AGU ADVANCING EARTH AND SPACE SCIENCE

Space Weather

RESEARCH ARTICLE
10.1029/2020SW002478

Key Points:
- Machine learning models are evaluated for the prediction of solar wind (SW)speed measured at Lagrangian Point 1 between the Sun and Earth
- Without imposing physics rules, deep neural networks (DNNs) can be trained on extreme UV images of the solar corona to forecast SW speed
- Our DNN WindNet significantly outperforms simpler models and pays attention to coronal holes in 193 Å for fast wind prediction

Correspondence to:
V. Upendran,
uvishal@iucaa.in

Citation:
Upendran, V., Cheung, M. C. M., Hanasoge, S., & Krishnamurthi, G. (2020). Solar wind prediction using deep learning. *Space Weather*, *18*, e2020SW002478. https://doi.org/10.1029/2020SW002478

Received 14 FEB 2020
Accepted 5 JUN 2020
Accepted article online 10 JUN 2020

Solar Wind Prediction Using Deep Learning

Vishal Upendran[1,2], Mark C. M. Cheung[3,4], Shravan Hanasoge[5], and Ganapathy Krishnamurthi[2]

[1]Inter-University Centre for Astronomy and Astrophysics, Pune, India, [2]Department of Engineering Design, Indian Institute of Technology-Madras, Chennai, India, [3]Lockheed Martin Solar and Astrophysics Laboratory, Palo Alto, CA, USA, [4]Hansen Experimental Physics Laboratory, Stanford University, Stanford, CA, USA, [5]Department of Astronomy and Astrophysics, Tata Institute of Fundamental Research, Mumbai, India

Abstract Emanating from the base of the Sun's corona, the solar wind fills the interplanetary medium with a magnetized stream of charged particles whose interaction with the Earth's magnetosphere has space weather consequences such as geomagnetic storms. Accurately predicting the solar wind through measurements of the spatiotemporally evolving conditions in the solar atmosphere is important but remains an unsolved problem in heliophysics and space weather research. In this work, we use deep learning for prediction of solar wind (SW) properties. We use extreme ultraviolet images of the solar corona from space-based observations to predict the SW speed from the National Aeronautics and Space Administration (NASA) OMNIWEB data set, measured at Lagragian Point 1. We evaluate our model against autoregressive and naive models and find that our model outperforms the benchmark models, obtaining a best fit correlation of 0.55 ± 0.03 with the observed data. Upon visualization and investigation of how the model uses data to make predictions, we find higher activation at the coronal holes for fast wind prediction ($\approx$3 to 4 days prior to prediction), and at the active regions for slow wind prediction. These trends bear an uncanny similarity to the influence of regions potentially being the sources of fast and slow wind, as reported in literature. This suggests that our model was able to learn some of the salient associations between coronal and solar wind structure without built-in physics knowledge. Such an approach may help us discover hitherto unknown relationships in heliophysics data sets.

Plain Language Summary The solar wind is a stream of particles coming from the Sun. The interaction of the solar wind with the Earth's magnetosphere gives rise to space weather effects, including geomagnetic storms, aurorae and disruptions to electrical distribution grids. Accurate prediction of the solar wind is of interest to government agencies and private industry. In this work, we explore the use of machine learning models to predict the solar wind speed as measured at the Lagrangian Point 1 (L1) between the Sun and Earth. The best performing method is a deep neural network that uses extreme ultraviolet (EUV) imagery data from National Aeronautics and Space Administration's (NASA's) Solar Dynamics Observatory (SDO) as input. Without explicitly building in physical relationships into the model, it is able to outperform a number of baseline models. We find the model pays attention to regions on the Sun that are in agreement with heuristics used in the literature (e.g. coronal holes for the fast solar wind). Such an approach may, in the future, help us discover new relationships in heliophysics.

1. Introduction

Space weather is defined by the U.S. National Space Weather Plan as the conditions on the Sun, in the solar wind (SW), and within Earth's magnetosphere, ionosphere, and thermosphere that can influence the performance and reliability of spaceborne and ground-based technological systems and can endanger human life or health (NASA, 2017). The solar wind that influences space weather comprises a continuous stream of magnetized charged particles emanating from the base of the solar corona and permeating the solar system (Schwenn, 2006). The influence of the SW on space weather arises due to its interaction with the Earth's magnetospehere, resulting in geomagnetic storms and aurorae. Such terrestrial effects of the SW make it extremely important to understand and identify its possible sources and perform a prediction as a forewarning against potential damages.

Owens et al. (2008) review SW prediction using empirical, physics-based, and hybrid approaches. Typically, a physics-based model uses synoptic magnetograms as the bottom boundary condition. An individual synoptic magnetogram is assembled by sampling the photospheric magnetic flux distribution near the central meridian over the course of a solar rotation. Such magnetograms can be used to extrapolate the surface field into the corona using potential-field source-surface (PFSS)(Altschuler & Newkirk, 1969) models or magnetohydrodynamics (MHD) models (see Riley et al., 2006, for a comparison between the two). The global coronal magnetic field (or certain derived properties thereof) may then be used as input for physics-based SW propagation models (e.g. Linker et al., 1999), or in the case of a hybrid approach, used for estimation of the SW at L1 using empirical relations. WSA-ENLIL and MAS-ENLIL (Owens et al., 2008; Schwenn, 2006) are among the most widely used SW models. The models provide SW properties such as the velocity, plasma density, magnetic field and temperature. Jian et al. (2015) perform a comparison of the various SW models and present a best correlation of 0.57 on hourly prediction using GONG-MAS Thermo-ENLIL model, and 0.50 on the same data set using GONG-WSA-ENLIL model.

The dependence of high-speed streams on the presence of Coronal Holes (CHs) was first shown in Krieger et al. (1973), using extrapolation methods. Krieger et al. (1973) interpreted this empirical relation as a manifestation of high-speed SW flowing along open magnetic flux regions. Wang and Sheeley Jr (1990) used the inverse of flux tube expansion factor for SW prediction, by identifying an inverse correlation between the flux tube expansion factor and the SW. Recently, there has been a surge in performing regression using fractional (CH) area extracted from EUV imagery data (Rotter et al., 2012, 2015; Temmer et al., 2018), and a correlations from $\approx$0.60 to $\approx$0.78 have been obtained between the CH areas and hourly solar wind speed. More recently, Yang et al. (2018) devised a Neural network based prediction scheme, taking PFSS model output among other parameters as input, and obtained a correlation of 0.74 on hourly solar wind speed data.

