Space Weather

RESEARCH ARTICLE
10.1029/2020SW002589

Key Points:
- Two artificial neural network (ANN) models are built to forecast SYM-H index 1 h ahead using interplanetary magnetic field measurements
- The developed models are based on two conceptually different neural networks: long short-term memory and convolutional neural network (CNN)
- CNN, used here for the first time for geomagnetic indices forecasting, has proved potentialities worth being further explored

Supporting Information:
- Supporting Information S1

Correspondence to:
R. Tozzi,
roberta.tozzi@ingv.it

Citation:
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

Received 15 JUL 2020
Accepted 16 NOV 2020
Accepted article online 21 NOV 2020

Forecasting *SYM-H* Index: A Comparison Between Long Short-Term Memory and Convolutional Neural Networks

F. Siciliano[1], G. Consolini[2], R. Tozzi[3], M. Gentili[1], F. Giannattasio[3], and P. De Michelis[3]

[1]Department of Computer, Control and Management Engineering Antonio Ruberti, Sapienza University of Rome, Rome, Italy, [2]INAF-Istituto di Astrofisica e Planetologia Spaziali, Rome, Italy, [3]Istituto Nazionale di Geofisica e Vulcanologia, Rome, Italy

Abstract Forecasting geomagnetic indices represents a key point to develop warning systems for the mitigation of possible effects of severe geomagnetic storms on critical ground infrastructures. Here we focus on *SYM-H* index, a proxy of the axially symmetric magnetic field disturbance at low and middle latitudes on the Earth's surface. To forecast *SYM-H*, we built two artificial neural network (ANN) models and trained both of them on two different sets of input parameters including interplanetary magnetic field components and magnitude and differing for the presence or not of previous *SYM-H* values. These ANN models differ in architecture being based on two conceptually different neural networks: the long short-term memory (LSTM) and the convolutional neural network (CNN). Both networks are trained, validated, and tested on a total of 42 geomagnetic storms among the most intense that occurred between 1998 and 2018. Performance comparison of the two ANN models shows that (1) both are able to well forecast *SYM-H* index 1 h in advance, with an accuracy of more than 95% in terms of the coefficient of determination R^2; (2) the model based on LSTM is slightly more accurate than that based on CNN when including *SYM-H* index at previous steps among the inputs; and (3) the model based on CNN has interesting potentialities being more accurate than that based on LSTM when not including *SYM-H* index among the inputs. Predictions made including *SYM-H* index among the inputs provide a root mean squared error on average 42% lower than that of predictions made without *SYM-H*.

Plain Language Summary Geomagnetic indices are proxies of geomagnetic disturbances observed on the ground during geomagnetic storms and substorms. This work deals with the forecasting of one of such indices, that is, *SYM-H* index, using two different artificial neural network architectures. Between the two, one has never been used for this purpose, being generally applied for image processing. Both the architectures provide good predictions. The capability to forecast high-resolution geomagnetic indices, such as *SYM-H* index, is crucial in issuing alerts for fast geomagnetic disturbances which can be responsible for the activation of geomagnetically induced currents (GICs), one of the most harmful ground effects of space weather events.

1. Introduction

It is well known that solar activity influences the state of the circumterrestrial space and can affect technological systems in many different ways and with different degrees of damage severity. For instance, variations in natural electromagnetic fields occurring during geomagnetic storms can disturb satellite navigation systems, or can also cause the building up of geomagnetically induced currents (GICs) (Hapgood, 2018). GICs can lead to the malfunctioning of power transformers, thus resulting in a reduction in the capacity of a power grid, and to blackouts at worst. Recovery from such events can last from a few hours to a few days, depending on the extent of the damage. These phenomena are significant especially in regions at high latitudes, but it has now been established that the risk to middle and low latitudes cannot be ignored as well (e.g., Carter et al., 2016; Gaunt & Coetzee, 2007; Moldwin & Tsu, 2016; Tozzi et al., 2019; Viljanen et al., 2014).

In order to find strategies for mitigating the damaging effects of space weather events on human technology, a lot of efforts are being devoted to forecast the occurrence and intensity of geomagnetic storms. In fact, it is still often impossible to completely protect electronic devices and critical infrastructures from the

effects of geomagnetic storms. For instance, in the case of expected adverse space weather conditions, hazard mitigation could include the temporary increase of the power networks resistance, the warning of users of satellite navigation systems in advance of the upcoming malfunction of the systems and of its duration, and the reset of digital systems to deal with any errors (Hapgood, 2018). Since the dependence of our society on digital technologies continues to grow, it is essential to be able to predict these space weather phenomena. In this framework, forecasting geomagnetic indices (inherently related to the enhancement of the geomagnetic disturbance) is one of the principal tasks to provide an alert to mitigate the space weather hazard. Geomagnetic indices are calculated from ground magnetic field measurements and are able to efficiently monitor the state of the Earth's magnetosphere and ionosphere as well as of the circumterrestrial electromagnetic environment. Details on the derivation and meaning of the most widely used geomagnetic indices can be found in Mayaud (1980).

In the past, several attempts have been made to forecast geomagnetic indices using, for instance, linear prediction models, differential equations (Burton et al., 1975; Wang et al., 2003), or Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) models (Boynton et al., 2011). However, linear models generally fail in giving a correct description of the evolution of the solar-wind-magnetosphere-ionosphere system due to the intrinsic nonlinear response of the circumterrestrial environment to changes of the interplanetary medium (see e.g., Consolini & Chang, 2001; Klimas et al., 1996). A way to cope with the need to capture the complex response of the magnetosphere to solar disturbances is to resort to models based on artificial neural networks (ANNs). Indeed, according to the universal approximation theorem (e.g., Hornik, 1991; Leshno et al., 1993), neural networks have the ability, under certain assumptions, to approximate any function.

The first investigations on the use of neural networks to forecast geomagnetic indices, mainly D_{st} and K_p indices, date back to 1990s. Among these, we mention the works by Gleisner et al. (1996), Lundstedt and Wintoft (1994), and Wu and Lundstedt (1997) on D_{st} index forecasting and by Costello (1998) on K_p index forecasting. Wu and Lundstedt (1997), as well as Lundstedt et al. (2002), used the Elman network that is a recurrent network indicated for time series prediction. Wu and Lundstedt (1997) also surveyed the inputs that improve the ANN's ability to predict D_{st} index and those that are instead irrelevant.

Most of these models use as inputs the measurements of the quantities driving the Earth's magnetospheric dynamics, that is, magnetic field and plasma parameters (solar wind density and velocity) of the interplanetary medium at the libration point L1. During intense interplanetary disturbances, however, some of them may be unavailable due to the saturation of measuring instruments, as for instance those for plasma parameters. In the attempt to overcome this issue, Pallocchia et al. (2006) developed an ANN model based on an Elman network that uses only interplanetary magnetic field (IMF) measurements to forecast, with a good performance, the D_{st} index (see also Amata et al., 2008). It turned out that this model displayed some limitation in the capability of predicting geomagnetic indices at a shorter time resolution, even using plasma parameters besides IMF measurements (Pallocchia et al., 2008).

Concerning prediction of D_{st} index, many advances have been achieved and very different approaches explored. Chandorkar et al. (2017) proposed a probabilistic prediction based on a Gaussian regression process. An important advantage of this approach is the possibility to associate error bars to predictions, a feature that is not available with other models as neural networks or linear models and that makes the probabilistic approach appealing for space weather operational tools. The performance of two models, one using only D_{st} index as input parameter and the other using also solar wind velocity and IMF B_z, has been tested on 63 geomagnetic storms that occurred between 1998 and 2006. A hybrid approach was proposed by Lazzús et al. (2017) who merged an ANN model with a particle swarm optimization to forecast D_{st} index from 1 to 6 h ahead, using as input parameters only past D_{st} values. When forecasting D_{st} index up to 1 and 3 h ahead, they obtained very low root mean squared errors (RMSEs) of 3.5 and 7.5 nT, respectively. Higher RMSE values are obtained for forecasts beyond 3 h, reaching 10.89 nT for a 6-h ahead prediction. Another hybrid approach has been adopted by Gruet et al. (2018) who combined the forecasting ability of a long short-term memory (LSTM) recurrent neural network (RNN) with the advantage to associate error bars and confidence intervals to predictions given by a Gaussian process model. For the first time, Gruet et al. (2018) included in the inputs also GPS data and forecast D_{st} index from 1 to 6 h in advance with RMSE ranging from 5.25 to 9.86 nT.

Wintoft et al. (2017), besides proposing different ANN models to predict K_p from IMF and solar wind data, investigated the role of various input parameters in the goodness of prediction by adding them one by one.

They started from IMF B_z and then added solar wind density, solar wind velocity, IMF total field B, and IMF B_y component. Tan et al. (2018) used an LSTM network to predict the occurrence of geomagnetic storms characterized by values of K_p index ≥5. Their inputs are IMF and solar wind data in addition to past K_p values, thus improving K_p forecasting capability in respect to previous works.

To conclude this brief overview, we mention the work by Bala and Reiff (2012) with their predictions of K_p, D_{st}, and *AE* indices 1 h ahead with minimum RMSE of 0.62, 8.84, and 92.9 nT, respectively, and using as inputs solar wind velocity and IMF magnitude *B*. For a more comprehensive picture of the different efforts made in the last decades in the field of D_{st} and K_p indices forecasting, refer, for instance, to Camporeale (2019) who gives also an interesting overview of the role of machine learning in space weather.

While a lot of work has been made to forecast D_{st} and K_p indices, only a few authors have approached *SYM-H* index forecasting. *SYM-H* is characterized by a time resolution higher than D_{st} and K_p (1 min instead of 1 and 3 h, respectively) (Iyemori, 1990; Wanliss & Showalter, 2006). So its forecasting poses a more difficult challenge due to its highly oscillating character. Cai et al. (2010) predicted, for the first time, 5-min averages of *SYM-H* index using a Nonlinear AutoRegressive with eXogenous inputs (NARX) neural network. They compared NARX forecasts with those produced by the Elman network over a sample of 73 geomagnetic storms that occurred between 1998 and 2006. More recently, again using the same type of network, Bhaskar and Vichare (2019) predicted not only *SYM-H* index but also *ASY-H* index using 92 geomagnetic storms that occurred between 1998 and 2013. The developed network, similar to that built by Cai et al. (2010), successfully predicts *SYM-H* and *ASY-H* indices about an hour prior to the start of the storm. The networks proposed by Cai et al. (2010) and by Bhaskar and Vichare (2019) both use, as input parameters, data of IMF magnitude (*B*), B_y and B_z components, and solar wind density and velocity. On the whole, *SYM-H* prediction performance of the two networks is very good both in terms of the average correlation coefficient, being 0.95 for Cai et al. (2010) and 0.9 for Bhaskar and Vichare (2019), and in terms of RMSE, being 14.6 nT for Cai et al. (2010) and 13.98 nT for Bhaskar and Vichare (2019). All models cited so far rely, among others, on solar wind density or velocity as input parameters. However, as already mentioned above, this represents a potential Achilles heel for the models oriented to operational forecasting.

