Supporting Information for "Probabilistic Geomagnetic Storm Forecasting via Deep Learning"

Adrian Tasistro-Hart[1,2], Alexander Grayver[1], Alexey Kuvshinov[1]

[1]Institute of Geophysics, ETH Zürich, Sonneggstrasse 5, 8092 Zürich, Switzerland
[2]Department of Earth Science, University of California, Santa Barbara, CA 93106, USA

Contents

Text S1: Data continuity and gap handling

OMNI data

We accessed the low-resolution OMNI dataset via the OMNIWeb interface. Within the OMNI data, all gaps of 72 hours or less were filled via linear interpolation.

CME data

The CME data were taken from the SOHO LASCO CME catalog.

Given that this project works with hourly data and multiple CMEs can occur within the same hour, the CME with the largest energy was taken in the cases when multiple CMEs did occur within an hour. Additionally, many events in the catalogue did not have all data fields filled, hence we used only the events for which all data were reported.

GOES data

The GOES x-ray flux data are provided by NOAA with one minute averaging. All the available files were downloaded for which primary and secondary satellites were specified and data from the satellite recommended by this relevant NOAA document was taken. These minute data were averaged in hourly bins to generate time series consistent with the other data sources. No gap exceeded 72 hours, and all gaps were interpolated linearly.

Text S2: Uncertainty in model parameters

One major paradigm of learning uncertainty in neural networks is to represent the network weights, or some internal aspect of the network, probabilistically. One of the first implementations was proposed by Blundell et al. (2015), who describe an architecture in which all of the network weights and biases are represented as distributions, and the problem becomes learning the parameters of those distributions. To achieve this, Blundell et al. (2015) outline a variational Bayesian framework. Variational refers to the fact that the true distribution over network weights is approximated

Corresponding author: Adrian Tasistro-Hart, adrian_tasistro-hart@ucsb.edu

by some simpler distribution, and Bayesian refers to the representation of this conditional distribution via Bayes' rule.

The cost function in this framework (Equation 1 depends only on the variational posterior, a prior over the weights (which we must specify), and a likelihood term dependent on the data. Blundell et al. (2015) propose a Gaussian mixture $p(\mathbf{w}) = \prod_i \pi \mathcal{N}(\mathbf{w}_i|0, \sigma_1^2) + (1-\pi)\mathcal{N}(\mathbf{w}_i|0, \sigma_2^2)$ for the prior, which we also utilize. This likelihood term is captured by a NN. Because the expectations are taken over the variational posterior, we can approximate them simply by sampling weights from their variational posteriors for given θ, and then we can update θ by differentiating the total loss against θ. This approach requires us to specify a functional form for $p(\mathbf{x}, \mathbf{y}|\mathbf{w})$, which effectively captures the level of anticipated noise in the training data.

$$\min_{\theta} \mathbb{E}_q \left[\log q(\mathbf{w}|\theta)\right] - \mathbb{E}_q \left[\log p(\mathbf{w})\right] - \mathbb{E}_q \left[\log p(\mathbf{x}, \mathbf{y}|\mathbf{w})\right] \tag{1}$$

This approach hinges on the choice of a simple variational posterior, and Blundell et al. (2015) suggest a Gaussian posterior over the weights. Since the Gaussian is a two-parameter model, this sort of architecture effectively double the number of parameters per weight, since the network learns a mean and a variance for each weight.

Given that the numerical differentiation can easily be done by TensorFlow, what remains is only to add the so-called"KL-losses" $\mathbb{E}_q \left[\log q(\mathbf{w}|\theta)\right] - \mathbb{E}_q \left[\log p(\mathbf{w})\right]$ to the negative log-likelihood loss of the data given weights sampled from the variational posterior while ensuring that the learnable parameters in the network are the θ parameterizing the variational posterior distributions over the weights. With a trained the network, arbitrarily many outputs can be sampled for a given input by sampling from the variational posterior over the weights, effectively simulating output from an ensemble of models. Then, statistics such as confidence intervals can be computed from this output.

This approach to probabilistic neural networks did not result in meaningful probabilistic forecasts for Est (Figure S1). Instead, the variational approach ended up learning a well-constrained posterior over the network weights, indicating that the network was quite confident that it had learned the optimal model. The confidence interval constructed from a Monte Carlo suite of models drawn from the variational posterior over the weights is barely visible around the mean posterior forecast, even though this confidence interval almost never contains the observed Est values. This approach simply demonstrates that the optimal model is confidently known by the network.

