.

4.1 Hyper-Parameter Optimisation

Hyper-parameter optimisation is an essential ingredient when training state-of-the-art ML models, due to their high complexity. Traditional random- or grid-search strategies have shown to be very inefficient when the number of hyper-parameters is larger than a few. Modern libraries exist which implement efficient algorithms for optimising costly functions. One of the most popular ones, which we adopted here, is the hyper-parameter optimisation framework "Optuna" (Akiba et al., 2019). In particular, we run Optuna to optimise the following hyper-parameters: the number of dense layers, the number of units of these layers, the learning rate, and the look-back parameter of the LSTM layer. We use Optuna's implementation of a Bayesian optimisation flavour called "tree-structured parzen estimator", the details of which are found in Bergstra et al. (2011).

Finding multiple local minima can be a problem in hyper-parameter search. In particular, because of the stochastic nature of gradient descent during training, there can be times when two identical trials result in a value of the loss function that varies more greatly than trials with different hyper-parameters would. For this reason, each trial is repeated five times. The mean and standard deviation of the loss function results for each trial with a set of hyper-parameters are calculated. Having done this, all trials with root-mean-square error (RMSE) standard deviations that overlap with the best (i.e. lowest) RMSE mean of all trials are labeled as *best trials*. This procedure allows us to explore flat directions in the hyper-parameter space, as discussed later in section 5.

4.2 Robustness of the LSTM model

Ensuring the robustness of state-of-the-art ML models, used to analyse and predict non-linear time-series data, is of critical importance for reliable and effective decision-making, especially on those with direct impact on society. This requires careful design as well as rigorous testing and validation, but the benefits in terms of reliability and effectiveness are significant. Non-linear time-series data can exhibit complex and dynamic behaviour, making it challenging to model and predict accurately.

As in many other works, we ensure the robustness of our LSTM model by using the hold-out validation technique and therefore splitting the full dataset into three different sub-datasets (training, validation and test) containing uniformly populated times-series data from different geomagnetic storms. This technique helps to identify weaknesses and guarantee the model to perform and generalise well on new and unseen data, even in the presence of various perturbations, such as data noise or changes in the distribution of the times-series data.

However, the performance estimate of the ML model may be highly dependent on the particular dataset split used. If the split is not representative of the overall dataset distribution, then the performance estimate may be biased. In our case the three sub-datasets are uniformly populated in terms of geomagnetic storms intensity and complexity and therefore no bias is expected. However, to evaluate this issue we reshuffled the original list of geomagnetic storms in the three sub-datasets shown in Table 1. Thus, we populated the new training sub-dataset with the 17 storms from the original test sub-dataset plus three storms from the original validation sub-dataset. To populate the new test sub-dataset, we used 17 storms from the original training sub-dataset. Finally, the new validation sub-dataset was filled with the remaining three storms from the original training sub-dataset plus two validation storms from the original validation set. A visual representation of the baseline and reshuffled lists of geomagnetic storms used for training, validation and test is shown in Figure 3. With the new reshuffled list of the geomagnetic storms, we trained an alternative LSTM model (with its own optimised hyper-parameters) and obtained compatible performance without observing over-fitting, under-fitting or biases.

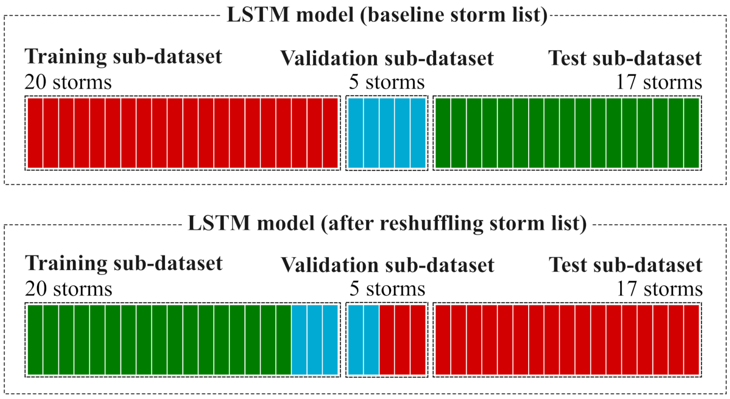


Figure 3. Visual representation of the two lists of geomagnetic storms used for training, validation and test for the baseline and the alternative LSTM models. The upper diagram represents the baseline list, as shown in Table 1 (i.e. same criteria as Siciliano et al. (2021)), while the lower diagram represents an alternative ordering where the list of storms is reshuffled. Each rectangle represents a different storm. Colours are assigned based on the baseline list of the geomagnetic storms, where red, blue and green represent the original storms in the training, validation and test sub-datasets, respectively.

Other techniques could also be used to enhance the robustness of LSTM models such as different flavours of cross-validation, data augmentation, model ensembling, adversarial training or regularisation, though we will deeply explore this in future works.

In any case, we want to point out that the regularisation technique is indeed used in this work for the estimation of uncertainties (see section 4.3 and Appendix A).

4.3 Estimation of Prediction Uncertainties

To estimate the uncertainties associated with our predictions, two main approaches can be followed: a frequentist approach, in particular adopting the "bootstrapping" method, or a Bayesian approach, where several state-of-the-art methods can be adopted depending on the needs and scope. In this work we have followed the two methods, and compared the results between them.

Bootstrapping is a series of techniques by which we obtain synthetic datasets out of the "real" (observed or simulated) dataset we have at our disposal. In doing this, both aleatoric and epistemic uncertainties are taken into account when making predictions, making bootstrapping equivalent to the principled Bayesian approach. In the physics community, the typical methodology is to: 1) propose a likelihood distribution of the data, and optimise its parameters by maximum likelihood estimate (MLE) method, 2) with these optimum parameters, use the proposed likelihood to sample a large number of synthetic datasets, identical in length to the original one, 3) find for each synthetic dataset the MLE parameters analogously as in step 1, and 4) each of the MLE parameters will lead to a different prediction, thus obtaining a distribution of predictions. While this technique works very well for many situations, it may be misleading when the assumed likelihood is very different from the true -unknown- underlying distribution of the data. For this reason, in the ML community there is another popular bootstrapping strategy, which consists in re-sampling a large number of times the real dataset directly, either with or without replacement[1]. This is equivalent to sampling from the empirical distribution, instead of assuming a particular parametric shape of the likelihood.

The traditional bootstrapping fails with time series because the sampling procedure breaks off the time dependence that concatenates adjacent samples in the sets. For this reason, a special consideration has to be made for our case. If we can divide the set in chunks of samples, and perform the bootstrap sampling procedure on these blocks instead of on the individual samples, we can conserve the time dependence up to the division of the blocks; for the present dataset, a natural way to divide the training set is by geomagnetic storms, in particular because we gain the advantage of explicitly breaking adjacent samples of different storms that are not expected to have a time dependency.

On the other hand, we have also followed a Bayesian approach for estimating the prediction uncertainties. In the case of deep neural networks one of the most popular strategies is the so-called "dropout" method (Gal & Ghahramani, 2016). More details on this can be found in Appendix A, where we also show the corresponding results as well as the comparison with respect to the bootstrap method. In summary, for this particular dataset we find that the bootstrap results perform better, especially around the peak of the storms, which is the most critical region. We thus retain the bootstrap predictions and corresponding uncertainties as our main results.

A note of caution is in order at this point. When reporting our prediction uncertainties, we are more specifically reporting the systematic (or epistemic) uncertainties of the expected (mean) values of the SYM-H index, which we calculate as the output of our LSTM network. Note that this is not the same as the total prediction uncertainties, which include the data noise (also known as aleatoric, or statistical uncertainties), and which we do not have available. Since the reported epistemic uncertainties decrease as the number of data points increase, it is perfectly consistent to have a very precise de-

[1] "With replacement" means that a particular instance of the real dataset can appear more than once in the synthetic dataset.

termination of the mean predictions of the SYM-H index, describing data whose noise is appreciably larger (and consequently having values beyond the corresponding interval of epistemic uncertainties). On the other hand, we are also not considering in this work the uncertainties related to the other input variables (related to the IMF B and its components). While we plan to include them in a future work, we nonetheless expect their impact on our results to be small, after checking that the uncertainties in B^2 are of few percent.

5 Results

After discussing the data and analysis setup in previous sections, we turn now to present the results of our analysis.

In the first subsection we present the optimisation of hyper-parameters needed to learn the evolution of the SYM-H index, the relative importance of each parameter in the final result and explore the possible correlation between hyper-parameters. We then present the overall performance of the trained algorithm when predicting the evolution of the SYM-H index.

5.1 Hyper-Parameter Tuning

Once we have chosen an LSTM as the basic architecture for the time-evolution analysis, the next step is to optimise the hyper-parameters of the learning structure. The results of this analysis are shown in Figures 4 and 5.

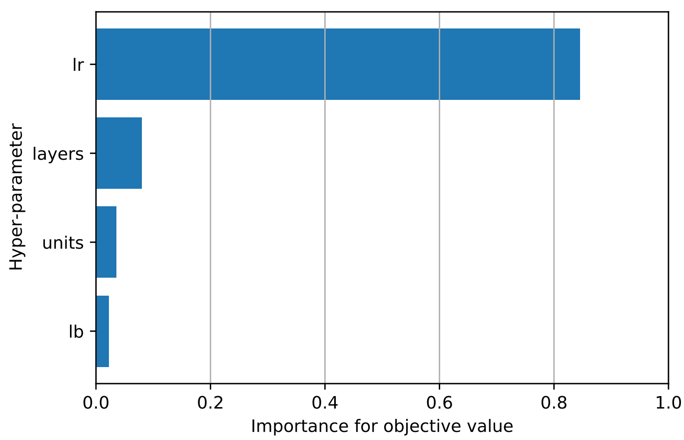


Figure 4. Hyper-parameter importance bars for learning rate (`lr`), number of dense hidden layers (`layers`), number of units in all hidden layers (`units`) and look-back (`lb`).

In particular, we vary the number of fully-connected layers which are placed after the LSTM architecture (`layers`), the number of neurons of these dense hidden layers (`units`) and the learning rate parameter (`lr`). We also explore different values of the look-back parameter (`lb`), the amount of previous data we allow the network to explore in order to predict the future evolution; this value is reported in terms of number of 5 min steps, unless otherwise indicated.

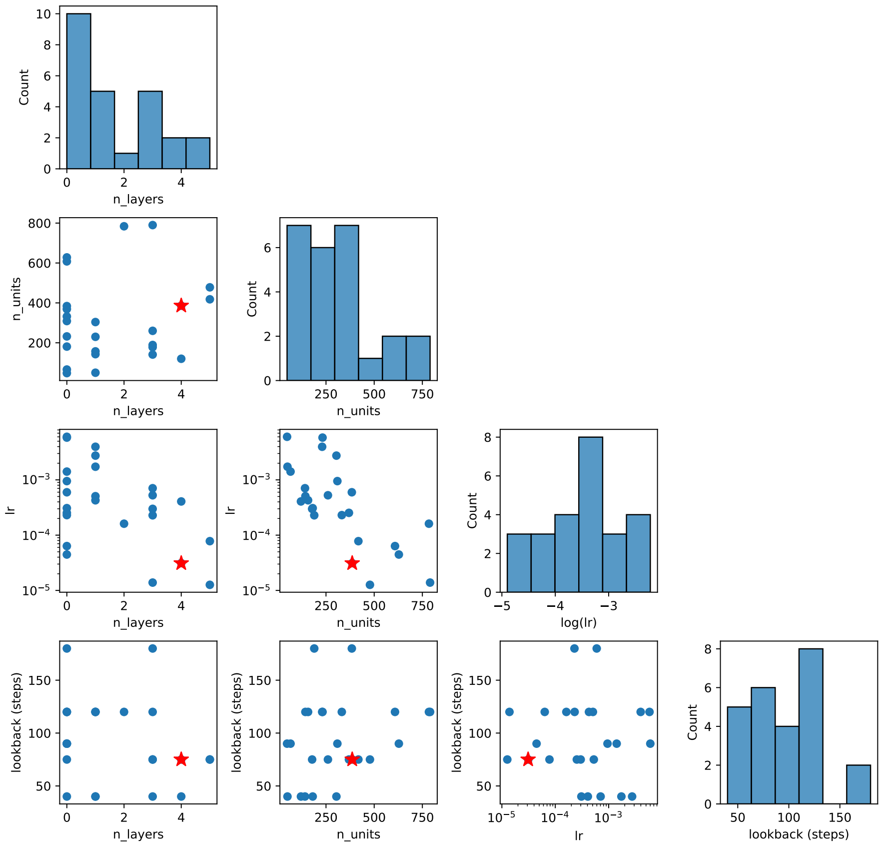


Figure 5. Pair-wise scatter-plots for the hyper-parameters optimised via Optuna for the LSTM architecture. Out of the total of 25 cases, the blue points correspond to hyper-parameter values which cover the minimum value of the MSE that results from using the global optimum values. Red stars indicate the reported optimum values. The histograms along the diagonal, for each hyper-parameter, are the result of marginalising all the points from the rest of hyper-parameters.

The ranges in which each hyper-parameter was optimised are summarised in Table 3. Variations of each of these parameters are not equally important, as shown in Figure 4.

Indeed, we found that the learning rate is key to the learning, whereas variations of the depth and width of the fully-connected layer (`n_layers, n_unit`) are much less important. This indicates that, once the LSTM is learning the time series, the particular characteristics of the additional dense layer are not that relevant. We also found that, in the range we explored, the look-back parameter was not an important handle. This would indicate that we have already chosen an optimal look-back range. Note, however, that if we were interested in describing other, less global, parameters than the SYM-H index, the look-back parameter may change. This optimisation is only valid for the output prediction we have chosen to describe.

The best hyper-parameter values (in terms of mean-square error (MSE)) according to Optuna are: (`n_layers, n_unit, lr, lb`) = (4, 386, 3.12×10^{-5}, 75 steps), which are shown in Table 3. However, while these specific values are indicated as optimal, one should keep in mind that slightly different values could lead to the same performance; there could be "flat directions", i.e. combinations of hyper-parameter values away from the reported optimum which produce equally low MSE. Most importantly the optimisation made using Optuna assumes that the hyper-parameters are uncorrelated; indeed, the hyper-parameters may be correlated to some extent, while the procedure assumes complete independence.

We have explored the impact of these caveats by performing a multidimensional scan of the hyper-parameters instead of assuming total uncorrelation. The results are summarised in Figure 5, where we show the pair-plots between the different hyper-parameters. The points shown in the scatter plots correspond to hyper-parameter values for which, upon repeating the trials five times, within their standard deviation, cover the minimum value of the MSE that results from using the global optimum values specified above. The optimum values are also shown. The histograms along the diagonal, for each hyper-parameter, are the result of marginalising all the points from the rest of hyper-parameters. For these scatter plots we observe no evident correlation between pairs of hyper-parameters, which validates the use of the Optuna procedure.

However, the flat directions are explicitly present in almost all axes. For example if using six hidden layers instead of one, while fixing the rest of hyper-parameters to values different from their "optimum", we get equally good results, statistically speaking. Analogously, this happens with the number of units per hidden layer, which can be as high as 800 (with respect to the reported optimum at 386), or the look-back parameter at 300 (with respect to the optimum at 75). In all cases we observe that each hyper-parameter can admit large excursions in combination of specific values of other hyper-parameters without sacrificing the figure of merit. This is nothing but the consequence of a highly complex parametric dependence of the loss function with respect to the hyper-parameters of the model, as is often the case with the large models used by the community nowadays.

5.2 Prediction of the SYM-H Index

To reproduce the results in Siciliano et al. (2021) where the SYM-H index is predicted from the IMF observations at L1 Lagrange point and from past SYM-H values, as shown in Table 2, the same storms and time intervals as in that work were used, as well as the same training-validation-test split of the storms (see Table 1).

In our case, the neural network architecture consists of an LSTM layer using the hyper-parameters configuration reported in Table 3.

Block bootstrap was performed and 200 bootstrap models were used to obtain estimations of the uncertainties of prediction values, RMSE and the coefficient of determination (R^2).

Table 4 shows the values of RMSE in nT for the target variable SYM-H, and the values of R^2 for the fits of the model to each of the storms in the test sub-dataset. The lower the RMSE, the better a model fits the test sub-dataset. The higher the R^2 value, the better a model fits the test sub-dataset. These values, which are directly comparable with those of Siciliano et al. (2021), are shown in Figure 6. With blue dots, we show the average of our predictions and the blue symmetric segment corresponds to the 95% confidence level (CL) of these predictions. The reported results from Siciliano et al. (2021), Collado-Villaverde et al. (2021) and Iong et al. (2022) are shown with orange dots, red crosses and green stars, respectively, which correspond to their best RMSE results. The same applies to the right panel in Figure 6, this time reporting R^2 values (the values for Iong et al. (2022) are not shown as they are not reported by these authors). Note that uncertainty bars are not shown in the results of other authors since they did not report them.

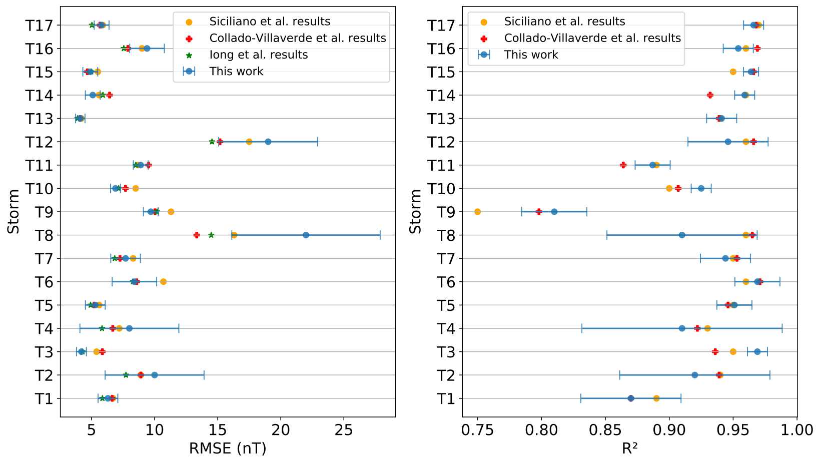


Figure 6. RMSE and R^2 values for the predicted SYM-H index for each one of the 17 test storms. The results of this work are shown in blue with 2σ uncertainty bars (i.e. 95% CL), while those from Siciliano et al. (2021), Collado-Villaverde et al. (2021) and Iong et al. (2022) are shown in orange circles, red crosses and green stars, respectively, which correspond to their best RMSE and R^2 results.

One should then compare the orange dots (which are the best predictions from a bunch of 20 predictions from Siciliano et al. (2021)) with either the lower RMSE or higher R^2 value of the blue range of our predictions. In most cases, our architecture leads to better performance, which we believe is mainly a manifestation of the achieved optimisation of hyper-parameters.

On the other hand, we also include in Figure 6 the comparison with two other more recent studies (see Collado-Villaverde et al. (2021); Iong et al. (2022) commented in section 2), which check the performance of their methods on the same storms as Siciliano et al. (2021). We can observe that those other studies in general improve over Siciliano et al. (2021), while for most of the test storms they still lie inside our RMSE intervals[2]. However, as we commented in section 2, note that contrary to the case of Siciliano et al. (2021), the models considered in Collado-Villaverde et al. (2021); Iong et al. (2022) are different from ours, either by using neural networks with different architectures or a completely different model. It is worth stressing again at this point that our aim in this work was not to build and optimise a robust model to be considered in terms of prediction performance, but to study the prediction uncertainties, while using a popular model which nonetheless, as we see, still gives very competitive results.

[2] For only four out of 12 storms their predictions are marginally better than our predictions, except storm T8, for which they are up to 25% better.

The results in Figure 6 are rather global measures of performance, as they evaluate the goodness of predictions during the whole storm. On the other hand, we may be interested to know how well the algorithm is performing during shorter periods of time, e.g. during the peaks of activity. To illustrate this point, in Figure 7 we show the prediction of the bootstrap models of the target variable SYM-H (in nT) for two of the 17 test storms[3]. In this figure, the orange band represents the 95% CL of the predictions coming from the bootstrap procedure, while the mean prediction is shown by the red dashed lines, and the actual test values are shown as a solid blue line. For each storm we also plot (bottom panels) the residuals, which are computed just subtracting the prediction mean from the observed values and orange bands. We observe in general a very good agreement between the predictions and the observations, where the regions around the peaks show, as expected, the largest deviations. Note how the prediction uncertainties are also larger around the peaks, as one would expect. These larger deviations around the peaks are mainly due to a difference in *timing* of the predictions with respect to the observations. This is indeed a common behaviour for LSTM models (and other models handling sequential data) using a limited training sub-dataset for predicting time-series data with a significant auto-correlation, which can make sometimes difficult for the model to accurately identify the underlying patterns and trends. In our case, for many storms (see Appendix A) we predict the drop in the SYM-H index to happen a bit before it actually happens, which then causes large positive residuals for instants of time before the observations start to drop as well. This is indeed the case at least for storms T1, T6, T7, T8, T11 and T16 featuring residuals around the peak in the range 50–100 nT. On the other hand, for storms T9, T12 and T14, having residuals around the latter (absolute) values, the timing oscillates between predicting in advance or with a small delay.

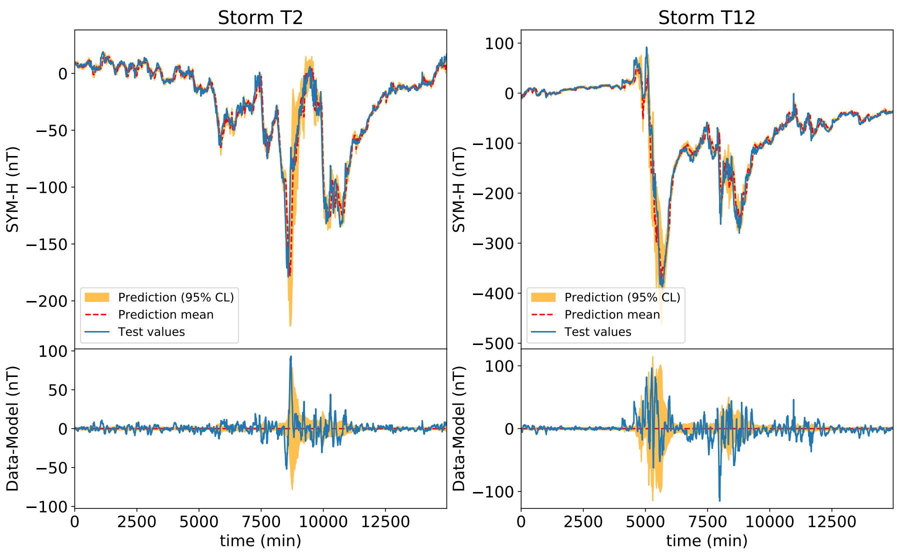


Figure 7. Time-series distributions of two of the 17 storms in the test sub-dataset, in particular, storms T2 and T12, showing in an orange band the 95% CL (corresponding to 2σ), in red dash line the mean for the one hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.