The present study grounds its roots in the work by Pallocchia et al. (2006) who proposed an ANN called empirical D_{st} data algorithm (EDDA), to forecast D_{st} index 1 h in advance. In terms of operational forecasting, the important advantage of EDDA is the use of IMF measurements only, acquired by the ACE satellite, placed in the Lagrangian point L1. According to the authors, the three magnetic inputs of the EDDA model, namely, IMF B_z, B^2, and B_y^2, can capture the vast majority of the relevant information needed to describe the relationship between the solar wind trigger and D_{st} index, especially under conditions of improved geomagnetic activity.

In the same way, we will use as input parameters only ACE IMF measurements, but, differently from Pallocchia et al. (2006), we will (1) focus on the forecasting of *SYM-H* index (instead of D_{st}) and (2) build two ANN models using LSTM and convolutional neural networks (CNNs), instead of the Elman network. The choice to move from D_{st} to *SYM-H* index is motivated by the long-term purpose of this work: both D_{st} and *SYM-H* indices provide information on the intensity of the ring current, but in the perspective of being able to issue alerts for GICs, a natural next step is to attempt the forecasting of a high-resolution geomagnetic index. Indeed, it is well known that GIC intensity is highly correlated to the rate of change of the Earth's magnetic field at subhour timescales (Welling et al., 2018). Thus, the forecasting of the high-resolution *SYM-H* index could represent a first step toward the estimation of the expected variation of the magnetic disturbance on the ground at low and middle latitudes.

Another choice that distinguishes this work from that done by Pallocchia et al. (2006) and from that already done on *SYM-H* forecasting (Bhaskar & Vichare, 2019; Cai et al., 2010) is the use of the LSTM and CNN neural networks. The former is a sophisticated RNN (the family to which Elman network belongs), while the latter is more frequently used for image processing and has never been adopted in geomagnetic indices forecasting. LSTM has proved to have a very good performance with respect to previous models in D_{st} index forecasting. Differently, so far, CNN has not received much attention within the space weather community as also highlighted by Camporeale (2019): despite CNN being "one of the most successful trends in machine learning (LeCun et al., 2015), it has been barely touched in this community."

Details on the selected neural networks and on the way they have been implemented are the topics of section 2. Data used are described in section 3. Discussion of findings and comparison of the performance of the two developed ANN models are presented in section 4. Main conclusions are drawn in section 5.

2. ANN Models Description

As mentioned in section 1, two ANN models are developed to forecast *SYM-H* index: one is based on the LSTM neural network and the other on CNN.

LSTM (Hochreiter & Schmidhuber, 1997) belongs to the class of RNNs that are specifically designed for sequence prediction problems, such as time series analysis or speech recognition. The main feature of RNNs is their ability to keep information about the past, thanks to the presence of loops inside them. An RNN can be thought of as multiple copies of the same network, each passing a message to the next as shown in Figure 1a for the specific case of an LSTM network. RNNs receive the data sequentially, so a higher weight is given to the data belonging to the more recent temporal instants. With respect to simple RNNs, LSTM better models complex temporal dependencies distinguishing between long-term and short-term time memory. In addition, LSTM networks do not suffer from the vanishing of the cost function gradient. This issue occurs during the training of simple RNNs using back propagation. In practice, the partial derivative of the cost function with respect to the current weights becomes so small that it prevents the weights from changing their values and the training from progressing.

CNNs, first introduced in 1990 by LeCun et al. (1990) as an effective technique for handwritten digit recognition, are now widely used in image analysis (LeCun et al., 2015). They distinguish elements in an array by assigning to the various features present in it different importance levels through weights and biases. Differently from LSTM networks, CNNs receive the input all at once, that is, all temporal instants at the same time. The network ignores which data are temporally closer to the data to be predicted and must therefore understand autonomously which are more related to the output to be predicted.

Both the ANN models described in what follows have been implemented using Keras, a deep learning Application Programming Interface (API) written in Python, running on top of the machine learning platform TensorFlow.

2.1. The LSTM Neural Network

The ANN model based on the LSTM neural network is composed of an LSTM layer and four dense layers. With the term dense layer, alternatively called fully connected layer, we refer to the basic structure of neural networks, that is, a series of neurons that receive the input from all the neurons of the previous layer. A simplified sketch is shown in Figure 4.

The idea behind LSTM network is that, for each time, the network receives the inputs one by one, makes an aggregate of the past information, and updates it with the new data. In this way, the output depends both on the current inputs and the aggregate of the past states. The LSTM basic module has a more complex structure than a simple RNN being composed of three interacting gates.

Figure 1b represents a sketch of the LSTM basic cell that consists of four hidden layers (hidden meaning everything in between the input and the output layers): the cell state and three gates. To describe the cell state, the image of a conveyor belt that transports information across the cells and interacts only linearly with what gets through the gates is often used. Information is added or removed from the cell state based on the operations performed within the gates that have the role to "decide" what information is allowed to reach the conveyor belt. Before entering into the details of the LSTM module functioning, the following notation is introduced here: x_t is the input at time t, h_t is the output from the cell that treated x_t, C_t is the cell state at time t, and W and b indicate weights and biases, respectively (a subscript will indicate the hidden layer they belong to).

The first gate is the forget gate layer, highlighted by the red dashed rectangle in Figure 1b. It contains a *sigmoid* function, σ, which outputs values in [0,1] and has the role to decide how much of h_{t-1} should get through the gate and arrive to the cell state C_{t-1}, based on the value of both x_t and h_{t-1}. This is done calculating f_t as follows:

$$f_t = \sigma\left(W_f(h_{t-1}, x_t) + b_f\right) \tag{1}$$

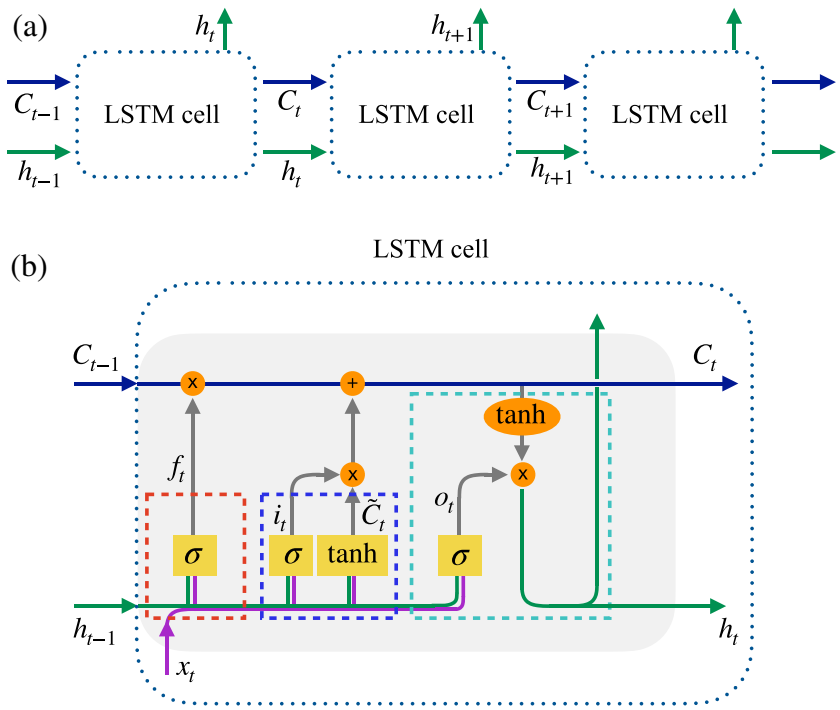

Figure 1. (a) An RNN made of LSTM cells (unfolded version). (b) Structure of the LSTM cell: cell state C_t (blue line), cell output h_t (green line), cell input x_t (purple line), forget gate (dashed red rectangle), input gate (dashed blue rectangle), output gate (dashed aquamarine rectangle), activation functions (yellow rectangles), other operators (orange circles), and gates outputs (gray lines).

where the subscript f stands for "forget." f_t ranges in $[0, 1]$: 0 represents the decision to completely forget the previous cell state, and 1 represents that to keep the previous state unchanged. The argument of the *sigmoid* function is given by the sum of the input at time t, x_t, the LSTM cell output at time $t-1$, h_{t-1}, both multiplied by the layer weights W_f, and added together with the layer bias b_f.

The second gate is the input gate layer that decides the new information that can reach the cell state. The input gate is highlighted by the blue dashed rectangle in Figure 1b. In this gate, a tanh function computes the new information to add to the cell state, that is, $\tilde{C}_t$, through

$$\tilde{C}_t = \tanh(W_C(h_{t-1}, x_t) + b_C) \tag{2}$$

$\tilde{C}_t$ is a linear combination of x_t and h_{t-1} through its weights W_C and bias b_C. A *sigmoid* function decides how much of this new candidate to take into consideration. This is done through

$$i_t = \sigma(W_i(h_{t-1}, x_t) + b_i) \tag{3}$$

where the subscript i stands for "input." Once the forget and input gates have accomplished their tasks, it is possible to update the cell state. The new cell state C_t is given by

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \tag{4}$$

The last gate is the output gate highlighted by the aquamarine dashed rectangle in Figure 1b. Here, the *sigmoid* function computes o_t, to decide the fraction of the cell state to output, in detail:

$$o_t = \sigma(W_o(h_{t-1}, x_t) + b_o) \tag{5}$$

where the subscript o stands for "output." So, the final output h_t is given by

$$h_t = o_t * \tanh(C_t) \tag{6}$$

This result will be used by the next layer in the network and/or by the LSTM cell in the next time step.