In machine learning (ML) and statistical-learning parlance, the aforementioned traditional empirical models use so-called hand-engineered features as input for their models (e.g., CH area or CH expansion factor). These hand-engineered features are often inspired by some insights from physics-based models, or simply from correlations reported in the literature. Deep learning (DL) is an umbrella term for a broad class of techniques that use neural networks with multiple hidden layers for performing supervised or unsupervised learning tasks. Due to the increased availability of data, and perhaps more importantly, of inexpensive computation, DL has been widely applied in many domains. In areas of engineering and science dealing with ambiguous features, DL has been found to outperform ML algorithms that use hand-engineered features (Goodfellow et al., 2016). Generally, DL involves solving supervised learning tasks such as classification (associating a particular label with a given set of inputs, such as in image classification) or regression (continuous-value prediction). Often, these algorithms need no prior information regarding the exact input-output mapping but instead try to discover underlying relations by iteratively updating the model parameters by minimizing a loss function (e.g., mean square error). Prior information can be built into the model in a number of ways, for example, (1) by providing hand-engineered input features (which are generally physics based), (2) assembling a neural net with some layers that have been pretrained and whose weights are kept fixed during training (known as transfer learning, although the amount of prior information shared is limited by the kind of pretraining performed), and (3) specially customized initialization of weights (usually to accelerate convergence).

Two of the most prominent architectures used in deep learning are Convolutional Neural Networks (ConvNets) and Recurrent Neural Networks (RNNs). ConvNets are a set of deep neural nets, which have been successfully applied to different kinds of classification and regression problems (Ciresan et al., 2011; Deng & Yu, 2014; LeCun et al., 2015) for image data. These networks work by detecting local patterns at multiple scales in the input, and map them to the appropriate class (classification) or continuous output (regression). RNNs, on the other hand, are a class of deep neural nets designed for understanding the structure of data with a sequential ordering. These have been used extensively for text prediction, natural language processing and regression (Hochreiter & Schmidhuber, 1997; Sutskever et al., 2014).

In this work, we use a ConvNet (Szegedy et al., 2015) pretrained on the ImageNet database (Deng et al., 2009), and couple it with a trainable Long-Short Term Memory cell (LSTM) implementation of an RNN (Hochreiter & Schmidhuber, 1997). We use EUV images in 193 and 211 Å from Atmospheric Imaging Assembly (AIA) onboard SDO as input, making features like CHs and ARs clearly visible, and predict the SW speed present in the NASA OMNIWEB data set. The network is not given any prior information about the physical mapping

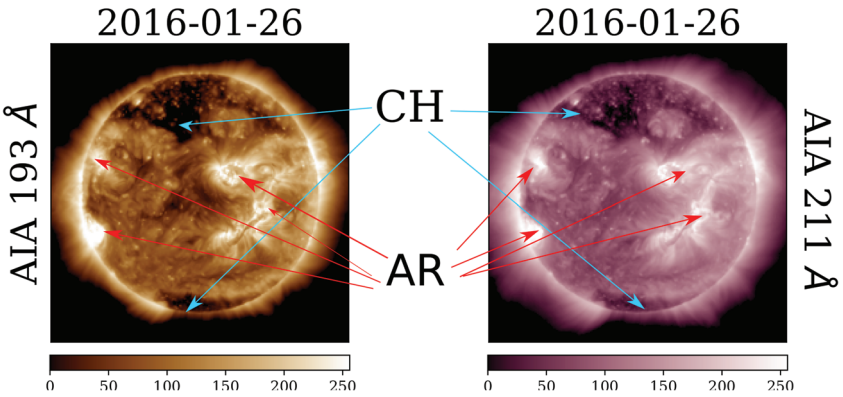

Figure 1. Representative AIA data in the 193 and 211 Å channels with CHs and ARs marked. This is the final data used in our analysis.

between the EUV image data and SW speed, and a direct regression is performed from a time series of AIA images to the SW speed.

This work is presented as follows: In section 2 we describe how the input and output data are preprocessed, partitioned into training and test sets, and define some control parameters and evaluation metrics. Then, in section 3, we briefly introduce the various algorithms used as our benchmarks. We detail our proposed model WindNet, and the visualization technique used for generating activation map. The segmentation algorithms used for the generation of binary masks for the computation of mean activation values, are also described. In section 4, we summarize our model predictions vis-a-vis our benchmarks, present the trends of mean activation, and draw conclusions in section 5.

2. Data and Metrics

2.1. EUV Data Set

The AIA (Lemen et al., 2012) onboard the SDO (Pesnell et al., 2012) is a four-telescope array observing the full Sun in visible, UV and EUV wavelengths. An ML data set curated from SDO instruments (hereafter, SDOML) has been made publicly available (Galvez et al., 2019) (https://purl.stanford.edu/jc488jb7715 and links therein). AIA images in the data set have been resampled onto a grid of 512×512 pixels with 4.8 arc sec pixel spacing and are available at 6 min cadence. SDOML images are stored as binary arrays in the Python numpy format (Walt et al., 2011). Our training and testing data include AIA images taken at 00 : 00 UTC for each day. The selected image forms a proxy for the whole day of observation. However, if the image at 00 : 00 does not exist (as is the case with many days), the closest image to 00 : 00, from that day, is taken as a proxy for that day.

Even during nonflaring times, solar EUV images can have a dynamic range that greatly exceeds the 8 bits per channel dynamic range typical of most computer-vision data sets. For this reason, the input AIA images are first preprocessed by performing log-scaling to bring out fainter features. The images are then passed through a threshold and saturation, which limits the dynamic range of pixel values. This was done to limit the prediction to contribution from Solar disc alone. Furthermore, it was seen that the model performance was better with thresholded and saturated images—thus, the dynamic range was limited. A general sweep of threshold and saturation was performed for a particular combination of history and delay (2.4), for the 193 Å data. The correlation of predicted SW speed with observed SW speed 0.48 ± 0.03 for the best set, with higher thresholds $(\log(250), \log(10000))$ giving us 0.46 ± 0.02 and lower thresholds $(\log(100), \log(1000))$ giving us 0.35 ± 0.02. A coarse search was performed to find the threshold values. The thresholds for 193 Å data were scaled to 211 Å through a ratio of maximum intensities on a given day selected randomly. Equations 1 and 2 specify the threshold and saturation operations for log scaled 193 and 211 Å channel images, respectively. AIA 193 and 211 Å data, with CHs and ARs marked, after preprocessing are shown in Figure 1.

$$x(193) = \begin{cases} \log(125.0) & \text{if } x \leq \log(125.0) \\ \log(5000.0) & \text{if } x \geq \log(5000.0) \\ x & \text{else} \end{cases} \quad (1)$$