[3] See Figure A1 in Appendix A for all the storms.

Finally, it is important to note that, as commented in section 4.3, the orange bands only represent the epistemic uncertainties (the uncertainties on the expected mean), reason for which there may be observed values lying outside the bands, which may be in part related to the intrinsic data noise, not represented in this figure (because we do not have access to it; see also Appendix A).

5.3 Feature Importance

Neural networks are often considered black-box algorithms though some external inference techniques can be used to extract useful information that can help to understand deep-learning models. Computing feature importance in LSTM models is indeed an important aspect of model interpretation and understanding. Feature importance is a measure of how much a particular input variable (or feature) contributes to the output of the model. Indeed, understanding feature importance can help to identify and select the input features that are most relevant for a given prediction model. It can also provide valuable insights into the underlying patterns, dynamics and relationships present in the considered time-series data. There are several techniques that are commonly used to compute feature importance in LSTM models. Some of these techniques are the "input permutation" (Breiman, 2001; Fisher et al., 2019), "SHapley Additive exPlanations" (Lundberg & Lee, 2017), "Leave-One-Feature-Out", "gradient-based method", "layer-wise relevance propagation" and "activation-based methods" among others.

In this work, the approach used to compute the feature importance in our LSTM model is based on the "input permutation" technique. We repeated the training procedure, using the same optimised hyper-parameters already discussed in section 5.1, but adding disturbances in the input data (i.e. IMF data and past SYM-H values). Thus, for each of the four input features, the values of all of the other features were shuffled, new predictions were calculated using the original test data, and RMSE was calculated. This procedure was performed 15 times for each variable; this is a total of 60 training sessions. The average value of the RMSE for each case is compared to a baseline value calculated with no shuffling (i.e. with the average RMSE value of the RMSE values shown in Table 4). The output of the feature importance results are shown in Figure 8. In the followed method, the most important features are the ones that, when all other variables are shuffled, result in an RMSE closer to the baseline average RMSE value. Thus, from the obtained results, we conclude that past SYM-H values represent the most important feature for our LSTM model, similarly to Siciliano et al. (2021).

It is important to point out that the interpretation of feature importance in LSTM models can be challenging, as these models are inherently complex and exhibit dynamic and non-linear behaviour. Additionally, the results can be influenced by the data preprocessing (e.g. interpolation approach for data gap filling), the choice of input scaling and normalisation as discussed in section 3, as well as the choice of model architecture and the optimisation of the training hyper-parameters as discussed in section 4.1.

6 Discussion and Outlook

In this paper we have explored the use of a deep-learning model to predict the evolution of an activity index during geomagnetic storms, and proposed ways to estimate the uncertainties of these predictions. In particular, we focused on the SYM-H index, a quantity whose variation during a storm is a good summary of its strength. As input parameters, we used IMF data from the ACE spacecraft located at the L1 Lagrange point together with historic SYM-H values.

We chose the SYM-H index to be able to compare with an existing study using deep learning and LSTM architectures in Siciliano et al. (2021). With this comparison, we can

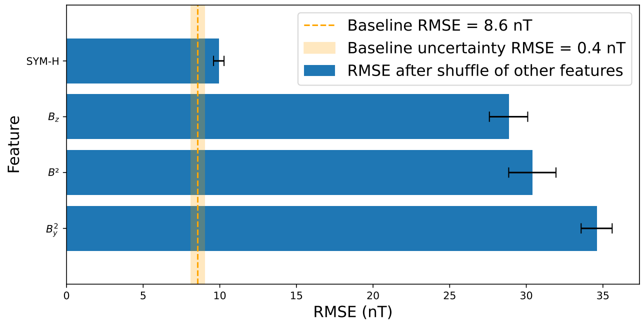


Figure 8. Ranking of the feature importance using an approach based on the "input permutation" technique (the smaller the value, the more important the variable is). Each bar represents the mean value of the RMSE evaluated over all test storms after having shuffled all except the indicated feature variable. The uncertainty bars represent the standard deviation, and the vertical orange line represents the baseline value calculated with no shuffling along with its own standard deviation (that can be computed by averaging values from Table 4).

illustrate the impact of the improvements we propose in both learning optimisation and uncertainty estimation.

We found an overall improvement of the best predictions for the SYM-H index due to hyper-parameter optimisation, as shown in Figure 6, where our lower limit of the RMSE range is lower than the reported best RMSE value in Siciliano et al. (2021), with the exception of test storm T8.

Moreover, we proposed a robust statistical procedure to compute uncertainties in the predictions based on block-bootstrapping. With those uncertainties we produce a prediction with an uncertainty band corresponding to a chosen confidence interval and examine the goodness of our predictions at different times during the storm. See Figure 7 for an illustration of how this uncertainty band evolves with time, and the comparison with the observed values of the SYM-H index.

The strategy described in this work could be applied to other architectures and target parameters, such as the evolution of the geomagnetic or geoelectric fields in the ground.

Reproducing the prediction of the SYM-H global geomagnetic activity index of Siciliano et al. (2021) has served to match the needs of a group of scientists working in SW with the experience of a group working on ML techniques applied to problems related to particle physics. The improvement in prediction performance obtained with this test augurs well for our ultimate goal, which is to be able to predict the variations of the geomagnetic or geoelectric field on the ground at a specific location (Spain). The challenge is important because it involves adding the effect of the field induced by the three-dimensional structure of the electrical resistivity of the lithosphere to the complexity of the sources of these variations. Since we have models for this three-dimensional structure of the resistivity (Torta et al., 2021), we should also be able to predict the variations of the geoelectric field and, by combining them with the models of electrical admittances of our national power grid also described in Torta et al. (2021), derive the expected GICs.

Future work will include ground-level magnetic field forecasting using data from Ebre Observatory, or better, also with those of the other geomagnetic observatories on the Iberian Peninsula. We are also interested in forecasting the time derivative of the geomagnetic field, since this variable is usually the most directly responsible for driving the geoelectric field and, therefore, the GICs. The ultimate goal will be to reformulate the problem in terms of an advanced deep-learning model that provides an alarm system against GICs in Spain. Moreover, our ML architecture can be made more robust and elaborated by including other developments such as a more sophisticated interpolation method to fill data gaps, a cross-validation technique for further improving the model robustness, and adding an attention layer in combination with LSTM.

Data Availability Statement

Raw data are obtained from the NASA's OMNIWeb page (https://omniweb.gsfc.nasa.gov). Processed data, high-resolution plots, and prediction models (for both bootstrap and dropout) in `h5` format can be downloaded at https://zenodo.org/record/7695656 (SpaceWeather-IFIC, 2023).

Table 1. List of the sub-datasets with the most relevant information of the geomagnetic storms: label assigned to the storm, starting date, duration in days and minimum value of the SYM-H index during the geomagnetic storm period. The distribution of the storms among the different sub-datasets follows the same criteria as Siciliano et al. (2021).

Training sub-dataset

Label	Start date	Duration (days)	SYM-H (nT)
TR1	14/02/1998	8	−119*
TR2	02/08/1998	6	−168*
TR3	19/09/1998	10	−213
TR4	16/02/1999	8	−127*
TR5	15/10/1999	10	−218
TR6	09/07/2000	10	−347
TR7	06/08/2000	10	−235*
TR8	15/09/2000	10	−196*
TR9	01/11/2000	14	−174*
TR10	14/03/2001	10	−165*
TR11	06/04/2001	10	−275
TR12	17/10/2001	10	−210
TR13	31/10/2001	10	−320
TR14	17/05/2002	10	−116*
TR15	15/11/2003	10	−490
TR16	20/07/2004	10	−208
TR17	10/05/2005	10	−302*
TR18	09/04/2006	10	−110*
TR19	09/12/2006	10	−211*
TR20	01/03/2012	10	−149

Validation sub−dataset

Label	Start Date	Duration (day)	SYM-H (nT)
V1	28/04/1998	10	−268
V2	19/09/1999	7	−160
V3	25/10/2003	9	−432*
V4	18/06/2015	10	−207*
V5	01/09/2017	10	−146*

Test sub-dataset

Label	Start Date	Duration (day)	SYM-H (nT)
T1	22/06/1998	8	−120
T2	02/11/1998	10	−179*
T3	09/01/1999	9	−111
T4	13/04/1999	6	−122
T5	16/01/2000	10	−101*
T6	02/04/2000	10	−315
T7	19/05/2000	9	−159*
T8	26/03/2001	9	−437
T9	26/05/2003	11	−162*
T10	08/07/2003	10	−125*
T11	18/01/2004	9	−137*
T12	04/11/2004	10	−394*
T13	10/09/2012	25	−138
T14	28/05/2013	7	−134
T15	26/06/2013	8	−110
T16	11/03/2015	10	−234
T17	22/08/2018	12	−205

* Geomagnetic storms with multiple depressions.

Table 2. Variables used in the analysis.

Training variables	B^2	B_y^2	B_z	SYM-H
Forecasted variable				SYM-H

Table 3. Range in which each hyper-parameter was optimised, and chosen value.

Hyper-parameter	Search range	Chosen value
Number of layers	[0, 10]	4
Number of units	[0, 1000]	386
Learning rate	$[10^{-6}, 10^{-1}]$	3.12×10^{-5}
Look-back (steps)	[40, 75, 90 120, 180, 360]	75

Table 4. RMSE and R^2 values for the predicted SYM-H index with their respective standard deviations for each of the storms in the test sub-dataset for our neural network architecture using an LSTM model, and the IMF variables and past SYM-H values as input features for the training.

Set	RMSE (nT)	R^2
T1	6.3 ± 0.4	0.87 ± 0.02
T2	10 ± 2	0.92 ± 0.03
T3	4.2 ± 0.2	0.969 ± 0.004
T4	8.0 ± 2.0	0.91 ± 0.04
T5	5.3 ± 0.4	0.951 ± 0.007
T6	8.4 ± 0.9	0.969 ± 0.090
T7	7.7 ± 0.6	0.944 ± 0.010
T8	22 ± 3	0.91 ± 0.03
T9	9.7 ± 0.3	0.810 ± 0.013
T10	6.9 ± 0.2	0.925 ± 0.004
T11	8.9 ± 0.3	0.887 ± 0.007
T12	19 ± 2	0.946 ± 0.016
T13	4.11 ± 0.19	0.941 ± 0.006
T14	5.1 ± 0.3	0.959 ± 0.004
T15	4.9 ± 0.3	0.964 ± 0.003
T16	9.4 ± 0.7	0.954 ± 0.006
T17	5.8 ± 0.3	0.966 ± 0.004
Total dataset	8.6 ± 0.4	0.929 ± 0.013

Appendix A Dropout method for estimating the prediction uncertainties

In this appendix we discuss in more detail the dropout method as an alternative approach for estimating the prediction uncertainties. We also compare the corresponding results with those obtained from the bootstrap method (see section 4.3).

Roughly speaking, the idea consists in randomly turning off units of the different neural network layers. This has an immediate utility as a regulariser procedure; this is the reason for which dropout is commonly used at the training phase in order to control over-fitting. However, as pointed out in (Gal & Ghahramani, 2016), such a procedure is mathematically equivalent to a variational inference algorithm, with a specific choice of the variational distribution. In particular, if dropout is also used at the test phase, the probability distribution of the predictions would be equivalent to the ones that would be obtained by computing the standard predictive distribution of the Bayesian approach, under the chosen variational approximation.

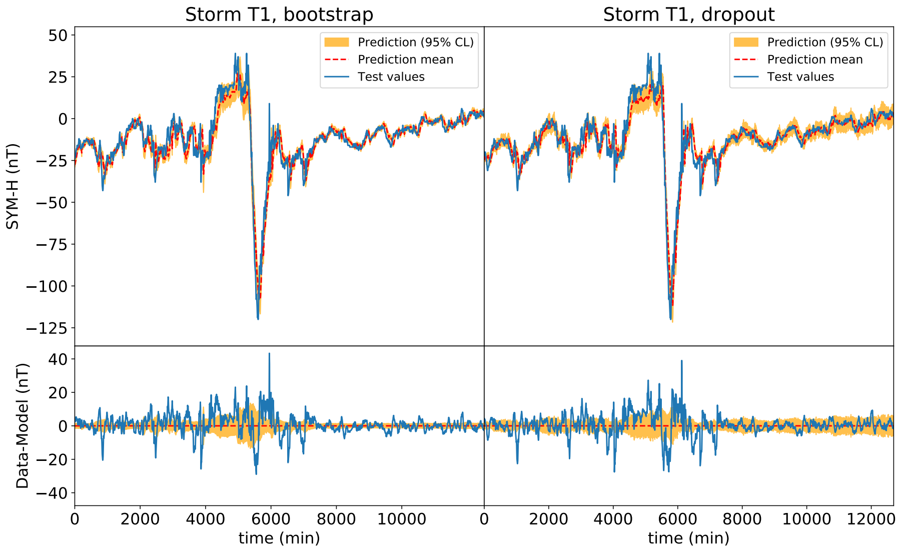
An essential parameter in the dropout implementation is the dropout probability p. Formally, p is the probability for a Bernoulli (binary) random variable to take value equal to 1; so by sampling from the Bernoulli distribution, once for every unit in a hidden layer, such a unit is turned off with a probability of $1-p$. Traditionally, p is considered as an important hyper-parameter to be optimised, e.g. by grid-search, which can be computationally expensive in largely parameterised models. This is the motivation behind "concrete dropout" cited from (Gal et al., 2017), which modifies the traditional dropout algorithm in such a way that p becomes an optimisable parameter during the normal training period. This is done by modifying the loss function so that it has an explicit -and differentiable- dependence on p, which is the result of approximating the Bernoulli distribution by its continuous relaxation using the concrete distribution. In our neural network architecture, we have implemented the concrete dropout method for the dense layers following the LSTM layer, and consequently the associated dropout probability p is automatically optimised during the training process. However, for the LSTM layer itself we stick to the traditional implementation of dropout, where the parameter p is in this case included as an hyper-parameter optimisable with the Optuna procedure. The resulting optimal value for the LSTM dropout probability is $p = 0.0128$.

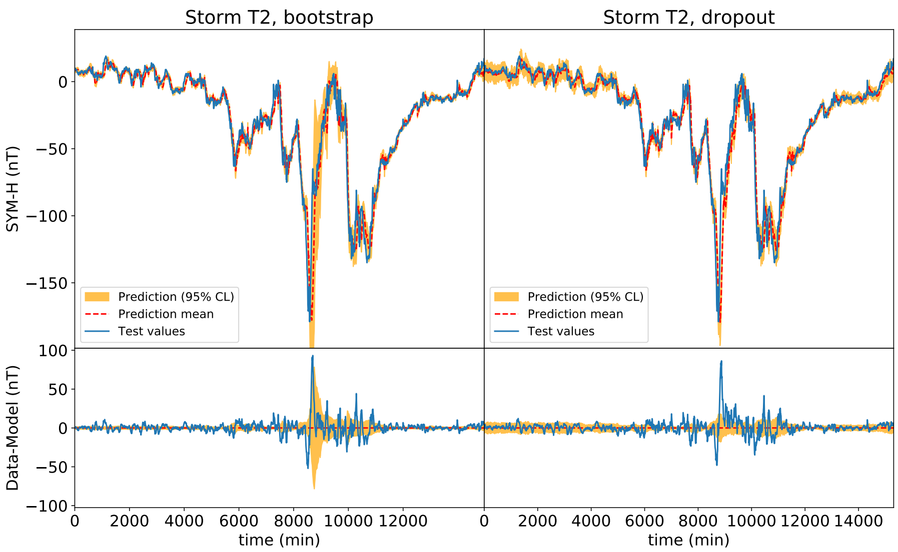
The dropout results are shown in Figure A1 (right panels) for all the 17 different storms of our test sub-dataset, in terms of the prediction with its associated uncertainty of the SYM-H index as a function of time. We compare side by side with the bootstrap results[4] (left panel in the figure).

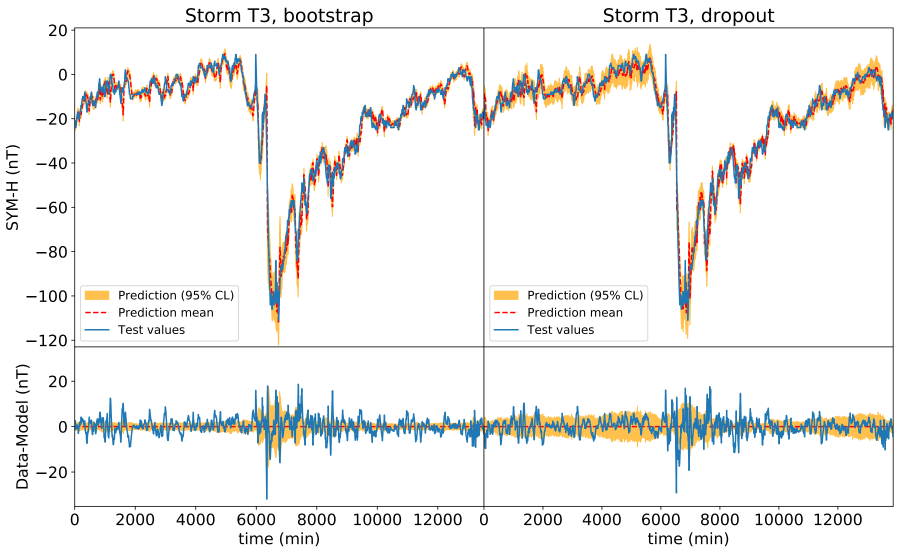
The first thing we note from these results is that both methods give similarly good results, on average, for the mean predictions (red dashed lines in the figures). This can be checked by the bottom panels of each storm, where we represent the residuals "Data - Model". Some exceptions occur, mainly around the peaks of the storms, where one method is noticeably better than the other (see e.g. storms T6 and T11, where dropout is better). On the other hand, concerning the prediction uncertainties, we see more differences, and it is worth noting that, as commented in section 4.3, what we report here are uncertainties on the expected values (means) of the SYM-H, and not on the variable itself. In other words, these uncertainties are not the total ones resulting from adding the data noise, which we do not have. Coming back to Figure A1, typically the uncertainties on regions away from the peaks are larger (or at most similar) for dropout than for bootstrap. However, the opposite is true when focusing on the regions around the peaks, and in general it is bootstrap the method giving larger (or at most similar) uncertainties than dropout. In Figure A2 we simply zoom-in around the peaks of maximum activity for two

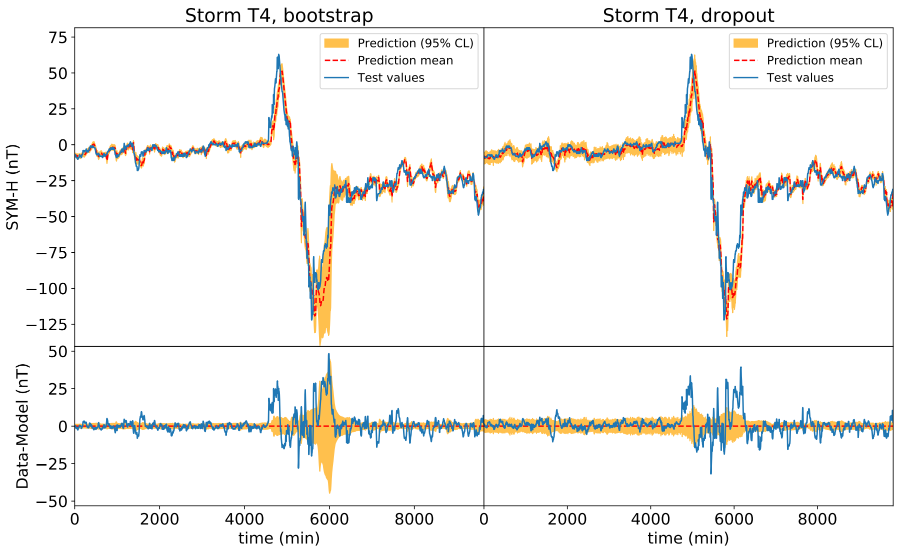
[4] Test storms T2 and T12 are the ones included in Figure 7.

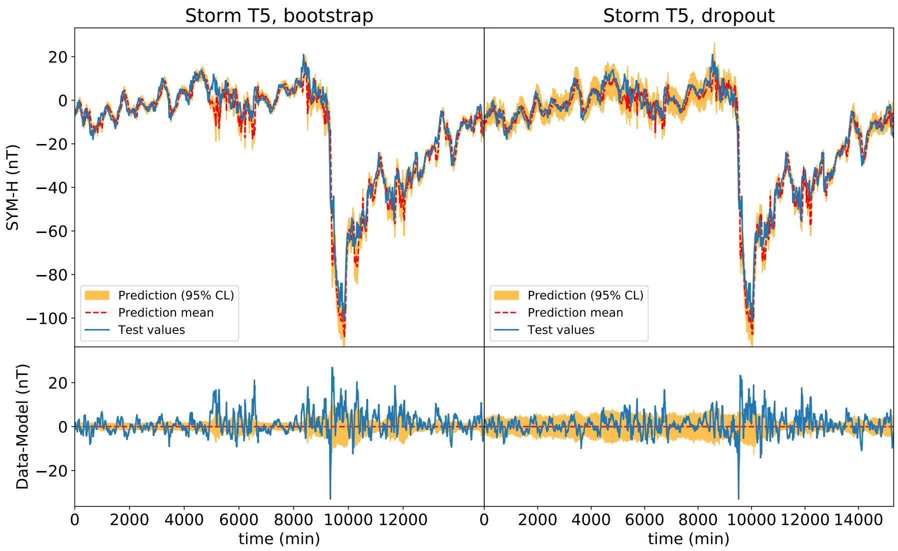
particular storms, T7 and T8, where this feature is more evident. Taking into account that the critical period of time of a storm is precisely when the peaks occur, the best procedure is chosen to be the one giving better results in that region of the storms. Here better means not only good predictions, but also conservative prediction uncertainties. For that reason, we have selected bootstrap to be the main procedure for obtaining the predictions in this work.

