Space Weather

RESEARCH ARTICLE
10.1029/2021SW002748

Key Points:

- Deep Neural Networks using convolutional and Long Short-Term Memory layers achieve great accuracy for SYM/ASY-H indices forecasting up to 2 h
- The network only uses the magnetic field data measured by ACE and the own index as inputs
- The ASY-H index is significantly harder to forecast than the SYM-H index

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

A. Collado-Villaverde,
armando.collado@uah.es

Citation:

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*, e2021SW002748. https://doi.org/10.1029/2021SW002748

Received 23 FEB 2021
Accepted 27 APR 2021

Deep Neural Networks With Convolutional and LSTM Layers for SYM-H and ASY-H Forecasting

Armando Collado-Villaverde[1], Pablo Muñoz[1], and Consuelo Cid[2]

[1]Departamento de Automática, Escuela Politécnica Superior, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain,
[2]Departamento de Física y Matemáticas, Universidad de Alcalá, Alcalá de Henares, Madrid, Spain

Abstract Geomagnetic indices quantify the disturbance caused by the solar activity on a planetary scale or in particular regions of the Earth. Among them, the SYM-H and ASY-H indices represent the (longitudinally) symmetric and asymmetric geomagnetic disturbance of the horizontal component of the magnetic field at midlatitude with a 1-min resolution. Their resolution, along with their relation to the solar wind parameters, makes the forecasting of the geomagnetic indices a problem that can be addressed through the use of Deep Learning, particularly using Deep Neural Networks (DNNs). In this work, we present two DNNs developed to forecast respectively the SYM-H and ASY-H indices. Both networks have been trained using the Interplanetary Magnetic Field (IMF) and the related index for the solar storms occurred in the last two solar cycles. As a result, the networks are able to accurately forecast the indices 2 h in advance, considering the IMF and indices values for the previous 200 min. The evaluation of both networks reveals a great forecasting precision, including good predictions for large storms that occurred during the solar cycle 23 and comparing with the persistence model for the period 2013–2020.

Plain Language Summary Machine Learning applications are conquering new areas taking advantage of the availability of large amounts of data. Considering space probes such as ACE that are providing information about the solar wind at the L1 point for more than 25 years, it is feasible that Machine Learning approaches can be applied to the space weather context. In this work, we present a SYM-H and ASY-H indices (that quantify the disturbance of the Earth's magnetosphere) forecasting system for the next 1 and 2 h. To do so, we apply the state of the art in Artificial Neural Networks. The goal of the work is to drive an operational geomagnetic indices forecasting system that can minimize the damages caused by geomagnetic storms, providing an alert ahead enough to take containment measures.

1. Introduction

Geomagnetic indices quantify the disturbance of the terrestrial magnetosphere due to interplanetary transients. Among all the existing indices (see review by Menvielle et al., 2011), the disturbance storm time, Dst, index (Mayaud, 1968; Sugiura, 1964), is commonly used to describe the geomagnetic storm strength. Taking advantage of this, the Dst index is used to drive operational forecasts of various geospace systems, including the thermosphere for mass density and satellite drag (e.g., Licata et al., 2020, and references therein) or the ionosphere for storm propagation (e.g., Hu et al., 1998). Sometimes, SYM-H index (Iyemori, 1990; Iyemori et al., 1992) is used over the Dst index because of its advantage of having higher time resolution (1-min) data as compared to hourly Dst index. Indeed, both Dst and SYM-H are proxies of the symmetric ring current, while the longitudinally asymmetric part of geomagnetic disturbance field at low latitude to midlatitude is quantified by ASY indices (Iyemori, 1990; Iyemori et al., 1992). For example, SYM-H index is used by Forsyth et al. (2020) to construct forecasts of the >2 MeV flux or by Tshisaphungo et al. (2018) to model the ionospheric F2 layer (foF2) changes during geomagnetic storms.

The official values of several indices, including Dst and SYM-H, are obtained from ground magnetometers and provided through the WDC Kyoto (http://wdc.kugi.kyoto-u.ac.jp/), but provisional Dst is available with 1-h latency and SYM-H is not available in real time. This delay in the nowcasting of the geomagnetic indices makes their forecasting even more necessary to provide the proper input to drive other space weather parameters in an operational situation. For this purpose, real-time solar wind parameters measured at L1 point are used to drive forecast of the indices. The forecasting methods based on physical or empirical relationships between the solar wind and the indices start with the works by Burton et al. (1975) and O'Brien and McPherron (2000), which achieve a reasonable level of accuracy. Rastätter et al. (2013) compares the performance of 30 models forecasting a 1-min Dst index (equivalent to SYM-H) during four events: two events representing highly disturbed times and the other two events represent quieter times. None of the models consistently performed best for all the events.

In this scenario, where accurate forecasting of geomagnetic indices is needed for operations, using Machine Learning (ML) techniques is a current trend due to the promising results that can be found in the literature. In this way, Camporeale (2019) presents a survey in the current state of the art of ML applied to space weather nowcasting and forecasting. This work provides an insight into the most important contributions that have been made in recent years, as well as the most common ML metrics that could be applicable to the field.

Regarding to the metrics, it is also important to highlight the work by Liemohn et al. (2018), where guidelines for geomagnetic index forecasting are provided, establishing which metrics are the most appropriate for each index. They recommend it in two processes: metrics focused on the accuracy of the prediction (Root Mean Squared Error [RMSE] and MAE) and metrics that assess the relationship between the predicted and the real value (Pearson correlation coefficient), that is, how close is the prediction to the original values, and metrics for assessing the event detection performance (HSS, POD, and FAR), which indicates how many events would have been detected if we only relied on our prediction. In a similar direction, Welling et al. (2018) made several recommendations to validate ground magnetic perturbation forecasting systems based on ML.

Focusing on geomagnetic indices prediction, one of the first ML approaches is proposed by Hernandez et al. (1993) for the AL index forecasting, which uses feed-forward networks and Autoregressive Moving Average Models with nonlinear filters. Novel works attempt to forecast the Kp index, such as Zhelavskaya et al. (2019) and Shprits et al. (2019), where different methods were evaluated, achieving the highest performance using a feed-forward neural network with one hidden layer, scoring a RMSE of 0.55 for the nowcasting and 0.7 for the 3-h horizon.

One of the most studied indices is the Dst. For this index, several mathematical and ML forecasting systems have been developed. We can highlight the work by Gruet et al. (2018) that combines the usage of Long Short-Term Memory (LSTM) layers in a Recurrent Neural Network (RNN) with a Gaussian process to provide a forecast up to 6 h ahead of the Dst. This model used hourly data from the OMNIWeb and Global Positioning System databases. Their model obtained great accuracy, with a correlation coefficient higher than 0.873 and an RMSE lower than 9.86. However, despite the overall performance, it is unable to obtain accurate predictions of intense storms where the Dst reaches values lower than −250 nT.

Lazzús et al. (2019) explored and compared several ML techniques for the Dst index forecast problem. In his work, several Artificial Neural Networks (ANNs) are studied, as well as its combination with bio-inspired algorithms, such as particle swarm optimization, genetic algorithms, and a hybridization of both, to improve the system's accuracy. In order to evaluate the models' performance, the RMSE and the correlation coefficient metrics were used, which are included in the previously mentioned recommendations. Their model achieved a RMSE lower than 5 nT when forecasting up to 3 h in advance and a RMSE lower than 7 when forecasting up to 6 h ahead.

ML algorithms can take advantage of high-resolution data. In this sense, the hourly nature of the Dst can lead to a loss of relevant information. As a consequence, the latest research is focused on forecasting the SYM-H and ASY-H indices, which have 1-min resolution, so the amount of information that may be lost due to the averaging is almost negligible. For the forecast of these indices, we can highlight the work of Bhaskar and Vichare (2019) where a Nonlinear Autoregressive Network with exogenous input solution provides the SYM-H and ASY-H prediction. Their model considers an input history of 30 min and an output feedback of 120 min. As for the inputs, they consider velocity, density, and Interplanetary Magnetic Field (IMF). For the training data, SYM-H and ASY-H indices during geomagnetic storms from 1998 to 2013 are used. However, only storms when the SYM-H reaches values below −85 nT are used as the target for training the networks. Predictions for both indices during nine geomagnetic storms of the 24th solar cycle are showcased, including the large storm that occurred on St. Patrick's Day in 2015, presenting the model capabilities for predicting these indices about an hour before they happen.