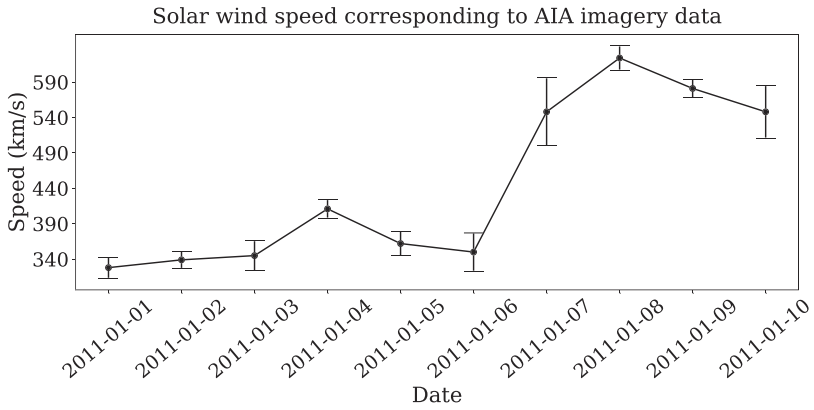

Figure 2. Ten days of SW speed from NASA OMNIWEB data set.

$$x(211) = \begin{cases} \log(25.0) & \text{if } x \leq \log(25.0) \\ \log(2500.0) & \text{if } x \geq \log(2500.0) \\ x & \text{else} \end{cases} \quad (2)$$

The pixel values are then rescaled between 0.0 and 255.0. This is done since our feature extractor expects inputs within this range of values.

2.2. SW Data Set

The target output of the prediction models is daily averaged SW speed measured at L1. Daily averages are used here since the variation of wind speed over a day is not large, and the variation across the mean value sets the uncertainty in wind speed value. The variation (or the standard deviation σ) is calculated as the variance in hourly measurements over the day, at the OMNIWEB archive (available online at https://omniweb.gsfc.nasa.gov/form/dx1.html). A representative variation in SW speed data over 10 days is plotted in Figure 2. The distribution of SW speed and the corresponding σ for the entire data set is shown in Figure 3.

There are gaps in the AIA EUV data (30 days of missing data in 211 Å and 31 days of missing data in 193 Å) for *00:00 UTC*, owing to various reasons ranging from calibration maneuvers to recoveries from instrument anomalies. Thus, the SW speed during these gaps have been removed to form sets of {image, wind speed}.

2.3. Data Set Partitioning and Cross Validation

Data are available from 1 January 2011 to 9 December 2018. Given the presence of a background solar cycle and events in the Sun, which might systematically bias our model to perform only for a particular phase of cycle, the whole data set was partitioned into batches comprising 20 contiguous days of data. If during batch formation, there exists a single discontinuity in the batch, the data from the day prior to discontinuity to 20 days prior is sampled and placed in the same place as the previous batch, thereby removing any data

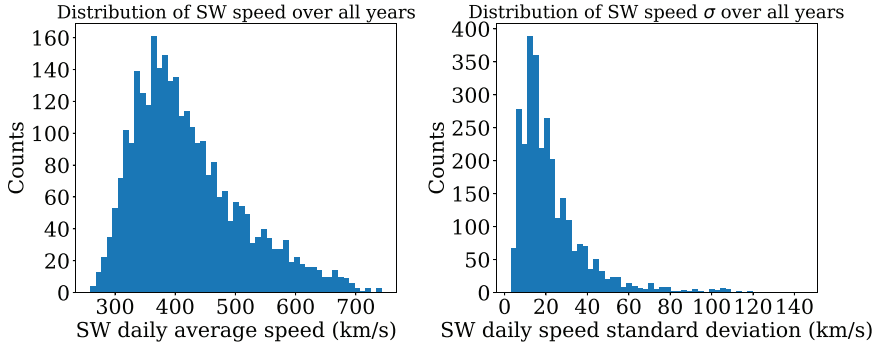

Figure 3. Left: Distribution of SW speed. Right: Distribution of the associated σ. The distributions are computed over the entire data set.

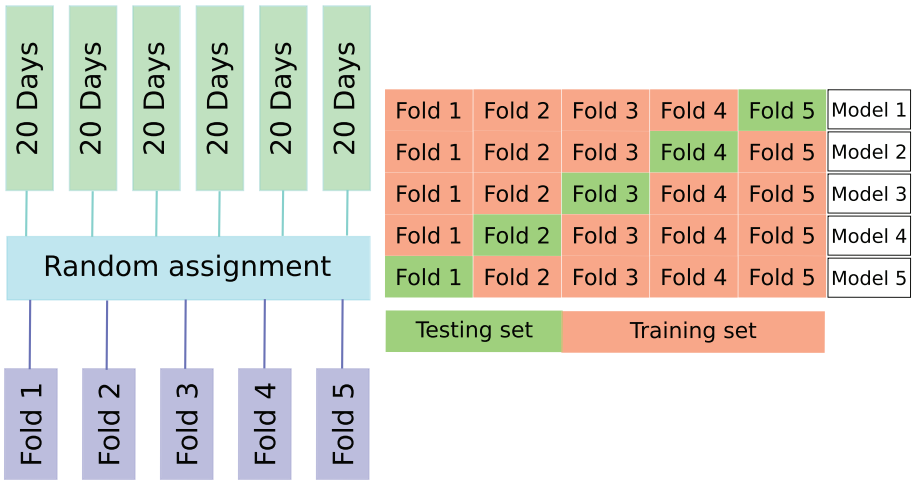

Figure 4. Training and test split for cross validation. First, the data are split into batches of 20 days each and each batch is randomly assigned to one of the cross-validation folds. Then, for one particular model (i.e., for one particular combination of history and delay), one of the folds is marked as the test set and the remaining as training sets. A circular permutation is performed till each fold is used as a test set. In our case, at the end of a training exercise, we will have five different variants of the particular model, from which we derive the mean and standard deviation of fitting metrics.

leak. If there exist multiple discontinuities (there are only two instances of such an event in either of the data sets), that particular window between the discontinuities is discarded. This results in 157 batches for 211 Å and 158 batches for 193 Å data (courtesy the one missing data, which resulted in a new batch). These batches were randomly sorted into five folds with equal probability, and these five folds were used to perform cross validation. The data set partitioning scheme is shown in Figure 4.

In cross validation, if there are [1,N] folds of data, a cross-validation set is constructed by holding the fold i as the test set, and the remaining in training set. Such a construction is done for all folds of the batches. Our models are evaluated against this cross-validation data set, thereby providing us with a mean value of the metric and a standard deviation. Henceforth, any standard deviation associated with the predictions are to be taken as evaluated on the cross-validation data set.

The image data are centered using the mean pixel value of the training data set per cross validation fold. The images are resized to 224×224 pixels using OpenCV's (Bradski, 2000) default Linear Interpolation, and each image replicated into three RGB channels. This was performed as our pretrained network demands the input images to be of dimensions $224 \times 224 \times 3$, since terrestrial images generally have red, green, and blue as the color basis. These images are then finally used for training our network. The solar wind speed data are scaled between 0 and 1 using the training data statistics (max and min values) of each cross validation fold.