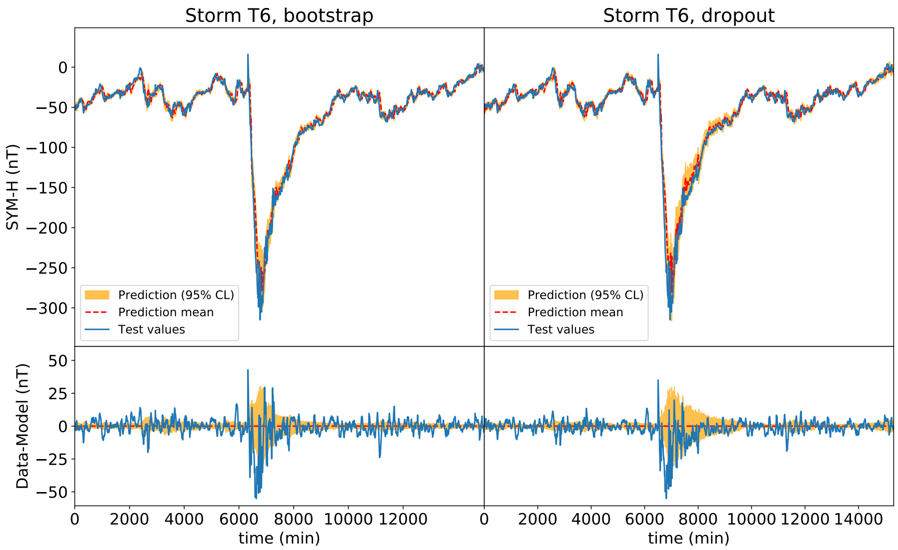


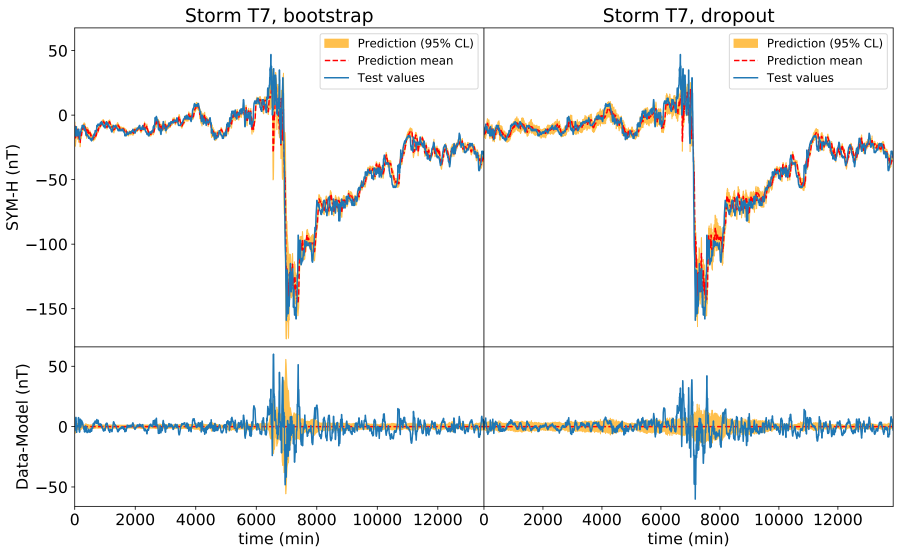


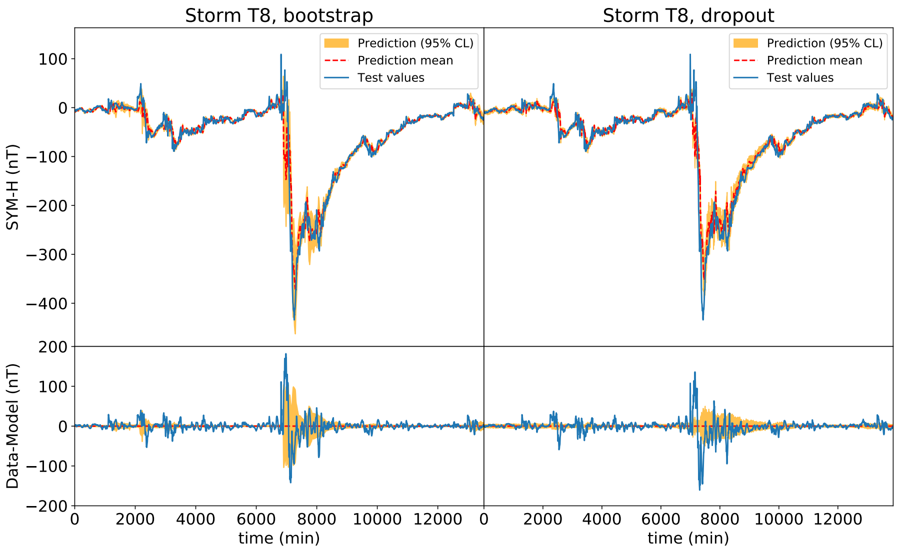

Storm T3, bootstrap
Storm T3, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

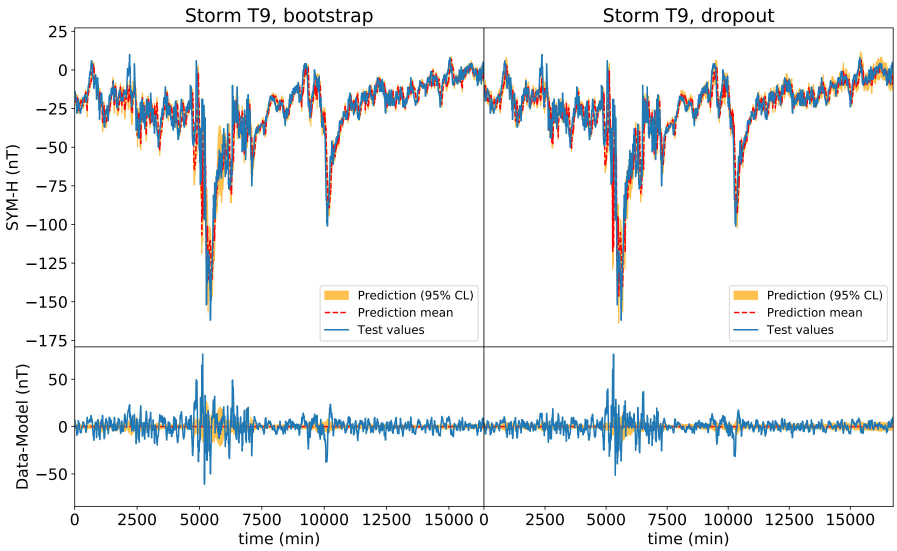

Storm T4, bootstrap
Storm T4, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

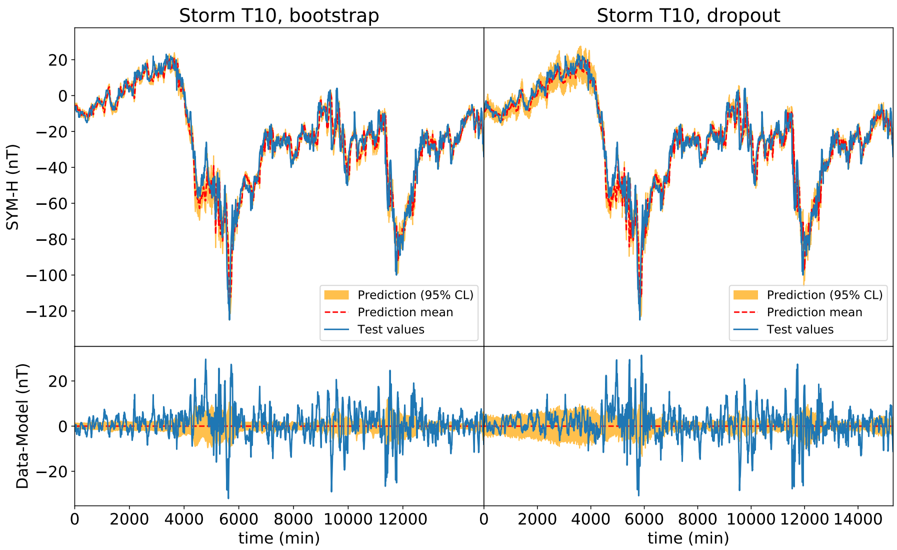

Storm T5, bootstrap
Storm T5, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

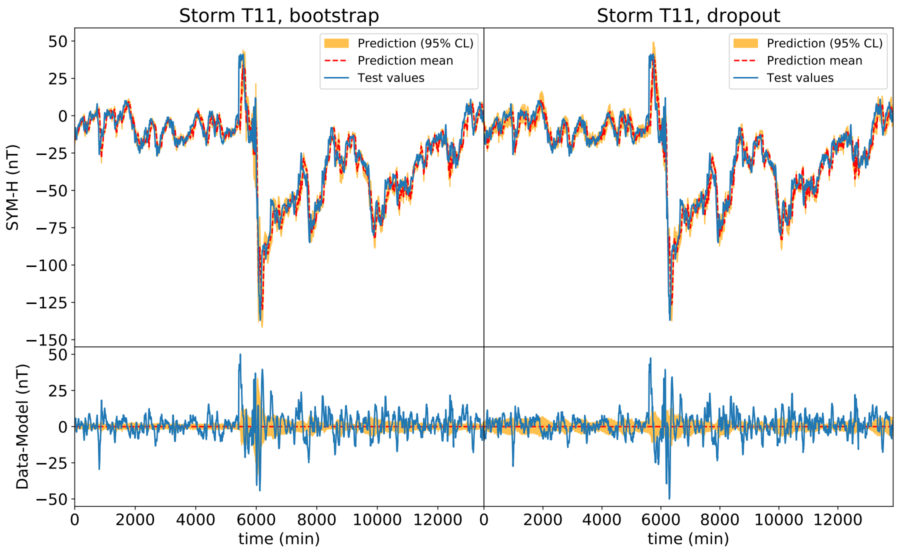

Storm T6, bootstrap
Storm T6, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

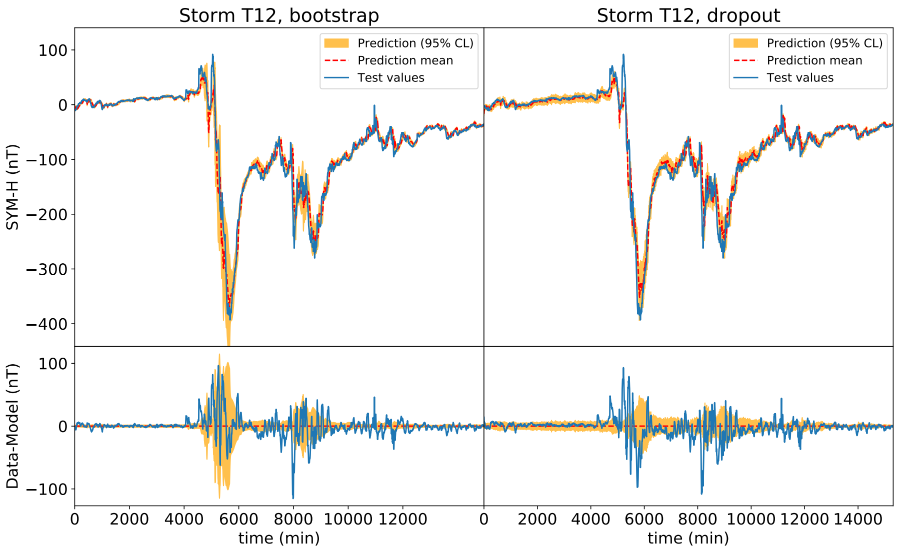

Storm T7, bootstrap
Storm T7, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)

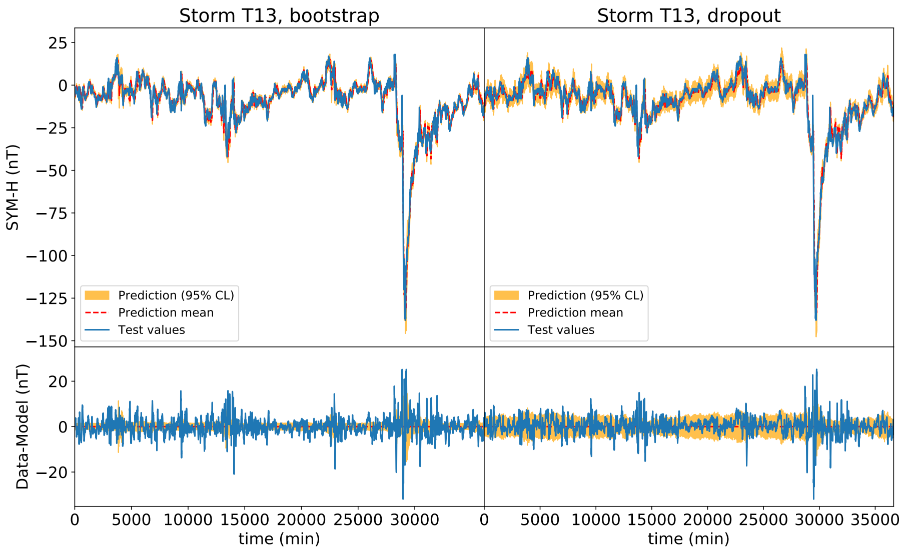

Storm T8, bootstrap
Storm T8, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)

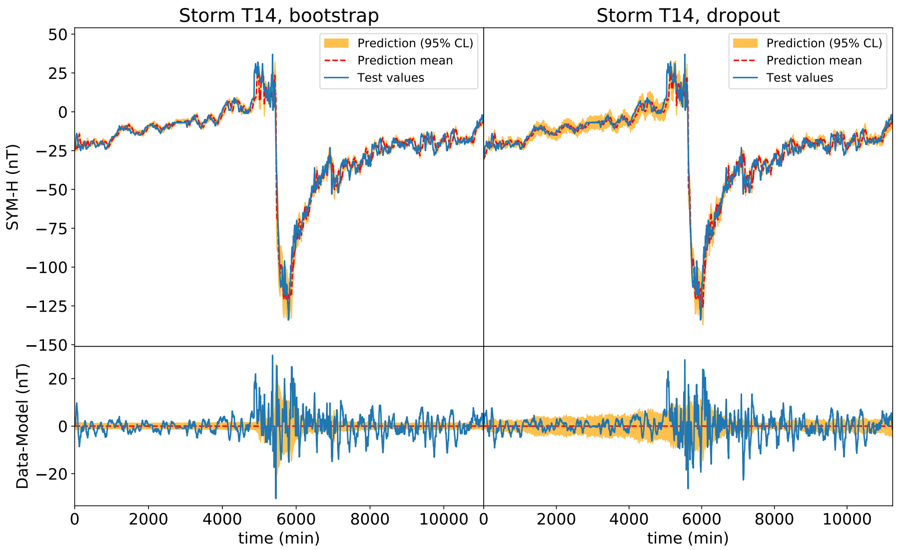

Storm T9, bootstrap
Storm T9, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

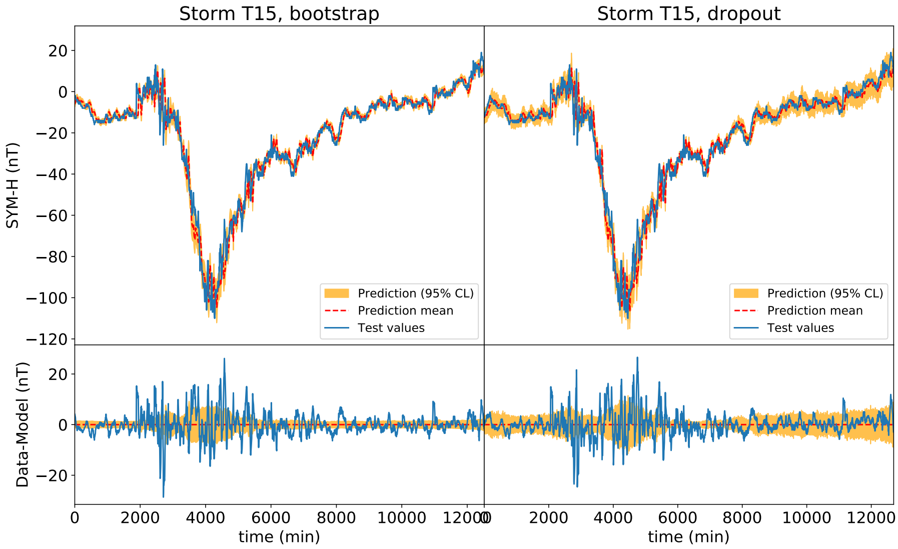

Storm T10, bootstrap
Storm T10, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

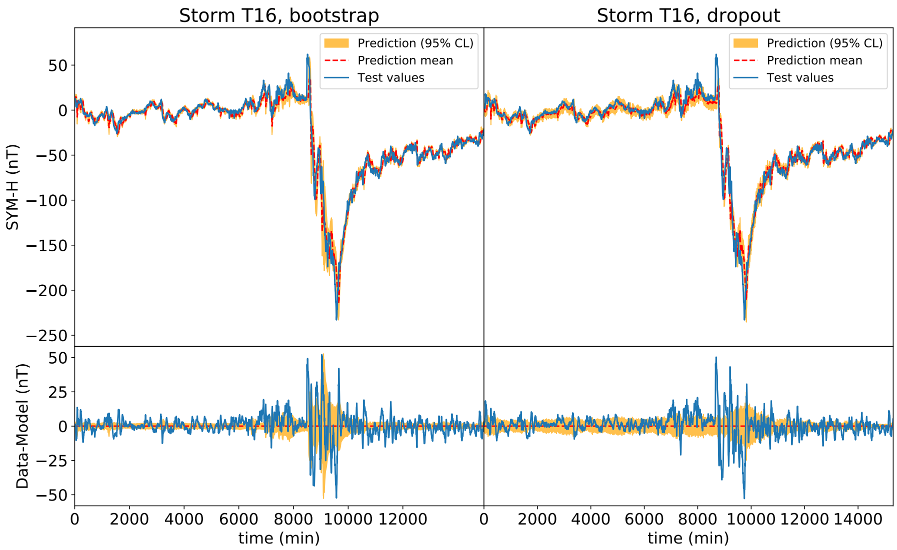

Storm T11, bootstrap
Storm T11, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)


Storm T12, bootstrap
Storm T12, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)


Storm T13, bootstrap
Storm T13, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T14, bootstrap
Storm T14, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T15, bootstrap
Storm T15, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)
time (min)


Storm T16, bootstrap
Storm T16, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)
time (min)

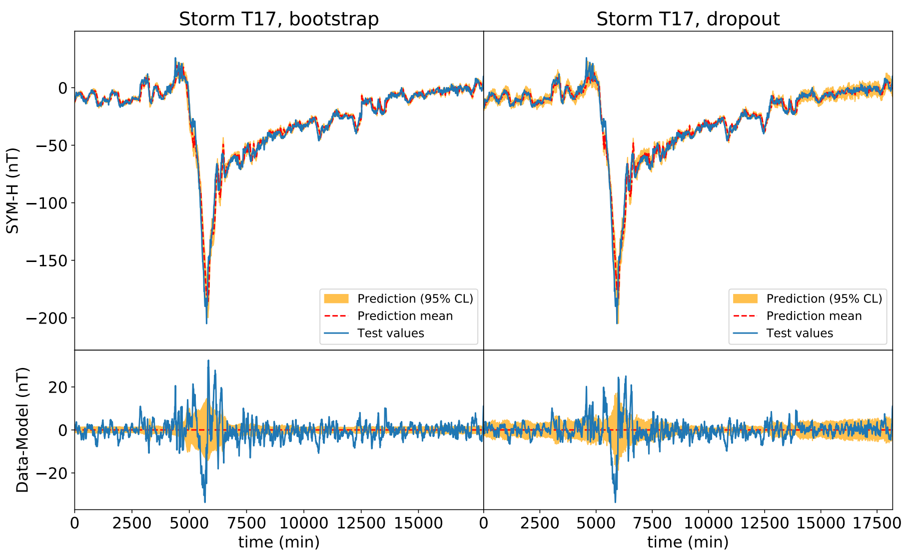


Figure A1. Time-series distributions for all 17 storms in the test sub-dataset, showing the results using the bootstrap method (left) and the dropout method (right). In all distributions, we show in an orange band the 95% CL (corresponding to 2σ), in red dashed line the mean for the one-hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.

Acknowledgments

We acknowledge use of NASA/GSFC's Space Physics Data Facility's OMNIWeb (or CDAWeb or ftp) service, and OMNI data. We also gratefully acknowledge the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana (Spain) as well as the technical support provided by the Instituto de Física Corpuscular (CSIC-UV). We thank our colleagues at the Institut de Recerca Geomodels from the Universitat de Barcelona for their expertise in SW, GIC and geoelectrical modelling, for their guidance in the use of the data and for their constructive comments and advice. Furthermore, the authors are grateful to the Spanish research grants PID2020-113135RB-C32 and PID2020-113135RB-C33 funded by MCIN/AEI/10.13039/501100011033 that supports this work. We also acknowledge the support from Generalitat Valenciana of the PROMETEO (ref. PROMETEO/2021/083) and GenT (ref. CIDEGENT/2020/055) research excellence programmes as well as support from MCIN/AEI of the "Ramon y Cajal" programme (ref. RYC2020-030254-I).

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. doi: https://doi.org/10.48550/arXiv.1907.10902

Bailey, R. L., Leonhardt, R., Möstl, C., Beggan, C., Reiss, M. A., Bhaskar, A., & Weiss, A. J. (2022). Forecasting gics and geoelectric fields from solar wind data using lstms: Application in austria. *Space Weather*, *20*(3), e2021SW002907. doi: https://doi.org/10.1029/2021SW002907

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 24). Curran Associates, Inc.