Table 1
Storms Used to Train the DNN Models

Train subset					
Storm no.	Start date	End date	MP	Min SYM-H (nT)	Max ASY-H (nT)
1	14/02/1998	22/02/1998	Y	−119	188
2	02/08/1998	08/08/1998	Y	−168	136
3	19/09/1998	29/09/1998	N	−213	395
4	16/02/1999	24/02/1999	Y	−127	246
5	15/10/1999	25/10/1999	N	−218	214
6	09/07/2000	19/07/2000	N	−335	380
7	06/08/2000	16/08/2000	Y	−235	206
8	15/09/2000	25/09/2000	Y	−196	219
9	01/11/2000	15/11/2000	Y	−174	130
10	14/03/2001	24/03/2001	Y	−165	225
11	06/04/2001	16/04/2001	N	−275	422
12	17/10/2001	22/10/2001	N	−210	154
13	31/10/2001	10/11/2001	N	−313	329
14	17/05/2002	27/05/2002	Y	−113	204
15	15/11/2003	25/11/2003	N	−488	374
16	20/07/2004	30/07/2004	Y	−208	294
17	10/05/2005	20/05/2005	N	−302	250
18	09/04/2006	19/04/2006	N	−110	162
19	09/10/2006	19/12/2006	N	−206	267
20	01/03/2012	11/03/2012	Y	−149	229

Note. From left to right: number used to identify the storm, start and end days, occurrence (Y) or not (N) of a multidip (MP) storm, SYM-H index minimum value, and ASY-H maximum value. For training and validating the ASY-H, the storm 18 from the training set is swapped for the storm 23 in the validation set, to better distribute the instances of each set according to this particular index.

Table 2
Storms Used to Validate the DNN Models

Validation subset					
Storm no.	Start date	End date	MP	Min SYM-H (nT)	Max ASY-H (nT)
21	28/04/1998	08/05/1998	N	−268	415
22	19/09/1999	26/09/1999	N	−160	157
23	25/10/2003	03/11/2003	Y	−427	828
24	18/06/2015	28/06/2015	Y	−207	348
25	01/09/2017	11/09/2017	Y	−144	230

Note. For training and validating the ASY-H, the storm 18 from the training set is swapped for the storm 23 in the validation set, to better distribute the instances of each set according to this particular index.

Deep Learning (DL) and ANNs have been used as well to forecast the onset of magnetic storms, as presented in the work of Maimaiti et al. (2019). They use a time history of solar wind speed, proton density, and IMF as inputs to forecast the occurrence probability of the onset of magnetic substorms over the next hour. The model, trained using a data set from the SuperMAG list of magnetic substorm onsets, is able to identify substorms 75% of the time, which is a significant improvement over the previous state of the art algorithms for this particular problem that only identified around 21% of the substorms in the same data set.

Siciliano et al. (2020) made a comparison of two ANNs for the SYM-H index forecasting: one based on LSTM layers and another one based on Convolutional Neural Network (CNN) layers. With the objective of developing operational models, they only use the IMF measured by ACE's magnetometer (they discard variables such as speed or density due to the data gaps occurred as a consequence of the instruments saturation during intense storms) to forecast storms in which the SYM-H index achieved values lower than −100 nT. They also compare the performance of the networks both with and without the index itself as an input parameter.

This paper proposes a Deep Neural Network (DNN) based on LSTM and single-dimension convolutional layers to forecast the SYM-H and ASY-H indices using the IMF measured by ACE and the indices themselves. Meanwhile the use of LSTM and CNN layers has been explored in the literature, the combination of both is a novel approach in this field that increases the forecasting accuracy.

The paper is structured as follows. Next section describes the database and variables used to train our neural networks. Section 3 presents the network architecture developed to predict the geomagnetic indices. Section 4 showcases the prediction accuracy of our networks, comparing it with previous works. Finally, some conclusions are outlined.

2. Database

For training, validating, and testing the DNN presented in the next section, we use the 42 geomagnetic storms that occurred between 1998 and 2018 in which the SYM-H index achieved a minimum value of less than −100 nT. For most storms, the time interval consists of 10 days surrounding the storm minus a few exceptions, as shown in Tables 1–3. We retrieve the IMF parameters and the SYM-H and ASY-H indices from https://cdaweb.gsfc.nasa.gov/index.html (King, 2005). The indices are retrieved in 5-min averages (OMNI_HRO_5MIN), whereas the IMF parameters are retrieved from the level 2 ACE data (AC_H3_MFI) sampled at 1-s intervals, recorded at L1. The selected parameters of the IMF are its magnitude and its three Geocentric Solar Magnetospheric (GSM) components.

Since the time resolutions of the indices and the IMF parameters are different, the IMF data have been grouped using the same resolution as the indices. When grouping the IMF data, the mean and standard deviation of each variable were computed. The aim of computing the standard deviation is to obtain information about the fluctuation of the IMF during the period, which would be lost if only the average is used.

The choice of limiting the input variables to the IMF data is motivated by the limited data availability of the solar wind plasma parameters

Table 3
Storms Used to Test the DNN Models

Test subset					
Storm no.	Start date	End date	MP	Min SYM-H (nT)	Max ASY-H (nT)
26	22/06/1998	30/06/1998	N	−120	127
27	02/11/1998	12/11/1998	Y	−179	196
28	09/01/1999	18/01/1999	N	−111	147
29	13/04/1999	19/04/1999	N	−122	138
30	16/01/2000	26/01/2000	Y	−101	99
31	02/04/2000	12/04/2000	N	−315	612
32	19/05/2000	28/05/2000	Y	−159	225
33	26/03/2001	04/04/2001	N	−434	352
34	26/05/2003	06/06/2003	Y	−162	377
35	08/07/2003	18/07/2003	Y	−125	172
36	18/01/2004	27/01/2004	Y	−137	150
37	04/11/2004	14/11/2004	Y	−393	339
38	10/09/2012	05/10/2012	N	−138	130
39	28/05/2013	04/06/2013	N	−134	154
40	26/06/2013	04/07/2013	N	−110	165
41	11/03/2015	21/03/2015	N	−233	250
42	22/08/2018	03/09/2018	N	−205	197

for effective implementation of any ML algorithm. Figure 1 in Larrodera and Cid (2020) shows data coverage of the main solar wind parameters measured by the ACE spacecraft from the time it is operational until the end of the year 2017. Data availability for IMF and solar wind velocity is over 98% for any year in the sample (even if some data gaps appear in solar wind velocity due to saturation of SWEPAM instrument during important storm events), but the coverage drops substantially for solar wind temperature or density. Indeed, the proton density and the temperature became increasingly sparse starting in 2010 as the detectors aged. An operational improvement has significantly increased the frequency of good-quality SWEPAM observations since October 23, 2012, but in this scenario, we choose to consider the solar wind parameter with almost full data availability. In fact, the IMF gaps are limited to a few seconds in most cases, which have been filled using linear interpolation. Additionally, by averaging into 5-min windows, the impact of the absence of a few seconds is greatly diminished.

Regarding to the indices, they serve a double purpose: they will be used both as the input of the network and as the output. Thus, the input of the network will be a discrete-time window of the indices in conjunction with the other variables. The output will be the predicted values for the SYM-H or ASY-H indices for 1 and 2 h in advance. Let us introduce a key point in our approach: each index is predicted by a specific DNN, that is, we train two separate neural networks using the IMF variables and one of the indices, the SYM-H or ASY-H separately.

Therefore, each network will receive nine input variables:

- 5-min average and standard deviation of the IMF strength, expressed in nT.
- 5-min average and standard deviation of the X, Y, and Z GSM components of the IMF, expressed in nT.
- SYM-H index, expressed in nT.
- ASY-H index, expressed in nT.