2.4. Control Hyperparameters

Hyperparameters are free parameters, which give a handle in controlling the whole algorithm. We define two control hyperparameters: *history* H (number of days of input data required for one prediction) and *delay* D (time from the latest input data point to the day of SW prediction). For example, if the day of prediction is T, and data from T-3 to T-6 are used as input, our *history* is defined as 4 and the *delay* as 3. We have trained models with different combinations of delay ($D = 1$ to 4) and history ($H = 1$ to 4), resulting in 16 variants of the WindNet model. The meaning of the two control hyperparameters are illustrated in Figure 5.

2.5. Metrics for Comparison

Quantitatively, a set of metrics need to be defined to unambiguously quantify if the fit is good or bad. We define three metrics to estimate the goodness of fit ($\hat{y}$ = Prediction, y = Observation):

1. Mean square error (χ^2 value):

$$\chi^2 = \frac{1}{N}\sum_{i}^{N}(\hat{y}_i - y_i)^2, \tag{3}$$

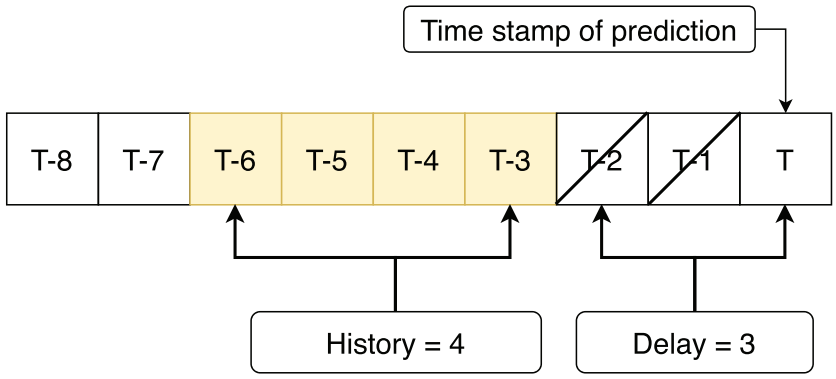

Figure 5. Each variant of the WindNet model is trained to predict the SW speed on day T, using input data (SW and SDO/AIA) from days in the range $[T - H - D + 1, T - D]$, where H and D denote the history and delay control hyperparameters, respectively.

where N is the no. of data points. However, we present the root-mean-square error (RMSE), defined as $\sqrt{\chi^2}$, in the units of km/s.

2. Reduced mean square error (χ^2_{red}):

$$\chi^2_{red} = \frac{1}{N}\sum_{i}^{N}\frac{(\hat{y}_i - y_i)^2}{\sigma_i^2}, \tag{4}$$

where σ_i denotes the standard deviation associated with each observation y_i. The standard deviation is computed over OMNI measurements for each day and reported in the data set.

3. Pearson correlation coefficient (r): The standard definition of correlation is

$$r = \frac{\sum_{i}^{N}(y_i - \bar{y})(\hat{y}_i - \hat{\bar{y}})}{\sigma_y \sigma_{\hat{y}}}, \tag{5}$$

where $\bar{y}$ and $\hat{\bar{y}}$ represent the mean values and $\sigma_y, \sigma_{\hat{y}}$ represent the standard deviation of the data set in consideration. To perform an average of correlation across all folds, we transform the data to Fischer's z space, perform the averaging, and then transform back—to prevent bias while performing average (Corey et al., 1998). The standard deviations are calculated in Fischer's z space, and propagated back to correlation.

The three metrics defined have their own advantages and drawbacks.

1. The predicted data are scaled between 0 and 1. Hence, even a large deviation, when squared, seems very small, if both the prediction and observation are <1. Thus, while χ^2 is a good minimizing function for training, it fails to perform well as a metric for good fit.
2. To counter the above case, the χ^2_{red} metric is used. This takes into account the inherent error in each measurement and scales the fit accordingly. A bad fit for a high error data point is acceptable, as the observation itself has high uncertainty, while a bad prediction on a low error data point is bad, since it serves as a much better point of comparison.
3. If the output were a naive mean value of the batch, the χ^2 and χ^2_{red} would still be reasonable—however, there would be no variance in the fit. Hence, the Pearson correlation r is used to understand the trend captured by the fitted curve. The exact fit values may not match, but if the trend is captured, the model is fairly good according to this metric.

Summarizing, Pearson r captures the trend but ignores any scaling error. χ^2 captures scaling errors but does not perform well on scaled data <1. And χ^2_{red} captures the errors by weighing them vis-a-vis the variance of observed data. These three metrics are used for comparing the models—that is, the r value of our proposed model should be higher, and the χ^2 and χ^2_{red} lesser than the benchmark models. Please note that errors (or spread) reported (for both the metrics, and activation evaluation) later on correspond to standard error (or uncertainty in the estimated mean), defined as

$$S(x) := \frac{\sigma(x)}{\sqrt{N(x)}},$$

where $\sigma(x)$ is the standard deviation derived from the sample, and $N(x)$ is the number of samples in the set.

Model performance, while accounting for timing errors, is an importance marker for capturing the response to dynamic events in the SW. Thus, we also compare the performance of our models through its ability to capture high-speed enhancements (HSE), as used in several texts (Bu et al., 2019; Owens et al., 2005; Reiss et al., 2016). We use the method as outlined in Jian et al. (2015) for finding out HSE. This is performed as follows:

1. Mark all time points that are more than 50 km/s faster than 1 day earlier.
2. Group each contiguous block of marked points as a distinct HSE and find the start and end time of each HSE.
3. For each HSE, find the minimum speed starting 2 days ahead of the HSE till the start of the HSE and mark it as the minimum speed (Vmin) of the HSE; find the maximum speed starting from the beginning of the HSE through 1 day after the HSE and mark it as the maximum speed (Vmax) of the HSE.
4. For each HSE, find the last time reaching Vmin and the first time reaching Vmax and mark them as the start and end time of an SIR.
5. For the regrouped SIRs, find the Vmin and Vmax for each SIR and mark the last time of highest speed gradient as the stream interface (SI), the boundary between slow and fast wind. Eliminate SIRs with the redundant SI time.
6. Reject any SIRs with Vmin faster than 500 km/s, or Vmax slower than 400 km, or speed increase less than 100 km/s.

Each HSE present in the observation, and captured by the model is called a true positive (TP), and those not captured by the model are called false negative (FN). Spurious HSE predictions by model are called false positives (FP). With these, we define the metric of comparison threat score (TS) as

$$TS = \frac{TP}{TP + FN + FP}. \tag{6}$$

Threat score is a proxy for the accuracy of forecast of any model. A model which predicts all the HSE perfectly (while not predicting any spurious HSE) has a TS of 1—thus, lower the TS, worse the model. For every cross validation set per model, the HSEs are identified and TS calculated—thereby giving us a mean TS and its uncertainty per model. Note that if the HSE (i.e., the peak of the enhancement) occurs very near the boundary, it would be missed by the algorithm due to our data partitioning scheme. Such HSE are discarded by benchmarking the $H = 1$, $D = 1$ Persistence model to give a TS = 1.0.