Bhaskar, & Vichare. (2019). Forecasting of sym-h and asy-h indices for geomagnetic storms of solar cycle 24 including st. patricks day, 2015 storm using narx neural network. *Journal of Space Weather and Space Climate*, *9*(A12). doi: https://doi.org/10.1051/swsc/2019007

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. doi: 10.1023/A:1010933404324

Burton, R. K., McPherron, R. L., & Russell, C. T. (1975). An empirical relationship between interplanetary conditions and dst. *Journal of Geophysical Research (1896-1977)*, *80*(31), 4204-4214. doi: https://doi.org/10.1029/JA080i031p04204

Cai, L., Ma, S. Y., & Zhou, Y. L. (2010). Prediction of sym-h index during large storms by narx neural network from imf and solar wind data. *Annales Geophysicae*, *28*(2), 381–393. doi: https://doi.org/10.5194/angeo-28-381-2010

Camporeale, E. (2019). The challenge of machine learning in space weather: Nowcasting and forecasting. *Space Weather*, *17*(8), 1166-1207. doi: https://doi.org/10.1029/2018SW002061

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM. doi: https://doi.org/10.1145/2939672.2939785

Collado-Villaverde, A., Muñoz, P., & Cid, C. (2021). Deep neural networks with convolutional and lstm layers for sym-h and asy-h forecasting. *Space Weather*, *19*(6), e2021SW002748. doi: https://doi.org/10.1029/2021SW002748

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81. Retrieved from http://jmlr.org/papers/v20/18-760.html

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Journal of Machine Learning Research Workshop and Conference Proceedings*, *48*. Retrieved from `http://proceedings.mlr.press/v48/gal16.pdf`

Gal, Y., Hron, J., & Kendall, A. (2017). Concrete dropout. doi: https://doi.org/10.48550/arXiv.1705.07832

Gleisner, H., H. Lundstedt, & Wintoft, P. (1996). Predicting geomagnetic storms from solar-wind data using time-delay neural networks. *Ann. Geophys.*, *14*, 679–686. doi: https://doi.org/10.1007/s00585-996-0679-1

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

Iong, D., Chen, Y., Toth, G., Zou, S., Pulkkinen, T., Ren, J., ... Gombosi, T. (2022). New findings from explainable sym-h forecasting using gradient boosting machines. *Space Weather*, *20*(8), e2021SW002928. doi: https://doi.org/10.1029/2021SW002928

Iyemori, T. (1990). Storm-time magnetospheric currents inferred from mid-latitude geomagnetic field variations. *J. Geomagn. Geoelectr*, *42*(11), 1249–1265. doi: http://dx.doi.org/10.5636/jgg.42.1249.

Kellinsalmi, M., Viljanen, A., Juusola, L., & Käki, S. (2022). The time derivative of the geomagnetic field has a short memory. *Annales Geophysicae*, *40*(4), 545–562. doi: https://doi.org/10.5194/angeo-40-545-2022

King, J., & Papitashvili, N. (2005). Solar wind spatial scales in and comparisons of hourly wind and ace plasma and magnetic field data. *Journal of Geophysical Research: Space Physics*, *110*(A2). doi: http://dx.doi.org/10.1029/2004JA010649

Leontaritis, I. J., & Billings, S. A. (1985). Input-output parametric models for non-linear systems part ii: stochastic non-linear systems. *International Journal of Control*, *41*(2), 329-344. doi: https://doi.org/10.1080/0020718508961130

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`

Lundstedt, H., & Wintoft, P. (1994). Prediction of geomagnetic storms from solar wind data with the use of a neural network. *Ann. Geophys.*, *12*, 19–24. doi: https://doi.org/10.1007/s00585-994-0019-2

Madsen, F. D., Beggan, C. D., & Whaler, K. A. (2022). Forecasting changes of the magnetic field in the united kingdom from l1 lagrange solar wind measurements. *Frontiers in Physics*, *10*. doi: https://doi.org/10.3389/fphy.2022.1017781

Marsal, S., & Curto, J. (2009). A new approach to the hourly mean computation problem when dealing with missing data. *Earth, Planets, and Space*, *61*, 945-956. doi: https://doi.org/10.1186/BF03352945

Mayaud, P. N. (1980). Introduction. In *Derivation, meaning, and use of geomagnetic indices* (p. 1-2). American Geophysical Union (AGU). doi: https://doi.org/10.1002/9781118663837.ch1

Papitashvili, N. E., & King, J. H. (2023a). *Omni 1-min data [data set]. nasa space physics data facility.* https://doi.org/10.48322/45bb-8792. (Last accessed on March 3, 2023)

Papitashvili, N. E., & King, J. H. (2023b). *Omni 5-min data [data set]. nasa space physics data facility.* https://doi.org/10.48322/gbpg-5r77. (Last accessed on March 3, 2023)

Patowary, R., Singh, S., & Bhuyan, K. (2013). A study of seasonal variation of geomagnetic activity. *Research Journal of Physical and Applied Sciences*, *2*, 1-11.

Pinto, V. A., Keesee, A. M., Coughlan, M., Mukundan, R., Johnson, J. W., Ngwira, C. M., & Connor, H. K. (2022). Revisiting the ground magnetic field perturbations challenge: A machine learning perspective. *Frontiers in Astronomy and Space Sciences*, *9*. doi: https://doi.org/10.3389/fspas.2022.869740

Qin, Z., Denton, R. E., Tsyganenko, N. A., & Wolf, S. (2007). Solar wind parameters for magnetospheric magnetic field modeling. *Space Weather*, *5*(11). doi: https://doi.org/10.1029/2006SW000296

Rumelhart, D. E., & McClelland, J. L. (1987). Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations* (p. 318-362).

Siciliano, F., Consolini, G., Tozzi, R., Gentili, M., Giannattasio, F., & De Michelis, P. (2021). Forecasting sym-h index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, *19*(2), e2020SW002589. doi: https://doi.org/10.1029/2020SW002589

SpaceWeather-IFIC. (2023). Spaceweather-ific/open_data: v1.0. doi: https://doi.org/10.5281/zenodo.7695656

Torta, J. M., Marcuello, A., Campanyà, J., Marsal, S., Queralt, P., & Ledo, J. (2017). Improving the modeling of geomagnetically induced currents in spain. *Space Weather*, *15*(5), 691-703. doi: https://doi.org/10.1002/2017SW001628

Torta, J. M., Marsal, S., Ledo, J., Queralt, P., Canillas-Pérez, V., Piña-Varas, P., ... Martí, A. (2021). New detailed modeling of gics in the spanish power transmission grid. *Space Weather*, *19*(9), e2021SW002805. doi: https://doi.org/10.1029/2021SW002805

Wanliss, J. (2005). Fractal properties of sym-h during quiet and active times. *J. Geophys. Res*, *110*. doi: https://doi.org/10.1029/2004JA010544

Wanliss, J., & Uritsky, V. (2010). Understanding bursty behavior in midlatitude geomagnetic activity. *Journal of Geophysical Research: Space Physics*, *115*(A3). doi: https://doi.org/10.1029/2009JA014642

Zhang, W. (1988). Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of annual conference of the japan society of applied physics.*

Zhang, W., Itoh, K., Tanida, J., & Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Appl. Opt.*, *29*(32), 4790–4797. doi: https://doi.org/10.1364/AO.29.004790

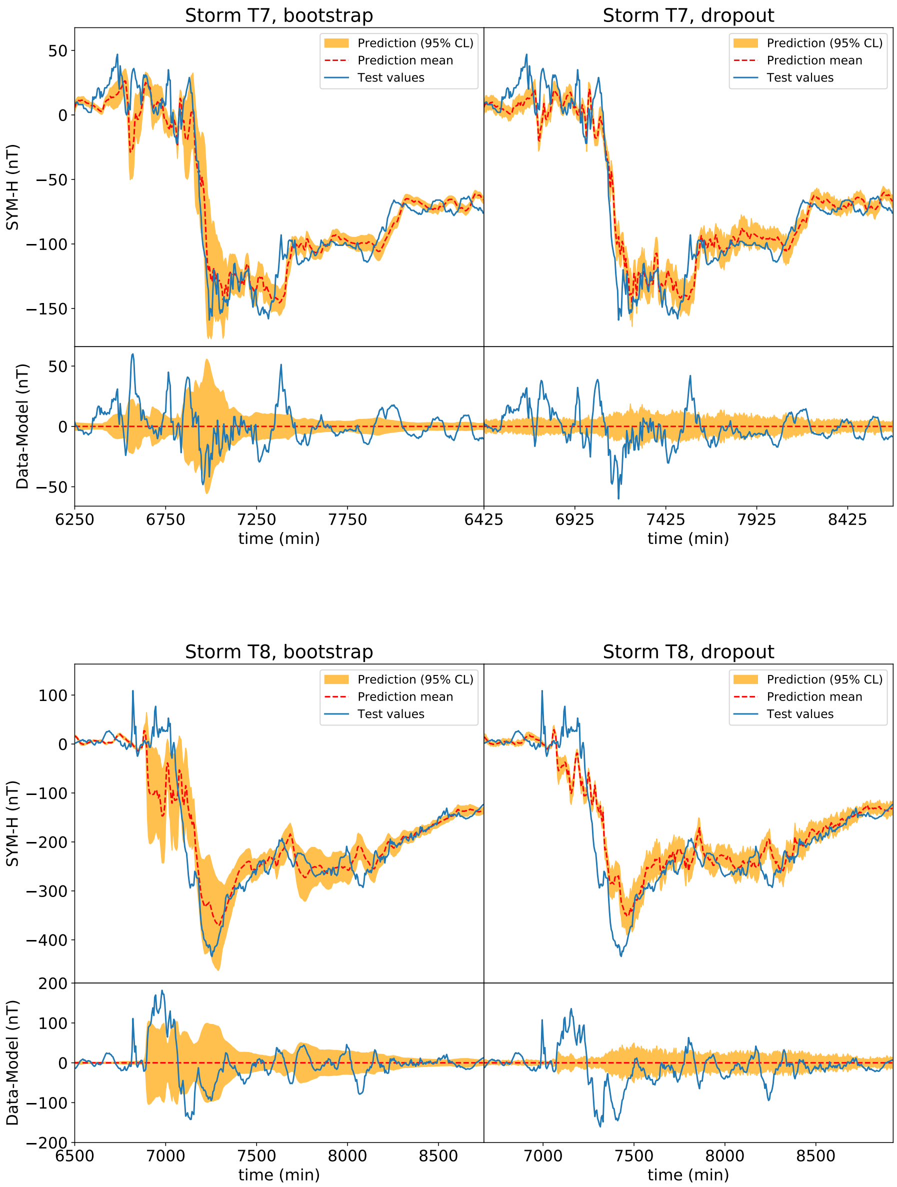


Figure A2. Zoom-in of the time-series distributions around the peaks of maximum activity for storms T7 and T8 in the test sub-dataset, showing the results using the bootstrap method (left) and the dropout method (right). In these distributions, we show in an orange band the 95% CL (corresponding to 2σ), in red dashed line the mean for the one-hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.

Forecasting Geomagnetic Storm Disturbances and Their Uncertainties using Deep Learning

D. Conde[1], F. L. Castillo[2], C. Escobar[1], C. García[1], J. E. García[1], V. Sanz[1,3], B. Zaldívar[1], J. J. Curto[4], S. Marsal[4], J. M. Torta[4]

[1]Instituto de Física Corpuscular (IFIC), Centro mixto CSIC - Universitat de València, Valencia, Spain
[2]Laboratoire d'Annecy de Physique des Particules (LAPP), Université Grenoble Alpes, Université Savoie Mont Blanc, CNRS/IN2P3, Annecy, France
[3]Department of Physics and Astronomy, University of Sussex, Brighton BN1 9QH, United Kingdom
[4]Observatori de l'Ebre (OE), CSIC - Universitat Ramon Llull, Roquetes, Spain

Key Points:

- An LSTM model is built to forecast the SYM-H index using interplanetary magnetic field measurements and past SYM-H values.
- The hyper-parameter optimisation and the robustness of the LSTM model is ensured by using dedicated algorithms and methods.
- Prediction uncertainties from the LSTM model are estimated and turn out to be considerable in the critical phases of geomagnetic storms.

Corresponding author: D. Conde, `Daniel.Conde@ific.uv.es`

Abstract

Severe space weather produced by disturbed conditions on the Sun results in harmful effects both for humans in space and in high-latitude commercial flights, and for technological systems such as spacecraft or communications. Also, geomagnetically induced currents flowing on long ground-based conductors, such as power networks or pipelines, potentially threaten critical infrastructures on Earth. The first step in developing an alarm system against geomagnetically induced currents is to forecast them. This is a challenging task, though, given the highly non-linear dependencies of the response of the magnetosphere to these perturbations. In the last few years, modern machine-learning models have shown to be very good at predicting magnetic activity indices as the SYM-H. However, such complex models are on the one hand difficult to tune, and on the other hand they are known to bring along potentially large prediction uncertainties which are generally difficult to estimate. In this work we aim at predicting the SYM-H index characterising geomagnetic storms one hour in advance, using public interplanetary magnetic field data from the Sun–Earth L1 Lagrange point and SYM-H. We implement a type of machine-learning model called long short-term memory networks. Our scope is to estimate -for the first time to our knowledge- the prediction uncertainties coming from a deep-learning model in the context of space weather. The resulting uncertainties turn out to be sizeable at the critical stages of the geomagnetic storms. Our methodology includes as well an efficient optimisation of important hyper-parameters of the long short-term memory network and robustness tests.

Plain Language Summary

Geomagnetic storms are disturbances of the geomagnetic field caused by interactions between the solar wind and particle populations mainly in the Earth's magnetosphere. These time-varying magnetic fields induce electrical currents on long ground-based conductors that can damage power transmission grids and other critical infrastructures on Earth. As a first step to forecast the ground magnetic perturbations caused by geomagnetic storms at specific mid-latitude locations, the objective of this work is to predict the SYM-H activity index, which is generated from ground observations of the geomagnetic field at low and mid-latitudes, and which provides a measure of the strength and duration of geomagnetic storms. We use the interplanetary magnetic field data measured by the ACE spacecraft at the L1 Lagrangian point and past SYM-H values to forecast the behavior and severity of geomagnetic storms one hour in advance. This forecasting is done using a type of artificial neural network model called long short-term memory. We also propose ways to estimate the uncertainties of these predictions, which help us to better understand machine-learning models in space weather prediction and could lead to more accurate and reliable forecasting of geomagnetic storms and their ground effects in the near future.

1 Introduction

In the last decades, our society has become more interdependent and complex than ever before. Local impacts can cause global issues, as the COVID-19 pandemic clearly showed, affecting the health of millions of human beings. Our society is highly dependent on relevant technological structures, such as communications, transport, or power transmission networks, which can be very vulnerable to the effects of space weather (SW). The latter has its origin in the solar activity and their associated events, such as coronal mass ejections and co-rotating interaction regions. Among other effects, these phenomena have an impact on the electrical current systems surrounding the Earth, enhancing them and thus causing large magnetic field fluctuations that propagate down to the ground. The electric field associated with these fluctuations, which is influenced by the interaction with the conductive earth, induces telluric currents in the uppermost solid layers and geomagnetically induced currents (GICs) in long conductors running on the surface. These GICs may cause disturbances, interruptions, and even long-term damage to critical infrastructures such as railways, oil and gas pipelines and power grids, with drastic social, economic and even political consequences. The intensity of the GICs is determined by the strength of the geoelectric field, but the latter measurements are rarely available. Because GICs are driven by temporal changes in the magnetic field, if we have an estimate of the resistivity structure below a specific location, variations in the magnetic field measured by ground magnetometers can in principle be used as the input parameter for deriving the GICs built up locally in a power grid (e.g. Torta et al. (2017)). However, because of the three-dimensional lithospheric resistivity structure, the behaviour of the time derivative of the geomagnetic field to which the ground electric fields are associated is complex and, consequently, has proven to be very difficult to predict (Kellinsalmi et al., 2022). Predicting geomagnetic indices, which attempt to condense a rich set of information about the status of the magnetosphere in a single number, is simpler and has always been a very attractive area for machine-learning (ML) applications (Camporeale, 2019). Although attempts to forecast geomagnetic indices started several decades ago (e.g. Burton et al. (1975)), they feature highly non-linear dependencies which are not yet well understood, and their forecasting is still an open and intensive area of research. Perhaps not surprisingly, recent efforts have been exploiting the large expressiveness offered by modern ML models, and their ability to characterise complicated multidimensional datasets. The present work follows such a trend by investigating advanced ML techniques to predict the behaviour of geomagnetic storms.

More specifically, our scope is to predict, at a given time in advance, the SYM-H index, which describes the geomagnetic disturbances at low and middle latitudes in terms of longitudinally symmetric disturbances of the horizontal component of the geomagnetic field (Iyemori, 1990). The SYM-H index is known to track very well the evolution in time, the topology and intensity of geomagnetic storms and their relation with solar source phenomena (Wanliss, 2005; Wanliss & Uritsky, 2010). We use time-series data from the Sun–Earth L1 Lagrange point tracking several covariates describing the interplanetary magnetic field (IMF) and its different components in addition to the SYM-H index. For this purpose, we predict the SYM-H index with a type of artificial neural network model called long short-term memory (LSTM) neural network (Hochreiter & Schmidhuber, 1997) especially conceived for describing, among others, non-linear time-series data.

Highly-parameterised neural networks as the ones we use in this work (as well as in recent literature) carry an important amount of intrinsic prediction uncertainty, called among statisticians "epistemic" uncertainty. This should be taken into account, where possible, in any scientific application, even more in those with direct impact on society as the present study. Yet another issue with those models is the presence of parameters (named "hyper-parameters" in the ML community) which are not directly optimised during the fitting processes, but whose impact on the predictions are potentially very large. Consequently, some sort of extra optimisation should be performed, which is typically computationally costly.

Furthermore, robustness in non-linear time-series predictions obtained from ML models (including LSTM) can be challenging due to their complex and often unpredictable nature. However, several techniques exist and can be used to test and improve the robustness of the models.

While studies on the prediction of geomagnetic indices with ML techniques have been conducted recently (see section 2), the novelty of our work is two-fold:

- For the chosen ML model (LSTM in this case), we report our predictions for the SYM-H index with associated uncertainties.
- We optimise the hyper-parameters of our model, in particular, following an efficient Bayesian optimisation strategy.

- The robustness of our LSTM model is evaluated not only with the standard hold-out method but also by reshuffling the list of geomagnetic storms.

2 Related Work

Efforts to forecast geomagnetic indices date back to the 1980's (Mayaud, 1980), which started using linear prediction models which were unable to capture well enough the complexity of the response of the magnetosphere to SW. For this reason, the community started to rely on the arbitrarily high expressiveness of neural network models (e.g. Lundstedt and Wintoft (1994); Gleisner and Wintoft (1996)).

Among these works, the one developed by Siciliano et al. (2021) constitutes a valuable reference from which we have started our study. They forecast the SYM-H index, for which one can have a priori finer time granularity with respect to other indices, thus being advantageous from the point of view of an alert system. Siciliano et al. (2021) compared the SYM-H predictions using two different neural network models: the LSTM and the convolutional neural network (CNN), the latter being typically used for image recognition tasks (Zhang, 1988; Zhang et al., 1990). While they have obtained good performances with the CNN compared to the LSTM (in some cases even slightly better), in our study we concentrate on the LSTM only, which for us delivered similar performances as the CNN. However, as commented above, we address the important issue of the uncertainty estimation, along with a detailed and explicit hyper-parameter optimisation together with an additional robustness test.

Posterior work by Collado-Villaverde et al. (2021) revolves on the same idea, but using a neural network architecture which actually combines CNN and LSTM transformations to predict not only the SYM-H index but also the complementary ASY-H index. With respect to our work, their architecture is different in that we use a standard LSTM model. Bhaskar and Vichare (2019) also predict both indices using a non-linear autoregressive exogenous (NARX) model (Leontaritis & Billings, 1985). On the other hand, Bailey et al. (2022) aim at forecasting the geoelectric field with LSTMs as well. While Pinto et al. (2022) forecast the ground magnetic field time derivative with LSTMs, Madsen et al. (2022) forecast both the ground magnetic field and its time derivative with LSTM networks and hybrid CNN-LSTMs. None of the works mentioned above, nor others less related to our study but in the same context, estimate the prediction uncertainties, nor have they thoroughly optimised their hyper-parameters. The only exception we were able to find was the very recent work by Iong et al. (2022), which studied the SYM-H index by using not neural networks, but another ML model belonging to ensemble methods (in particular, using a regularising gradient boosting framework; the eXtreme Gradient Boosting (XGBoost) library (Chen & Guestrin, 2016)), obtaining very good performance as well. In their case, while not estimating their prediction uncertainties, the hyper-parameter optimisation was actually performed, using a gradient-free “black box” optimisation method. The latter is a generic algorithm most convenient for situations where little or no information is known about the structure of the function to optimise. In our study, on the other hand, we use an optimisation algorithm particularly suitable for the type of objective function we have, so it is arguably more efficient.

3 Dataset Selection and Processing

The dataset used in this work corresponds to a sample of geomagnetic storms that occurred between 1998 and 2018, which were recorded at ground-based geomagnetic observatories, and were preceded by changes in the magnetic field and plasma parameters of the interplanetary medium, which were measured at the L1 Lagrange point by NASA's Advanced Composition Explorer (ACE) spacecraft. The geomagnetic storms have been selected following the same criteria as in Siciliano et al. (2021), in order to make a direct comparison with this previous work. The sample contains 42 of the most intense geomagnetic storms, distributed in two solar cycles. The intensity of the storms is defined by the SYM-H index. This index can be considered as a proxy of the response of the Earth's magnetosphere (especially the ring current) to solar activity and it is computed from data of a network of six magnetic observatories distributed in longitude across the low and middle-latitude region, with a time resolution of 1 min and precision of 1 nT. All the geomagnetic storms selected have a minimum SYM-H index lower than -100 nT, so they can be considered as either severe or extreme (Patowary et al., 2013). This ensures a high signal-to-noise ratio. Indeed, 55% of all these geomagnetic storms (23 out of 42) have a minimum SYM-H value between -200 nT and -100 nT, while the rest (i.e. 19 geomagnetic storms) have a minimum SYM-H value below -200 nT.