Text S3: Output distribution selection

Basic statistics of Est observations can inform the choice of output distribution. For instance, the empirical histogram of Est shows an asymmetry defined by a large tail at negative values (Figure S2B). Symmetric two-parameter distributions like the Gaussian and Laplace distributions fail to capture this asymmetry, and the Gaussian distribution furthermore fails to capture the heavy negative tail. These results are demonstrated by the quantile-quantile plots, which show that the Gumbel distribution is most appropriate for modeling the empirical distribution of Est (Figure S2A). The Gumbel distribution is from the family of generalized extreme value distributions often used to model phenomena with heavy tails such as earthquakes or flooding events. In this sense, the distribution of Est captures the extreme value nature of geomagnetic storms and motivates the utilization of the Gumbel distribution as an output distribution for probabilistic forecasting of Est.

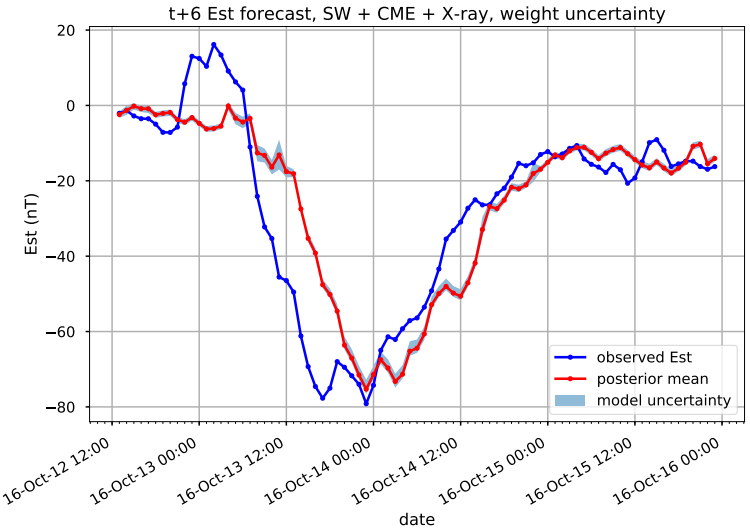

Figure S1. Model output for at 6 hour ahead forecast from the probabilistic network modeling uncertainty over network weights. The confidence interval was computed by sampling several thousand weights from the trained network and taking the 2.5-97.5% interval of output values.

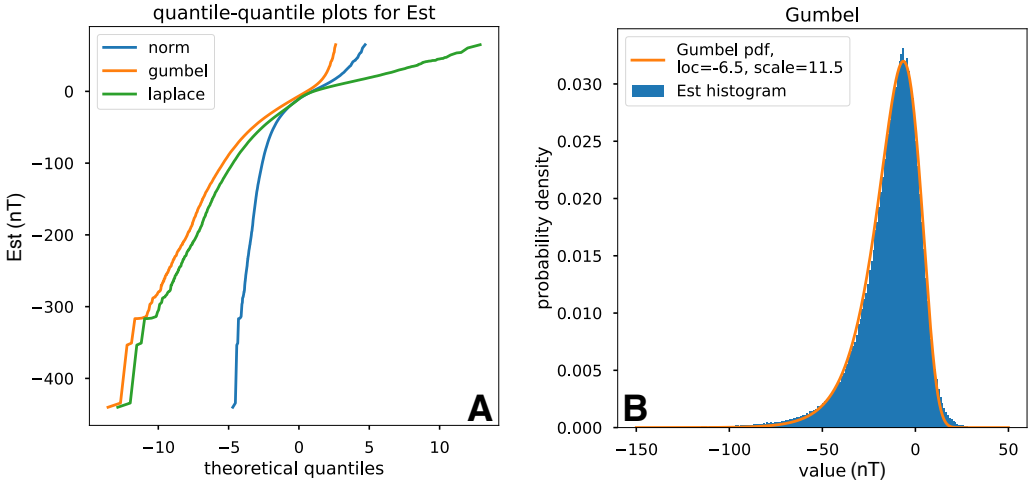

Figure S2. (A) Quantile-quantile plots for Est for three two-parameter distributions. The Gumbel distribution is closest to linear. (B) Empirical histogram of Est and a well-fitting Gumbel probability density showing how the asymmetry of the Gumbel distribution is capable of capturing the long tail of negative Est values.

However, while the marginal distribution of Est might be well-approximated by a Gumbel distribution, the forecast itself is a much more complicated distribution conditioned on the time history of the input data, neural network weights and biases, and the internal memory of the LSTM cell. The nature of this conditional output distribution is unknown. While it could be framed in a Bayesian sense, where the prior could be specified as a Gumbel distribution, the evidence and likelihood terms in this framework are not at all obvious to construct. Instead, we argue for regularization of the cost function, which can be heuristically motivated. This regularization is necessary because the choice of output distribution and cost function strongly impact the qualitative nature of network output (Figure S3) and forecast reliability (Figure 3 in manuscript).