This study does not account for the effect of ICMEs (near-Earth interplanetary coronal mass ejections). There are 170 ICMEs reported within the time range considered in this study, affecting solar wind measurements in 336 days. In both model training and evaluation, we did not remove days for which there were ICMEs. The prediction of solar eruptions leading to CMEs and ICMEs is outside the scope of this study. Nevertheless, their occurrence impacts the solar wind measurements at L1. So for evaluation of the solar wind models in this paper, we decided to include even the days when ICMEs were present.

3. Modeling and Methods

3.1. Benchmark Models

We next describe various models taken as benchmarks for our proposed WindNet model. These benchmark models all operate as autoregressive models on the SW data only and do not use AIA images as input. The models (except 27 day persistence) are all corrected for the data gaps, thereby making the comparison reasonable.

- Naive mean value model.
- N day and 27 day Persistence model.
- Autoregression with XGBoost (Chen & Guestrin, 2016).
- Autoregression with support vector machines (SVMs).

3.1.1. Autoregression Using a "Mean Value"

One of the most basic benchmarks for any model is the comparison of the fit with a mean value model. This benchmark takes in the SW data and outputs the mean value of the whole batch. This model serves as the lowest benchmark that the proposed model should surpass, since untrained models output mean values.

Table 1
XGBoost Parameter Selection Using Grid Search

Parameter	Value
eta	[0.001, 0.01, 0.1, 0.8, 0.9, 1.0]
seed	0
objective	reg:linear
max_depth	200
lambda	[50, 10, 5, 1, 0.5, 0.05]

3.1.2. Persistence Model

The second benchmark model is persistence. The SW speed is fed in as input, and the same output is obtained. Such a model would show how long the data persists through time.

The N day persistence is calculated from $H+D-1$ days prior to prediction, to the day of prediction. As such, there is no individual dependence of the persistence model on H or D—rather, the dependence is on the combined value, thereby having degeneracy. This model is primarily used for determining how far into the future our models consistently give a good prediction, given an observation today, or observations starting today.

We also benchmark our results against 27 day persistence for 1 Carrington rotation, as it has been shown to be a good benchmark model in Owens et al. (2013). The 27 day persistence model operated on the complete SW data set (devoid of any gaps).

3.1.3. Autoregression Using XGBoost

The SW speed is autoregressed for different H and D using the XGboost algorithm (Chen & Guestrin, 2016). That is, the prediction $\hat{y}_{T+1}$ is given as $\hat{y}_{T+1} = f(\mathbf{x})$, where model input is $\mathbf{x} = (y_{T-H-D+1}, y_{T-H-D}, \cdots, y_{T-D})$, and the function $f()$ comprises the gradient-boosted decision trees. The various parameters set for the algorithm are shown in Table 1. The best model from the swept set of parameters is selected based on the lowest χ^2 value.

3.1.4. Autoregression Using SVMs

SVM is also used as a benchmark for good fit, since it has more non-linearity than decision trees due to the presence of kernels. Three kernels are used for benchmarking—Radial Basis function, Linear, and Polynomial kernel of degree 5. We use the Scikit-learn (Pedregosa et al., 2011) implementation of SVM in this work. The parameters were selected by grid search using the χ^2 value as the comparison metric. The best fitting parameters are shown in Table 2.

3.2. Proposed SW Model

The methodology followed here is to define a feature extractor, which reduces the dimensionality of AIA images into a set of generic features, and a regressor which then takes these abstract features to regress against the solar wind speed. The proposed model *WindNet* is a deep learning model constructed using a ConvNet and a RNN. Here, a pretrained GoogLeNet (Szegedy et al., 2015) model is used as a ConvNet feature extractor, and the obtained embeddings are fed into a variant of a RNN, called LSTM (Hochreiter & Schmidhuber, 1997) model (GoogLeNet weights were obtained from http://www.deeplearningmodel.net/).

GoogLeNet is a ConvNet (Szegedy et al., 2015) developed for the ImageNet (Deng et al., 2009) competition. This competition provides a huge database of labeled images with the objective of classifying them into different categories. As mentioned in section 1, ConvNets work by detecting patterns at multiple scales

Table 2
Support-Vector Regression-Parameter Selection

Kernel	Parameter	Value
RBF	C	1e+4
RBF	gamma	0.001
Linear	C	1e+4
Polynomial	C	1e+4
Polynomial	degree	5

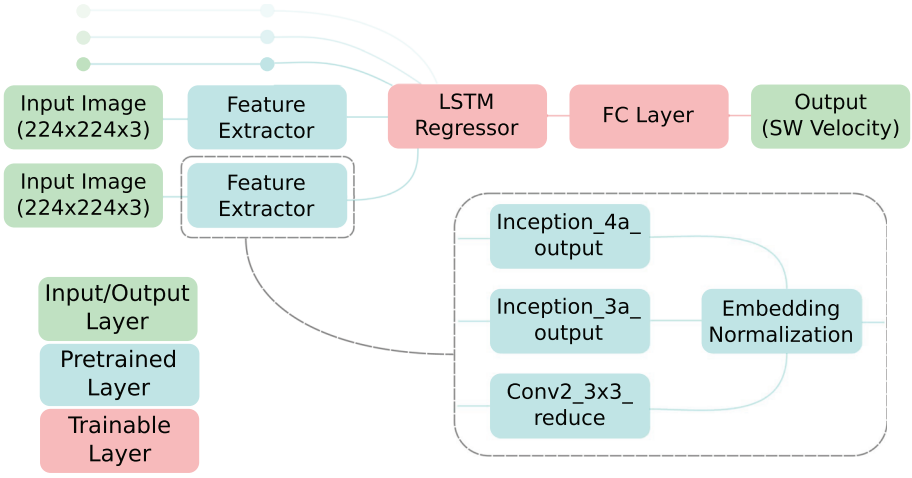

Figure 6. WindNet architecture using GoogLeNet and an LSTM.

in the input. This scale over which patterns are detected is given by the size of the convolution kernel. ConvNets generally have convolutions performed sequentially—thus, at a given layer, the sensitivity is only to a particular scale. However, GoogLeNet, for the first time, introduces us to the concept of the *Inception module* (Szegedy et al., 2015). Essentially, this module has, at each layer, convolutions using different kernel sizes done in parallel. Thus, it provides sensitivity at multiple scales at the same time. This has been shown to outperform other models on the ImageNet14 data set (Szegedy et al., 2015). GoogLeNet has been trained on everyday objects. However, given the large volume of training data in ImageNet14, the initial layers of the network capture generic global features in the images (Goodfellow et al., 2016). As one goes deeper, the network captures features specific to the data set—which is not relevant to our data set. Thus, we use this pretrained network and generate an embedding corresponding to the AIA data. This technique is known in the literature as Transfer learning (Yosinski et al., 2014). We adopt a "multiresolution approach" to generate the embeddings—that is, responses from layers at different depths are taken, normalized and concatenated. The embeddings are then fed to an LSTM for regression against the SW speed. GoogLeNet has its weights fixed, while the LSTM (and a fully connected layer at the end) are trained. We use a single LSTM cell in our work. The model is developed using the Tensorflow package for Python (Abadi et al., 2015) and has been summarized in Figure 6.