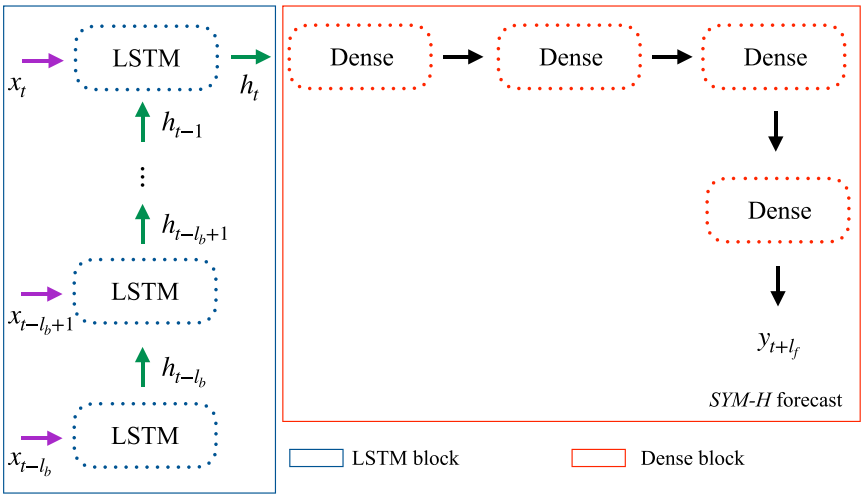

Figure 2. Scheme of the ANN model based on LSTM used for *SYM-H* index forecasting (x_t is the input at the time t, h_t is the output from the cell that treated x_t, l_b is look-back, l_f look-forward, and y_{t+l_f} the prediction).

So far, we have considered as input a vector x_t, but we actually work with a matrix of dimension $N \times l_b \times H$, where N represents the number of samples (i.e., the length of the time series), l_b is the look-back that defines how many time instants in the past we want the network to have access to, and H is the number of input parameters.

In our case, the total number of samples is 118,316, but the value of N decreases depending on the value used for look-back. H is equal to 3 when the inputs are IMF B_z, B_y^2, and B^2 or to 4 with the additional input of *SYM-H* index. Instead of a priori setting l_b, we run the model for different look-back values. In detail, we start considering values of l_b corresponding to 90, 120, 180, 360, 540, 720, 1,080, and 1,440 min for both cases of $H = 3$ and $H = 4$. Based on the results obtained for the case of $H = 3$ (i.e., no *SYM-H* index as input), we investigate the behavior of the model also for values of l_b corresponding to 1,800, 2,160, 2,520, and 2,880 min. We will return on this choice in section 4. We specify that when running the models, the actual values of look-back are those listed above divided by the time sampling. A sketch of the logical flow of information in the LSTM network used in this work is presented in Figure 2. This figure displays the LSTM block (composed of LSTM layers) and the dense block (composed of four dense layers).

According to Figure 2, when forecasting the value y_{t+l_f}, the input to the model consists of values of the input parameters from $t - l_b$ to t. l_f represents how many time instants in the future from t *SYM-H* index will be predicted. The LSTM layer computes the output h_t as explained above, processing the l_b inputs one by one and returning the final value h_t. It should be noted that h_t is a vector, since the LSTM layer has a $n_R = 32$ number of neurons. The vector h_t is now passed to the dense block consisting of four dense layers. Each neuron of the dense layers receives the input from all the neurons of the previous layer, calculates a linear combination of the input data, adds a bias, and then passes the result to an activation function. In detail, the first three layers j ($j = 1,..., 3$) perform the following operation:

$$h_t^j = \phi_{D_j}(W_{D_j} \cdot h_t^{j-1} + b_{D_j}) \tag{7}$$

where h_t^j is the output of the jth dense layer, W_{D_j} and b_{D_j} are its weights matrix and bias, respectively, ϕ_{D_j} is its activation function, and h_t^0 is the output of the LSTM layer. Here, h_t^j is a vector of length n_{D_j} =32, being n_{D_j} the number of neurons of the dense layers. Finally, the last dense layer provides its output y_{t+l_f} defined by

$$y_{t+l_f} = W_{D_f} \cdot h_t^3 + b_{D_f} \tag{8}$$

where W_{D_f} and b_{D_f} are the weights matrix and bias for the final dense layer. As can be deduced from Equation 8, the last dense layer has no activation function.

It is worth underlining that the number of neurons n_R and n_D, as well as the type of the activation function used in LSTM and dense layers, have been tuned to obtain the optimal performance of the ANN model.

Concerning n_R and n_D, we considered the same number of neurons for all layers (LSTM and dense), testing values of 8, 16, 32, and 64. Concerning the activation function, we compared tanh and relu (defined as $\text{relu}(x) = max(0, x)$) for both the LSTM and dense block (testing all possible combinations), obtaining the best performance using the tanh function for the LSTM and the relu function for the dense layers. To summarize, the ANN model based on LSTM used in this study has either 3 or 4 original external input nodes (depending on the use or not of *SYM-H* index), 32 hidden neurons per layer, and 1 output neuron.

2.2. The Convolutional Neural Network

CNNs are a class of ANNs most commonly used for image analysis and characterized by the ability to successfully capture spatial and temporal dependencies. Unlike RNNs, like LSTM, a layer of the CNN is able to receive the entire input matrix and process it all at once. More in general, they are conceived to process data stored in the form of multiple arrays, and for this reason, they are very flexible. Images can be thought of three (one for each RGB color channel) matrices containing pixel intensities in each cell of the matrix. A CNN input is in fact a matrix of shape $iW \times iH \times iD$ (here iW, iH, and iD are the image width, height, and depth, respectively). The last dimension is often understood as the number of filters of an image, but it can also represent the number of input parameters. Since here we are going to analyze time series instead of images, $iH = 1$ and iW represents the number of time instants, but for the sake of a more complete explanation, the CNN will be explained keeping iH as generic. Differently from the LSTM network, CNN architecture does not use loops, but it consists of a sequence of layers.

Generally, CNNs are composed of two main layers: a convolutional layer and a pooling layer. In addition to these two layers, other structures and/or techniques are often used in CNNs, for example, batch normalization, local response normalization, attention, and inception. The role of the convolutional layer is to analyze small portions of the matrix to find the important features, while that of the pooling layer is usually to synthesize the information.

To characterize the convolutional layer, five elements must be defined: kernel size (K), stride value (s), padding (p), activation function (ϕ), and number of filters (F). The kernel (alternatively named filter) is the element that performs the convolutional operation; in practice, it is a matrix of weights with dimension $K \times K$.

The kernel is moved over the image, every time performing a matrix multiplication between its weights and the portion of the image that must have the same size of the kernel, until it parses the complete image. The stride value defines how many matrix cells the kernel moves each time it performs its operation. In this way, if the stride value is greater than 1, the matrix resulting from this operation has a smaller dimension than the original one. Padding, that is, preliminary adding zeros to the sides of the image, can avoid decreasing the size of the image.

The activation function transforms the value obtained from the convolution of the image with the kernel into the final output. It is also possible to use a number F of kernels, simultaneously. Each has its own different weights, meaning that the operations described above are repeated F times.

Figure 3, as an example, displays the case of $iW = iH = 6, iD = 2, K = 2, s = 1$, relu as activation function, $F = 1$, and padding applied just to a border 1 cell wide, that is, $p = 1$. Once the element-wise product between the kernel matrix and the portion of the input channel with size 2×2 is made for both kernel channels, resulting values are summed together with the bias. They are then passed to the activation function whose output is used to fill the matrix that represents the output of the convolutional layer. The kernel is then moved 1 cell on the left and the operation is repeated from the beginning.

The other essential part of a CNN is the pooling layer. Its objective is to reduce the size of the input, mainly to decrease the computational effort of data processing. As for the convolutional layer, the pooling layer is characterized by a kernel, of variable size, that moves through the entire image, according to the stride value. However, unlike the kernel of the convolutional layer, it is not aimed at weighting but performs only one operation on the area selected by the kernel. Most frequently used pooling layers select either the maximum or the average of the values present in the area.

An example of pooling layer functioning is displayed in Figure 4 where the case of the use of two pooling layers, with size 2×2, is represented. Given the matrix 6×6, that is the output of the convolutional layer,

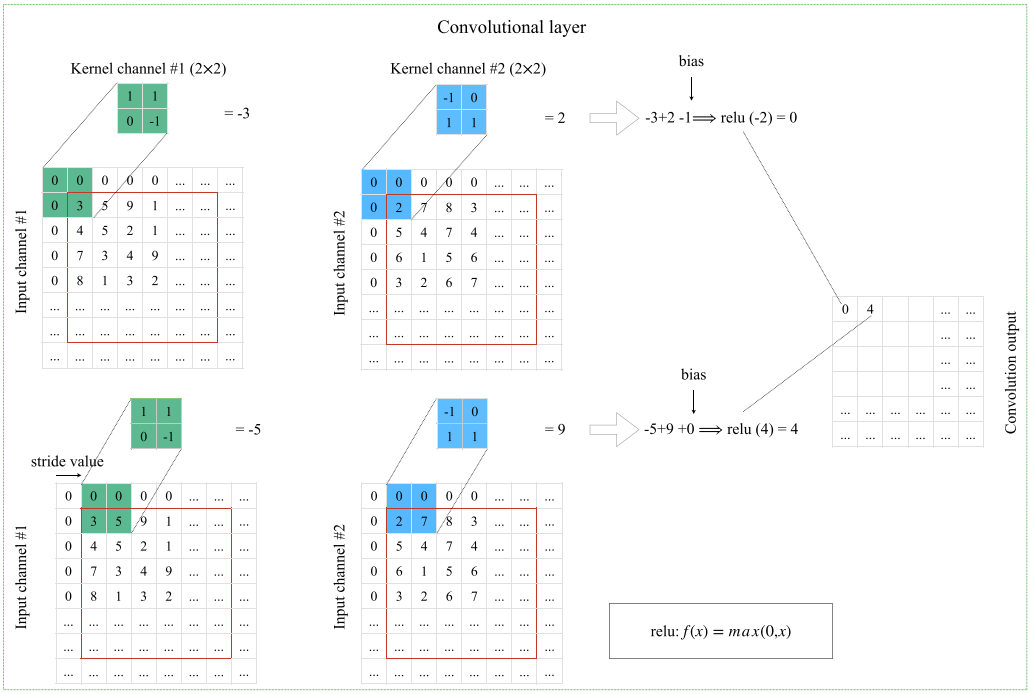

Figure 3. Simplified sketch of the convolutional layer functioning in the case of two channels ($iD = 2$), $F = 1$ kernel with dimension $K = 2$, relu as activation function, padding applied just to a border 1-cell wide ($p = 1$), and $s = 1$. The red rectangle denotes the original 6×6 matrix.

the maximum pooling is moved across the entire matrix, thus producing as output a matrix with size 3×3. The same is done for the average pooling.