As Siciliano et al. (2021), we follow the commonly used hold-out method for training a ML model which is the process of dividing the full dataset into different splits and then using one split for training the model and other splits to validate and test it. Table 1 lists the geomagnetic storms classified in three sub-datasets containing data from different storms. These three sub-datasets are uniformly populated in terms of geomagnetic storm intensity and complexity. The training sub-dataset is used to train the LSTM model, the validation sub-dataset stops the network training and prevent over-fitting, while the test (also known as hold-out) sub-dataset is used as a proxy to evaluate the performance of the model on unseen data.

The length of the time interval of the considered geomagnetic storms range from 6 to 25 days, with an average of 10 days. This choice allows us to consider not only the main phase periods but also the initial and recovery phases, as well as previous and later quiet periods. The three sub-sets are uniformly populated in terms of geomagnetic storm intensity and complexity, the latter measured by the presence of multiple depressions of the magnetic field.

As already mentioned by Siciliano et al. (2021), a larger number of geomagnetic storms can be considered, as done for instance by Cai et al. (2010) and Bhaskar and Vichare (2019), though the additional ones are just either weak or moderate geomagnetic storms, adding no further predictive power to our LSTM model. This is due to the fact that all storm phases including quiet periods are already considered in all three sub-datasets.

The independent variables (commonly named "features" in ML) used for training the LSTM model are the squared value of the IMF magnitude B, the squared value of the IMF B_y component, the IMF B_z component (all these in GSM coordinates recorded at L1 Lagrange point by the ACE satellite) and the SYM-H index. The forecasting variable is the SYM-H index as mentioned above. All these variables are shown in table 2. All data are extracted from the NASA's OMNIWeb page (https://omniweb.gsfc.nasa.gov) with time resolution of 5 min (Papitashvili & King, 2023b). Although the data are available with a resolution of 1 min (Papitashvili & King, 2023a), the election of a lower resolution allows to reduce the computation time without reducing predictive power, allowing also a direct comparison with the results of Siciliano et al. (2021). The 5 min sample is computed by averaging the 1 min samples, so that the data at minute 0 corresponds to the average from minutes 0 to 4.

The IMF variables are propagated to the nominal magnetospheric bow shock following the method described in the OMNIWeb site (King & Papitashvili, 2005). It is important to note in this context that in this study (and also in that of Siciliano et al. (2021)) the information available on SYM-H at the Earth's surface is assumed to be simultaneous with that of the IMF projected at the bow shock. However, the spacecraft measuring the IMF is located upstream of the solar wind at the L1 Lagrange point, which allows these data to be known some time in advance (typically between 15 and 60 min). This advantageous position, which is therefore not exploited here, is expected to have a significant role in the efficiency of the SYM-H predictions.

The IMF data overflows are removed from the full sample of geomagnetic storms and the remaining empty gaps are filled using a linear interpolation method. Geomagnetic storms generally have short periods with IMF data overflows but in few cases (e.g. training storm TR13 and validation storm V3) the overflows are large and occur near the peak of the storm activity. While linear interpolation is one possible way to address the problem of gaps, it is clearly not optimal when these are located in periods of high activity. We are aware that more sophisticated approaches could be exploited (e.g. the interpolation schemes proposed by Qin et al. (2007) or Marsal and Curto (2009) or even a ML-based method we are currently developing), but we used the same approach as Siciliano et al. (2021) to perform a direct comparison between their results and ours. Figure 1 illustrates the removal of overflows in the IMF variables (B^2, B_y^2, B_z) followed by a linear interpolation to fill the resulting gaps for a particular storm, TR13, of the training sub-dataset. It was decided to keep the storms with overflows near maximum of activity in the study, both to train and to stop training the network, since they represent a real possibility when part of the data is lost due to measurement errors or overflows or even detector failures.

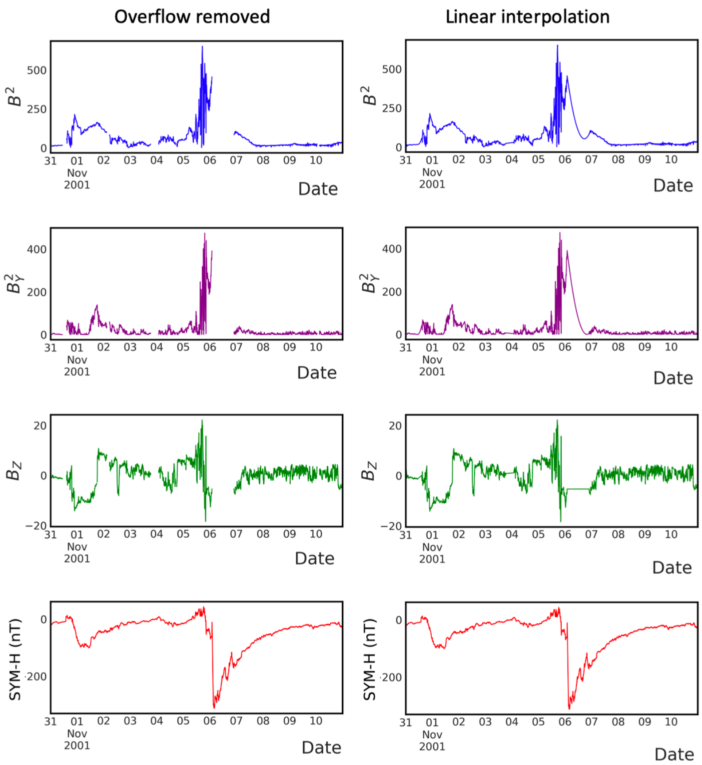


Figure 1. Training variables (B^2, B_y^2, B_z, SYM-H) for storm TR13 of the training sub-dataset, after overflow removal (left) and after linear interpolation to fill the gaps (right). The SYM-H index is also shown for completeness as it is the fourth variable used in the training.

Each geomagnetic storm time series is standardised using the associated mean and standard deviation, similar to Siciliano et al. (2021). However, in our case, the mean and standard deviation are just estimated from the training sub-dataset, given that this is the only data previously known for the study. With the standardisation the values are centred around the mean and given a unit standard deviation using the equation:

$$X_S = \frac{X - \mu_{\mathrm{TR}}}{\sigma_{\mathrm{TR}}}\,, \tag{1}$$

where X_S is the standardised time-series data, X is the original time-series data, μ_{TR} is the mean of the training series and σ_{TR} its standard deviation. Making each time series similar to a normal distribution, it is less sensitive to the scale of features and more consistent with each other, thus allowing the model to predict outputs more accurately. In addition, this scaling method is more resilient to outliers than the more common normalisation between $[-1, 1]$ or $[0, 1]$, which only considers the minimum and maximum values instead of the overall statistics of the data.

4 Deep-Learning Model

Humans, as intelligent beings, do not have to learn how to speak, walk or cook from scratch every time. Our previous experiences on these tasks endure, and the ability to do it improves after several repetitions. Traditional neural networks do not have this feature, and it is a major deficiency for some specific tasks such as time-series forecasting or natural language processing. Recurrent neural networks (RNNs) (Rumelhart & McClelland, 1987) address this problem. RNNs integrate cyclic connections, allowing information to persist, making them a more powerful tool to model sequential data than the traditional feed-forward neural networks. LSTM networks are a variety of RNNs, capable of learning long-term dependencies and also forgetting irrelevant information, especially in sequence prediction problems. RNNs read the data sequentially, and attribute higher weights to the recent information. The back-propagation training of RNNs suffers from the so-called "vanishing gradient" problem, which limits the learning capabilities of the network. LSTMs address this problem by recognising between long-term and short-term memory through a gating mechanism that regulates the flow of information. This allows the model to selectively retain or forget information based on its relevance, making it more robust and able to handle complex sequential patterns. In this work we use one such type of neural network with a typical architecture shown in Figure 2, which we briefly describe below.

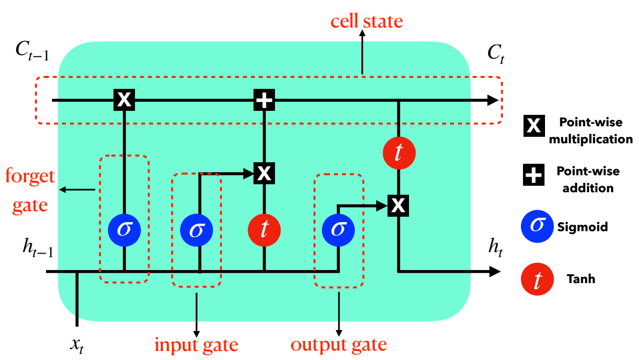


Figure 2. LSTM "cell" based on four interacting layers (cell state, input gate, forget gate and output gate). An LSTM network consists of repetitions of such a cell for every step t of the time series.

An LSTM network consists of a series of non-linear transformations for each time step with shared parameters (weights). The transformation for a generic input x_t at time step t is schematically represented in Figure 2. For each t, there are two outputs: the cell state c_t and the hidden output h_t, which are computed from four quantities. The first quantity is the result of the "forget gate" f_t, consisting typically of a sigmoid function applied to the linear function $W_f x_t + U_f h_{t-1} + b_f$. Here x_t is the input, h_{t-1} the hidden output of the previous time step, while W_f and U_f are the weight matrices and b_f is the bias vector. The motivation for the forget layer -whose value is a number in the range $[0, 1]$- is to decide how much to "forget" about the previous time step's cell state c_{t-1} (i.e. "0" forgets the previous data and "1" uses the previous data). The second quantity is the "cell input" $\tilde{c}_t$, similar in structure to the forget gate, but with its own set of weights W_c, U_c and b_c, and using a tanh function instead of a sigmoid. It represents the new information that would potentially be included in the new cell state c_t. How much of such information to be retained is determined by the third quantity: the result of the "input gate" i_t, defined as another sigmoid transformation, but with its corresponding weights W_i, U_i and b_i. With the above three quantities and their interpretations, the new cell state is defined in an intuitive way as: $c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t$, with $c_0 = 0$ by definition. Finally, the fourth quantity is the result of the "output gate" o_t, being yet another sigmoid with weights W_o, U_o and b_o. It has the role of a weight factor for the hidden output, computed as $h_t = o_t \times \tanh(c_t)$, with $h_0 = 0$.

The above series of transformations are consecutively applied to a finite number of time steps, from $t - l_b$ to t, with l_b being an optimisable hyper-parameter called the "look-back". The final aim is to predict an observable y_{t+l_f} at a future time $t+l_f$, with l_f being the "look-ahead", representing how much time in advance we want to make a prediction, which is fixed by the domain needs. In order to make such a prediction, right after the last LSTM cell applied on t and resulting in the hidden output h_t, a number of dense network layers are applied to transform the vector h_t into the predicted value of y_{t+l_f}, while potentially adding more expressiveness to the model. The number of dense layers is also another hyper-parameter to be optimised.

4.1 Hyper-Parameter Optimisation

Hyper-parameter optimisation is an essential ingredient when training state-of-the-art ML models, due to their high complexity. Traditional random- or grid-search strategies have shown to be very inefficient when the number of hyper-parameters is larger than a few. Modern libraries exist which implement efficient algorithms for optimising costly functions. One of the most popular ones, which we adopted here, is the hyper-parameter optimisation framework "Optuna" (Akiba et al., 2019). In particular, we run Optuna to optimise the following hyper-parameters: the number of dense layers, the number of units of these layers, the learning rate, and the look-back parameter of the LSTM layer. We use Optuna's implementation of a Bayesian optimisation flavour called "tree-structured parzen estimator", the details of which are found in Bergstra et al. (2011).

Finding multiple local minima can be a problem in hyper-parameter search. In particular, because of the stochastic nature of gradient descent during training, there can be times when two identical trials result in a value of the loss function that varies more greatly than trials with different hyper-parameters would. For this reason, each trial is repeated five times. The mean and standard deviation of the loss function results for each trial with a set of hyper-parameters are calculated. Having done this, all trials with root-mean-square error (RMSE) standard deviations that overlap with the best (i.e. lowest) RMSE mean of all trials are labeled as *best trials*. This procedure allows us to explore flat directions in the hyper-parameter space, as discussed later in section 5.

4.2 Robustness of the LSTM model

Ensuring the robustness of state-of-the-art ML models, used to analyse and predict non-linear time-series data, is of critical importance for reliable and effective decision-making, especially on those with direct impact on society. This requires careful design as well as rigorous testing and validation, but the benefits in terms of reliability and effectiveness are significant. Non-linear time-series data can exhibit complex and dynamic behaviour, making it challenging to model and predict accurately.

As in many other works, we ensure the robustness of our LSTM model by using the hold-out validation technique and therefore splitting the full dataset into three different sub-datasets (training, validation and test) containing uniformly populated times-series data from different geomagnetic storms. This technique helps to identify weaknesses and guarantee the model to perform and generalise well on new and unseen data, even in the presence of various perturbations, such as data noise or changes in the distribution of the times-series data.

However, the performance estimate of the ML model may be highly dependent on the particular dataset split used. If the split is not representative of the overall dataset distribution, then the performance estimate may be biased. In our case the three sub-datasets are uniformly populated in terms of geomagnetic storms intensity and complexity and therefore no bias is expected. However, to evaluate this issue we reshuffled the original list of geomagnetic storms in the three sub-datasets shown in Table 1. Thus, we populated the new training sub-dataset with the 17 storms from the original test sub-dataset plus three storms from the original validation sub-dataset. To populate the new test sub-dataset, we used 17 storms from the original training sub-dataset. Finally, the new validation sub-dataset was filled with the remaining three storms from the original training sub-dataset plus two validation storms from the original validation set. A visual representation of the baseline and reshuffled lists of geomagnetic storms used for training, validation and test is shown in Figure 3. With the new reshuffled list of the geomagnetic storms, we trained an alternative LSTM model (with its own optimised hyper-parameters) and obtained compatible performance without observing over-fitting, under-fitting or biases.

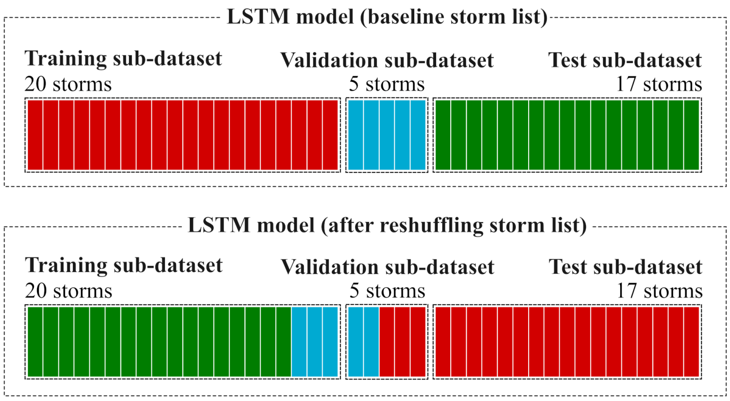


Figure 3. Visual representation of the two lists of geomagnetic storms used for training, validation and test for the baseline and the alternative LSTM models. The upper diagram represents the baseline list, as shown in Table 1 (i.e. same criteria as Siciliano et al. (2021)), while the lower diagram represents an alternative ordering where the list of storms is reshuffled. Each rectangle represents a different storm. Colours are assigned based on the baseline list of the geomagnetic storms, where red, blue and green represent the original storms in the training, validation and test sub-datasets, respectively.

Other techniques could also be used to enhance the robustness of LSTM models such as different flavours of cross-validation, data augmentation, model ensembling, adversarial training or regularisation, though we will deeply explore this in future works.

In any case, we want to point out that the regularisation technique is indeed used in this work for the estimation of uncertainties (see section 4.3 and Appendix A).

4.3 Estimation of Prediction Uncertainties

To estimate the uncertainties associated with our predictions, two main approaches can be followed: a frequentist approach, in particular adopting the "bootstrapping" method, or a Bayesian approach, where several state-of-the-art methods can be adopted depending on the needs and scope. In this work we have followed the two methods, and compared the results between them.

Bootstrapping is a series of techniques by which we obtain synthetic datasets out of the "real" (observed or simulated) dataset we have at our disposal. In doing this, both aleatoric and epistemic uncertainties are taken into account when making predictions, making bootstrapping equivalent to the principled Bayesian approach. In the physics community, the typical methodology is to: 1) propose a likelihood distribution of the data, and optimise its parameters by maximum likelihood estimate (MLE) method, 2) with these optimum parameters, use the proposed likelihood to sample a large number of synthetic datasets, identical in length to the original one, 3) find for each synthetic dataset the MLE parameters analogously as in step 1, and 4) each of the MLE parameters will lead to a different prediction, thus obtaining a distribution of predictions. While this technique works very well for many situations, it may be misleading when the assumed likelihood is very different from the true -unknown- underlying distribution of the data. For this reason, in the ML community there is another popular bootstrapping strategy, which consists in re-sampling a large number of times the real dataset directly, either with or without replacement[1]. This is equivalent to sampling from the empirical distribution, instead of assuming a particular parametric shape of the likelihood.

The traditional bootstrapping fails with time series because the sampling procedure breaks off the time dependence that concatenates adjacent samples in the sets. For this reason, a special consideration has to be made for our case. If we can divide the set in chunks of samples, and perform the bootstrap sampling procedure on these blocks instead of on the individual samples, we can conserve the time dependence up to the division of the blocks; for the present dataset, a natural way to divide the training set is by geomagnetic storms, in particular because we gain the advantage of explicitly breaking adjacent samples of different storms that are not expected to have a time dependency.

On the other hand, we have also followed a Bayesian approach for estimating the prediction uncertainties. In the case of deep neural networks one of the most popular strategies is the so-called "dropout" method (Gal & Ghahramani, 2016). More details on this can be found in Appendix A, where we also show the corresponding results as well as the comparison with respect to the bootstrap method. In summary, for this particular dataset we find that the bootstrap results perform better, especially around the peak of the storms, which is the most critical region. We thus retain the bootstrap predictions and corresponding uncertainties as our main results.

A note of caution is in order at this point. When reporting our prediction uncertainties, we are more specifically reporting the systematic (or epistemic) uncertainties of the expected (mean) values of the SYM-H index, which we calculate as the output of our LSTM network. Note that this is not the same as the total prediction uncertainties, which include the data noise (also known as aleatoric, or statistical uncertainties), and which we do not have available. Since the reported epistemic uncertainties decrease as the number of data points increase, it is perfectly consistent to have a very precise de-

[1] "With replacement" means that a particular instance of the real dataset can appear more than once in the synthetic dataset.

termination of the mean predictions of the SYM-H index, describing data whose noise is appreciably larger (and consequently having values beyond the corresponding interval of epistemic uncertainties). On the other hand, we are also not considering in this work the uncertainties related to the other input variables (related to the IMF B and its components). While we plan to include them in a future work, we nonetheless expect their impact on our results to be small, after checking that the uncertainties in B^2 are of few percent.

5 Results

After discussing the data and analysis setup in previous sections, we turn now to present the results of our analysis.

In the first subsection we present the optimisation of hyper-parameters needed to learn the evolution of the SYM-H index, the relative importance of each parameter in the final result and explore the possible correlation between hyper-parameters. We then present the overall performance of the trained algorithm when predicting the evolution of the SYM-H index.

5.1 Hyper-Parameter Tuning

Once we have chosen an LSTM as the basic architecture for the time-evolution analysis, the next step is to optimise the hyper-parameters of the learning structure. The results of this analysis are shown in Figures 4 and 5.

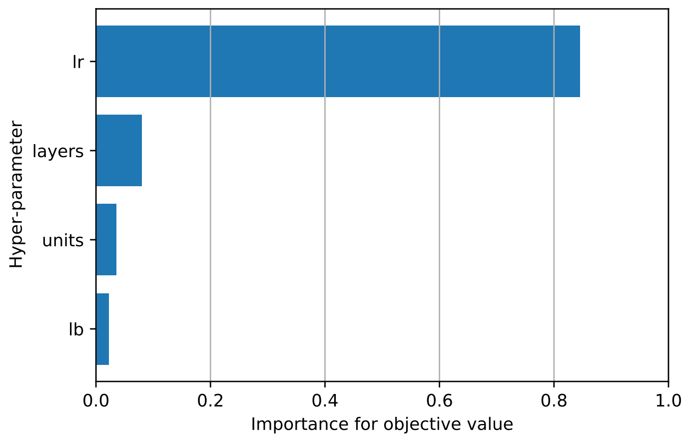


Figure 4. Hyper-parameter importance bars for learning rate (`lr`), number of dense hidden layers (`layers`), number of units in all hidden layers (`units`) and look-back (`lb`).

In particular, we vary the number of fully-connected layers which are placed after the LSTM architecture (`layers`), the number of neurons of these dense hidden layers (`units`) and the learning rate parameter (`lr`). We also explore different values of the look-back parameter (`lb`), the amount of previous data we allow the network to explore in order to predict the future evolution; this value is reported in terms of number of 5 min steps, unless otherwise indicated.