We considered three cases:

1. Gumbel output distribution with Gumbel as likelihood cost function.
2. Gaussian output distribution with Gaussian as likelihood cost function.
3. Gaussian output with regularized Gaussian as likelihood cost function.

The first two cases follow the paradigm that the output distribution simultaneously serves as the likelihood distribution that is the cost function in this probabilistic framework. Output from each of these is shown for a storm in Figure S3, and the overall reliability of these networks demonstrates that the third, regularized architecture is most reliable (Figure 3 in main manuscript).

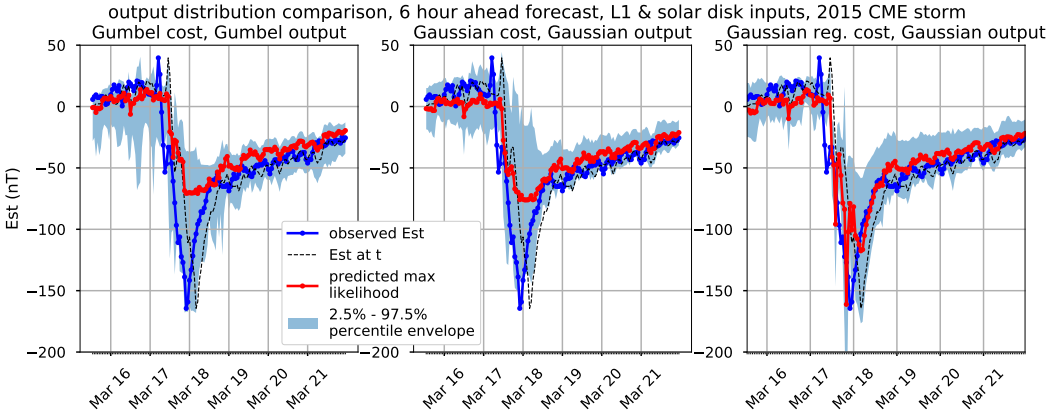

Figure S3. Network output for a 6 hour ahead Est forecast for three different cost functions. Left: output over Gumbel distributions with the Gumbel likelihood cost (Equation 2). Middle/Right: both networks learned Gaussian outputs, but the right plot shows output for the network that utilized the regularized Gaussian cost function (Equation 4), while the middle one utilized the basic Gaussian likelihood function (Equation 3). All networks were trained on data from both the solar disk (x-ray fluxes, CMEs) as well as solar wind observations from the L1 point.

The cost functions corresponding to these cases are negative log-likelihoods, whose expressions are

$$C_{\text{Gumbel}}(y,\mu,\sigma) = \log\,\sigma - \frac{y-\mu}{\sigma} + e^{\frac{y-\mu}{\sigma}} \tag{2}$$

$$C_{\text{Gaussian}}(y,\mu,\sigma) = \log\left(\sqrt{2\,\pi}\,\sigma\right) + \frac{(y-\mu)^2}{2\,\sigma^2} \tag{3}$$

$$C_{\text{Gaussian, regularized}}(y,\mu,\sigma) = \log\left(\sqrt{2\pi}\,\sigma\right) + \frac{(y-\mu)^2}{2\,\sigma^2} + \alpha\,(y-\mu)^2 + \beta\,\frac{1}{\sigma^2} \tag{4}$$

The cost functions in Equations 2 and 3, which are both negative log-likelihoods, produced networks that would not forecast mean values less than -100 nT. This effect is quite apparent in the first panel of Figure S3, where the Gumbel network forecasts storm main phase values at most -75 nT despite observed values exceeding -150 nT. Given that this network is generating a 6 hour ahead forecast, observations of Est from 6 hours ago should contribute information about reasonable magnitudes for the current forecast, meaning that once observed values decreased beyond -75 nT, one might expect the network to forecast more negative values if it perceives the storm to be continuing. However, this is not the case for the storm shown, during which Est exceeded -75 nT for roughly 20 hours. What is apparent is that instead of moving the output distribution location, the network favored increasing the output uncertainty, in general capturing the observed range of variability within the 95% confidence interval. While this result demonstrates that the network is aware of its forecast uncertainty during the main phase of the storm, the inability of the network to move its maximum likelihood estimate to forecast large storm magnitudes diminishes its operational reliability. The corresponding reliabilities demonstrate the reduced utility of the probabilistic forecasts for large storm amplitudes from the Gaussian and Gumbel networks: while they are quite reliable at smaller Est thresholds, forecasting Est beyond -75 nT becomes less reliable.