The training details for the algorithm are summarized in Table 3.

3.3. Activation Visualization

There exist techniques in the DL literature to visualize neurons in hidden layers which are preferentially activated for a given input—this activation can be extrapolated back to the given input to understand, which regions of the input data have large impact on the prediction. These methods rely primarily on the gradient of output with respect to each input pixel, thereby providing an approximation of regions most responsible

Table 3
WindNet Parameter Selection

Parameter	Value
Cost function	$\chi^2(\hat{y}, y)+\chi^2_{red}(\hat{y}, y)$
Optimizer	Adam
Learning rate	5e−4
Dropout for LSTM	0.5
L2 Norm coefficient	1e−6
No. of hidden units in one LSTM cell	400
No. of iterations	300
Feature length from GoogleNet	832

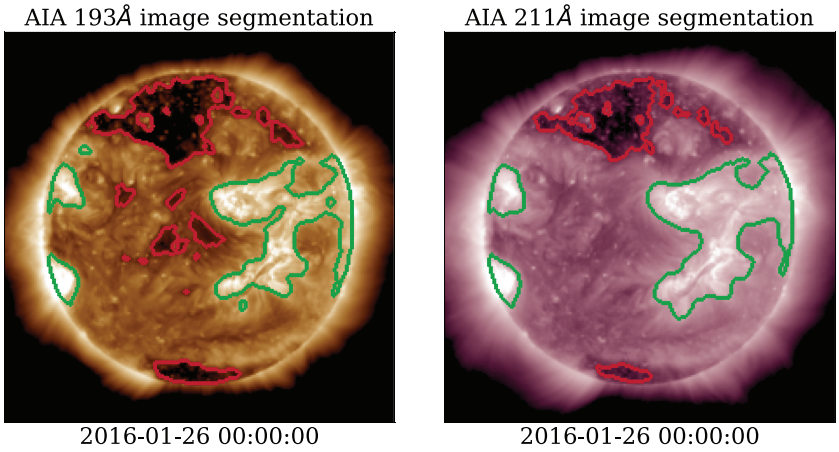

Figure 7. A representative visualization of the segmentation map of 193 Å channel (left) and 211 Å channel (right) using classical computer vision algorithms. The overplotted green contours enclose AR, and the red contours CH. The segmentation maps are created separately for the AR and CH.

for an increase or decrease in the output. The methods, while not being perfect visualizers, are a window into the workings of the network. In this work, we use Gradient Class Activation Maps (Grad-CAM) (Selvaraju et al., 2017) as a visualization technique. Grad-CAM are maps generated by a pointwise multiplication of the average gradient per channel of output vis-a-vis a given convolution layer with the corresponding ConvNet layer activation. The obtained map is then passed through a Rectified Linear Unit (ReLU, namely, $f(x) = max([0, x])$) activation function to obtain the activation map. The maps are averaged across channels, and then scaled up to the dimensions of the input image for comparison. This method produces activation maps of the model on the input data. These activation maps are subsequently used to generate a metric for the determination of influence of the CHs and ARs.

3.3.1. Generating Binary Masks

A simple metric for understanding the influence of a particular set of features for a regression problem would be to look at the mean value of the activation on that particular set of features across all data points and look for the variation of this mean value over days leading to prediction. Therefore, it is of great importance to segment out the CHs and the ARs to generate binary maps. To obtain the CH segmentation map, we use Otsu thresholding (Otsu, 1979). This thresholding assumes the presence of two distinct classes of pixels intensities, essentially Gaussian and tries to find an intensity value which would maximize the interclass variance (or alternatively, minimize the intraclass variance). We use stacked thresholding—that is, a preliminary threshold to segment out the approximate region of the coronal holes first, and then another threshold to segment out the coronal holes from this subset of the image.

The AR segmentation is far more nontrivial. Otsu thresholding picks out spurious areas as "active regions." Hence, we apply a five-class Gaussian Mixture Model (Pedregosa et al., 2011) on the pixel intensities to segment out the ARs. The Gaussian with the highest mean is found to segment out the ARs well. A representative set of segmentation maps overplotted on the EUV data is shown in Figure 7.

With these binary maps, we simply perform a pointwise multiplication of our activation values on a given image with its CH and AR map respectively, while also scaling by the total area of segmentation. The scaling by area of CH and AR is done to remove dependence on the absolute size of these regions, and obtain a normalized quantity. We then take the mean value over the image, and across all data sets to obtain a single scalar to quantify the activation at ARs and CHs, across the days of history for both fast and slow SW. The activation plots are constructed for the training set (for better statistics), since the generalizability of the model is captured in its performance on the test set (or the cross-validation set).

4. Results

4.1. Model Benchmarking

From Tables 4 through 7, we have summarized the performance of WindNet, as well as the benchmark autoregressive models for the metrics defined—correlation (r), RMSE, χ^2_{red}, and TS, respectively. We see that

Table 4
Correlation Comparison of Our Model Predictions With the Benchmark Models