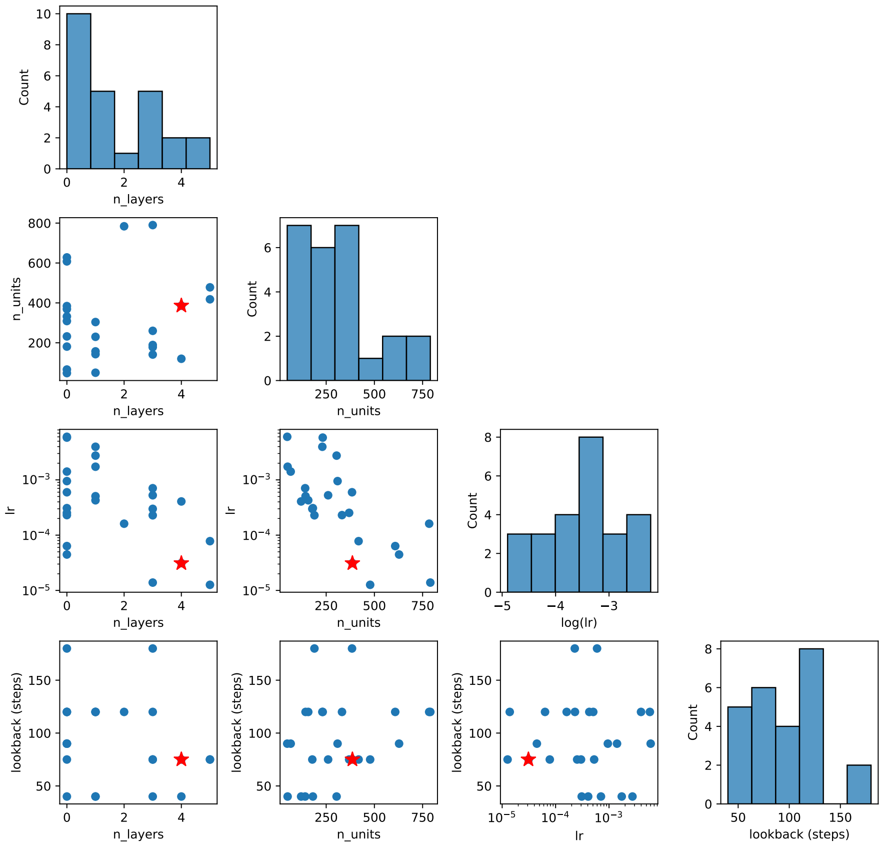


Figure 5. Pair-wise scatter-plots for the hyper-parameters optimised via Optuna for the LSTM architecture. Out of the total of 25 cases, the blue points correspond to hyper-parameter values which cover the minimum value of the MSE that results from using the global optimum values. Red stars indicate the reported optimum values. The histograms along the diagonal, for each hyper-parameter, are the result of marginalising all the points from the rest of hyper-parameters.

The ranges in which each hyper-parameter was optimised are summarised in Table 3. Variations of each of these parameters are not equally important, as shown in Figure 4.

Indeed, we found that the learning rate is key to the learning, whereas variations of the depth and width of the fully-connected layer (`n_layers, n_unit`) are much less important. This indicates that, once the LSTM is learning the time series, the particular characteristics of the additional dense layer are not that relevant. We also found that, in the range we explored, the look-back parameter was not an important handle. This would indicate that we have already chosen an optimal look-back range. Note, however, that if we were interested in describing other, less global, parameters than the SYM-H index, the look-back parameter may change. This optimisation is only valid for the output prediction we have chosen to describe.

The best hyper-parameter values (in terms of mean-square error (MSE)) according to Optuna are: (`n_layers, n_unit, lr, lb`) = (4, 386, 3.12×10^{-5}, 75 steps), which are shown in Table 3. However, while these specific values are indicated as optimal, one should keep in mind that slightly different values could lead to the same performance; there could be "flat directions", i.e. combinations of hyper-parameter values away from the reported optimum which produce equally low MSE. Most importantly the optimisation made using Optuna assumes that the hyper-parameters are uncorrelated; indeed, the hyper-parameters may be correlated to some extent, while the procedure assumes complete independence.

We have explored the impact of these caveats by performing a multidimensional scan of the hyper-parameters instead of assuming total uncorrelation. The results are summarised in Figure 5, where we show the pair-plots between the different hyper-parameters. The points shown in the scatter plots correspond to hyper-parameter values for which, upon repeating the trials five times, within their standard deviation, cover the minimum value of the MSE that results from using the global optimum values specified above. The optimum values are also shown. The histograms along the diagonal, for each hyper-parameter, are the result of marginalising all the points from the rest of hyper-parameters. For these scatter plots we observe no evident correlation between pairs of hyper-parameters, which validates the use of the Optuna procedure.

However, the flat directions are explicitly present in almost all axes. For example if using six hidden layers instead of one, while fixing the rest of hyper-parameters to values different from their "optimum", we get equally good results, statistically speaking. Analogously, this happens with the number of units per hidden layer, which can be as high as 800 (with respect to the reported optimum at 386), or the look-back parameter at 300 (with respect to the optimum at 75). In all cases we observe that each hyper-parameter can admit large excursions in combination of specific values of other hyper-parameters without sacrificing the figure of merit. This is nothing but the consequence of a highly complex parametric dependence of the loss function with respect to the hyper-parameters of the model, as is often the case with the large models used by the community nowadays.

5.2 Prediction of the SYM-H Index

To reproduce the results in Siciliano et al. (2021) where the SYM-H index is predicted from the IMF observations at L1 Lagrange point and from past SYM-H values, as shown in Table 2, the same storms and time intervals as in that work were used, as well as the same training-validation-test split of the storms (see Table 1).

In our case, the neural network architecture consists of an LSTM layer using the hyper-parameters configuration reported in Table 3.

Block bootstrap was performed and 200 bootstrap models were used to obtain estimations of the uncertainties of prediction values, RMSE and the coefficient of determination (R^2).

Table 4 shows the values of RMSE in nT for the target variable SYM-H, and the values of R^2 for the fits of the model to each of the storms in the test sub-dataset. The lower the RMSE, the better a model fits the test sub-dataset. The higher the R^2 value, the better a model fits the test sub-dataset. These values, which are directly comparable with those of Siciliano et al. (2021), are shown in Figure 6. With blue dots, we show the average of our predictions and the blue symmetric segment corresponds to the 95% confidence level (CL) of these predictions. The reported results from Siciliano et al. (2021), Collado-Villaverde et al. (2021) and Iong et al. (2022) are shown with orange dots, red crosses and green stars, respectively, which correspond to their best RMSE results. The same applies to the right panel in Figure 6, this time reporting R^2 values (the values for Iong et al. (2022) are not shown as they are not reported by these authors). Note that uncertainty bars are not shown in the results of other authors since they did not report them.

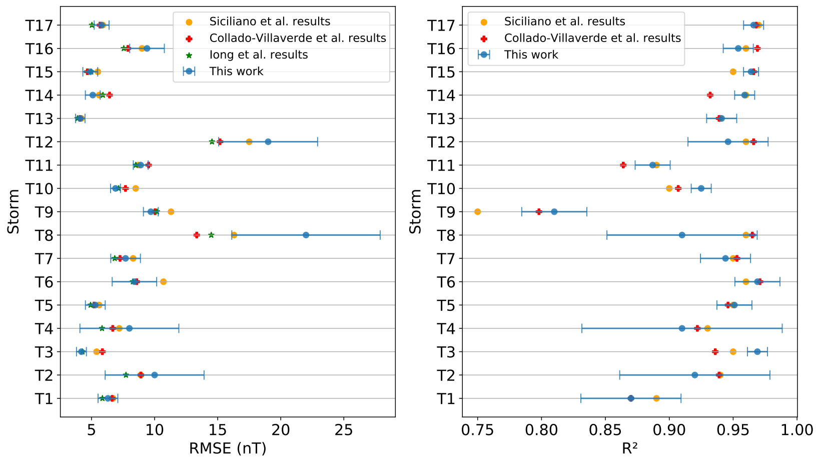


Figure 6. RMSE and R^2 values for the predicted SYM-H index for each one of the 17 test storms. The results of this work are shown in blue with 2σ uncertainty bars (i.e. 95% CL), while those from Siciliano et al. (2021), Collado-Villaverde et al. (2021) and Iong et al. (2022) are shown in orange circles, red crosses and green stars, respectively, which correspond to their best RMSE and R^2 results.

One should then compare the orange dots (which are the best predictions from a bunch of 20 predictions from Siciliano et al. (2021)) with either the lower RMSE or higher R^2 value of the blue range of our predictions. In most cases, our architecture leads to better performance, which we believe is mainly a manifestation of the achieved optimisation of hyper-parameters.

On the other hand, we also include in Figure 6 the comparison with two other more recent studies (see Collado-Villaverde et al. (2021); Iong et al. (2022) commented in section 2), which check the performance of their methods on the same storms as Siciliano et al. (2021). We can observe that those other studies in general improve over Siciliano et al. (2021), while for most of the test storms they still lie inside our RMSE intervals[2]. However, as we commented in section 2, note that contrary to the case of Siciliano et al. (2021), the models considered in Collado-Villaverde et al. (2021); Iong et al. (2022) are different from ours, either by using neural networks with different architectures or a completely different model. It is worth stressing again at this point that our aim in this work was not to build and optimise a robust model to be considered in terms of prediction performance, but to study the prediction uncertainties, while using a popular model which nonetheless, as we see, still gives very competitive results.

[2] For only four out of 12 storms their predictions are marginally better than our predictions, except storm T8, for which they are up to 25% better.

The results in Figure 6 are rather global measures of performance, as they evaluate the goodness of predictions during the whole storm. On the other hand, we may be interested to know how well the algorithm is performing during shorter periods of time, e.g. during the peaks of activity. To illustrate this point, in Figure 7 we show the prediction of the bootstrap models of the target variable SYM-H (in nT) for two of the 17 test storms[3]. In this figure, the orange band represents the 95% CL of the predictions coming from the bootstrap procedure, while the mean prediction is shown by the red dashed lines, and the actual test values are shown as a solid blue line. For each storm we also plot (bottom panels) the residuals, which are computed just subtracting the prediction mean from the observed values and orange bands. We observe in general a very good agreement between the predictions and the observations, where the regions around the peaks show, as expected, the largest deviations. Note how the prediction uncertainties are also larger around the peaks, as one would expect. These larger deviations around the peaks are mainly due to a difference in *timing* of the predictions with respect to the observations. This is indeed a common behaviour for LSTM models (and other models handling sequential data) using a limited training sub-dataset for predicting time-series data with a significant auto-correlation, which can make sometimes difficult for the model to accurately identify the underlying patterns and trends. In our case, for many storms (see Appendix A) we predict the drop in the SYM-H index to happen a bit before it actually happens, which then causes large positive residuals for instants of time before the observations start to drop as well. This is indeed the case at least for storms T1, T6, T7, T8, T11 and T16 featuring residuals around the peak in the range 50–100 nT. On the other hand, for storms T9, T12 and T14, having residuals around the latter (absolute) values, the timing oscillates between predicting in advance or with a small delay.

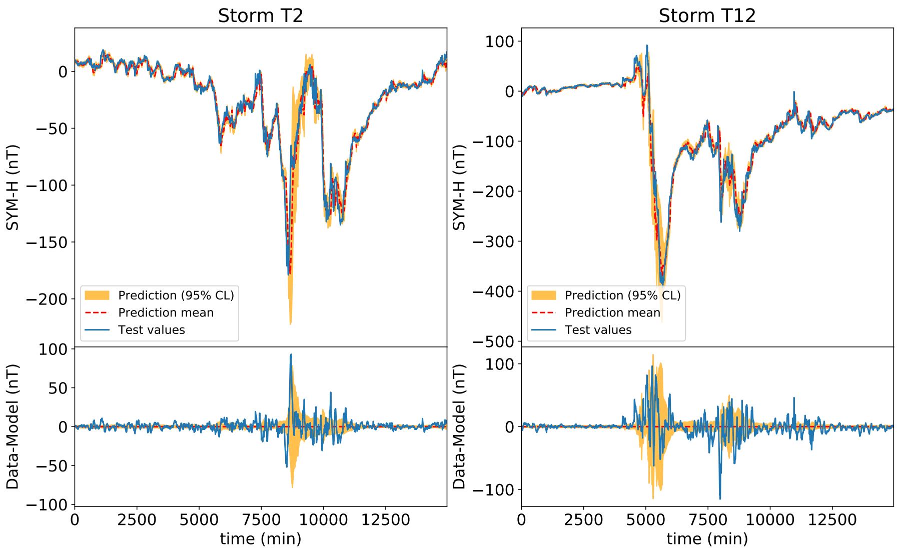


Figure 7. Time-series distributions of two of the 17 storms in the test sub-dataset, in particular, storms T2 and T12, showing in an orange band the 95% CL (corresponding to 2σ), in red dash line the mean for the one hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.

[3] See Figure A1 in Appendix A for all the storms.

Finally, it is important to note that, as commented in section 4.3, the orange bands only represent the epistemic uncertainties (the uncertainties on the expected mean), reason for which there may be observed values lying outside the bands, which may be in part related to the intrinsic data noise, not represented in this figure (because we do not have access to it; see also Appendix A).

5.3 Feature Importance

Neural networks are often considered black-box algorithms though some external inference techniques can be used to extract useful information that can help to understand deep-learning models. Computing feature importance in LSTM models is indeed an important aspect of model interpretation and understanding. Feature importance is a measure of how much a particular input variable (or feature) contributes to the output of the model. Indeed, understanding feature importance can help to identify and select the input features that are most relevant for a given prediction model. It can also provide valuable insights into the underlying patterns, dynamics and relationships present in the considered time-series data. There are several techniques that are commonly used to compute feature importance in LSTM models. Some of these techniques are the "input permutation" (Breiman, 2001; Fisher et al., 2019), "SHapley Additive exPlanations" (Lundberg & Lee, 2017), "Leave-One-Feature-Out", "gradient-based method", "layer-wise relevance propagation" and "activation-based methods" among others.

In this work, the approach used to compute the feature importance in our LSTM model is based on the "input permutation" technique. We repeated the training procedure, using the same optimised hyper-parameters already discussed in section 5.1, but adding disturbances in the input data (i.e. IMF data and past SYM-H values). Thus, for each of the four input features, the values of all of the other features were shuffled, new predictions were calculated using the original test data, and RMSE was calculated. This procedure was performed 15 times for each variable; this is a total of 60 training sessions. The average value of the RMSE for each case is compared to a baseline value calculated with no shuffling (i.e. with the average RMSE value of the RMSE values shown in Table 4). The output of the feature importance results are shown in Figure 8. In the followed method, the most important features are the ones that, when all other variables are shuffled, result in an RMSE closer to the baseline average RMSE value. Thus, from the obtained results, we conclude that past SYM-H values represent the most important feature for our LSTM model, similarly to Siciliano et al. (2021).

It is important to point out that the interpretation of feature importance in LSTM models can be challenging, as these models are inherently complex and exhibit dynamic and non-linear behaviour. Additionally, the results can be influenced by the data preprocessing (e.g. interpolation approach for data gap filling), the choice of input scaling and normalisation as discussed in section 3, as well as the choice of model architecture and the optimisation of the training hyper-parameters as discussed in section 4.1.

6 Discussion and Outlook

In this paper we have explored the use of a deep-learning model to predict the evolution of an activity index during geomagnetic storms, and proposed ways to estimate the uncertainties of these predictions. In particular, we focused on the SYM-H index, a quantity whose variation during a storm is a good summary of its strength. As input parameters, we used IMF data from the ACE spacecraft located at the L1 Lagrange point together with historic SYM-H values.

We chose the SYM-H index to be able to compare with an existing study using deep learning and LSTM architectures in Siciliano et al. (2021). With this comparison, we can

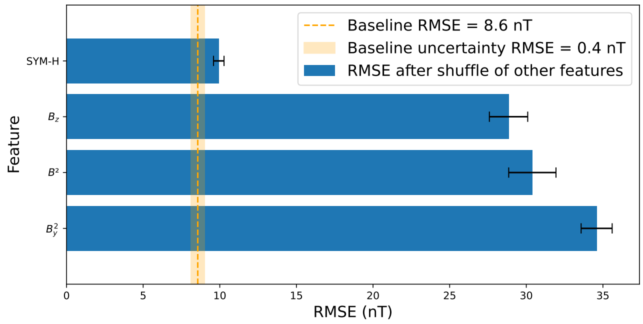


Figure 8. Ranking of the feature importance using an approach based on the "input permutation" technique (the smaller the value, the more important the variable is). Each bar represents the mean value of the RMSE evaluated over all test storms after having shuffled all except the indicated feature variable. The uncertainty bars represent the standard deviation, and the vertical orange line represents the baseline value calculated with no shuffling along with its own standard deviation (that can be computed by averaging values from Table 4).

illustrate the impact of the improvements we propose in both learning optimisation and uncertainty estimation.

We found an overall improvement of the best predictions for the SYM-H index due to hyper-parameter optimisation, as shown in Figure 6, where our lower limit of the RMSE range is lower than the reported best RMSE value in Siciliano et al. (2021), with the exception of test storm T8.

Moreover, we proposed a robust statistical procedure to compute uncertainties in the predictions based on block-bootstrapping. With those uncertainties we produce a prediction with an uncertainty band corresponding to a chosen confidence interval and examine the goodness of our predictions at different times during the storm. See Figure 7 for an illustration of how this uncertainty band evolves with time, and the comparison with the observed values of the SYM-H index.

The strategy described in this work could be applied to other architectures and target parameters, such as the evolution of the geomagnetic or geoelectric fields in the ground.

Reproducing the prediction of the SYM-H global geomagnetic activity index of Siciliano et al. (2021) has served to match the needs of a group of scientists working in SW with the experience of a group working on ML techniques applied to problems related to particle physics. The improvement in prediction performance obtained with this test augurs well for our ultimate goal, which is to be able to predict the variations of the geomagnetic or geoelectric field on the ground at a specific location (Spain). The challenge is important because it involves adding the effect of the field induced by the three-dimensional structure of the electrical resistivity of the lithosphere to the complexity of the sources of these variations. Since we have models for this three-dimensional structure of the resistivity (Torta et al., 2021), we should also be able to predict the variations of the geoelectric field and, by combining them with the models of electrical admittances of our national power grid also described in Torta et al. (2021), derive the expected GICs.

Future work will include ground-level magnetic field forecasting using data from Ebre Observatory, or better, also with those of the other geomagnetic observatories on the Iberian Peninsula. We are also interested in forecasting the time derivative of the geomagnetic field, since this variable is usually the most directly responsible for driving the geoelectric field and, therefore, the GICs. The ultimate goal will be to reformulate the problem in terms of an advanced deep-learning model that provides an alarm system against GICs in Spain. Moreover, our ML architecture can be made more robust and elaborated by including other developments such as a more sophisticated interpolation method to fill data gaps, a cross-validation technique for further improving the model robustness, and adding an attention layer in combination with LSTM.

Data Availability Statement

Raw data are obtained from the NASA's OMNIWeb page (https://omniweb.gsfc.nasa.gov). Processed data, high-resolution plots, and prediction models (for both bootstrap and dropout) in `h5` format can be downloaded at https://zenodo.org/record/7695656 (SpaceWeather-IFIC, 2023).

Table 1. List of the sub-datasets with the most relevant information of the geomagnetic storms: label assigned to the storm, starting date, duration in days and minimum value of the SYM-H index during the geomagnetic storm period. The distribution of the storms among the different sub-datasets follows the same criteria as Siciliano et al. (2021).

Training sub-dataset

Label	Start date	Duration (days)	SYM-H (nT)
TR1	14/02/1998	8	−119*
TR2	02/08/1998	6	−168*
TR3	19/09/1998	10	−213
TR4	16/02/1999	8	−127*
TR5	15/10/1999	10	−218
TR6	09/07/2000	10	−347
TR7	06/08/2000	10	−235*
TR8	15/09/2000	10	−196*
TR9	01/11/2000	14	−174*
TR10	14/03/2001	10	−165*
TR11	06/04/2001	10	−275
TR12	17/10/2001	10	−210
TR13	31/10/2001	10	−320
TR14	17/05/2002	10	−116*
TR15	15/11/2003	10	−490
TR16	20/07/2004	10	−208
TR17	10/05/2005	10	−302*
TR18	09/04/2006	10	−110*
TR19	09/12/2006	10	−211*
TR20	01/03/2012	10	−149

Validation sub−dataset

Label	Start Date	Duration (day)	SYM-H (nT)
V1	28/04/1998	10	−268
V2	19/09/1999	7	−160
V3	25/10/2003	9	−432*
V4	18/06/2015	10	−207*
V5	01/09/2017	10	−146*

Test sub-dataset

Label	Start Date	Duration (day)	SYM-H (nT)
T1	22/06/1998	8	−120
T2	02/11/1998	10	−179*
T3	09/01/1999	9	−111
T4	13/04/1999	6	−122
T5	16/01/2000	10	−101*
T6	02/04/2000	10	−315
T7	19/05/2000	9	−159*
T8	26/03/2001	9	−437
T9	26/05/2003	11	−162*
T10	08/07/2003	10	−125*
T11	18/01/2004	9	−137*
T12	04/11/2004	10	−394*
T13	10/09/2012	25	−138
T14	28/05/2013	7	−134
T15	26/06/2013	8	−110
T16	11/03/2015	10	−234
T17	22/08/2018	12	−205

* Geomagnetic storms with multiple depressions.

Table 2. Variables used in the analysis.

Training variables	B^2	B_y^2	B_z	SYM-H
Forecasted variable				SYM-H

Table 3. Range in which each hyper-parameter was optimised, and chosen value.

Hyper-parameter	Search range	Chosen value
Number of layers	[0, 10]	4
Number of units	[0, 1000]	386
Learning rate	$[10^{-6}, 10^{-1}]$	3.12×10^{-5}
Look-back (steps)	[40, 75, 90 120, 180, 360]	75

Table 4. RMSE and R^2 values for the predicted SYM-H index with their respective standard deviations for each of the storms in the test sub-dataset for our neural network architecture using an LSTM model, and the IMF variables and past SYM-H values as input features for the training.

Set	RMSE (nT)	R^2
T1	6.3 ± 0.4	0.87 ± 0.02
T2	10 ± 2	0.92 ± 0.03
T3	4.2 ± 0.2	0.969 ± 0.004
T4	8.0 ± 2.0	0.91 ± 0.04
T5	5.3 ± 0.4	0.951 ± 0.007
T6	8.4 ± 0.9	0.969 ± 0.090
T7	7.7 ± 0.6	0.944 ± 0.010
T8	22 ± 3	0.91 ± 0.03
T9	9.7 ± 0.3	0.810 ± 0.013
T10	6.9 ± 0.2	0.925 ± 0.004
T11	8.9 ± 0.3	0.887 ± 0.007
T12	19 ± 2	0.946 ± 0.016
T13	4.11 ± 0.19	0.941 ± 0.006
T14	5.1 ± 0.3	0.959 ± 0.004
T15	4.9 ± 0.3	0.964 ± 0.003
T16	9.4 ± 0.7	0.954 ± 0.006
T17	5.8 ± 0.3	0.966 ± 0.004
Total dataset	8.6 ± 0.4	0.929 ± 0.013

Appendix A Dropout method for estimating the prediction uncertainties

In this appendix we discuss in more detail the dropout method as an alternative approach for estimating the prediction uncertainties. We also compare the corresponding results with those obtained from the bootstrap method (see section 4.3).