It must be considered that, in general and especially to achieve the computational effort reduction, the stride value for pooling layers is equal to the size of the pooling kernel and padding is not used. However, it is also possible to use stride and padding values to keep the image size unchanged after the pooling operation. The use of two different pooling layers allows avoiding the loss of potentially important information. After these layers, typical of CNNs, other layers are used, usually a flatten layer that "flattens" the dimension of the previous layers. For instance, in the example of Figure 4, the two matrices that are the output of the two pooling layers are piled up and their elements concatenated to create a new vector (h_t^0) that will be the input for the fully connected part consisting of the dense layers.

The ANN model based on CNN, implemented for *SYM-H* index forecasting, consists of a convolutional block and a fully connected part and follows the scheme of Figures 3 and 4. The input of the network X is formed by the values of the H input parameters at l_b times of the input time series, so it would have a dimension of $l_b \times 1 \times H$. However, since we are feeding the network with all the samples, X has actually the dimension $N \times l_b \times 1 \times H$, where N is the number of samples.

The convolutional block built for *SYM-H* forecasting is composed, in sequence, of one convolutional layer that receives in input the values of the time series at l_b times, two different pooling layers (one average pooling layer and one maximum pooling layer), and one flatten layer that receives the concatenated output of the two pooling layers. This last layer "flattens" the dimensions $l_b \times 1 \times (F \cdot 2)$ of the obtained series to obtain an array with a dimension of $l_b \cdot 1 \cdot F \cdot 2$ that can be managed by the fully connected part. This last part coincides in structure with the dense part of the ANN model based on LSTM network, consisting of four dense layers.

As well as for the model based on LSTM, also in this case, the model forecasts the value y_{t+l_f} from the input consisting of values of the input parameters from $t - l_b$ to t.

In the case of CNN, the adjustable parameters that have been tuned for a better performance of the network are the activation functions (ϕ_C and ϕ_D), padding (p_C), number of filters (F), number of neurons (n_D), kernel and pool size (K_C and p_P, respectively), stride values (s_C and s_P), and look-back (l_b), where the subscript C

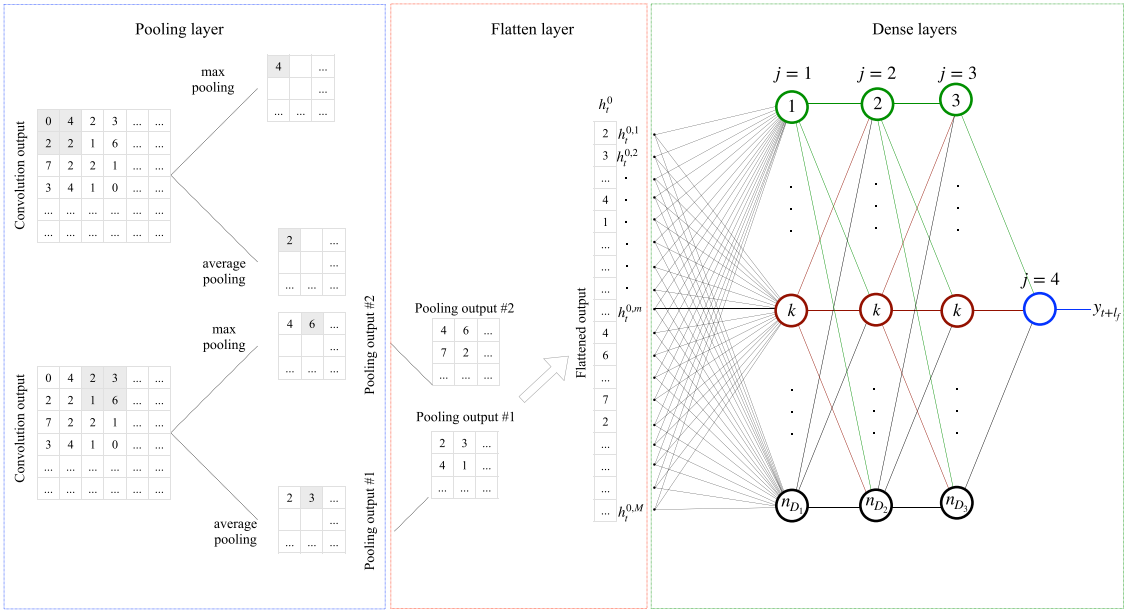

Figure 4. Simplified sketch of the pooling, flatten, and dense layers functioning. Colored circles in the dense layers represent the layer neurons.

stands for convolutional, D for dense, and P for pooling. Concerning the activation function, similarly to what was done for the ANN model based on the LSTM network, we evaluated tanh and relu functions testing all possible combinations for the two blocks. Performance comparison has shown that the best activation function to use is relu for both the convolutional and dense layers. Concerning the number of filters for the convolutional block and the number of neurons for the dense block, we have tested values of 8, 16, 32, and 64, obtaining the best performance with $F = n_D = 32$. Concerning kernel size for convolutional layers and pool size for pooling layers, we have tested values of 3, 5, and 7, obtaining the best performance with $K_C = p_P = 5$. Stride value has been changed in relation to the size of the kernels and pools, so that it results $s_C = \left\lfloor \frac{K_C}{2} \right\rfloor$ and $s_P = \left\lfloor \frac{p_P}{2} \right\rfloor$. In this way, the models have been tested for values of both s_C and s_P equal to 1, 2, and 3, obtaining best results with $s_C = s_P = 2$.

The look-forward l_f is set to a value corresponding to 60 min, while the look-back parameter l_b is tested for values corresponding to 90, 120, 180, 360, 540, 720, 1,080, 1,440, 1,800, 2,160, 2,520, and 2,880 min, that have to be divided by the time sampling. The look-back values over 1,440 min are tested only for the network that does not use *SYM-H* index as input. The choice to set l_f equal to 60 min and hence obtaining forecasts 1 h ahead derives from the fact that this time corresponds approximately to that needed by the solar wind to propagate from 1 AU to the Earth and also that needed for the directly driven response of the magnetosphere to manifest (see e.g., Alberti et al., 2017). Moreover, while a prediction for values of $l_f < 60$ min would make no sense for alerting purposes, it would be worth exploring values of $l_f > 60$ min. However, so far, we prefer to focus on a single value of l_f.

2.3. Performance Evaluation

To evaluate the performance of the two ANN models used to forecast *SYM-H*, we estimate the root mean squared error, *RMSE*, and the so-called coefficient of determination R^2 that are defined as follows:

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{9}$$

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{10}$$

where y represents the observed values, $\bar{y}$ their mean, $\hat{y}$ represents the predicted values, and N is the sample size.

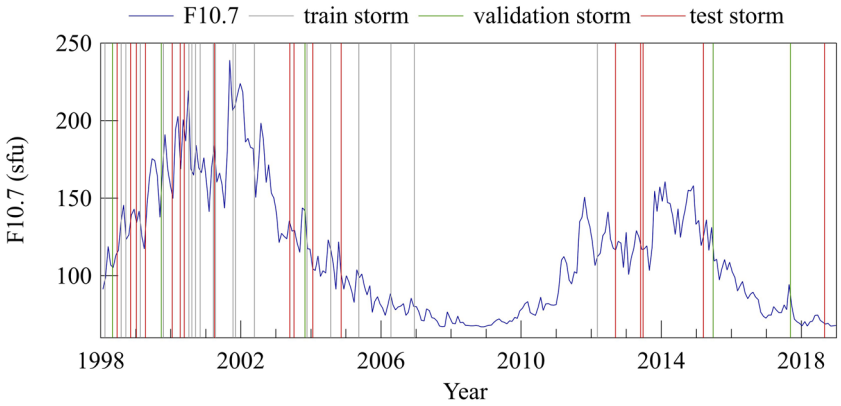

Figure 5. Solar cycle as represented by 27-day averages of F10.7 index (blue). Vertical lines indicate the times of the storms used to train (gray), validate (green), and test (red) the ANN models.

RMSE allows comparing our results with those found by other authors, although attention should be paid to the fact that *RMSE* depends on the scale of data used. So, for models with the same predictive ability, it can provide different values when forecasting storms of different intensities. R^2, known as coefficient of determination, represents the amount of variance of the actual data that is explained by the predicted data. It is a very simple indicator, which usually varies between 0 and 1, although it can assume negative values. Values of $R^2 = 1$ mean that the model is perfect, while values of $R^2 = 0$ mean that using the model is equivalent to forecasting the average of the dependent variable. Values of $R^2 < 0$ indicate that the model performs worse than predicting the average of the dependent variable.

Unlike many other studies on magnetic indices forecasting, which use the correlation coefficient, we have chosen to use the metrics described above to calculate the goodness of our models because they are better suited to evaluate prediction models. In fact, the correlation coefficient does not take into account the residuals between the forecast and the real data but only the correlation between the two. Therefore, a model that completely underestimates/overestimates the real data but that follows perfectly its fluctuations could provide $R = 1$, that is perfect, while the prediction is not.

3. Data: Description and Pre-Processing

The database used to train, validate, and test the developed ANN models consists of 42 geomagnetic storms that occurred between 1998 and 2018. These are indicated in Figure 5; Table 1 lists the most relevant information on the database. In detail, the columns display (from left to right) the number assigned to the storm to identify it, the start and end date of selected data (hours are not specified since whole days of data are used), *SYM-H* minimum value to provide an indication of the intensity of the storm, and the occurrence (Y) or not (N) of a multiple-dip (MP) geomagnetic storm (i.e., the presence of multiple depressions in *SYM-H* index) to give an idea of the storm complexity.