(H, D)	WindNet 193	WindNet 211	XGBoost	Persistence	SVM Linear	SVM RBF	SVM Polynomial	Naive mean
(1, 1)	0.28 ± 0.03	0.34 ± 0.02	0.73 ± 0.01	0.76 ± 0.01	0.76 ± 0.01	0.76 ± 0.01	0.56 ± 0.01	—
(1, 2)	0.37 ± 0.03	0.42 ± 0.03	0.36 ± 0.02	0.43 ± 0.02	0.43 ± 0.02	0.43 ± 0.02	0.34 ± 0.01	—
(1, 3)	0.47 ± 0.01	0.48 ± 0.02	0.12 ± 0.02	0.19 ± 0.02	0.19 ± 0.03	0.19 ± 0.02	0.18 ± 0.03	—
(1, 4)	0.46 ± 0.03	0.52 ± 0.04	0.02 ± 0.03	0.07 ± 0.03	0.08 ± 0.03	0.07 ± 0.03	0.07 ± 0.03	—
(2, 1)	0.37 ± 0.05	0.42 ± 0.02	0.76 ± 0.01	0.43 ± 0.02	0.79 ± 0.01	0.79 ± 0.01	0.56 ± 0.01	—
(2, 2)	0.47 ± 0.02	0.39 ± 0.06	0.40 ± 0.02	0.19 ± 0.02	0.47 ± 0.02	0.47 ± 0.02	0.34 ± 0.02	—
(2, 3)	0.46 ± 0.03	0.53 ± 0.03	0.16 ± 0.02	0.07 ± 0.03	0.22 ± 0.02	0.21 ± 0.02	0.18 ± 0.03	—
(2, 4)	0.51 ± 0.03	0.48 ± 0.03	0.01 ± 0.03	0.03 ± 0.03	0.08 ± 0.03	0.08 ± 0.03	0.05 ± 0.03	—
(3, 1)	0.41 ± 0.04	0.51 ± 0.03	0.76 ± 0.01	0.19 ± 0.02	0.79 ± 0.01	0.79 ± 0.01	0.57 ± 0.01	—
(3, 2)	0.46 ± 0.03	0.47 ± 0.02	0.39 ± 0.02	0.07 ± 0.03	0.47 ± 0.02	0.47 ± 0.02	0.34 ± 0.01	—
(3, 3)	0.47 ± 0.03	0.53 ± 0.03	0.15 ± 0.02	0.03 ± 0.03	0.22 ± 0.02	0.22 ± 0.02	0.16 ± 0.02	—
(3, 4)	0.46 ± 0.03	0.54 ± 0.03	0.03 ± 0.04	0.01 ± 0.03	0.08 ± 0.03	0.09 ± 0.03	0.05 ± 0.03	—
(4, 1)	0.47 ± 0.04	0.54 ± 0.03	0.75 ± 0.01	0.07 ± 0.03	0.79 ± 0.01	0.79 ± 0.01	0.57 ± 0.01	—
(4, 2)	0.48 ± 0.03	0.52 ± 0.02	0.38 ± 0.01	0.03 ± 0.03	0.47 ± 0.02	0.47 ± 0.02	0.34 ± 0.01	—
(4, 3)	0.45 ± 0.04	0.55 ± 0.03	0.16 ± 0.02	0.01 ± 0.03	0.22 ± 0.02	0.22 ± 0.01	0.15 ± 0.03	—
(4, 4)	0.48 ± 0.04	0.50 ± 0.03	0.04 ± 0.04	-0.02 ± 0.03	0.09 ± 0.03	0.09 ± 0.03	0.06 ± 0.04	—

Note. The 27 day persistence gives a correlation of 0.456 ± 0.02. Models that do not have a correlation value are given "—." The p values are all less than 10^{-7} for WindNet variants, and less than 10^{-2} for the benchmark models.

WindNet outperforms the benchmarks over combinations where delay is generally more than 1—that is, where the autoregressive models do not have the immediately preceding solar wind speed available. In fact, for larger delays and histories, WindNet shows consistent performance, while other models fail to perform a reasonable prediction. The best performance of WindNet is for a history-delay combination of (4, 3), wherein the correlation is ≈0.55, and the spread is 0.03—his is for 211 Å. Similarly, the best fit using 193 Å data occurs for a combination of (2, 4), with a correlation of 0.51, with a spread of 0.03.

The Naive mean model has no variance, so there would be no correlation associated with it—however, it is presented for the sake of completeness. Autoregressive SVM using an RBF kernel seems to perform better given the SW speed closer to the day of prediction, but falter as more delay is induced. The linear SVM performs as well as the nonlinear RBF kernel, but the polynomial kernel fails to get a good fit. The 27 day persistence is a set of just five models—thus, this performance is stated in the caption of the respective Tables.

4.2. WindNet Prediction

In this section we investigate the variation in prediction for our WindNet models. The model with highest correlation, as mentioned previously, is for a history of 4 and delay of 3 for 211 Å. As can be seen in the Table 4, there seems to be a subtle trend of an increase in correlation with history for a given delay for short delays. The predictions for models with delay smaller than history seem mostly consistent within the error bars. For the 193 Å model in Table 4, it can be seen that an increase in delay for a given history results in almost a consistent prediction correlation for high history models (again, within the error bars—though the mean values do not seem to follow an ordered trend), except in the case of 1 day history, where the correlation increases. This trend of increase in delay for a given history is largely followed in the 211 Å data, though the 4 day history seems to be the most consistent in this case within the errors, and the best performing. In general, the expectation would be an increase in correlation with increasing history, and some form of variation due to an increase in delay. The variation in performance with history for small delays is fairly consistent between both 193 and 211 Å with only the actual correlation values being different—however, larger delay models do not have the same variation in performance for 193 and 211 Å. The 211 Å, in fact, seems to be a better channel for SW prediction, since the corresponding models have higher correlation means and smaller standard deviations. Short-delay and short-history models (e.g., 1 day history and delay) do not perform as well as models with larger history and delay (e.g., 4 day history and 3 day delay), since the solar wind is yet to arrive at L1. The 193 Å data show a peak in correlation at 2 day history and 4 day delay. The 211 Å data shows a similar peak at 4 day history and 3 day delay.