Roughly speaking, the idea consists in randomly turning off units of the different neural network layers. This has an immediate utility as a regulariser procedure; this is the reason for which dropout is commonly used at the training phase in order to control over-fitting. However, as pointed out in (Gal & Ghahramani, 2016), such a procedure is mathematically equivalent to a variational inference algorithm, with a specific choice of the variational distribution. In particular, if dropout is also used at the test phase, the probability distribution of the predictions would be equivalent to the ones that would be obtained by computing the standard predictive distribution of the Bayesian approach, under the chosen variational approximation.

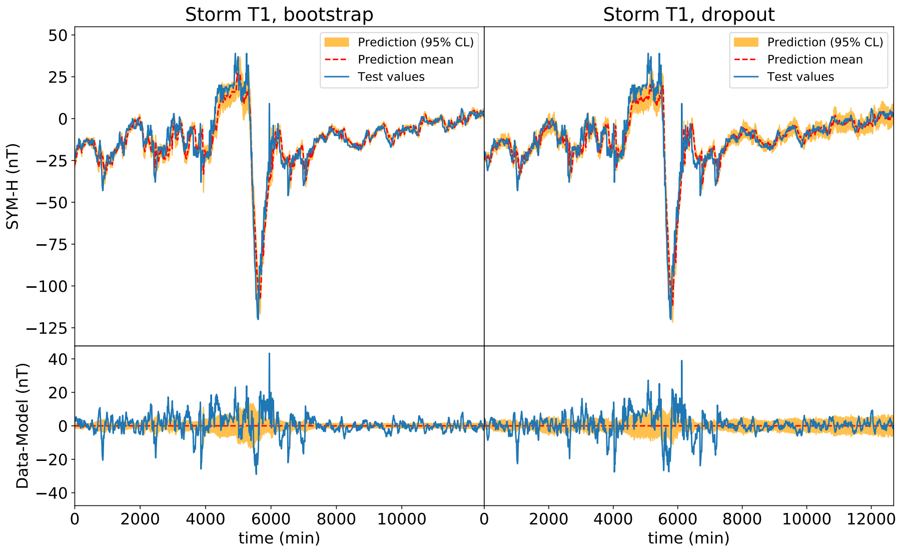
An essential parameter in the dropout implementation is the dropout probability p. Formally, p is the probability for a Bernoulli (binary) random variable to take value equal to 1; so by sampling from the Bernoulli distribution, once for every unit in a hidden layer, such a unit is turned off with a probability of $1-p$. Traditionally, p is considered as an important hyper-parameter to be optimised, e.g. by grid-search, which can be computationally expensive in largely parameterised models. This is the motivation behind “concrete dropout” cited from (Gal et al., 2017), which modifies the traditional dropout algorithm in such a way that p becomes an optimisable parameter during the normal training period. This is done by modifying the loss function so that it has an explicit -and differentiable- dependence on p, which is the result of approximating the Bernoulli distribution by its continuous relaxation using the concrete distribution. In our neural network architecture, we have implemented the concrete dropout method for the dense layers following the LSTM layer, and consequently the associated dropout probability p is automatically optimised during the training process. However, for the LSTM layer itself we stick to the traditional implementation of dropout, where the parameter p is in this case included as an hyper-parameter optimisable with the Optuna procedure. The resulting optimal value for the LSTM dropout probability is $p = 0.0128$.

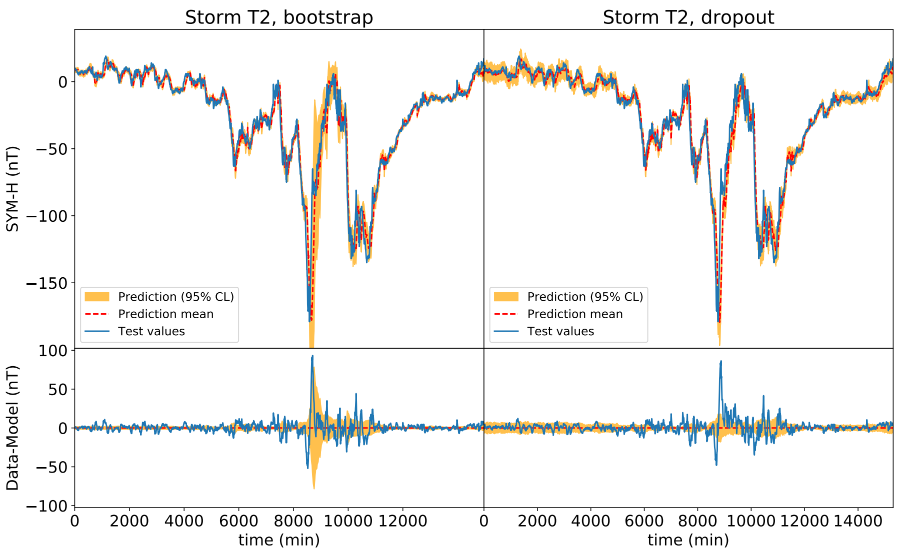
The dropout results are shown in Figure A1 (right panels) for all the 17 different storms of our test sub-dataset, in terms of the prediction with its associated uncertainty of the SYM-H index as a function of time. We compare side by side with the bootstrap results[4] (left panel in the figure).

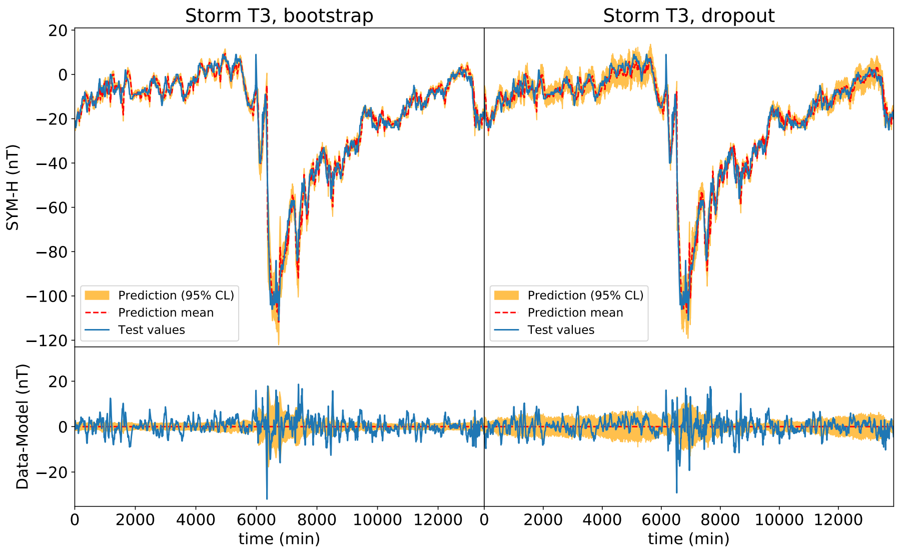
The first thing we note from these results is that both methods give similarly good results, on average, for the mean predictions (red dashed lines in the figures). This can be checked by the bottom panels of each storm, where we represent the residuals “Data - Model”. Some exceptions occur, mainly around the peaks of the storms, where one method is noticeably better than the other (see e.g. storms T6 and T11, where dropout is better). On the other hand, concerning the prediction uncertainties, we see more differences, and it is worth noting that, as commented in section 4.3, what we report here are uncertainties on the expected values (means) of the SYM-H, and not on the variable itself. In other words, these uncertainties are not the total ones resulting from adding the data noise, which we do not have. Coming back to Figure A1, typically the uncertainties on regions away from the peaks are larger (or at most similar) for dropout than for bootstrap. However, the opposite is true when focusing on the regions around the peaks, and in general it is bootstrap the method giving larger (or at most similar) uncertainties than dropout. In Figure A2 we simply zoom-in around the peaks of maximum activity for two

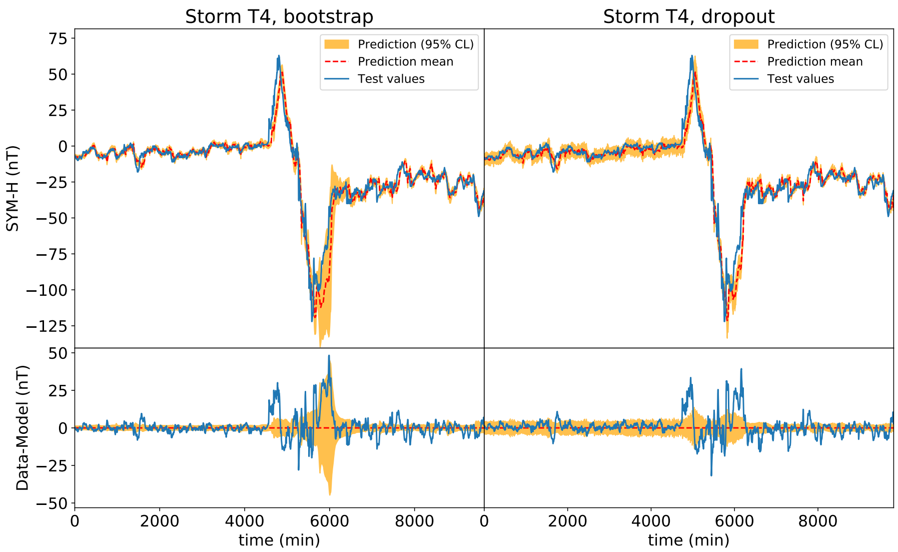
[4] Test storms T2 and T12 are the ones included in Figure 7.

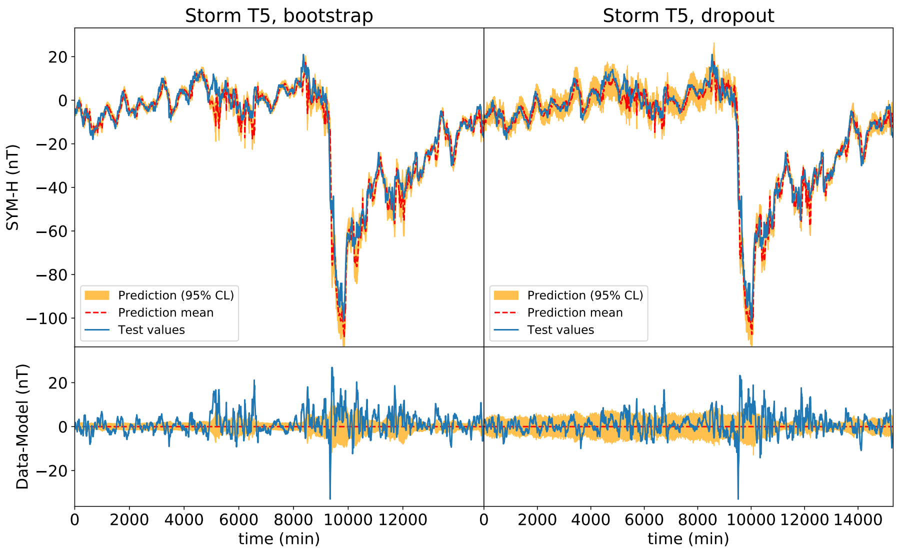
665 particular storms, T7 and T8, where this feature is more evident. Taking into account
666 that the critical period of time of a storm is precisely when the peaks occur, the best pro-
667 cedure is chosen to be the one giving better results in that region of the storms. Here
668 better means not only good predictions, but also conservative prediction uncertainties.
669 For that reason, we have selected bootstrap to be the main procedure for obtaining the
670 predictions in this work.

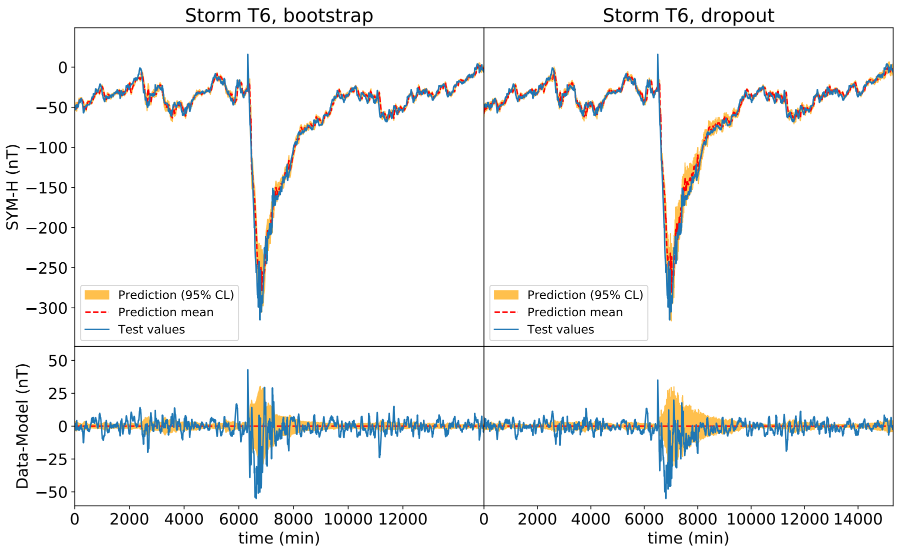


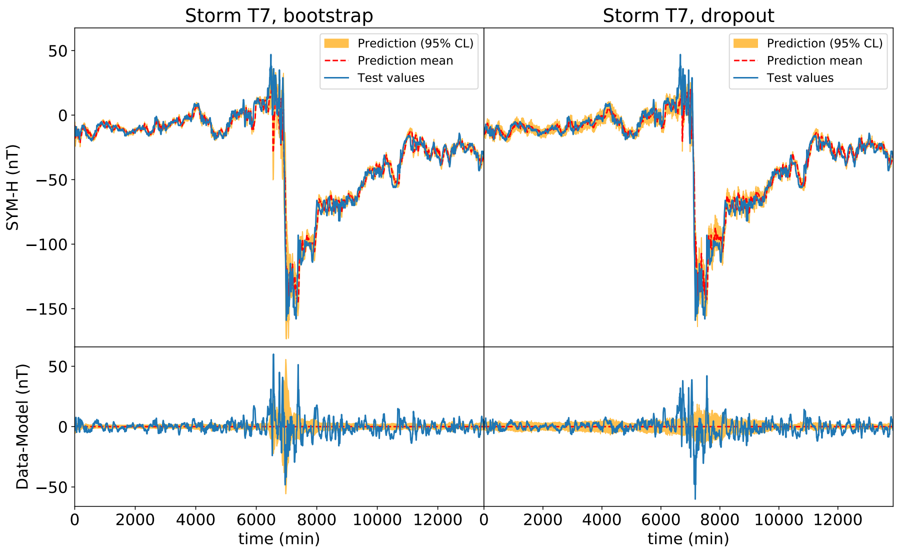



Storm T3, bootstrap
Storm T3, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

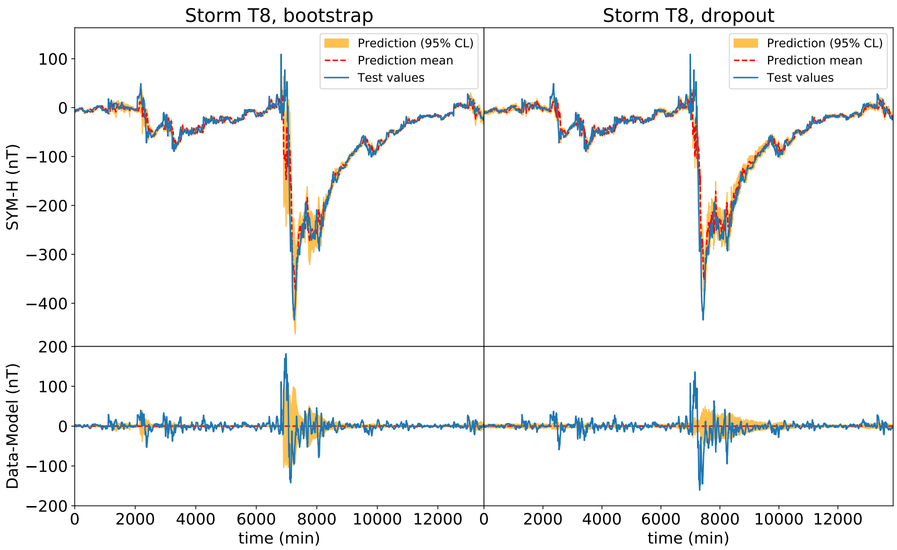

Storm T4, bootstrap
Storm T4, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T5, bootstrap
Storm T5, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

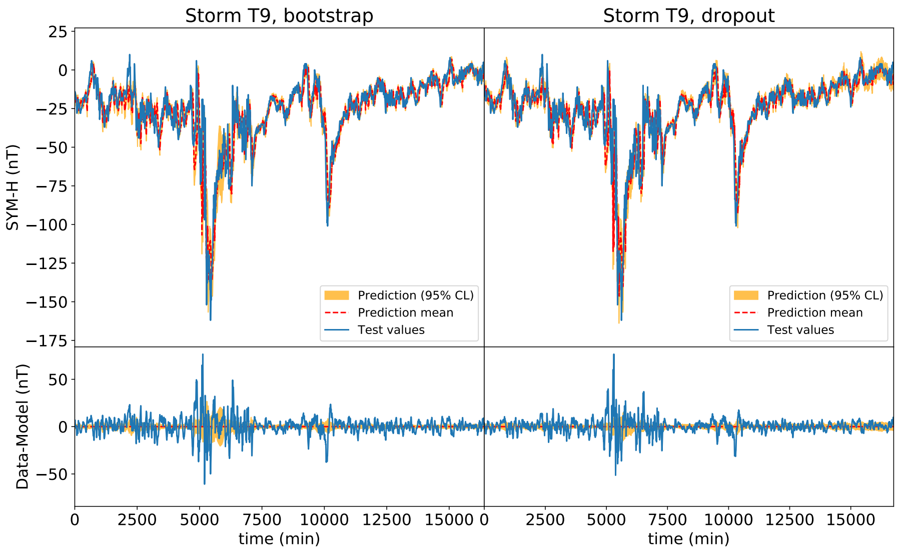

Storm T6, bootstrap
Storm T6, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

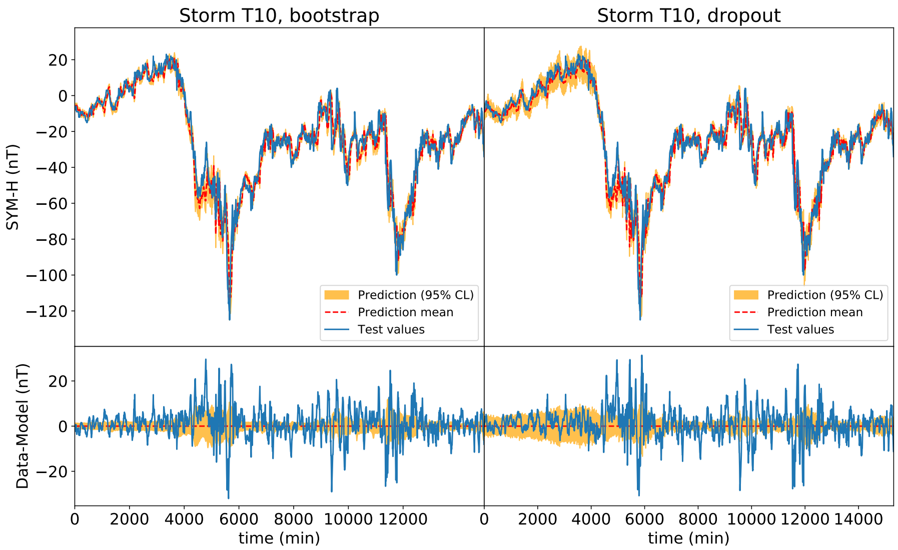


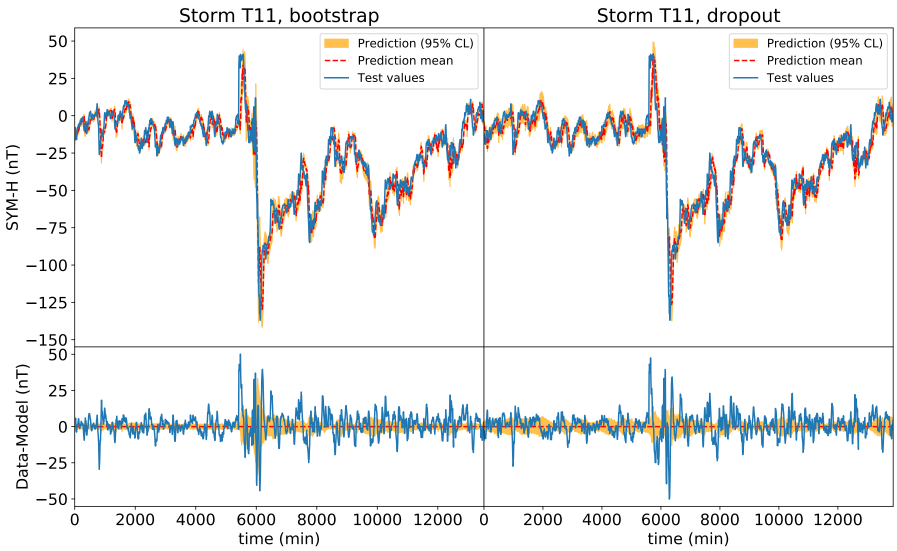
Storm T7, bootstrap
Storm T7, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)