Data used consist of 5-min averages of (1) IMF B_z component and squared values of IMF magnitude B and of B_y component, in GSM coordinates recorded at L1 from ACE satellite and (2) *SYM-H* index. Both kinds of data have been downloaded online (https://cdaweb.gsfc.nasa.gov/cgi-bin/eval1.cgi), but while *SYM-H* index values have been retrieved directly in the desired sampling (OMNI_HRO_5MIN), 5-min averages of IMF have been computed from ACE magnetic field 1-s level 2 data (AC_H3_MFI). The choice to use a resolution of 5 min is motivated by both the decrease of computation time that is a fundamental parameter for a real-time forecasting and the removal of the noise present in 1-min resolution data. Moreover, we recall that *SYM-H* index is characterized by a precision of 1 nT and can therefore assume only integer values: the use of this lower resolution for *SYM-H* provides the advantage to work with an index whose values span into a more continuous, less quantized, range. Considering the few gaps present in our database, we collected a total of 118,316 samples. Linear interpolation is used to fill gaps encountered in IMF data, in most cases of only few consecutive seconds.

Other authors have performed their analysis on a larger number of storms (e.g., Bhaskar & Vichare, 2019; Cai et al., 2010). We limit our analysis to 42 geomagnetic storms. Further expanding the database would have

Table 1
Details of the Storms Used to Develop the ANN Model

Train subset				
Storm N.	Start date	End date	$SYM\text{-}H_{min}$ (nT)	MP
1	14/02/1998	22/02/1998	−119	Y
2	02/08/1998	08/08/1998	−168	Y
3	19/09/1998	29/09/1998	−213	N
4	16/02/1999	24/02/1999	−127	Y
5	15/10/1999	25/10/1999	−218	N
6	09/07/2000	19/07/2000	−347	N
7	06/08/2000	16/08/2000	−235	Y
8	15/09/2000	25/09/2000	−196	Y
9	01/11/2000	15/11/2000	−174	Y
10	14/03/2001	24/03/2001	−165	Y
11	06/04/2001	16/04/2001	−275	N
12	17/10/2001	22/10/2001	−210	N
13	31/10/2001	10/11/2001	−320	N
14	17/05/2002	27/05/2002	−116	Y
15	15/11/2003	25/11/2003	−490	N
16	20/07/2004	30/07/2004	−208	Y
17	10/05/2005	20/05/2005	−302	N
18	09/04/2006	19/04/2006	−110	N
19	09/12/2006	19/12/2006	−211	N
20	01/03/2012	11/03/2012	−149	Y
Validation subset				
Storm N.	Start Date	End Date	$SYM\text{-}H_{min}$ (nT)	MP
21	28/04/1998	08/05/1998	−268	N
22	19/09/1999	26/09/1999	−160	N
23	25/10/2003	03/11/2003	−432	Y
24	18/06/2015	28/06/2015	−207	Y
25	01/09/2017	11/09/2017	−146	Y
Test subset				
Storm N.	Start Date	End Date	$SYM\text{-}H_{min}$ (nT)	MP
26	22/06/1998	30/06/1998	−120	N
27	02/11/1998	12/11/1998	−179	Y
28	09/01/1999	18/01/1999	−111	N
29	13/04/1999	19/04/1999	−122	N
30	16/01/2000	26/01/2000	−101	Y
31	02/04/2000	12/04/2000	−315	N
32	19/05/2000	28/05/2000	−159	Y
33	26/03/2001	04/04/2001	−437	N
34	26/05/2003	06/06/2003	−162	Y
35	08/07/2003	18/07/2003	−125	Y
36	18/01/2004	27/01/2004	−137	Y
37	04/11/2004	14/11/2004	−394	Y
38	10/09/2012	05/10/2012	−138	N
39	28/05/2013	04/06/2013	−134	N
40	26/06/2013	04/07/2013	−110	N

Table 1 *(continued)*

Test subset				
Storm N.	Start date	End date	$SYM\text{-}H_{min}$ (nT)	MP
41	11/03/2015	21/03/2015	−234	N
42	22/08/2018	03/09/2018	−205	N

Note. From left to right: number assigned to the storm, start and end time of selected data (hours are not specified since whole days of data are used), *SYM-H* minimum value, and occurrence (Y) or not (N) of a multiple-dip (MP) geomagnetic storm. From top to bottom: train, validation, and test subsets.

meant just to increase the number of weak and moderate geomagnetic storms with respect to the number of severe and extreme ones. We selected all the geomagnetic storms with available ACE IMF data characterized by a minimum value of *SYM-H* index less than −200 nT (19 out of 42 geomagnetic storms). The remaining 23 geomagnetic storms are characterized by minimum values of *SYM-H* between −200 and −100 nT. For most of the considered storms, the length of the time interval is of 10 days, apart from a few exceptions (refer to Table 1 for details). The use of long time intervals allows training, validating, and testing the network not only on main phase periods but also on quiet and recovery phase periods.

Selected storms listed in Table 1 are divided into three subsets according to the role that each storm has played in the ANN model implementation. From top to bottom, the train (48% of 42, i.e., 20 storms), validation (12% of 42, i.e., five storms), and test (40% of 42, i.e., 17 storms) subsets are listed. The only criteria used to assign geomagnetic storms to each set have been those to uniformly populate the three sets in terms of geomagnetic storm intensity and multiple-dip occurrence. Training data are used to train the ANN models defining weights matrices and biases, the validation data are used to stop the network training and prevent overfitting, while the test data are used to evaluate the performance of the considered models.

To help the model converge faster and achieve better results, time series are also standardized using the associated mean and standard deviation as follows:

$$Z = \frac{X - \mu_X}{\sigma_X} \tag{11}$$

where Z is the standardized data, X is the original time series, and μ_X and σ_X are the associated mean and standard deviation, respectively. Standardization has been preferred to normalization between [−1,1] or [0,1] in order not to compress too much the data in a small range. Most of the time series we considered are characterized by long periods of quiet conditions (*SYM-H* greater than −50 nT) and by sudden peaks of activity when *SYM-H* index reaches values even less than −400 nT. Thus, using the normalization in a specific range, as for instance $[0, 1]$, more than 80% of the data would have been compressed in a very small range, approximately $[0, 0.2]$. Moreover, in case of normalization between 0 and 1, the *sigmoid* function is the one that makes more sense to use. This would have meant that predicted values would have taken values only between the minimum and the maximum values of the starting data set. In practice, forecasting new geomagnetic storms with values of *SYM-H* index going beyond the range defined by the storms used to train, validate, and test the model would result in a sure underestimation of the corresponding absolute values. Differently, standardization and the use of a linear output activation function guarantees that the generated output values cover the whole $\mathbb{R}$ space.

4. Discussion of Results

The first step in *SYM-H* forecasting consists of training the two ANN models. Training is achieved via back propagation and gradient descent using Adam (short name for adaptive moment estimation) algorithm (Kingma & Ba, 2014) and mean squared error as loss metric. This is done using data from the geomagnetic storms listed in Table 1. To avoid overfitting the train data, a small set of geomagnetic storms is used for validation, that is, they are not used directly to drive the gradient descent, but only to decide when to stop the training. Specifically, at the end of each epoch, the loss on validation is calculated, and if for 50 epochs it has not decreased, the computation is stopped and the network configuration of 50 epochs before is recovered (i.e., the best configuration obtained as to the loss on validation). ANN models are trained both without and with *SYM-H* among the input parameters to investigate how much the prediction changes when only IMF data are used.

Table 2
Root Mean Squared Errors (RMSEs) and R^2 Obtained for Geomagnetic Storms in the Test Subset Without Using SYM-H Index for Prediction Among the Input Parameters

	LSTM		CNN	
Storm N.	RMSE (nT)	R^2	RMSE (nT)	R^2
26	18.0	0.28	19.8	0.13
27	16.8	0.79	23.4	0.58
28	18.6	0.49	14.4	0.70
29	21.1	0.51	20.0	0.56
30	24.2	0.11	25.8	–0.01
31	32.5	0.66	32.1	0.67
32	23.4	0.63	18.9	0.76
33	33.8	0.85	26.7	0.91
34	17.9	0.42	16.6	0.50
35	21.3	0.34	18.6	0.50
36	20.4	0.41	21.4	0.35
37	42.6	0.73	36.9	0.80
38	18.6	–0.15	13.0	0.44
39	20.3	0.56	16.5	0.71
40	13.6	0.74	9.2	0.88
41	27.3	0.69	25.4	0.73
42	17.8	0.73	16.7	0.76
Total data set	23.6	0.73	21.4	0.78

Results for all the test storms and both ANN models (LSTM and CNN) are summarized in Tables 2 and 3. These tables list, for each test storm, the RMSE and R^2 for the best forecast among 20 simulations with random weights initialization, obtained without (Table 2) and with (Table 3) *SYM-H* index among the input parameters. Note that the value referring to the total data set reported in these tables is not an arithmetical average, but it is calculated on all the data points (all storms) at once.

Figure 6 displays the results for four test storms with different intensity, that is, characterized by a minimum *SYM-H* around –400, –300, –200, and –100 nT and different structure. Results for all the 17 test storms can be found in Figures S1 to S17 of the supporting information. Figure 6 contains a panel for each geomagnetic storm, ordered by the storm date. In detail, panel (a) refers to storm N. 27 (data from 2 to 12 November, 1998), panel (b) to storm N. 31 (data from 2 to 12 April, 2000), panel (c) to storm N. 37 (data from 4 to 14 November, 2004), and panel (d) to storm N. 38 (data from 10 September to 5 October, 2012). For each panel, plots on the left side refer to predictions made by the LSTM model (green) and plots on the right side to those made by the CNN model (red). The top row displays the observed *SYM-H* index (gray) and the predicted one when relying only on IMF measurements, while the bottom row refers to predictions made using *SYM-H* index among the input parameters.

Figure 6c displays the worst prediction, in terms of RMSE, and one of the best, in terms of R^2, among test storms. The storm that occurred in November 2004 is one of the most intense and complex among the 42 selected. It is characterized by multiple negative depressions in the observed *SYM-H* index displaying one first dip reaching about –400 nT and a second one of about –300 nT. Indeed, forecasting "nonstandard" geomagnetic storms is quite a challenging task for the ANN models here proposed since, between one dip and the next, *SYM-H* index does not return to its quiet state. We recall that we do not include in the input parameters those describing solar wind plasma, so it is a lot more complicated to assess the influence of IMF alone on a magnetosphere whose status is already disturbed.