The structure of the cost functions for a given true Est value of -150 nT (Figure S4) shows why the unregularized networks favor expanding uncertainty rather than shifting the central value for the forecast: the cost functions are much less sensitive to forecasted μ if the forecasted σ is large, meaning that the network will tend to learn to increase σ without having a strong incentive to learn a reasonable μ. Thus, the idea to regularize is motivated by the desire to incentivize the network to learn more reasonable estimates for μ. A simple way to include this incentive is to add another least squares cost that is not normalized by the forecasted uncertainty, as shown in Equation 4, where the quantity α is a new hyper-parameter that dictates the strength of this regularization.

This additional cost forces the network to learn more reasonable forecasts for μ while still allowing it to change σ quite freely for a given μ. As can be seen in Figures S3 and 3 (in manuscript), this regularization term significantly improves the forecast. The maximum likelihood forecast more closely overlaps the observed Est values, and the forecast uncertainty still exhibits meaningful behavior, with large uncertainties associated with storm arrivals and smaller uncertainties during storm recovery and quiet times.

We also add a second regularizing term, $\beta \frac{1}{\sigma^2}$, which permits adjustment of the forecasted uncertainty. Positive values of β penalize smaller forecast uncertainties, whereas negative values of β encourage smaller forecast uncertainties.

Text S4: Learning parameters

This section briefly lists all other learning parameters used during training of the neural networks. The optimizer of choice is the so-called "adam" optimizer, which uses both momentum (i.e., memory of the previous update direction), and scaling by the inverse of the second moment of the gradient to generate first order updates in stochastic gradient descent optimization (Kingma & Ba, 2014). This optimizer has three hyperparameters, and the only one that was changed is the step-size multiplier. The value of this parameter was varied between 0.001 and 0.005. All networks were trained with batch sizes of 1000 hours, meaning that the trained LSTM cells developed memories relevant to processes operating on the timescale of months, which is more than enough for learning storm-scale dynamics on the timescales of days to weeks. Networks were trained over 1000-2000 epochs, where an epoch reflects one entire pass

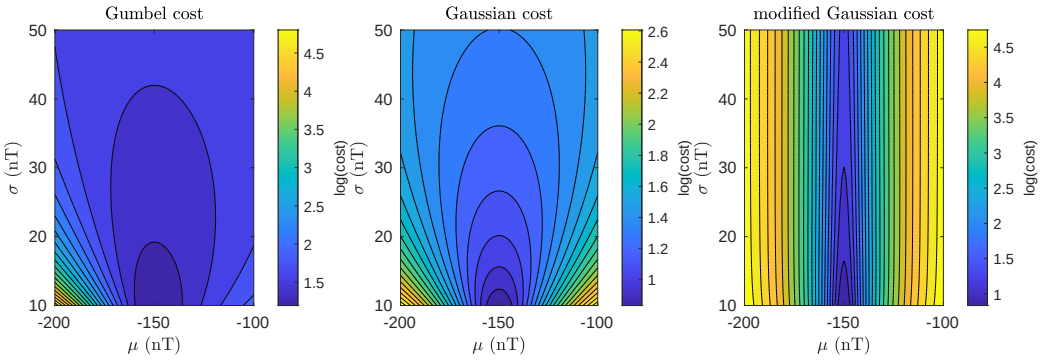

Figure S4. Cost functions for an observed output of $y = -150$ nT, with each panel corresponding to the cost functions used to train the networks whose output is shown in Figures S3. Note that the contours show the logarithm of the cost functions.

through the training dataset. We selected $\alpha = 0.1$ and $\beta = 1$ for the regularizing coefficient values in Equation 4.

Text S5: Forecast accuracy sensitivity to input data

In order to assess the benefit of incorporating observations from the solar disk, we trained two separate networks to generate probabilistic forecasts for Est: one network was trained on both data from the solar disk and solar wind observations from the L1 point (i.e., utilizing the GOES, CME, and OMNI data), and the second network was trained only on the solar wind observations from the L1 point (i.e., utilizing only the OMNI data). Both networks had identical architectures that differed only by the dimensionalities of the weights and biases necessary to accommodate the differing input dimensionalities.

This section demonstrates that utilizing observations of the solar disk in addition to observations from the L1 point as input data, forecasting of storm main phase timing and amplitude does not significantly improve, although the estimated of the uncertainty envelope becomes more reliable (Figures S5-S6), especially for multiple hours ahead forecast. Nevertheless, more opportunities remain for developing neural network architectures capable of utilizing sparse, impulse-like solar disk observations.

For both networks, storm onset remains difficult to predict, with the network not recognizing that a storm has begun until it receives as input the storm onset from the L1 point (Figures S5 & S6). Once the networks have felt the storm onset, however, they both dramatically expand uncertainty in their forecasts. The networks with only L1 inputs expand uncertainty more during the storm main phase and less during recovery and quiet times compared to the networks with both L1 and solar disk inputs, resulting in the observed reliability curves that demonstrate reduced reliability for low amplitude storms but slightly higher reliability for larger amplitude storms (Figure S7). For probable, high amplitude storms, then, the networks with only L1 inputs are slightly more reliable, while for smaller amplitude storms the networks with both L1 and solar disk inputs are more reliable.