The split of the storms into the training, validation, and testing sets has been done following the approach proposed by Siciliano et al. (2020). The training set consists of 20 storms (48% of the whole 42 storms, 60,808 samples, Table 1), 5 storms have been saved for validation (12%, 14,688 samples, Table 2), and the remaining 17 storms (40%, 55,072 samples, Table 3) have been used for testing. The split has been done so each set is uniformly populated regarding the intensity of the storms. However, despite that this split is suitable for the SYM-H index, it is not for the ASY-H one, since the two most intense storms for this index would be unseen by the network during training. Trying to improve the split for the ASY-H index, we have swapped the storm number 18 of the training set with the storm number 23 of the validation set, only for the ASY-H index training, validation, and testing, so the SYM-H split remains unchanged.

Additionally, to ease the convergence of the network and improve the forecasting, each variable in the input data has been standardized: (i) subtracting the mean (centering the data on −1) and (ii) dividing it by the standard deviation (scaling the data so it has unit variance). This is shown in Equation 1, where Z is the standardized data, x are the original values, μ is the mean of that variable, and σ its standard deviation. Standardization is preferred in the DL domain since it makes each feature similar to a normal distribution. Additionally, this scaling method is more resilient to outliers than the more common normalization between [-1, 1] and [0, 1], which only considers the minimum and maximum values instead of the overall statistics of the data.

$$Z = \frac{(x - \mu_x)}{\sigma_x} \tag{1}$$

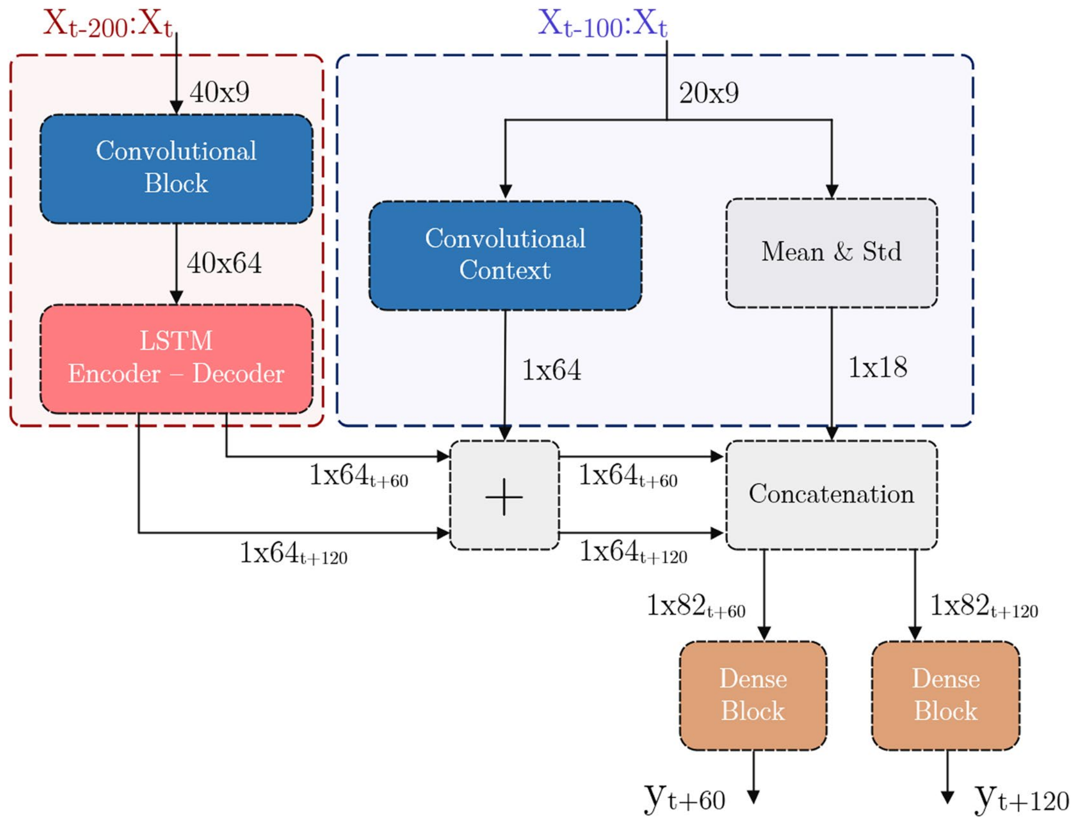

Figure 1. ANN architecture for SYM-H or ASY-H index prediction. Input and output shapes are expressed in t ± minutes. Shapes between layers are expressed in time steps × features. ANN, Artificial Neural Network.

3. Neural Network Architecture

The prediction of the SYM/ASY-H indices can be seen as a time-series forecasting problem due to the temporal nature of the data. It is well known that the prediction's accuracy is inversely proportional to the look ahead horizon (Bontempi et al., 2013), representing a key constraint in the definition and modeling of forecasting algorithms. In our work, we propose a DNN model for forecasting the SYM-H and ASY-H indices. Therefore, we have to define the temporal scope and forecasting horizon as both properties define the input and output layers of our DNNs.

The networks presented in this section will look back 200 min into the past to make the predictions, forecasting the indices at the subsequent 1 and 2 h. As stated in the previous section, we use 5-min resolution data, so the 200 min used to make our prediction are grouped into 40 time steps. That decision has a great impact on the complexity of the network: it greatly reduces the amount of data needed to make each prediction, which reduces the model size and the required training resources (time and memory). The output will be the predicted value of the indices in the next 1 and 2 h.

Figure 1 depicts a sketch of the proposed network for the indices forecasting. It is built by a combination of single-dimensional CNN layers (Koprinska et al., 2018), LSTM layers (Hochreiter & Schmidhuber, 1997), and multilayer perceptron (Gardner & Dorling, 1998, also known as fully connected layers, from now on referenced as dense layers). The boxes shaded in gray represent operations without learnable parameters, whereas the colored ones have weight and bias parameters that need to be optimized.

The CNNs are designed to capture spatial and temporal dependencies. The most used layer of this family in the DL field is the two-dimensional convolutional layer, whose main application is image processing. Although we are not extracting image information, our time-series data are very similar to an image from the programmatic perspective: images are two-dimensional matrices of pixels (width, height, and color channels), while our input variables are matrices whose height is 1, its width is the time steps, and the data

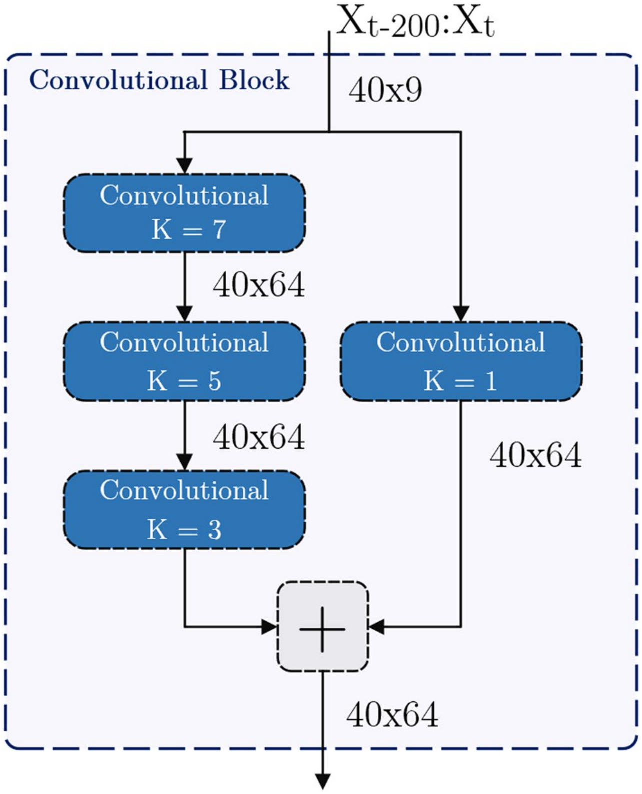

Figure 2. Residual convolutional block of the general architecture shown in Figure 1. K denotes the kernel size of each layer.

variables correspond to the color channels in the images. When dealing with data structured in this way, we can take advantage of the extensively researched convolutional layers. Under the hood, this layer applies the convolution operation over several subsequent time steps for each variable, extracting features from them, while maintaining the temporal structure.