If we consider both validation and test storms, the worst prediction is that obtained for storm N. 23, occurring on October 2003. RMSEs obtained for this storm, without using *SYM-H* among the input parameters, are of 45.5 ($R^2 = 0.75$) and 57.1 nT ($R^2 = 0.61$) for LSTM and CNN models, respectively. RMSEs obtained for this storm considering *SYM-H* among the input parameters definitely improve, being of 24.6 ($R^2 = 0.92$) and

Table 3
Root Mean Squared Errors (RMSEs) and R^2 Obtained for Geomagnetic Storms in the Test Subset Using SYM-H Index for Prediction Among the Input Parameters

	LSTM		CNN	
Storm N.	RMSE (nT)	R^2	RMSE (nT)	R^2
26	6.7	0.89	7.2	0.87
27	8.9	0.94	10.5	0.92
28	5.4	0.95	5.6	0.95
29	7.2	0.93	7.7	0.92
30	5.6	0.95	6.5	0.93
31	10.7	0.96	9.6	0.97
32	8.3	0.95	8.2	0.95
33	16.3	0.96	19.1	0.95
34	11.3	0.75	12.4	0.70
35	8.5	0.90	8.8	0.89
36	8.7	0.89	10.5	0.84
37	17.5	0.96	17.3	0.96
38	4.2	0.94	4.6	0.93
39	5.6	0.96	6.8	0.94
40	5.5	0.95	5.9	0.95
41	9.0	0.96	9.4	0.96
42	5.9	0.97	6.3	0.96
Total data set	9.0	0.96	9.7	0.95

26.3 nT ($R^2 = 0.91$), again for LSTM and CNN models, respectively. The remaining validation storms are all characterized by RMSE values that in both cases (without and with *SYM-H* index) are on average far lower than those obtained for the October 2003 geomagnetic storm.

Figure 6d displays the forecast obtained for a moderate storm. In this case, the data set includes also a long interval (approximately 3 weeks) characterized by quiet conditions before the storm onset. This has the purpose to highlight the behavior of the two models also under quiet conditions. Indeed, a feature of quiet periods predictions obtained using IMF data only is that the discrepancy between observed and predicted *SYM-H* values is larger than during disturbed periods. This feature is evident in all the storms displayed in Figure 6, but best emerges in panel (d), where a long period of quiet data is considered. During quiet periods, the model tends to predict values of *SYM-H* around ∼−25 nT probably because it considers this value as the most indicative for an undisturbed magnetosphere.

On average, looking at the values summarized in Tables 2 and 3, the CNN model seems to perform slightly better than LSTM when not resorting to the knowledge of previous values of *SYM-H* index. Indeed, CNN average RMSE is of 21.4 nT with a value of $R^2 = 0.78$ against an RMSE of 23.6 nT with $R^2 = 0.73$ for the LSTM model. Differently, when *SYM-H* index is considered among the inputs, the situation is reversed with the LSTM model behaving slightly better than the CNN one in terms of the selected error metrics. If, however, we look at the skill of the two models in predicting peculiar characteristics of *SYM-H*index, we realize that the CNN model shows a higher ability in catching also storm sudden commencements and sudden impulses than LSTM model, as displayed in Figures S1 to S17. To better focus on the performances of the developed models in predicting the peak intensity of the storms, in Figure 7, we show the scatter plots of the predicted relative minima of *SYM-H*index as a function of the corresponding observed values for both LSTM (green squares) and CNN (red circles) models. In detail, panel (a) of Figure 7 displays what is obtained for predictions made without *SYM-H* index among the input parameters while panel (b) for predictions made using *SYM-H* index among the input parameters. It is worth here clarifying that we do not consider only one peak for geomagnetic storm, that is, that characterized by the lowest value reached by *SYM-H* index, but we consider also multiple peaks for the most complicated geomagnetic storms. This is the reason why the number of points in Figure 7 is larger than 17, that is, the number of test storms. Figure 7 shows that our models

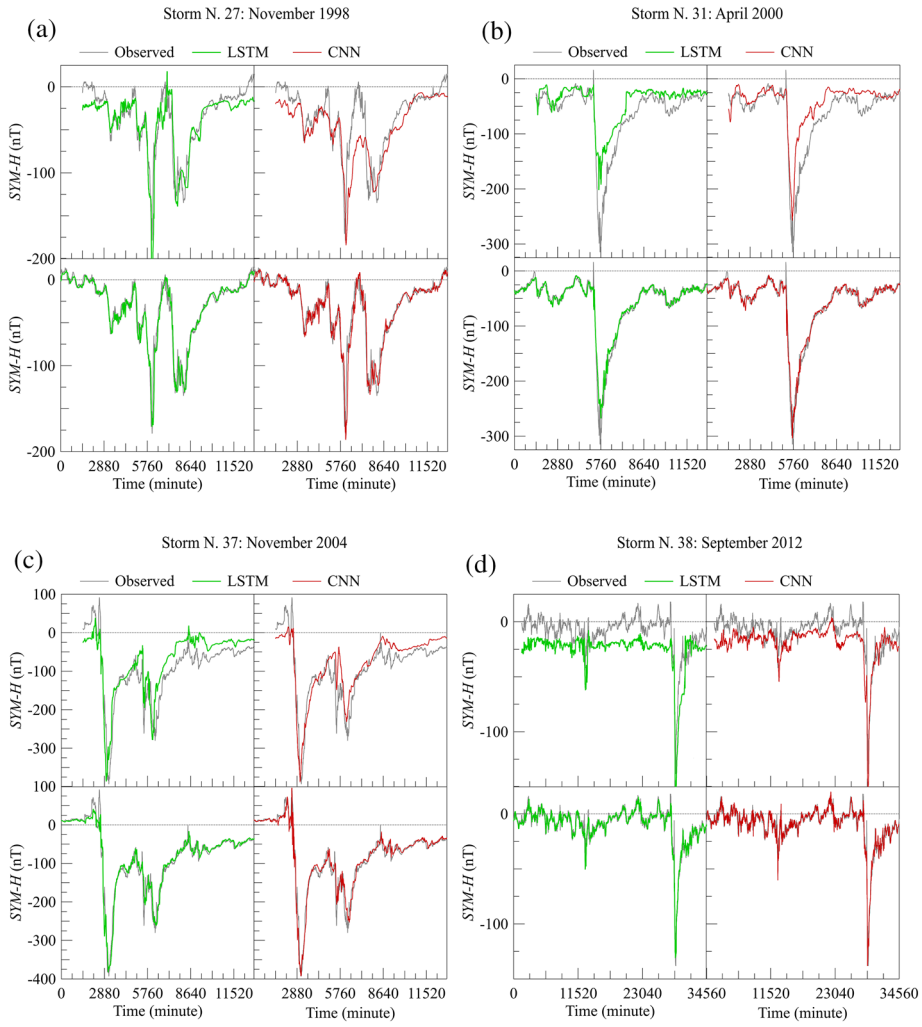

Figure 6. Observed and predicted *SYM-H* index in the case of four test geomagnetic storms: (a) storm N. 27 (data from 2 to 12 November), (b) storm N. 31 (data from 2 to 12 April), (c) storm N. 37 (data from 4 to 14 November), and (d) storm N. 38 (data from 10 September to 5 October). Plots in each panel correspond to LSTM (top left) and CNN (top right) prediction without *SYM-H* index among the input parameters and LSTM (bottom left) and CNN (bottom right) prediction with *SYM-H* index among the input parameters.

tend to underestimate the peak intensity with a performance that improves with decreasing peak intensity. As expected when using *SYM-H* index among the input parameters, the discrepancy between predictions and observations diminishes. A possible explanation of this could be given in terms of the composition of the train subset. Indeed, out of 20 train storms, nine are characterized by $-200\,\text{nT} < SYM\text{-}H_{min} < -100\,\text{nT}$ while the minimum *SYM-H* of the remaining 11 ranges approximately between −200 and −500 nT. In this way, the network has an experience on values lower than −200 nT that is more limited than that on values higher than −200 nT.

To quantify the average discrepancy between predicted and observed peaks amplitude for moderate ($-200\,\text{nT} < SYM\text{-}H_{min} < -100\,\text{nT}$) and severe ($SYM - H_{min} < -200\,\text{nT}$) storms, we have estimated the RMSE for the peaks plotted in Figure 7 for these two intensity categories. For moderate intensity, we find that the RMSE for LSTM and CNN models is comparable; in detail, it is ∼7–8 nT when *SYM-H* index is not among the input parameters, decreasing to ∼5–6 nT when *SYM-H* index is considered among the inputs. For severe intensity, we find that the values of RMSE is ∼38 nT for the LSTM model and ∼30 nT for the CNN model when not using *SYM-H* as input and ∼20–21 nT when considering it as an input.

Some interesting considerations can be made on the role of *SYM-H* index as input parameter. Our results evidence the substantial difference in the accuracy between predictions obtained using or not using *SYM-H*

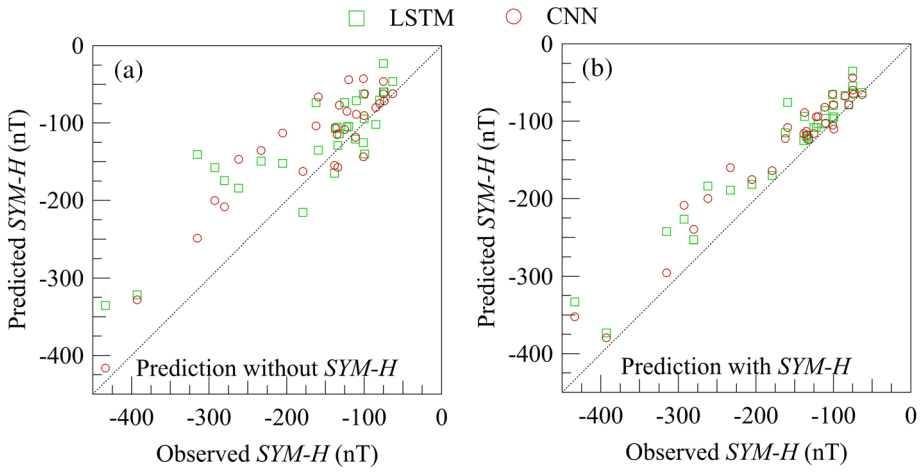

Figure 7. Scatter plots of the predicted relative minima of *SYM-H* index as a function of the corresponding observed values for both LSTM (green squares) and CNN (red circles) models. Panel (a) displays results for predictions made without *SYM-H* index among the input parameters and panel (b) those for predictions made using *SYM-H* index among the input parameters.

index as an input, latter predictions being much worse than the former. This can be explained in terms of the lack of information in IMF measurements about the magnetospheric internal dynamical state, which is instead included in *SYM-H* index data. Moreover, the model that receives only IMF data has more difficulty in predicting the storm recovery phase once it exits the look-back interval. This is very clear in the top plots of the panels shown in Figure 6, obtained using a value of l_b corresponding to 1,800 min. In these plots, especially in panels (a) and (b), we observe a sort of plateau after the main phase of the storm, missing the recovery phase. This can be explained as follows: when predicting the recovery phase, if look-back is such not to include data describing the main phase, the neural network behaves as if back in the quiet state, thus generating this sudden return of *SYM-H* index to pre-storm values. It is worth mentioning that in both cases of without and with *SYM-H* as input parameter, the ANN models are not able, in the quiet state, to predict the fluctuations of the index, limiting themselves to reproduce the received input data.