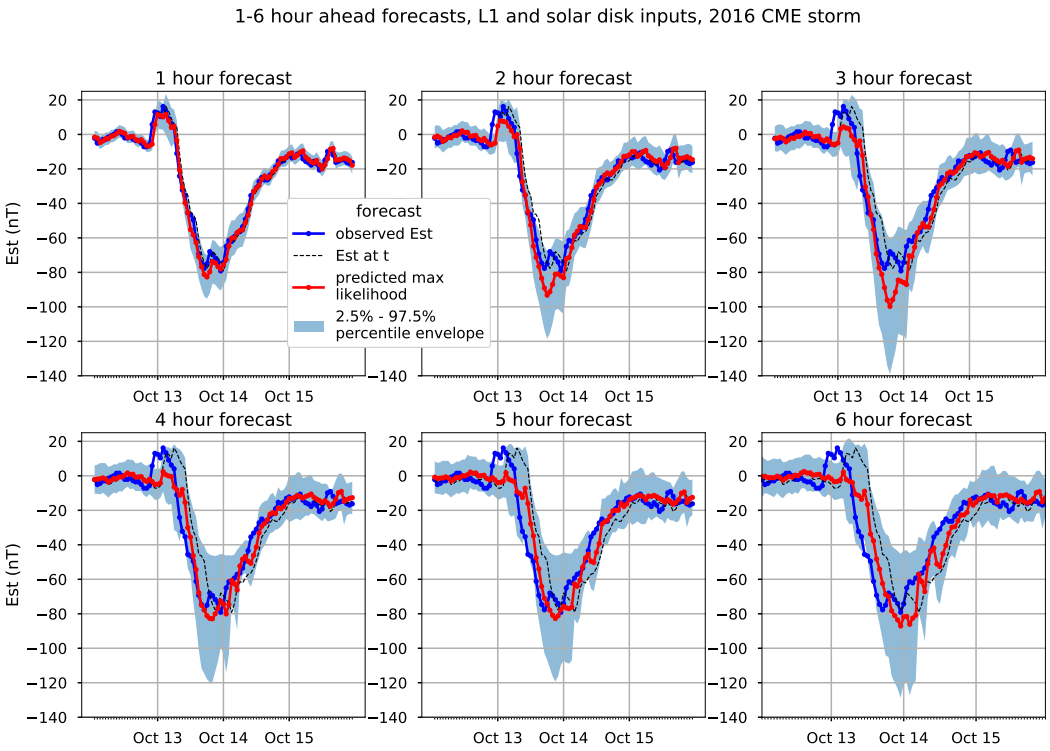

Figure S5. 1-6 hour ahead forecasts for the 2016 CME storm for the networks with observations from both the solar disk and L1 point as input.