The LSTM layer belongs to the family of the RNN (Jain, 2000). This family of layers is designed to solve the temporal dependency in the data. Instead of analyzing each time step individually or the time series as a whole, the layer loops over each time step, processing the input in a timely manner. For each time step, the relevant information is saved and combined with the one obtained from the previous time steps, while simultaneously discarding the features that are no longer relevant. This approach inherently takes into account the temporal dependency of the data, allowing the forecasting of time series with outstanding accuracy.

Focusing on our DNN, we have split the input data into two branches as can be seen in Figure 1: The left branch of the network (dashed red square) considers the whole input sequence, that is, the 40 time steps or 200 min history, whereas the right branch (dashed blue square) focuses on the last 20 time steps or last 100 min of data, that is, the recent history of the input data. This split allows one section of the network to have a more complete vision of the sequence fully analyzing it, meanwhile the other section focuses on the more recent input data.

The left branch consists of a CNN block with residual connections, similar to those used for image recognition (He et al., 2015). It is composed of three stacked CNN layers with different kernel sizes (7, 5, and 3) operating over the input features. Parallel to that stack, there is another CNN layer with a kernel size of 1 is also operating over the input data. The result of each CNN layers is added together as can be seen in Figure 2. This kind of architecture has proven to increase the performance over the usage of a single CNN layer, both in the image recognition task and in this particular forecasting problem.

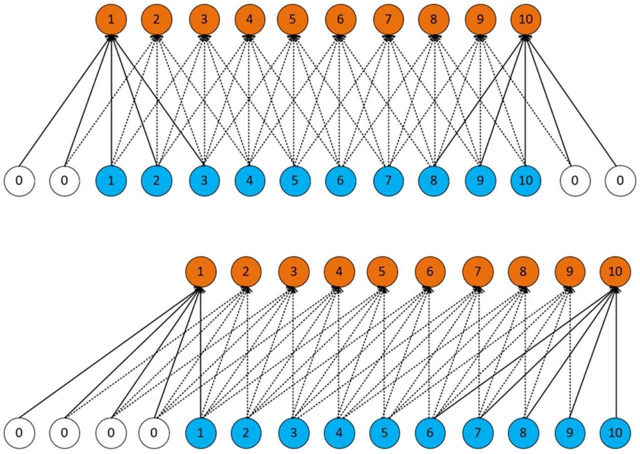

Figure 3. *Same* (top) and *causal* (bottom) padding comparison for a kernel size of 5 and a sequence length of 10. Calculated filters are represented in orange, input variables are represented in blue, and the added zeros in white.

One issue that has to be addressed in the CNN is the padding of the input features to maintain the sequence length, allowing the later addition following the residual connections architecture (otherwise the left and right branch of the residual CNN block would have different shapes). This is generally done by adding zeros using one of the following approaches: *causal* or *same* padding, as can be seen in Figure 3. The last one, the *same* padding, adds zeros to both ends of the data, meanwhile the *causal* padding only adds zeros to the left. This means that, using the *causal* padding, the CNN filter is only computed using the current and previous time steps, maintaining the temporal order. Another valuable advantage regarding the usage of this kind of padding in time series, specially when the convolutional layers are placed before the recurrent ones, is that the last time step processed by the following recurrent layer does not have any padded zeros to the end (i.e., the last time step). When applying filters to data padded with zeros, they are inherently contaminated. For image processing, this is not a significant issue, but for the time-series forecasting problem, it has great importance, since the last time step is the one that carries most of the relevant information when the derivatives are calculated in order to back-propagate and optimize the network (Hochreiter, 1998). These particularities make the *causal* padding the most suitable one for time-series forecasting tasks and the one used in our model.

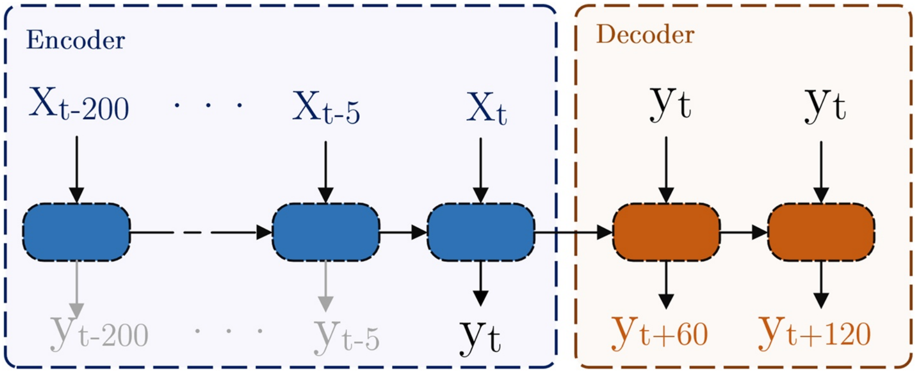

Figure 4. Encoder-Decoder block used in the prediction model (see Figure 1). Grayed outputs from the encoder are ignored. The encoder works as a sequence-to-vector network compressing the whole context into a vector. Then, the decoder produces the predictions from that encoder context.

Also, we have tested different activation functions for the CNN layers: The Rectified Linear Unit (ReLU) and Exponential Linear Unit (ELU), as well as no activation function. The best performance was achieved without using any activation function. However, the difference was very slight.

The decision to process the raw input data using CNN layers before the LSTM layers is motivated by the fact that CNNs are excellent feature extractors, so they can facilitate the work of the subsequent LSTM layers considerably.

The output of the convolutional block is then processed by a LSTM Encoder-Decoder architecture as shown in Figure 4. This block consists of two connected LSTM layers and is commonly used in Neural Machine Translation tasks. The first layer, called *encoder* is a sequence-to-vector network, that processes the added filters from the previous convolutional block, encoding the 40 time steps in a single vector, comprising all the important information during the whole sequence. Then, the next LSTM layer, called *decoder*, works as a vector-to-sequence network, recursively processing the context vector twice. The two resulting vectors for each iteration are used to predict the index value in the first and second hour, respectively. For both LSTM layers, the *tanh* function is used for the activation, meanwhile the *sigmoid* is used for the recurrent computations.

The output of the LSTM Encoder-Decoder block is combined with the output of the recent history processing block (right blue dashed box in Figure 1). This block works over the most recent time steps, that is, the last 20 (or last 100 min of input data). We do this separation, instead of working over the whole input sequence, because the recent history of the data has high significance for the forecasting.

The first part of this branch is composed of a single CNN layer, which we named convolutional context. It has a kernel size equal to the amount of time steps used as input. Using this configuration, each filter represents a summary of the last 20 time steps. Since we only want to obtain a single vector, no padding is needed. The second part of the branch consists of calculating the mean and standard deviation of each input variable for the last 20 time steps. This represents the amount of variability, and thus, disturbance, in the moments prior to the prediction.

For combining both branches of the architecture, first, the filters learned from the convolutional context are added to each resulting vector from the LSTM decoder. As the LSTM Decoder provides two output predictions (at $t + 60$ min and $t + 120$ min), we combine the filters from the convolutional context and the two predictions into two vectors, one for each predicted horizon, while maintaining the sequence shapes. This combination is inspired by the Residual Networks used in image recognition mentioned earlier.

Then, the mean and standard deviation previously computed are concatenated to each vector to add more information regarding the amount of disturbance of each variable in the latest time steps, increasing the length of each sequence by 18 (mean and standard deviation of the nine input variables).

Finally, each of the resulting vectors is processed through a dense block composed of three dense layers to output the final prediction as depicted in Figure 1. The first two dense layers apply the ELU activation function; it leads to a slower convergence, but usually it offers a better performance over the ReLU one. Finally, the last dense layer has only one neuron without an activation function to output the final prediction.