To better explain the role of look-back in the ANN models performance, Figures 8 and 9 display R^2 as a function of look-back, in both the cases of without and with *SYM-H* index as input, respectively. The shaded areas (green for the LSTM model and red for the CNN model) indicate the 95% confidence interval computed around the average of R^2 obtained from 20 tests performed for each look-back. Panel (a) of both figures displays R^2 estimated using all data points, while panel (b) displays R^2 estimated using only points falling in the half-peak interval. We defined this time interval as that around a relative minimum of *SYM-H* index whose extremes are the values of time, before and after this minimum, when *SYM-H* is equal to half of the minimum itself. This allows estimating the performance of the models around the peak intensity of the geomagnetic storm. Since the initialization of the weights of the neural network is random, performing multiple tests allows obtaining a more precise estimate of the error made depending on look-back.

Figure 8 shows that R^2 increases with look-back when *SYM-H* index is not used as input, while it decreases when *SYM-H* index is used (Figure 9). This latter result means that the knowledge of previous values of *SYM-H* allows obtaining the best prediction for the minimum used value of look-back; all further information seems to confuse the network. In Figure 9, the behavior of the persistence model is displayed for comparison. We recall that the persistence model is a simple prediction algorithm for which prediction at time $t+1$ is given by the value at time t. In our case, it would be equivalent to take the value of *SYM-H* index at time t as the prediction 1 h ahead. This model therefore does not have a real predictive ability but can be used as a benchmark and tells whether it is worth using the tested models. Figure 9 shows that the performance of the persistence model is always less than that of LSTM model and comparable with that of CNN model for values of look-back larger than 120 and 540 depending on the data set used (all points or half-peak, respectively). Moreover, when considering only points around peak intensity, the performance of the proposed models is definitely better than that of the persistence model, meaning that the models are

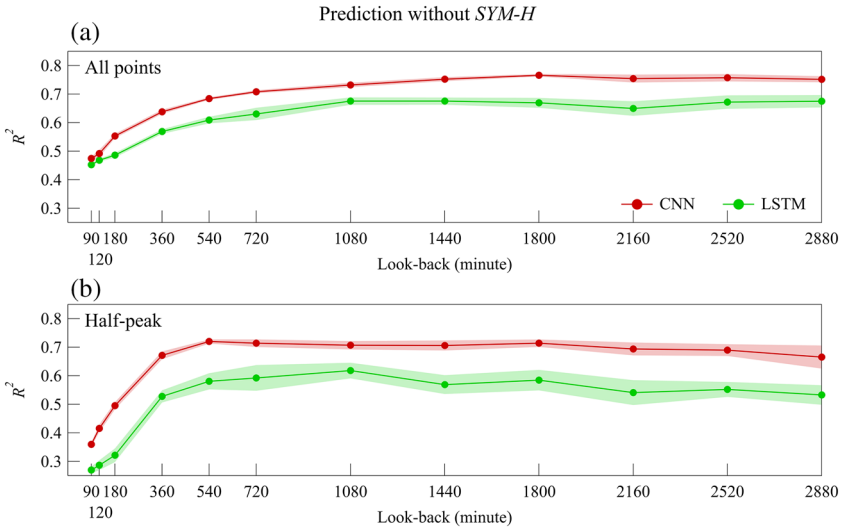

Figure 8. Prediction without using *SYM-H* index among the input parameters for (a) all data and (b) data in the half-peak interval. R^2 for different look-back values for both CNN and LSTM ANN models.

cleverer at predicting disturbed periods than quiet periods. Indeed, when estimating R^2 using points corresponding to quiet periods only, the performance of the persistence model by far exceeds that of the CNN model and equals that of the LSTM one (data not shown). By the way, the increase observed in Figure 8a is the reason why, only for the case of predictions made without *SYM-H*, we extended the investigated values of look-back going beyond the value we initially set as maximum look-back, that is, that corresponding to 1,440 min. At $l_b = 1{,}440$ min, the curve is still clearly increasing in the CNN case; it was therefore necessary to explore what happened beyond 1,440 min.

Figure 8 also confirms that when using only IMF measurements, the CNN model performs definitely better than LSTM, with the best performance obtained for $l_b = 1{,}800$ min when all points are considered (panel a) and for $l_b = 540$ min when only half-peak intervals are considered (panel b). Differently, when also *SYM-H* is used, the two models reach their (similar) highest value of R^2 for $l_b = 90$ min, afterwards prediction degrades, stabilizing from $l_b = 720$ min onwards. We observe also that CNN performance decreases faster as the look-back increases than it does for the LSTM; this happens because LSTM receives input data one at a

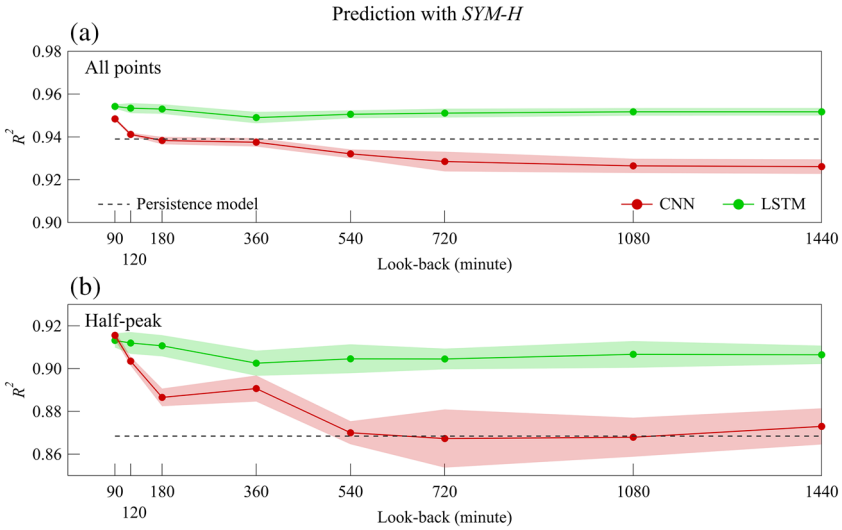

Figure 9. Prediction using *SYM-H* index among the input parameters for (a) all data and (b) data in the half-peak interval. R^2 for different look-back values for both CNN and LSTM ANN models and persistence model (dashed line).

time, so instants of time far from the output have a weak effect on the prediction. For CNN, which receives the data all at once, the exact opposite happens.

On the whole, our results suggest that the ANN models here presented, based on LSTM and CNNs, are able to well forecast a high-resolution index like *SYM-H* filling the gap, for instance, of the Elman network. A physical explanation could be that when switching to a higher time resolution forecast, that is, moving from D_{st} index to *SYM-H* index, the Elman network is no longer able to predict with sufficient accuracy, especially in the storm recovery phase, the changes in the index. This could be due to the inherent dual nature of the magnetospheric dynamics in response to interplanetary changes. As well documented in a series of recent papers (see, e.g., Alberti et al., 2017, 2018; Consolini et al., 2018), the magnetospheric dynamics is characterized by both directly driven processes and externally triggered internal ones. These two categories of processes are characterized by different timescales; furthermore, internal processes depend on the magnetospheric internal dynamical state. Thus, Elman-type networks seem to be unable to correctly reproduce the internal magnetospheric dynamics, which generally affects the short timescale fluctuations of the geomagnetic indices. Our idea is that IMF data do not contain the necessary information to predict these fluctuations, and additional data, such as solar wind data, are needed. Further studies on this side could be carried out.

To compare our results with previous studies on *SYM-H* forecasting, we refer to Cai et al. (2010) and Bhaskar and Vichare (2019), with each of whom we share a test storm. For instance, with Cai et al. (2010), we share the April 2000 storm; the difference is that they use 4 days only, while we use 11 days. In order to compare our RMSE with theirs, we computed it on their same interval. We obtain an RMSE of 39.93 (LSTM) and 42.85 nT (CNN) using only IMF data as input. RMSEs obtained for predictions made using *SYM-H* are of 14.82 (LSTM) and 12.72 nT (CNN). Using also solar wind data, that is, proton density of solar wind velocity, they obtained an RMSE of 19.34 nT. With Bhaskar and Vichare (2019), we share the storm of March 2015. In this case, unfortunately, they use more days than we do (a few days more during the quiet period before the storm onset), making a rigorous comparison not possible. In any case, they get an RMSE of 21.34 nT, against ours of 27.3 (LSTM) and 25.4 nT (CNN) without *SYM-H* and of 9.0 (LSTM) and 9.4 nT (CNN) with *SYM-H* as input. As done by Cai et al. (2010), Bhaskar and Vichare (2019) also add solar wind data as model input.

5. Conclusions

In this work, we built two ANN models to predict 5-min averages of *SYM-H* index 1 h in advance, that is the time strictly necessary to issue an alert. Differently from the bulk of models for geomagnetic indices forecasting that rely on the aid of solar wind parameters such as density and velocity, we used only ACE IMF data (B^2, B_y^2, and B_z). Certainly, the use of plasma data allows achieving better results than ours, but on the other hand, there is a high probability of missing data during severe space weather events. For this reason, models not using plasma data should be preferable when implementing operational tools for alerting the arrival of geomagnetic storms with all the correlated damages.