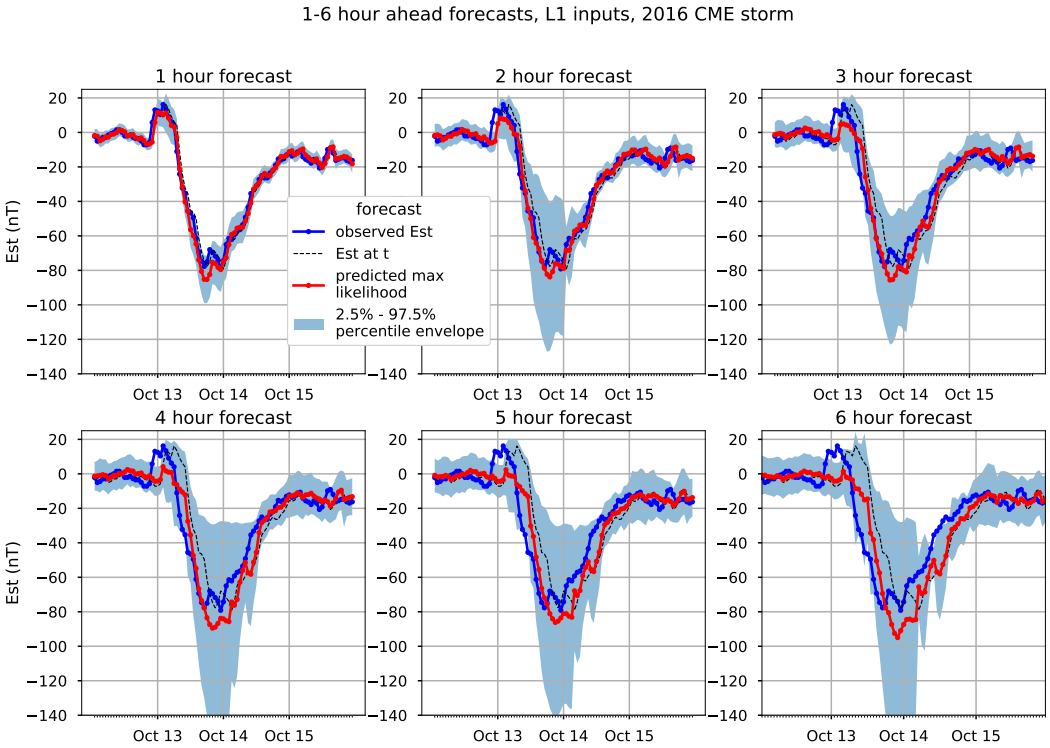

Figure S6. 1-6 hour ahead forecasts for the 2016 CME storm for the networks with only observations from the L1 point as input.

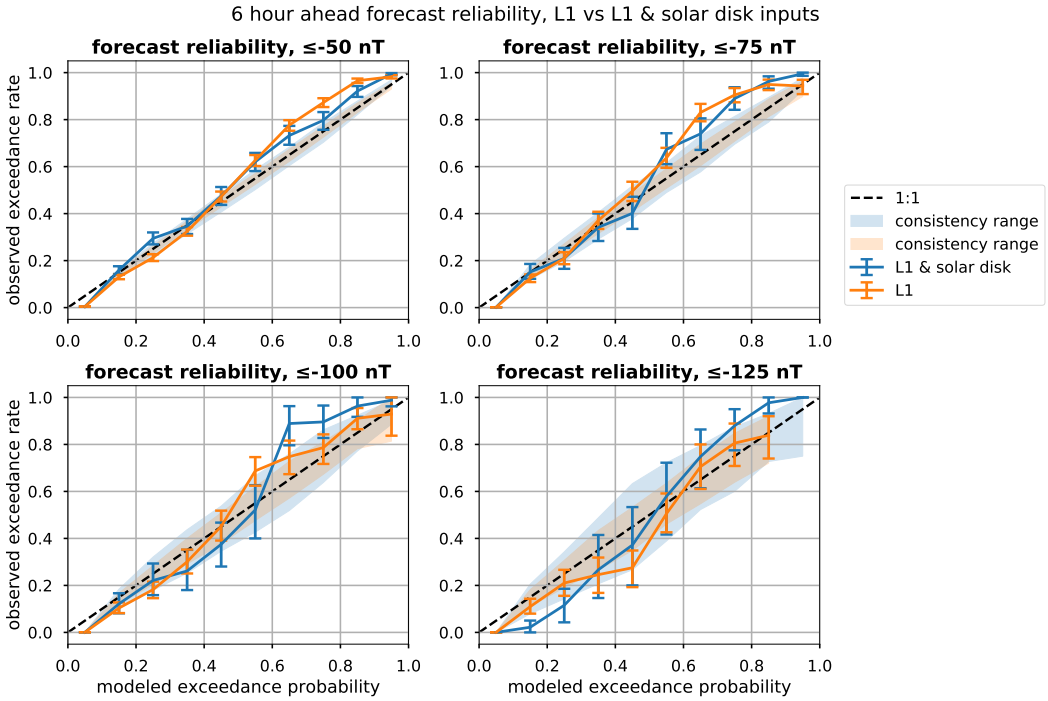

Figure S7. Reliability of 6 hour ahead forecast for network trained on L1 data only compared with the network trained on both L1 and solar disk inputs.

Text S6: Long Short-Term Memory

The complete set of equations for the LSTM is as follows

$$
\begin{aligned}
\mathbf{f}_t &= \sigma_g(W_f\,\mathbf{x}_t + U_f\,\mathbf{y}_{t-1} + \mathbf{b}_f) \\
\mathbf{i}_t &= \sigma_g(W_i\,\mathbf{x}_t + U_i\,\mathbf{y}_{t-1} + \mathbf{b}_i) \\
\mathbf{o}_t &= \sigma_g(W_o\,\mathbf{x}_t + U_o\,\mathbf{y}_{t-1} + \mathbf{b}_o) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \sigma_h(W_c\,\mathbf{x}_t + \mathbf{U}_c\,\mathbf{y}_{t-1} + \mathbf{b}_c) \\
\mathbf{z}_t &= \mathbf{o}_t \odot \sigma_h(\mathbf{c}_t)
\end{aligned}
$$