The ELU function is similar to the more common ReLU one, except on the negative inputs (the ReLU sets negative values to 0). The ELU function is calculated as in Equation 2, where α is a scale for the negative factor, which we fixed to 1, and z the input value.

$$\mathrm{ELU}(z) = \left\{ \begin{array}{cc} z & z \geq 0 \\ \alpha \cdot \left(e^{z} - 1 \right) & z < 0 \end{array} \right\} \tag{2}$$

Regarding the number of neurons of each layer, we have tested values of 32, 64, and 128, obtaining the best performance with 64 neurons. In total, 134.338 parameters need to be optimized when 64 neurons are used for each layer. The DNNs for both indices have been implemented using Keras, an open source library that provides a Python interface for the development of ANN and acts as an interface for the TensorFlow library.

3.1. Training and Validating

All neural networks tend to overfit the training set, losing their ability to generalize the predictions outside the training instances at some point. Thus, is of critical importance to decide which data will be used to train and test the network. The weights and biases of the connections between neurons are updated using the instances from the training set, iterating several times through it. Each iteration is known as epoch. After each epoch, the current weights and biases are evaluated against the validation set, without modifying the weights. This gives us an estimate of how well the network generalizes and when the network is simply overfitting the training data and thus, losing its ability to properly generalize to unseen, new data. By following this procedure, the validation set is only used to decide when to stop the training process.

During the training, the previously motioned weights and biases that govern the behavior of the network have to be modified. There are several optimization algorithms for controlling this process (Géron, 2017). Selecting which algorithm to use has a great impact in the training time, and, usually, the election is based on the experience. In our case, for the training process, we have used the Adam optimizer (Kingma & Ba, 2017). This optimizer is a stochastic gradient descent method, which computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. It is computationally efficient with a low memory requirement. This last makes it, from our perspective, the best choice for performing the training. The training convergence is controlled by a set of hyperparameters, whose are the following:

- Learning rate (0.001): step size at each iteration while moving toward minimizing the loss function.
- β_1 (0.9): the exponential decay rate for the first moment estimates.
- β_2 (0.999): the exponential decay rate for the second moment estimates.
- ϵ (1e−7): a small constant for numerical stability.

To finish the training, we have followed an early stop procedure in which, when the validation error has not improved for a defined number of epochs, the training finishes. We have set 50 epochs for the early stop and 200 epochs as a maximum number of epochs for the training process. Once the training process has ended, whether due to exhausting the number of epochs or because the validations loss ceased to improve, the weights and biases are reverted and fixed to those that achieved the best performance on the validation set. Then, the resulting model is evaluated on the test set, which has never been seen by the DNN.

4. Results

In this section, we will present an assessment of our DNN architecture for forecasting the SYM-H and ASY-H indices. Before presenting the results, next subsection introduces the metrics used.

4.1. Metrics

To evaluate the accuracy of the predictions made by the previously presented neural network, we have followed the evaluation guidelines for geomagnetic index predictions recommended by Liemohn et al. (2018). Such methodology implies computing several fit performance metrics between the predicted indices values and the real ones. The metrics used are the RMSE and the coefficient of determination R^2, which are also included in the survey by Camporeale (2019) for regression problems and commonly used other works in the field, for example, Yang et al. (2018) and Yang and Shen (2019). Both have been computed after reverting the standardization process done to the variables prior to using them in the network.

The first metric, RMSE (see Equation 3), gives an insight regarding how well the model captures the range of values of the index, highlighting the larger differences between the real values of the indices (y_i) and the predicted ones ($\hat{y}_i$) which mainly happens during active times where the error is farther from 0 and can be used to compare the results obtained by our model to those obtained by other authors.

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

The second metric, R^2 (see Equation 4), represents the amount of variance of the observed data, explained by the predicted, being $\overline{y}$ the mean of the N samples evaluated. This metrics output values usually ranged between 0 and 1. A $R^2 = 1$ means that the model perfectly predicts the target variable, meanwhile 0 implies that the model is equivalent to forecasting the mean of the target variable. It can also achieve negative values if the predictive model performs worse than simply predicting the average of the target variable.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \overline{y}\right)^2} \tag{4}$$

4.2. Model Evaluation

In this section, we present, discuss, and evaluate the predictions obtained using our DNN model for the 17 test storms (see Table 3) using the metrics discussed above, for both indices and both forecasting horizons (1 and 2 h). The metrics are displayed using three decimal places since we believe that level of precision is important regarding the metrics used. The results are summarized in Tables 4 and 5 for the SYM-H and ASY-H, respectively.

In Table 4, we show the computed RMSE and R^2 metrics for the SYM-H index, for each test storm, for both forecasting horizons. First, in the case of the 1-h horizon, the metrics are compared to the ones obtained by Siciliano et al. (2020), using their LSTM model (i.e., slightly better than their CNN model). If we compare the obtained results between each model, our proposed network achieves a significantly lower RMSE in 12 out of the 17 storms and behaves mostly the same in 2 out of the 17 and worse in 3 out of the 17. However, the mean is significantly lower. Regarding the R^2 metric, the mean results are the same, but the results presented by Siciliano et al. (2020) are provided with a lower precision, providing one decimal place for the RMSE and two for the R^2, meanwhile, we provide three decimal places in all cases, so the results may not be totally accurate.

Focusing on the predictions at 2 h, the results are compared against a persistence model, referenced as baseline (to our extent there are no other works that make predictions up to 2 h). The behavior of the persistence model is a simple prediction algorithm for which the prediction at both $t + 60$ and $t + 120$ is the value of the index at time t. This model has no predictive ability but serves as a basis for comparison. In this case, our model always outperforms the baseline model, the worst performing storm, number 30, only improves the baseline by a slight margin, but all the other storms are predicted significantly better.

Following, Table 5 depicts the computed metrics for the test storms for the ASY-H index for both forecasting horizons. In this case, the metrics are compared against the baseline model for both predictions horizon. The only storm in which our model behaved worse than the baseline is the prediction at 1 h for the storm number 30; in all the other cases, our model performs better.

Table 4
Metrics for the SYM-H Prediction Over the Test Storms Set (3), Comparing With the Work of Siciliano et al. for 1-h Predictions and the Baseline for the 2-h Prediction

	SYM-H 1-h prediction				SYM-H 2-h prediction			
	RMSE		R^2		RMSE		R^2	
Storm number	Siciliano et al.	Our DNN	Siciliano et al.	Our DNN	Baseline	Our DNN	Baseline	Our DNN
26	6.7	6.630	0.89	0.870	12.297	8.989	0.555	0.766
27	8.9	8.913	0.94	0.939	15.050	13.418	0.826	0.862
28	5.4	5.858	0.95	0.936	9.332	5.877	0.838	0.936
29	7.2	6.683	0.93	0.922	11.305	9.314	0.777	0.848
30	5.6	5.200	0.95	0.946	7.363	7.288	0.891	0.894
31	10.7	8.584	0.96	0.971	17.116	12.436	0.885	0.939
32	8.3	7.259	0.95	0.953	15.170	8.937	0.795	0.929
33	16.3	13.340	0.96	0.965	29.703	18.481	0.825	0.932
34	11.3	10.034	0.75	0.798	15.048	13.941	0.548	0.612
35	8.5	7.693	0.90	0.907	11.137	9.932	0.805	0.845
36	8.7	9.525	0.89	0.864	14.669	12.058	0.677	0.782
37	17.5	15.184	0.96	0.966	30.190	21.084	0.865	0.934
38	4.2	4.080	0.94	0.939	7.346	5.213	0.802	0.900
39	5.6	6.431	0.96	0.932	12.235	6.798	0.754	0.924
40	5.5	4.673	0.95	0.966	6.340	5.281	0.937	0.957
41	9.0	7.882	0.96	0.969	15.269	11.707	0.882	0.931
42	5.9	5.669	0.97	0.968	10.120	8.273	0.898	0.932
Mean	8.547	7.861	0.93	0.930	14.099	10.527	0.798	0.878

Note. The combined mean error is depicted in the last row.