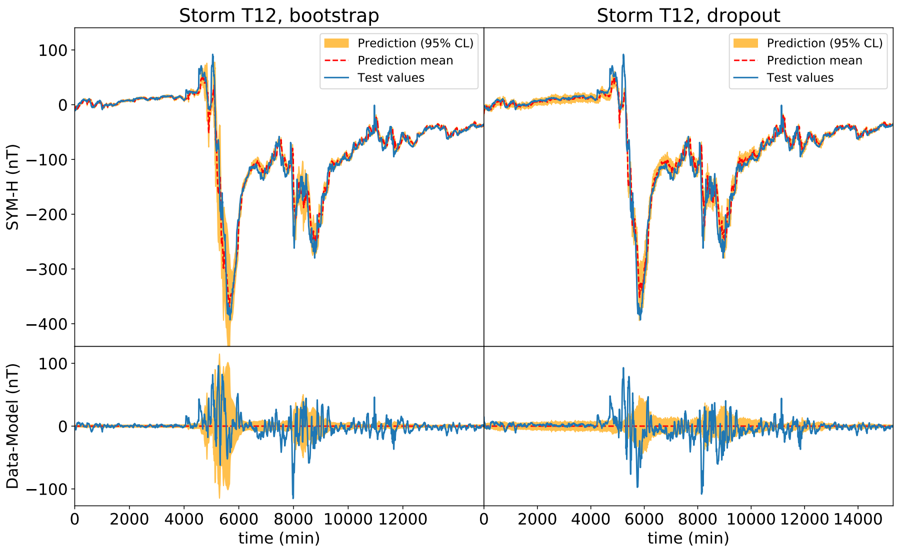


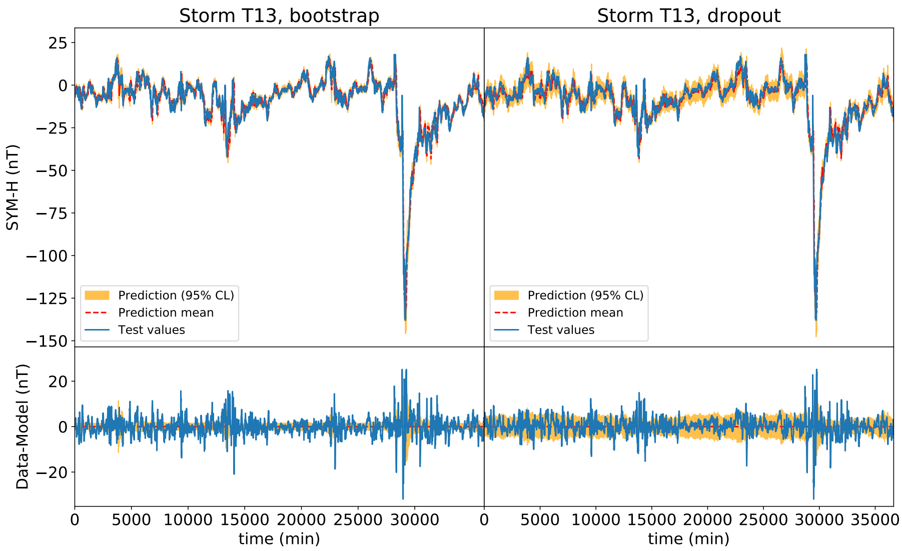
Storm T8, bootstrap
Storm T8, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)

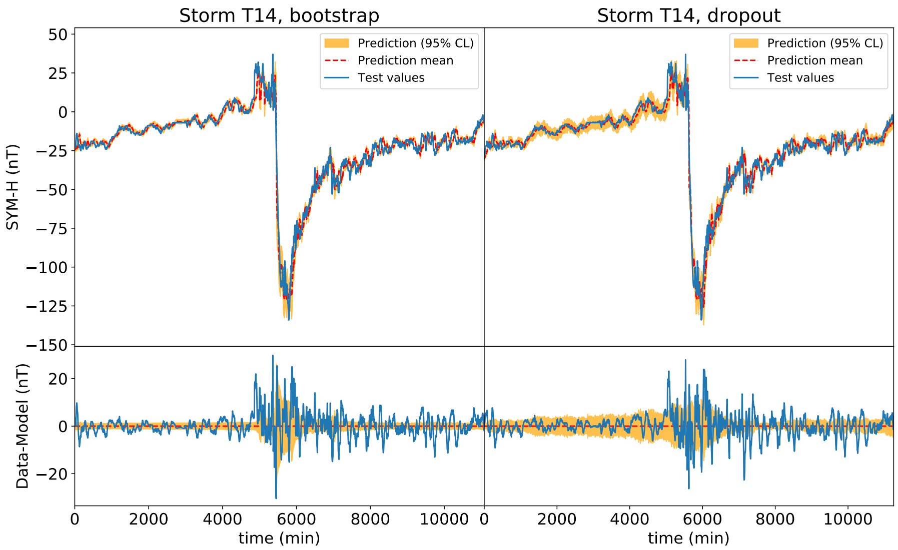

Storm T9, bootstrap
Storm T9, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

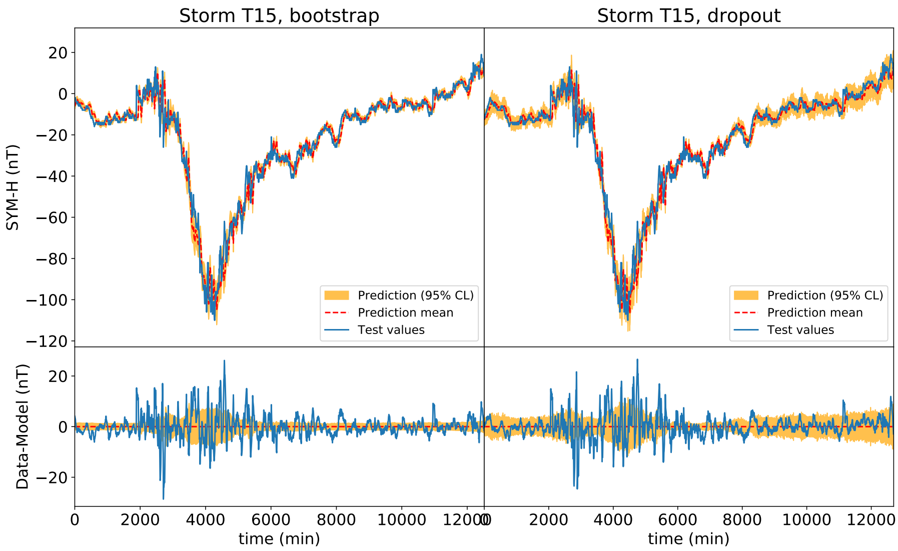

Storm T10, bootstrap
Storm T10, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

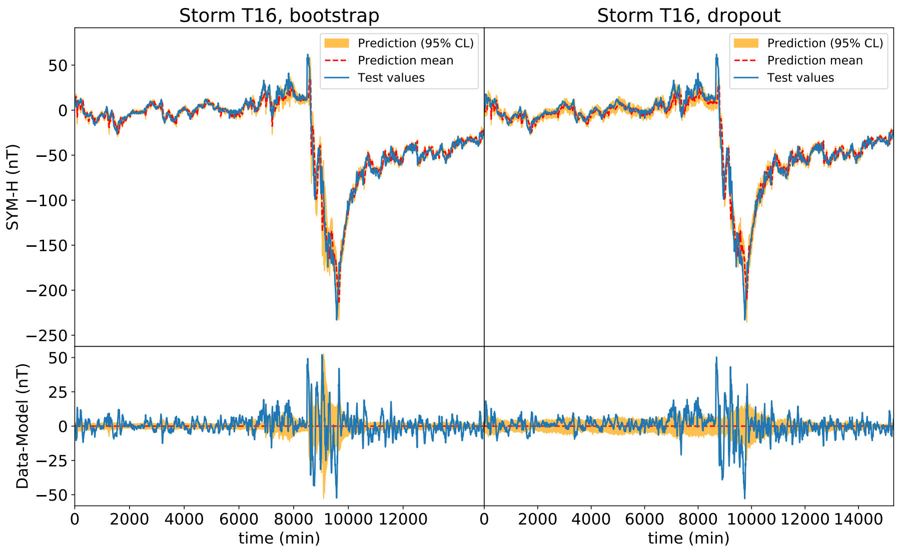

Storm T11, bootstrap
Storm T11, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)


Storm T12, bootstrap
Storm T12, dropout
Prediction (95% CL)
Prediction mean
Test values
SYM-H (nT)
Data-Model (nT)
time (min)


Storm T13, bootstrap
Storm T13, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T14, bootstrap
Storm T14, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T15, bootstrap
Storm T15, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)


Storm T16, bootstrap
Storm T16, dropout
SYM-H (nT)
Data-Model (nT)
Prediction (95% CL)
Prediction mean
Test values
time (min)

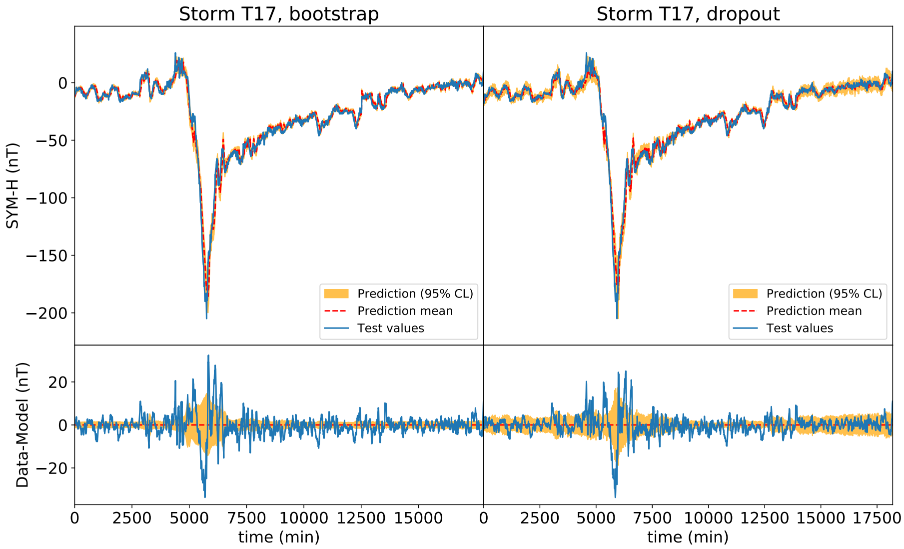


Figure A1. Time-series distributions for all 17 storms in the test sub-dataset, showing the results using the bootstrap method (left) and the dropout method (right). In all distributions, we show in an orange band the 95% CL (corresponding to 2σ), in red dashed line the mean for the one-hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.

Acknowledgments

We acknowledge use of NASA/GSFC's Space Physics Data Facility's OMNIWeb (or CDAWeb or ftp) service, and OMNI data. We also gratefully acknowledge the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana (Spain) as well as the technical support provided by the Instituto de Física Corpuscular (CSIC-UV). We thank our colleagues at the Institut de Recerca Geomodels from the Universitat de Barcelona for their expertise in SW, GIC and geoelectrical modelling, for their guidance in the use of the data and for their constructive comments and advice. Furthermore, the authors are grateful to the Spanish research grants PID2020-113135RB-C32 and PID2020-113135RB-C33 funded by MCIN/AEI/10.13039/501100011033 that supports this work. We also acknowledge the support from Generalitat Valenciana of the PROMETEO (ref. PROMETEO/2021/083) and GenT (ref. CIDEGENT/2020/055) research excellence programmes as well as support from MCIN/AEI of the "Ramon y Cajal" programme (ref. RYC2020-030254-I).

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. doi: https://doi.org/10.48550/arXiv.1907.10902

Bailey, R. L., Leonhardt, R., Möstl, C., Beggan, C., Reiss, M. A., Bhaskar, A., & Weiss, A. J. (2022). Forecasting gics and geoelectric fields from solar wind data using lstms: Application in austria. *Space Weather*, *20*(3), e2021SW002907. doi: https://doi.org/10.1029/2021SW002907

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 24). Curran Associates, Inc.

Bhaskar, & Vichare. (2019). Forecasting of sym-h and asy-h indices for geomagnetic storms of solar cycle 24 including st. patricks day, 2015 storm using narx neural network. *Journal of Space Weather and Space Climate*, *9*(A12). doi: https://doi.org/10.1051/swsc/2019007

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32. doi: 10.1023/A:1010933404324

Burton, R. K., McPherron, R. L., & Russell, C. T. (1975). An empirical relationship between interplanetary conditions and dst. *Journal of Geophysical Research (1896-1977)*, *80*(31), 4204-4214. doi: https://doi.org/10.1029/JA080i031p04204

Cai, L., Ma, S. Y., & Zhou, Y. L. (2010). Prediction of sym-h index during large storms by narx neural network from imf and solar wind data. *Annales Geophysicae*, *28*(2), 381–393. doi: https://doi.org/10.5194/angeo-28-381-2010

Camporeale, E. (2019). The challenge of machine learning in space weather: Nowcasting and forecasting. *Space Weather*, *17*(8), 1166-1207. doi: https://doi.org/10.1029/2018SW002061

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). ACM. doi: https://doi.org/10.1145/2939672.2939785

Collado-Villaverde, A., Muñoz, P., & Cid, C. (2021). Deep neural networks with convolutional and lstm layers for sym-h and asy-h forecasting. *Space Weather*, *19*(6), e2021SW002748. doi: https://doi.org/10.1029/2021SW002748

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81. Retrieved from `http://jmlr.org/papers/v20/18-760.html`

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *Journal of Machine Learning Research Workshop and Conference Proceedings*, *48*. Retrieved from `http://proceedings.mlr.press/v48/gal16.pdf`

Gal, Y., Hron, J., & Kendall, A. (2017). Concrete dropout. doi: https://doi.org/10.48550/arXiv.1705.07832

Gleisner, H., H. Lundstedt, & Wintoft, P. (1996). Predicting geomagnetic storms from solar-wind data using time-delay neural networks. *Ann. Geophys.*, *14*, 679–686. doi: https://doi.org/10.1007/s00585-996-0679-1

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. doi: https://doi.org/10.1162/neco.1997.9.8.1735

Iong, D., Chen, Y., Toth, G., Zou, S., Pulkkinen, T., Ren, J., ... Gombosi, T. (2022). New findings from explainable sym-h forecasting using gradient boosting machines. *Space Weather*, *20*(8), e2021SW002928. doi: https://doi.org/10.1029/2021SW002928

Iyemori, T. (1990). Storm-time magnetospheric currents inferred from mid-latitude geomagnetic field variations. *J. Geomagn. Geoelectr*, *42*(11), 1249–1265. doi: http://dx.doi.org/10.5636/jgg.42.1249.

Kellinsalmi, M., Viljanen, A., Juusola, L., & Käki, S. (2022). The time derivative of the geomagnetic field has a short memory. *Annales Geophysicae*, *40*(4), 545–562. doi: https://doi.org/10.5194/angeo-40-545-2022

King, J., & Papitashvili, N. (2005). Solar wind spatial scales in and comparisons of hourly wind and ace plasma and magnetic field data. *Journal of Geophysical Research: Space Physics*, *110*(A2). doi: http://dx.doi.org/10.1029/2004JA010649

Leontaritis, I. J., & Billings, S. A. (1985). Input-output parametric models for nonlinear systems part ii: stochastic non-linear systems. *International Journal of Control*, *41*(2), 329-344. doi: https://doi.org/10.1080/0020718508961130

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon et al. (Eds.), *Advances in neural information processing systems* (Vol. 30). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf`

Lundstedt, H., & Wintoft, P. (1994). Prediction of geomagnetic storms from solar wind data with the use of a neural network. *Ann. Geophys.*, *12*, 19–24. doi: https://doi.org/10.1007/s00585-994-0019-2

Madsen, F. D., Beggan, C. D., & Whaler, K. A. (2022). Forecasting changes of the magnetic field in the united kingdom from l1 lagrange solar wind measurements. *Frontiers in Physics*, *10*. doi: https://doi.org/10.3389/fphy.2022.1017781

Marsal, S., & Curto, J. (2009). A new approach to the hourly mean computation problem when dealing with missing data. *Earth, Planets, and Space*, *61*, 945-956. doi: https://doi.org/10.1186/BF03352945

Mayaud, P. N. (1980). Introduction. In *Derivation, meaning, and use of geomagnetic indices* (p. 1-2). American Geophysical Union (AGU). doi: https://doi.org/10.1002/9781118663837.ch1

Papitashvili, N. E., & King, J. H. (2023a). *Omni 1-min data [data set]. nasa space physics data facility.* https://doi.org/10.48322/45bb-8792. (Last accessed on March 3, 2023)

Papitashvili, N. E., & King, J. H. (2023b). *Omni 5-min data [data set]. nasa space physics data facility.* https://doi.org/10.48322/gbpg-5r77. (Last accessed on March 3, 2023)

Patowary, R., Singh, S., & Bhuyan, K. (2013). A study of seasonal variation of geomagnetic activity. *Research Journal of Physical and Applied Sciences*, *2*, 1-11.

Pinto, V. A., Keesee, A. M., Coughlan, M., Mukundan, R., Johnson, J. W., Ngwira, C. M., & Connor, H. K. (2022). Revisiting the ground magnetic field perturbations challenge: A machine learning perspective. *Frontiers in Astronomy and Space Sciences*, *9*. doi: https://doi.org/10.3389/fspas.2022.869740

Qin, Z., Denton, R. E., Tsyganenko, N. A., & Wolf, S. (2007). Solar wind parameters for magnetospheric magnetic field modeling. *Space Weather*, *5*(11). doi: https://doi.org/10.1029/2006SW000296

Rumelhart, D. E., & McClelland, J. L. (1987). Learning internal representations by error propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations* (p. 318-362).

Siciliano, F., Consolini, G., Tozzi, R., Gentili, M., Giannattasio, F., & De Michelis, P. (2021). Forecasting sym-h index: A comparison between long short-term memory and convolutional neural networks. *Space Weather*, *19*(2), e2020SW002589. doi: https://doi.org/10.1029/2020SW002589

SpaceWeather-IFIC. (2023). Spaceweather-ific/open_data: v1.0. doi: https://doi.org/10.5281/zenodo.7695656

Torta, J. M., Marcuello, A., Campanyà, J., Marsal, S., Queralt, P., & Ledo, J. (2017). Improving the modeling of geomagnetically induced currents in spain. *Space Weather*, *15*(5), 691-703. doi: https://doi.org/10.1002/2017SW001628

Torta, J. M., Marsal, S., Ledo, J., Queralt, P., Canillas-Pérez, V., Piña-Varas, P., ... Martí, A. (2021). New detailed modeling of gics in the spanish power transmission grid. *Space Weather*, *19*(9), e2021SW002805. doi: https://doi.org/10.1029/2021SW002805

Wanliss, J. (2005). Fractal properties of sym-h during quiet and active times. *J. Geophys. Res*, *110*. doi: https://doi.org/10.1029/2004JA010544

Wanliss, J., & Uritsky, V. (2010). Understanding bursty behavior in midlatitude geomagnetic activity. *Journal of Geophysical Research: Space Physics*, *115*(A3). doi: https://doi.org/10.1029/2009JA014642

Zhang, W. (1988). Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of annual conference of the japan society of applied physics.*

Zhang, W., Itoh, K., Tanida, J., & Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Appl. Opt.*, *29*(32), 4790–4797. doi: https://doi.org/10.1364/AO.29.004790

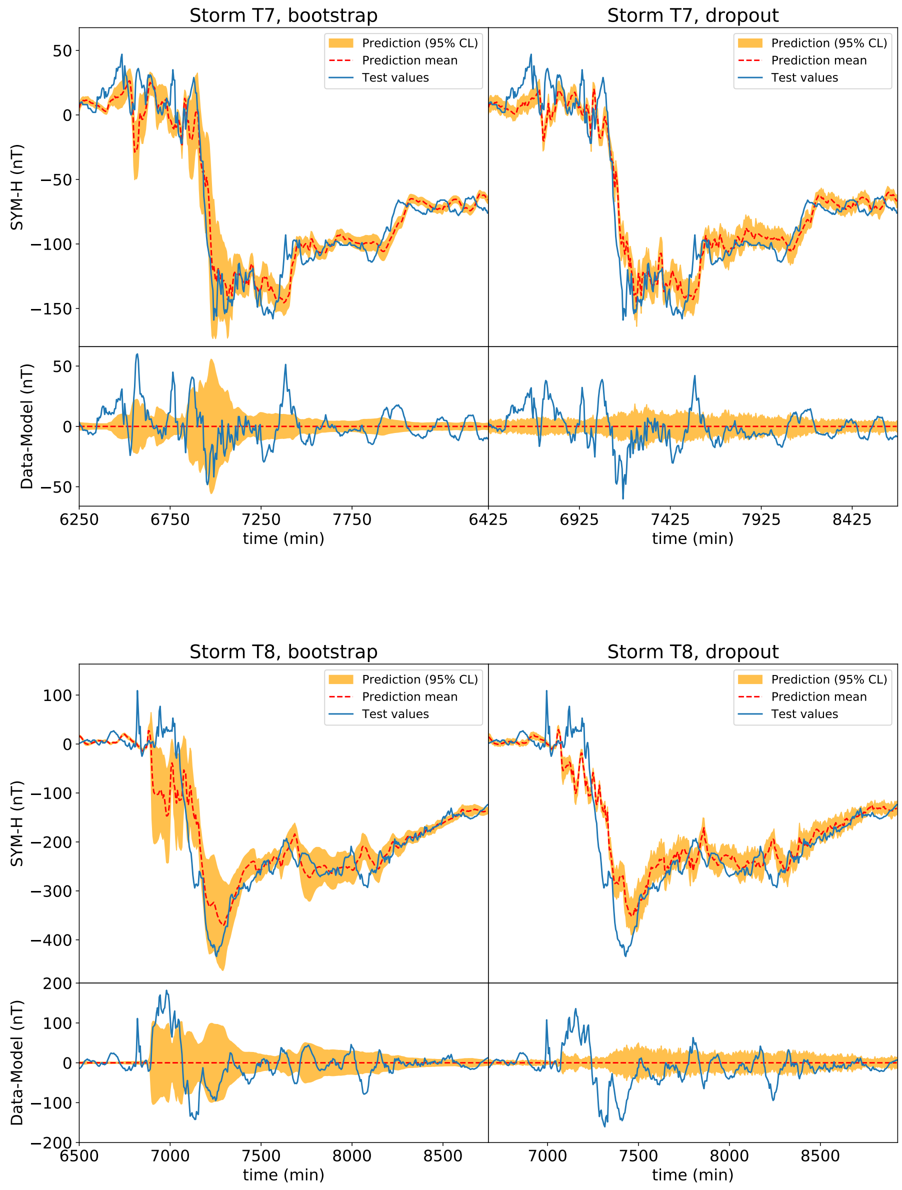


Figure A2. Zoom-in of the time-series distributions around the peaks of maximum activity for storms T7 and T8 in the test sub-dataset, showing the results using the bootstrap method (left) and the dropout method (right). In these distributions, we show in an orange band the 95% CL (corresponding to 2σ), in red dashed line the mean for the one-hour ahead predictions of the SYM-H index from the LSTM model, and the test data as a solid blue line. The lower panels represent the residuals with respect to the model prediction mean.