which is then again combined with the input to generate a new output. for input $\mathbf{x}_t \in \mathbb{R}^d$ at time t and $\mathbf{z}, \mathbf{c} \in \mathbb{R}^m$ where m is a hyperparameter defining the dimensionality of the cell memory $\mathbf{c}$ and output $\mathbf{z}$. The functions $\sigma_g(.)$ and $\sigma_h(.)$ are the sigmoid and hyperbolic tangent functions, respectively. The weights then have the following dimensions $W \in \mathbb{R}^{m\times d}$ and $U \in \mathbb{R}^{m\times m}$ and the biases are all in $\mathbb{R}^m$. The $\odot$ operator signifies an elementwise product. Thus, for given dimensions m and d, the total number of parameters to be learned within an LSTM cell is $4\,m\,d + 4\,m^2 + 4\,m$.

Other Supplementary Information

Ahead	Architecture	R	RMSE (nT)	Inputs	Database	Reference
t + 1	MLP, BP	0.92	15	n, v, Bz	1963–1983	Gleisner et al. (1996)
t + 1	Elman Recurrent, BP	0.91	16	n, v, Bz	1963–1987	Wu & Lundstedt (1996)
t + 1 t + 2 t + 3 t + 4 t + 5 t + 6 t + 7 t + 8	Elman Recurrent, BP	0.91 0.89 0.86 0.83 0.82 0.82 0.80 0.77	14.5 16.3 18.2 19.9 20 20.8 21.8 23.1	n, v, Bx, By, Bz	1963–1992	Wu & Lundstedt (1997)
t + 1	MLP, BP	0.95	11	n, v, Bx, By, Bz	1972–1982	Kugblenu et al. (1999)
t + 1	Elman Recurrent, BP	0.88		n, v, B, Bs	1968–1987	Munsami (2000)
t + 1 t + 2 t + 3 t + 4 t + 5 t + 6 t + 7 t + 8	MLP, BP	0.95 0.93 0.88 0.85 0.82 0.78 0.75 0.72		previous Dst	1983	Stepanova & Pérez (2000)
t + 1 t + 6 t + 12 t + 18	MLP, BP	0.93 0.73 0.69 0.66		Bx, By, Bz, B, n, v, dBx/dt, dBy/dt, dBz/dt, dv/dt, dn/dt	1998–1999	Jankovičová et al. (2002)
t + 1	MLP, BP	0.70		polar cap index, previous Dst	1997	Stepanova et al. (2005)
t + 1	Elman Recurrent, BP	0.83	13.9	By, Bz, B	1995–2005	Pallocchia et al. (2006)
t + 1 t + 2 t + 3 t + 4	Locally Linear Neuro-Fuzzy Model	0.98 0.95 0.95 0.87	4.38 7.43 10 11.83	n, v, Bs, Dst, dDst/dt	1995–1999	Sharifie et al. (2006)
t + 1	Radial Basis Function Network		18.45	n, v, Bs	1998	Wei et al. (2007)
t + 1 t + 3 t + 6	MLP, BP	0.86 0.84 0.80	8.84 9.40 10.34	Boyle index, Dp	1998–2009	Bala & Reiff (2012)
t + 1	MLP, BP	0.77		Bz, n, v, T	1998–2005	Revallo et al. (2014)
t + 1	Relevance Vector Machine	0.96	10	By, Bz, v, n, a:p, T, f10.7	1996–2007	Andriyas & Andriyas (2015)
t + 1 t + 2 t + 3 t + 4 t + 5 t + 6	MLP, Particle Swarm	0.978 0.936 0.895 0.857 0.825 0.788	3.57 5.97 7.54 8.82 9.75 10.89	previous Dst	1990–2016	Lazzús et al. (2017)
t + 1 t + 2 t + 3 t + 4 t + 5 t + 6	LSTM, BP	0.966 0.946 0.928 0.910 0.892 0.873	5.25 6.55 7.59 8.53 9.18 9.86	n, v, B, Bz, B-GPS, previous Dst	2001–2016	Gruet et al. (2018)

Table S1. Results of previous applications of neural networks to Dst forecasting, after Lazzús et al. (2017). MLP: multilayered perceptron. BP: backpropagation. LSTM: long short-term memory. n: solar wind density. v: solar wind velocity. a:p: alpha-to-proton ratio. Bs: southward-only component of interplanetary magnetic field. Dp: dynamic pressure = nv^2. Compare the values for R and $RMSE$ with those resulting from a simple persistence forecast shown in Figure S8.