The only other work that we can compare to regarding the ASY-H index is the one done by Bhaskar and Vichare (2019). In such work, they used a NARX network to predict the storms that happened in 2014 and 2015. This means that the only shared storm between both test subsets is the one that occurred in March 2015 (number 41 in Table 3). However, they considered different days for the start and end of the storm: they considered more than 20 days, meanwhile we only considered 10, so the comparison is not strictly fair, since the longer the period, more nondisturbed time will be predicted which will decrease the overall error metrics. Nevertheless, they reported a RMSE of 20.43, whereas our model achieved an error of 17.924 when forecasting 1 h ahead and 20.380 when the horizon was increased to 2 h.

Another experiment that we have performed is the evaluation of the models' predictions during large, mostly quite periods, that is, period that contained a small amount of storms or relative small storms that have never been seen by the DNN. Since a final objective is to deploy an operational forecasting system, we want to avoid false positives, increasing the reliability of our approach. To evaluate this behavior, we have computed the metrics for both indices, 1 and 2 h ahead, from January 2013, until December 2020. For such period, the minimum SYM-H is −233 (2015) and the maximum ASY-H is 348 (2015). It is important to note that none of the used training storms belong to this period (only two validation storms are in this range).

Table 6 shows the metrics computed for the SYM-H index over the predictions for each of the mentioned years. It is important to mention that since the number of disturbances on a yearly scale is low, the baseline model is harder to outperform. Still, only for the last 2 years, the 1-h look ahead model is slightly worse than the baseline model, whereas for the remaining years and for the 2 h look ahead, our model performs better than the baseline. Assessing the ASY-H reported in Table 7, only one year (2014) performs slightly worse than the baseline for the 1-h prediction, whereas for the remaining cases our model outperforms the baseline.

Table 5
Metrics for the ASY-H Prediction Over the Test Storms Set (3), Comparing With the Baseline for the 1- and 2-h Prediction

	ASY-H 1-h prediction				ASY-H 2-h prediction			
	RMSE		R^2		RMSE		R^2	
Storm number	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN
26	9.104	8.688	0.685	0.713	13.682	10.298	0.289	0.597
27	13.331	12.522	0.788	0.813	18.177	14.921	0.606	0.735
28	12.876	10.442	0.621	0.751	16.764	11.623	0.359	0.692
29	9.356	9.089	0.791	0.803	13.122	10.843	0.592	0.722
30	8.015	8.917	0.753	0.695	11.106	9.897	0.528	0.625
31	23.347	21.174	0.677	0.735	30.737	26.128	0.442	0.597
32	14.742	11.264	0.604	0.769	17.993	13.285	0.412	0.679
33	19.438	18.860	0.741	0.756	25.525	20.219	0.554	0.720
34	24.616	18.932	0.266	0.566	29.080	21.359	−0.021	0.449
35	17.054	13.979	0.494	0.660	21.331	16.273	0.209	0.540
36	16.590	14.116	0.344	0.525	20.247	16.274	0.026	0.371
37	29.021	23.115	0.692	0.805	33.708	27.533	0.586	0.724
38	7.386	7.335	0.623	0.628	9.892	8.203	0.324	0.535
39	12.126	10.639	0.655	0.734	15.143	10.412	0.464	0.747
40	10.536	9.163	0.779	0.833	11.943	9.826	0.718	0.809
41	17.818	17.924	0.731	0.728	21.462	20.380	0.611	0.649
42	11.126	9.944	0.802	0.842	13.962	11.144	0.689	0.802
Mean	15.087	13.300	0.650	0.727	19.051	15.213	0.435	0.647

Note. The combined mean error is depicted in the last row.

Finally, we provide the temporal profile of the SYM-H and ASY-H predictions for 1 and 2 h against the real values for the storms occurred in November 2004 as an example of an intense storm (Figure 5, storm number 37 in Table 3), and the storm occurred in June 2013 as an example of a moderate one (Figure 6, storm number 40 in Table 3). The plots for all the storms in the test set can be seen in the supporting information provided.

Table 6
Yearly SYM-H Prediction From 2013 to 2020 Compared to the Baseline

	SYM-H 1-h prediction				SYM-H 2-h prediction			
	RMSE		R^2		RMSE		R^2	
Year	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN
2013	4.891	4.805	0.905	0.908	7.021	6.235	0.804	0.845
2014	4.990	4.981	0.880	0.881	7.145	6.420	0.754	0.802
2015	6.383	5.967	0.913	0.924	9.195	7.829	0.820	0.870
2016	5.119	5.011	0.893	0.897	7.327	6.723	0.779	0.814
2017	5.294	5.135	0.881	0.888	7.586	6.833	0.755	0.801
2018	4.197	4.189	0.893	0.893	6.100	5.522	0.773	0.814
2019	3.849	3.892	0.860	0.856	5.552	5.140	0.708	0.750
2020	3.690	3.734	0.858	0.854	5.470	4.958	0.688	0.743
Mean	4.802	4.714	0.885	0.888	6.925	6.208	0.760	0.805

Table 7
Yearly ASY-H Prediction From 2013 to 2020 Compared to the Baseline

	ASY-H 1-h prediction				ASY-H 2-h prediction			
	RMSE		R^2		RMSE		R^2	
Year	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN	Baseline	Our DNN
2013	8.310	7.988	0.582	0.614	10.857	9.301	0.287	0.477
2014	8.264	8.321	0.498	0.491	10.952	9.819	0.119	0.292
2015	11.643	10.833	0.607	0.660	14.826	12.581	0.364	0.541
2016	10.604	9.758	0.447	0.531	13.109	11.090	0.153	0.394
2017	10.648	9.454	0.430	0.551	13.268	10.787	0.115	0.415
2018	7.990	7.524	0.503	0.560	10.279	8.640	0.178	0.419
2019	7.532	7.197	0.444	0.492	9.646	8.286	0.088	0.327
2020	6.579	6.597	0.464	0.461	8.719	7.781	0.058	0.250
Mean	8.946	8.459	0.497	0.545	11.457	9.786	0.170	0.389

Besides the great performance of our models, we still can find a few intervals where the predictions are not as accurate as expected. One of these examples is the about 6-h interval starting at 12 UT on April 16, 1999 (storm number 29) when the SYM-H (both 1- and 2-h) prediction does not accurately match with the observed SYM-H. The reason for this not-so-good performance seems to be related to the fact that the solar wind dynamic pressure is playing a major role in the solar wind–magnetosphere interaction. Figure 7 evidences that dynamic pressure increased during that interval, reaching about 30 nPa. Solar wind plasma parameters involved in dynamic pressure (proton density and flow speed) have not been considered as input parameters in our models. Therefore, the models are not expected to accurately reproduce the magnetospheric response (as seen by geomagnetic SYM-H and ASY-H indices) of an outstanding contribution of the solar wind plasma relative to the IMF contribution.

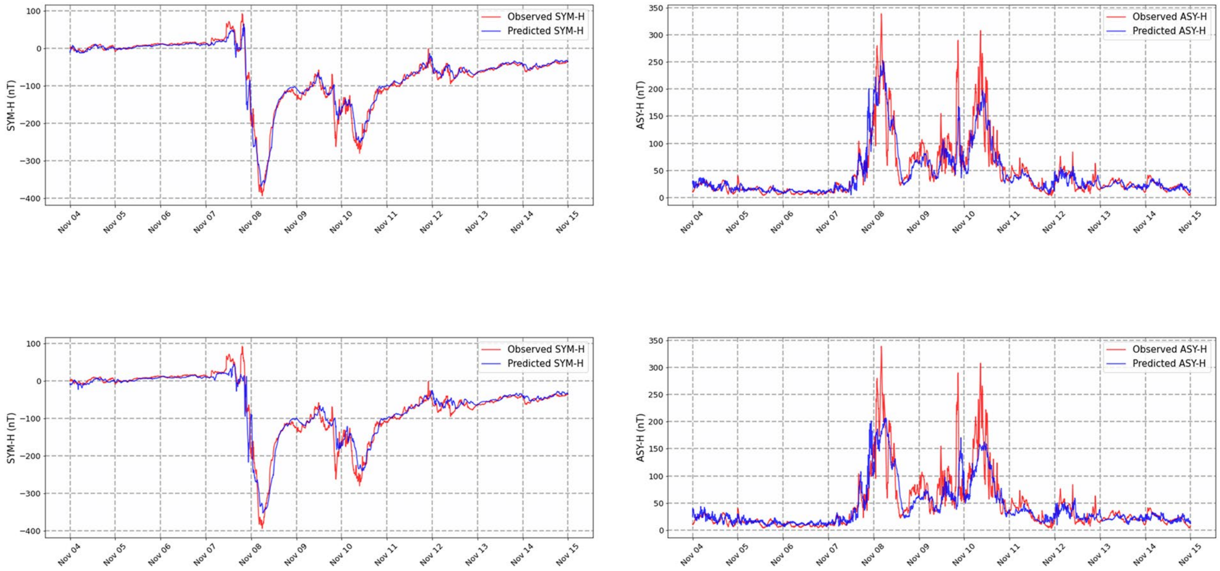

Figure 5. Predictions for the storm of November 2004. Top: 1-h prediction, SYM-H (left) and ASY-H (right). Bottom: 2-h prediction, SYM-H (left) and ASY-H (right).