The ANN models presented in this work are based on two neural networks, completely different from a conceptual point of view: LSTM and CNN. To our knowledge, this is the first attempt to forecast a geomagnetic index using a CNN. We have found that in terms of RMSE and R^2, the CNN model performs better than the LSTM when *SYM-H* index is not used as input parameter, the opposite occurring when *SYM-H* index is used. In this case, however, CNN seems to have a higher ability than LSTM to reproduce features as sudden storm commencements or sudden impulses. This suggests that convolutional networks can obtain results similar or superior to recursive networks, such as the LSTM, and evidences also the interesting potentialities of CNNs even for purposes by far different from those they have been thought for.

Another point that is worth mentioning concerns the approach used to choose the structure of the developed ANN models. Although we carried out an optimization of some of the hyperparameters of the two models, we decided to start from fairly simple basic structures. A future improvement on this side may involve the investigation of more complex neural network structures, with the help of neuroevolution techniques, in order to create structures more suitable for the purpose of geomagnetic indices forecasting. Based on the promising results provided by the ANN models here proposed, as a future work, we could deal with the forecasting of other indices that play a crucial role in GICs, for instance, the AL index to account for substorms (Freeman et al., 2019), and attempt the leap from the forecasting of geomagnetic indices to that of the

ground geomagnetic field and of its derivative. This could play a significant role in the mitigation of ground effects of space weather events.

Data Availability Statement

F10.7 data were obtained from the GSFC/SPDF OMNIWeb interface (https://omniweb.gsfc.nasa.gov).

Acknowledgments
The authors 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. Acknowledgments are also due to 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. The authors also thank Professor Stefano Leonardi for some useful discussion. G. C., R. T., F. G., and P. D. M. acknowledge financial support from the Italian MIUR-PRIN grant 2017APKP7T on "Circumterrestrial Environment: Impact of Sun-Earth Interaction." The authors acknowledge also two anonymous referees who, with their constructive comments, have contributed to improve the manuscript.

References

Alberti, T., Consolini, G., De Michelis, P., Laurenza, M., & Marcucci, M. F. (2018). On fast and slow Earth's magnetospheric dynamics during geomagnetic storms: A stochastic Langevin approach. *Journal of Space Weather and Space Climate*, *8*, A56. https://doi.org/10.1051/swsc/2018039

Alberti, T., Consolini, G., Lepreti, F., Laurenza, M., Vecchio, A., & Carbone, V. (2017). Timescale separation in the solar wind-magnetosphere coupling during St. Patrick's Day storms in 2013 and 2015. *Journal of Geophysical Research: Space Physics*, *122*, 4266–4283. https://doi.org/10.1002/2016JA023175

Amata, E., Pallocchia, G., Consolini, G., Marcucci, M. F., & Bertello, I. (2008). Comparison between three algorithms for D_{st} predictions over the 2003 2005 period. *Journal of Atmospheric and Solar-Terrestrial Physics*, *70*(2–4), 496–502. https://doi.org/10.1016/j.jastp.2007.08.041

Bala, R., & Reiff, P. (2012). Improvements in short-term forecasting of geomagnetic activity. *Space Weather*, *10*, S06001. https://doi.org/10.1029/2012SW000779

Bhaskar, A., & Vichare, G. (2019). Forecasting of SYM-H and ASY-H 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

Boynton, R. J., Balikhin, M. A., Billings, S. A., Sharma, A. S., & Amariutei, O. A. (2011). Data derived NARMAX D_{st} model. *Annales Geophysicae*, *29*(6), 965–971. https://doi.org/10.5194/angeo-29-965-2011

Burton, R. K., McPherron, R. L., & Russell, C. T. (1975). An empirical relationship between interplanetary conditions and D_{st}. *Journal of Geophysical Research*, *80*(31), 4204–4214. 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. 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*, 1166–1207. https://doi.org/10.1029/2018SW002061

Carter, B. A., Yizengaw, E., Pradipta, R., Weygand, J. M., Piersanti, M., Pulkkinen, A., et al. (2016). Geomagnetically induced currents around the world during the 17 March 2015 storm. *Journal of Geophysical Research: Space Physics*, *121*, 10,496–10,507. https://doi.org/10.1002/2016JA023344

Chandorkar, M., Camporeale, E., & Wing, S. (2017). Probabilistic forecasting of the disturbance storm time index: An autoregressive Gaussian process approach. *Space Weather*, *15*, 1004–1019. https://doi.org/10.1002/2017SW001627

Consolini, G., Alberti, T., & De Michelis, P. (2018). On the forecast horizon of magnetospheric dynamics: A scale-to-scale approach. *Journal of Geophysical Research: Space Physics*, *123*, 9065–9077. https://doi.org/10.1029/2018JA025952

Consolini, G., & Chang, T. S. (2001). Magnetic field topology and criticality in geotail dynamics: Relevance to substorm phenomena. *Space Science Reviews*, *95*(1–2), 309–321. https://doi.org/10.1023/A:1005252807049

Costello, K. A. (1998). Moving the rice MSFM into a real-time forecast mode using solar wind driven forecast modules (PhD Thesis). Rice University.

Freeman, M. P., Forsyth, C., & Rae, I. J. (2019). The influence of substorms on extreme rates of change of the surface horizontal magnetic field in the United Kingdom. *Space Weather*, *17*, 827–844. https://doi.org/10.1029/2018SW002148

Gaunt, C. T., & Coetzee, G. (2007). Transformer failures in regions incorrectly considered to have low GIC-risk. 2007 IEEE Lausanne Power Tech. https://doi.org/10.1109/PCT.2007.4538419

Gleisner, H., Lundstedt, H., & Wintoft, P. (1996). Predicting geomagnetic storms from solar-wind data using time-delay neural networks. *Annales Geophysicae*, *14*(7), 679–686. https://doi.org/10.1007/s00585-996-0679-1

Gruet, M. A., Chandorkar, M., Sicard, A., & Camporeale, E. (2018). Multiple-hour-ahead forecast of the D_{st} index using a combination of long short-term memory neural network and Gaussian process. *Space Weather*, *17*, 1882–1896. https://doi.org/10.1029/2018SW0018981

Hapgood, M. (2018). Societal and economic importance of space weather, *Machine Learning Techniques for Space Weather* (pp. 3–26). Elsevier.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*, 251–257. https://doi.org/10.1016/0893-6080(91)90009-T

Iyemori, T. (1990). Storm-time magnetospheric currents inferred from mid-latitude geomagnetic field variations. *Journal of Geomagnetism and Geoelectricity*, *42*(11), 1249–1265. https://doi.org/10.5636/jgg.42.1249

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. In *Paper Presented at the 3rd International Conference for Learning Representations, San Diego, 2015*.

Klimas, A. J., Vassiliadis, D., Baker, D. N., & Roberts, D. A. (1996). The organized nonlinear dynamics of the magnetosphere. *Journal of Geophysical Research*, *10*(A6), 13,089–13,113. https://doi.org/10.1029/96JA00563

Lazzús, J. A., Vega, P., Rojas, P., & Salfate, I. (2017). Forecasting the D_{st} index using a swarm-optimized neural network. *Space Weather*, *15*, 1068–1089. https://doi.org/10.1002/2017SW001608

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Boser, B. J., Denker, S., Howard, R. E., Habbard, W., Jackel, L. D., & Henderson, D. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems*, 396–404.

Leshno, M., Vladimir, Y. L., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, *6*, 861–867. https://doi.org/10.1016/S0893-6080(05)80131-5

Lundstedt, H., Gleisner, H., & Wintoft, P. (2002). Operational forecasts of the geomagnetic D_{st} index. *Geophysical Research Letters*, *29*(24), 2181. https://doi.org/10.1029/2002GL016151

Lundstedt, H., & Wintoft, P. (1994). Prediction of geomagnetic storms from solar wind data with the use of a neural network. *Annales Geophysicae*, *12*(1), 19–24. https://doi.org/10.1007/s00585-994-0019-2

Mayaud, P. N. (1980). *Derivation, meaning, and use of geomagnetic indices*. American Geophysical Union.

Moldwin, M. B., & Tsu, J. S. (2016). Stormtime equatorial electrojet ground-induced currents. *Ionospheric Space Weather* (pp. 33–40). American Geophysical Union (AGU). https://doi.org/10.1002/9781118929216.ch3

Pallocchia, G., Amata, E., Consolini, G., Marcucci, M. F., & Bertello, I. (2006). Geomagnetic D_{st} index forecast based on IMF data only. *Annales Geophysicae*, *24*(3), 989–999. https://doi.org/10.5194/angeo-24-989-2006

Pallocchia, G., Amata, E., Consolini, G., Marcucci, M. F., & Bertello, I. (2008). AE index forecast at different time scales through an ANN algorithm based on L1 IMF and plasma measurements. *Journal of Atmospheric and Solar-Terrestrial Physics*, *70*(2–4), 663–668. https://doi.org/10.1016/j.jastp.2007.08.038

Tan, Y., Hu, Q., Wang, Z., & Zhong, Q. (2018). Geomagnetic index K_p forecasting with LSTM. *Space Weather*, *16*, 406–416. https://doi.org/10.1002/2017SW001764

Tozzi, R., De Michelis, P., Coco, I., & Giannattasio, F. (2019). A preliminary risk assessment of geomagnetically induced currents over the Italian territory. *Space Weather*, *17*, 46–58. https://doi.org/10.1029/2018SW002065

Viljanen, A., Pirjola, R., Prácser, E., Katkalov, J., & Wik, M. (2014). Geomagnetically induced currents in Europe. *Journal of Space Weather and Space Climate*, *4*, A09. https://doi.org/10.1051/swsc/2014006

Wang, C. B., Chao, J. K., & Lin, C. H. (2003). Influence of the solar wind dynamic pressure on the decay and injection of the ring current. *Journal of Geophysical Research*, *108*(A9), 1341. https://doi.org/10.1029/2003JA009851

Wanliss, J. A., & Showalter, K. M. (2006). High-resolution global storm index: D_{st} versus SYM-H. *Journal of Geophysical Research*, *111*, A02202. https://doi.org/10.1029/2005JA011034

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

Wintoft, P., Wik, M., Matzka, J., & Shprits, Y. (2017). Forecasting K_p from solar wind data: Input parameter study using 3-hour averages and 3-hour range values. *Journal of Space Weather and Space Climate*, *7*, A29. https://doi.org/10.1051/swsc/2017027

Wu, J. G., & Lundstedt, H. (1997). Neural network modeling of solar wind magnetosphere interaction. *Journal of Geophysical Research*, *102*(A7), 14,457–14,466. https://doi.org/10.1029/97JA01081