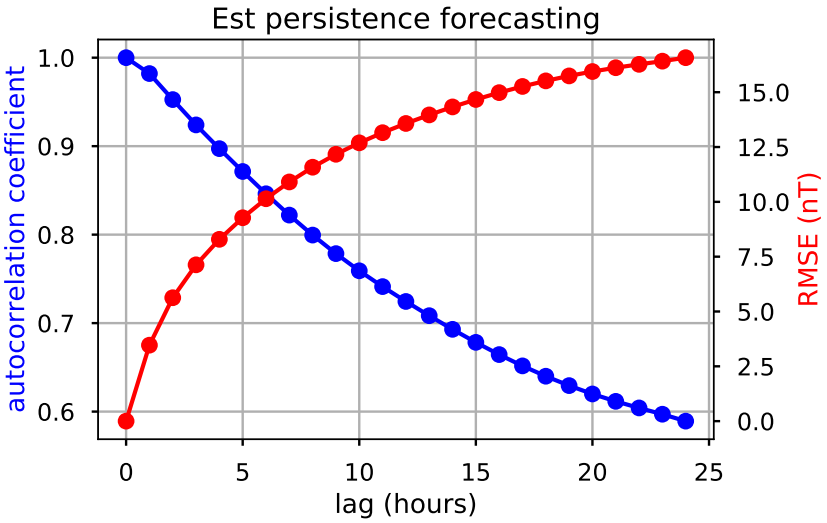

Figure S8. Statistics of persistence forecasting of Est with lags up to 24 hours, demonstrating how persistence forecasting even a day in advance results in deceptively correlative and accurate forecasts.

References

Andriyas, T., & Andriyas, S. (2015). Relevance vector machines as a tool for forecasting geomagnetic storms during years 1996–2007. *Journal of Atmospheric and Solar-Terrestrial Physics*, *125*, 10–20.

Bala, R., & Reiff, P. (2012). Improvements in short-term forecasting of geomagnetic activity. *Space Weather*, *10*(6).

Blundell, C., Cornebise, J., Kavukcuoglu, K., & Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Gleisner, H., Lundstedt, H., & Wintoft, P. (1996). Predicting geomagnetic storms from solar-wind data using time-delay neural networks. *Annales Geophysicae*, *14*, 679.

Gruet, M., 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*(11), 1882–1896.

Jankovičová, D., Dolinskỳ, P., Valach, F., & Vörös, Z. (2002). Neural network-based nonlinear prediction of magnetic storms. *Journal of atmospheric and solar-terrestrial physics*, *64*(5-6), 651–656.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kugblenu, S., Taguchi, S., & Okuzawa, T. (1999). Prediction of the geomagnetic storm associated Dst index using an artificial neural network algorithm. *Earth Planets Space*, *51*(307-313).

Lazzús, J., Vega, P., Rojas, P., & Salfate, I. (2017). Forecasting the Dst index using a swarm-optimized neural network. *Space Weather*, *15*(8), 1068–1089.

Munsami, V. (2000). Determination of the effects of substorms on the storm-time ring current using neural networks. *Journal of Geophysical Research: Space Physics*, *105*(A12), 27833–27840.

Pallocchia, G., Amata, E., Consolini, G., Marcucci, M., & Bertello, I. (2006). Geomagnetic Dst index forecast based on IMF data only. *Annales Geophysicae*, *24*(3), 989–999.

Revallo, M., Valach, F., Hejda, P., & Bochníček, J. (2014). A neural network Dst index model driven by input time histories of the solar wind-magnetosphere interaction. *Journal of Atmospheric and Solar-Terrestrial Physics*, *110*, 9–14.

Sharifie, J., Lucas, C., & Araabi, B. N. (2006). Locally linear neurofuzzy modeling and prediction of geomagnetic disturbances based on solar wind conditions. *Space Weather*, *4*(6).

Stepanova, M., Antonova, E., & Troshichev, O. (2005). Prediction of Dst variations from polar cap indices using time-delay neural network. *Journal of atmospheric and solar-terrestrial physics*, *67*(17-18), 1658–1664.

Stepanova, M., & Pérez, P. (2000). Autoprediction of Dst index using neural network techniques and relationship to the auroral geomagnetic indices. *GEOFISICA INTERNACIONAL-MEXICO-*, *39*(1), 143–146.

Wei, H.-L., Zhu, D.-Q., Billings, S. A., & Balikhin, M. A. (2007). Forecasting the geomagnetic activity of the Dst index using multiscale radial basis function networks. *Advances in Space Research*, *40*(12), 1863–1870.

Wu, J.-G., & Lundstedt, H. (1996). Prediction of geomagnetic storms from solar wind data using Elman recurrent neural networks. *Geophysical research letters*, *23*(4), 319–322.

Wu, J.-G., & Lundstedt, H. (1997). Geomagnetic storm predictions from solar wind data with the use of dynamic neural networks. *Journal of Geophysical Research: Space Physics*, *102*(A7), 14255–14268.