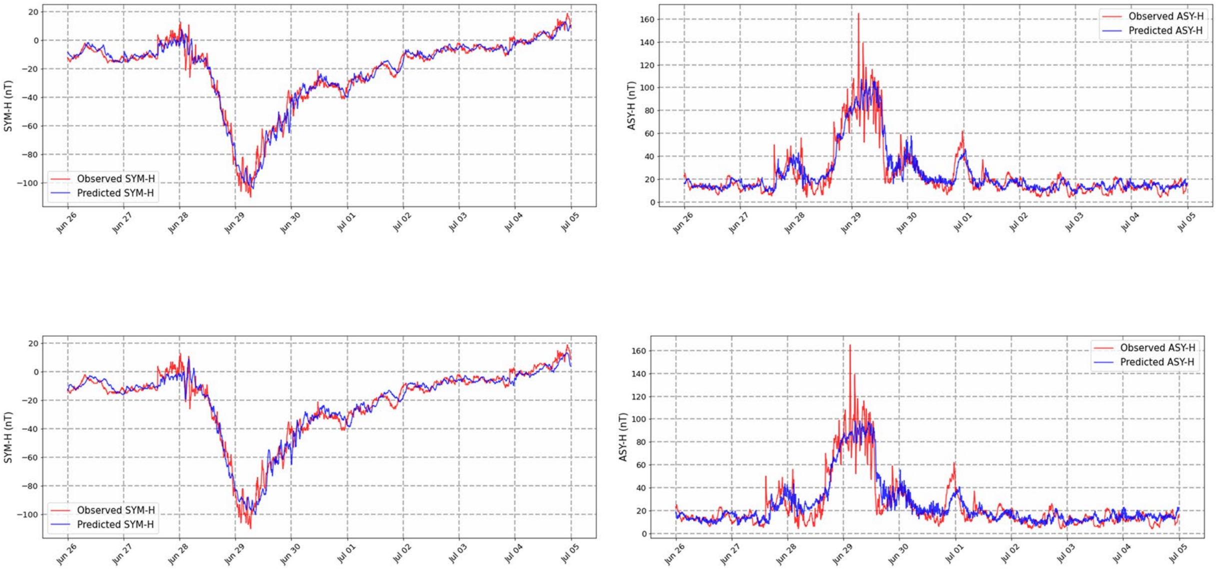

Figure 6. Predictions for the storm of June 2013. Top: 1-h prediction, SYM-H (left) and ASY-H (right). Bottom: 2-h prediction, SYM-H (left) and ASY-H (right).

5. Conclusions

The prediction of the SYM-H and ASY-H indices is of paramount importance to unravel the information about the response of the Earth's magnetosphere during geomagnetic storms. Given the importance of these predictions, a notable amount of work has been done trying to forecast its values. Even so, the problem has not been fully solved yet. Some forecasting systems cannot predict further enough into the future,

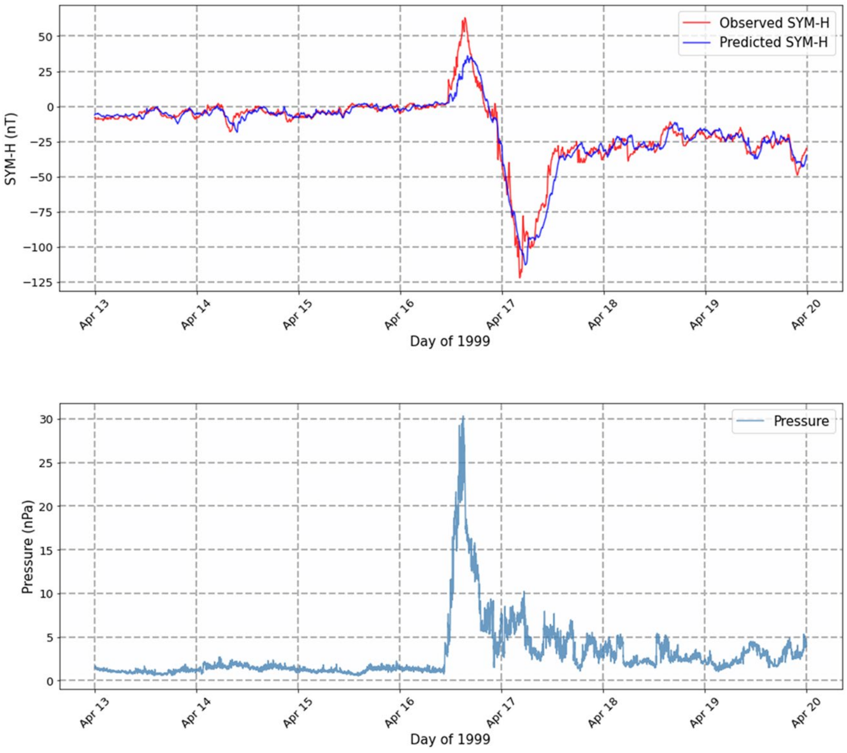

Figure 7. SYM-H 1-h prediction (top) and dynamic pressure (bottom) for the storm of April 1999.

the horizon in which the predictions are reliable has no real value yet, and when the prediction horizon is further enough into the future so it starts to become of great use, their accuracy is greatly hindered.

Thus, the present work proposes the usage of a DNN system to forecast them by exploiting the advantages of combining CNNs and LSTM layers. Using the IMF measured by ACE, and the index itself, the trained networks are able to predict both indices up to 2 h ahead. Although the overall metrics for both indices are quite accurate, the forecasting during the most intense storms can still be improved. However, the scarce number of events of such intensity, in conjunction with the voracity of neural networks for training data, increases the difficulty of this task.

Based on the evaluation of the network's prediction presented in the article, we can acknowledge that our predictions not only outperform the current state of the art forecasting models for the SYM-H index but also provide a good performance for the ASY-H when comparing to the persistence model. Thus, the proposed DNN will be implemented in a near future as a real-time forecasting product under the Spanish Space Weather Service (http://www.senmes.es). The implementation plan includes the real-time nowcasting of SYM-H and ASY-H.

We have noticed the influence of dynamic pressure in the SYM-H index in some intervals where predictions were not so accurate. We also observed that the ASY-H index is significantly harder to predict than the SYM-H, especially during the highest peaks. This suggests that this index may be influenced by more variables than the ones studied in this work. Future efforts will be devoted to include solar wind plasma parameters like proton density and flow speed among the input parameters of the DNN system. Even if there are several gaps in ACE plasma data during severe solar storm conditions, these problems could be solved by using solar wind plasma data from DSCVR mission.

Data Availability Statement

The predictions in *csv* format and high-resolution plots can be downloaded at https://zenodo.org/record/4562456 (Redxgit, 2021).

Acknowledgments
We acknowledge J. H. King and N. Papatashvilli at NASA and CDA Web for solar wind data (downloaded from https://cdaweb.gsfc.nasa.gov/index.html/) and the World Data Center for Geomagnetism (Kyoto) for geomagnetic indices data. We would also like to thank N. Ness of Bartol Research Institute, PI of ACE magnetic field instrument, and to D. J. McComas of Princeton/PPPL PI of ACE/SWEPAM Solar Wind Experiment. This work is cosupported by the JCLM project SBPLY/19/180501/000024 and the Spanish Ministry of Science and Innovation projects PID2019-109891RB-I00 and AYA2016-80881-P, all of them under the European Regional Development Fund (FEDER).

References

Bhaskar, A., & Vichare, G. (2019). Forecasting of SYMH and ASYH indices for geomagnetic storms of solar cycle 24 including St. Patrick's day, 2015 storm using NARX neural network. *Journal of Space Weather and Space Climate*, *9*, A12. https://doi.org/10.1051/swsc/2019007

Bontempi, G., Ben Taieb, S., & Le Borgne, Y.-A. (2013). Machine learning strategies for time series forecasting. In *Business intelligence* (pp. 62–77). Berlin/Heidelberg, Germany: Springer. https://doi.org/10.1007/978-3-642-36318-4_3

Burton, R. K., McPherron, R. L., & Russell, C. T. (1975). An empirical relationship between interplanetary conditions and Dst. *Journal of Geophysical Research*, *80*(31), 4204–4214.

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

Forsyth, C., Watt, C. E. J., Mooney, M. K., Rae, I. J., Walton, S. D., & Horne, R. B. (2020). Forecasting GOES 15 >2 MeV electron fluxes from solar wind data and geomagnetic indices. *Space Weather*, *18*, e2019SW002416. https://doi.org/10.1029/2019SW002416

Gardner, M. W., & Dorling, S. R. (1998). Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmospheric Environment*, *32*(14), 2627–2636. https://doi.org/10.1016/S1352-2310(97)00447-0

Géron, A. (2017). *Hands-on machine learning with Scikit-learn and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Newton, MA: O'Reilly Media.

Gruet, M. A., Chandorkar, M., Sicard, A., & Camporeale, E. (2018). Multiple-hour-ahead forecast of the Dst index using a combination of long short-term memory neural network and gaussian process. *Space Weather*, *16*, 1882–1896. https://doi.org/10.1029/2018SW001898

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*.

Hernandez, J. V., Tajima, T., & Horton, W. (1993). Neural net forecasting for geomagnetic activity. *Geophysical Research Letters*, *20*(23), 2707–2710. https://doi.org/10.1029/93GL02848

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*, 107–116. https://doi.org/10.1142/S0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu, S., Bhattacharjee, A., Hou, J., Sun, B., Roesler, D., Frierdich, S., et al. (1998). Ionospheric storm forecast for high-frequency communications. *Radio Science*, *33*(5), 1413–1428. https://doi.org/10.1029/98RS02219

Iyemori, T. (1990). Storm-time magnetospheric currents inferred from mid-latitude geomagnetic field variations. *Journal of Geomagnetism and Geoelectricity*, *42*(11), 1249–1265.

Iyemori, T., Araki, T., Kamei, T., & Takeda, M. (1992). *Mid-latitude geomagnetic indices ASY and SYM (provisional) no. 1 1989*. Kyoto, Japan: Data Analysis Center for Geomagnetism and Space Magnetism, Kyoto University.

Jain, L. C. (2000). *Recurrent neural networks: Design and applications*. Boca Raton, FL: CRC Press.

King, J. H. (2005). Solar wind spatial scales in and comparisons of hourly wind and ACE plasma and magnetic field data. *Journal of Geophysical Research*, *110*, A02104. https://doi.org/10.1029/2004JA010649

Kingma, D. P., & Ba, J. (2017). *Adam: A method for stochastic optimization*.

Koprinska, I., Wu, D., & Wang, Z. (2018). Convolutional neural networks for energy time series forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)* (p. 1–8). https://doi.org/10.1109/IJCNN.2018.8489399

Larrodera, C., & Cid, C. (2020). Bimodal distribution of the solar wind at 1 au. *Astronomy & Astrophysics*, *635*, A44.

Lazzús, J. A., Vega-Jorquera, P., Palma-Chilla, L., Stepanova, M. V., & Romanova, N. V. (2019). *Dst* index forecast based on ground-level data aided by bio-inspired algorithms. *Space Weather*, *17*, 1487–1506. https://doi.org/10.1029/2019SW002215

Licata, R. J., Tobiska, W. K., & Mehta, P. M. (2020). Benchmarking forecasting models for space weather drivers. *Space Weather*, *18*, e2020SW002496. https://doi.org/10.1029/2020SW002496

Liemohn, M. W., McCollough, J. P., Jordanova, V. K., Ngwira, C. M., Morley, S. K., Cid, C., et al. (2018). Model evaluation guidelines for geomagnetic index predictions. *Space Weather*, *16*, 2079–2102. https://doi.org/10.1029/2018SW002067

Maimaiti, M., Kunduri, B., Ruohoniemi, J. M., Baker, J. B. H., & House, L. L. (2019). A deep learning-based approach to forecast the onset of magnetic substorms. *Space Weather*, *17*, 1534–1552. https://doi.org/10.1029/2019SW002251

Mayaud, P. N. (1968). *Índices Kn, Ks et Km* (Tech. Rep.). Retrieved from http://isgi.unistra.fr/Documents/References/Mayaud{_}CNRS{_}1968.pdf

Menvielle, M., Iyemori, T., Marchaudon, A., & Nosé, M. (2011). Geomagnetic indices. In *Geomagnetic observations and models* (pp. 183–228). Dordrecht, The Netherlands: Springer. https://doi.org/10.1007/978-90-481-9858-0_8

O'Brien, T. P., & McPherron, R. L. (2000). An empirical phase space analysis of ring current dynamics: Solar wind control of injection and decay. *Journal of Geophysical Research*, *105*(A4), 7707–7719.

Rastätter, L., Kuznetsova, M. M., Glocer, A., Welling, D., Meng, X., Raeder, J., et al. (2013). Geospace environment modeling 2008–2009 challenge: Dst index. *Space Weather*, *11*, 187–205. https://doi.org/10.1002/swe.20036

Redxgit (2021). Redxgit/supporting_data_paper_sw: Space Weather Submission. Zenodo. https://doi.org/10.5281/zenodo.4562456

Shprits, Y. Y., Vasile, R., & Zhelavskaya, I. S. (2019). Nowcasting and predicting the Kp index using historical values and real-time observations. *Space Weather*, *17*, 1219–1229. https://doi.org/10.1029/2018SW002141

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*, e2020SW002589. https://doi.org/10.1029/2020SW002589

Sugiura, M. (1964). Hourly values of equatorial Dst for IGY. In *Annals of the International Geophysical Year* (Vol. 35, pp. 945–948). Oxford: Pergamon Press.

Tshisaphungo, M., Habarulema, J. B., & McKinnell, L.-A. (2018). Modeling ionospheric *foF*2 response during geomagnetic storms using neural network and linear regression techniques. *Advances in Space Research*, *61*(12), 2891–2903.

Welling, D. T., Ngwira, C. M., Opgenoorth, H., Haiducek, J. D., Savani, N. P., Morley, S. K., et al. (2018). Recommendations for next-generation ground magnetic perturbation validation. *Space Weather*, *16*, 1912–1920. https://doi.org/10.1029/2018SW002064

Yang, Y., & Shen, F. (2019). Modeling the global distribution of solar wind parameters on the source surface using multiple observations and the artificial neural network technique. *Solar Physics*, *294*(8), 111. https://doi.org/10.1007/s11207-019-1496-5

Yang, Y., Shen, F., Yang, Z., & Feng, X. (2018). Prediction of solar wind speed at 1 AU using an artificial neural network. *Space Weather*, *16*, 1227–1244. https://doi.org/10.1029/2018SW001955

Zhelavskaya, I. S., Vasile, R., Shprits, Y. Y., Stolle, C., & Matzka, J. (2019). Systematic analysis of machine learning and feature selection techniques for prediction of the Kp index. *Space Weather*, *17*, 1461–1486. https://doi.org/10.1029/2019SW002271