Table 5
RMSE Comparison of Our Model Predictions With the Benchmark Models

(*H*, *D*)	WindNet 193	WindNet 211	XGBoost	Persistence	SVM Linear	SVM RBF	SVM Polynomial	Naive mean
(1, 1)	97.01 ± 4.02	96.64 ± 4.27	60.31 ± 0.62	62.02 ± 1.05	57.68 ± 0.72	57.68 ± 0.73	74.17 ± 1.32	88.05 ± 2.08
(1, 2)	92.13 ± 2.88	89.45 ± 2.68	83.47 ± 1.00	95.35 ± 1.60	80.60 ± 1.28	80.55 ± 1.28	83.74 ± 1.75	87.77 ± 2.20
(1, 3)	83.70 ± 1.77	87.34 ± 3.85	90.33 ± 1.30	113.81 ± 2.39	87.77 ± 1.74	87.78 ± 1.75	87.97 ± 1.91	88.04 ± 2.42
(1, 4)	84.33 ± 2.31	85.94 ± 4.67	92.14 ± 1.73	122.20 ± 3.13	89.06 ± 1.98	89.07 ± 1.97	88.91 ± 1.94	88.29 ± 2.46
(2, 1)	96.31 ± 4.87	91.12 ± 2.30	57.87 ± 0.65	95.35 ± 1.60	54.27 ± 0.93	54.19 ± 0.92	74.64 ± 1.80	87.77 ± 2.20
(2, 2)	90.80 ± 2.85	102.85 ± 9.00	83.48 ± 0.83	113.81 ± 2.39	78.94 ± 1.58	78.98 ± 1.57	84.36 ± 1.90	88.04 ± 2.42
(2, 3)	86.21 ± 2.12	83.38 ± 2.78	91.86 ± 1.19	122.20 ± 3.13	87.09 ± 1.73	87.17 ± 1.70	87.91 ± 1.85	88.29 ± 2.46
(2, 4)	86.24 ± 2.63	86.53 ± 2.27	93.11 ± 1.14	125.16 ± 3.12	88.68 ± 1.95	88.72 ± 1.97	88.77 ± 1.88	88.86 ± 2.46
(3, 1)	93.35 ± 4.33	82.60 ± 1.75	57.80 ± 0.80	113.81 ± 2.39	54.40 ± 1.01	54.34 ± 1.00	73.84 ± 1.70	88.04 ± 2.42
(3, 2)	88.17 ± 1.81	85.46 ± 2.63	84.10 ± 0.86	122.20 ± 3.13	78.59 ± 1.67	78.55 ± 1.63	83.91 ± 1.85	88.29 ± 2.46
(3, 3)	87.04 ± 1.25	83.97 ± 3.04	91.58 ± 1.05	125.16 ± 3.12	86.68 ± 1.76	86.75 ± 1.73	87.91 ± 1.79	88.86 ± 2.46
(3, 4)	87.21 ± 2.17	81.21 ± 1.86	92.72 ± 1.28	126.79 ± 2.92	88.72 ± 1.75	88.62 ± 1.80	88.96 ± 1.67	89.14 ± 2.41
(4, 1)	84.19 ± 2.83	80.27 ± 2.07	59.14 ± 0.82	122.20 ± 3.13	54.52 ± 1.03	54.48 ± 1.04	74.28 ± 2.07	88.29 ± 2.46
(4, 2)	86.42 ± 1.98	83.06 ± 2.51	83.78 ± 0.74	125.16 ± 3.12	78.47 ± 1.81	78.45 ± 1.78	84.16 ± 1.88	88.86 ± 2.46
(4, 3)	88.32 ± 1.93	80.28 ± 3.05	91.00 ± 1.25	126.79 ± 2.92	86.81 ± 1.59	86.82 ± 1.63	88.25 ± 1.68	89.14 ± 2.41
(4, 4)	82.93 ± 1.72	85.34 ± 3.10	92.34 ± 1.34	128.23 ± 2.96	88.87 ± 1.46	88.78 ± 1.58	89.41 ± 1.40	89.43 ± 2.29

Note. The 27 day persistence gives a RMSE of 93.14 ± 4.43.

A summary of RMSE is shown in Table 5, and a similar summary of χ^2_{red} is shown in Table 6. The TS is tabulated in Table 7. The TS table shows that our proposed model has a maximum of 0.357 ± 0.03. The low TS may be explained better by a careful observation of Figure 8. This is a plot of one of the cross validation models using 211 Å, having the highest correlation, with 4 day history and 3 day delay of data. With 10 TP, 2 FP, and 13 FN, the model is seen to have a TS of 0.4, a correlation with observation of 0.61, RMSE of 76.4 km/s and χ^2_{red} of 19.35. Here, we see that there are many more HSE present in the observed wind speed, which seem to be missing from the prediction. However, upon careful observation,it may be seen that many of the observed HSE do correspond to an enhancement in the wind speed of the predictor—either at the exact time step, or with a lag/lead of 3 to 4 days. However, the predicted values do not show the drastic enhancement more than the prescribed thresholds. Thus, these events are not marked as HSE.

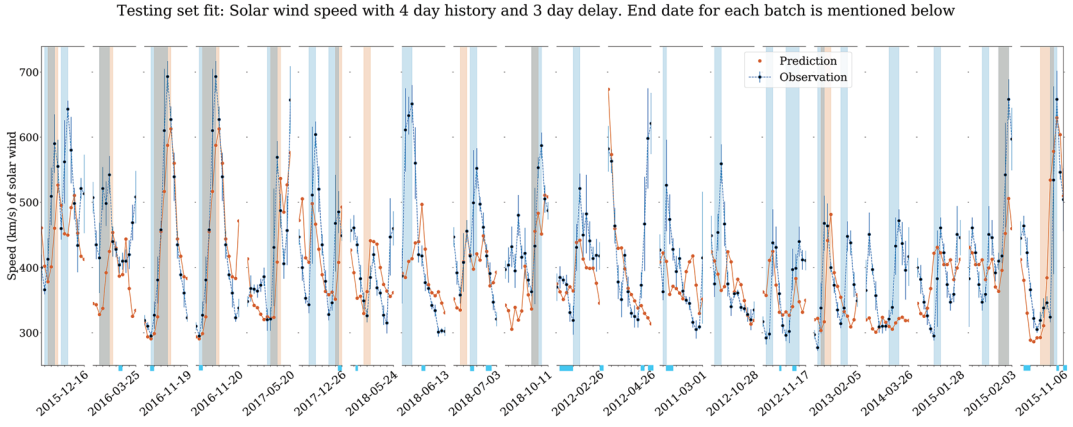

Figure 8. Wind speed prediction plot from one of the cross validation models, using 4 days of image data with 3 days of delay. On the *x* axis, the ending date of each batch is shown. Since batches are randomly assigned to each cross-validation fold, the dates are not kept in order. The model has a correlation value of 0.61, RMSE of 76.4 km/s, and χ^2_{red} of 19.35. The error bars are measurement errors of the wind speed observations. The HSE are highlighted by their start and end times—blue for the observed wind speed and red for the predicted wind speed. The blue bars below the plot indicate ICMEs.

Table 6
The χ^2_{red} Comparison of Our Model Predictions With the Benchmark Models

(*H*, *D*)	WindNet 193	WindNet 211	XGBoost	Persistence	SVM Linear	SVM RBF	SVM Polynomial	Naive mean
(1, 1)	33.38 ± 4.10	29.63 ± 1.36	23.18 ± 0.82	24.50 ± 0.73	21.20 ± 0.77	21.20 ± 0.77	35.17 ± 1.91	41.98 ± 3.71
(1, 2)	28.43 ± 1.98	30.80 ± 2.05	44.10 ± 1.75	57.63 ± 1.76	41.13 ± 1.77	41.09 ± 1.79	44.54 ± 2.65	39.16 ± 3.92
(1, 3)	25.82 ± 1.39	26.27 ± 1.97	51.67 ± 2.08	81.61 ± 2.74	48.79 ± 2.17	48.80 ± 2.20	49.09 ± 2.57	40.34 ± 3.25
(1, 4)	26.83 ± 2.31	26.01 ± 2.47	54.21 ± 2.55	95.19 ± 4.25	50.63 ± 2.40	50.64 ± 2.39	50.48 ± 2.48	44.99 ± 3.65
(2, 1)	48.07 ± 11.96	43.75 ± 7.58	21.17 ± 0.67	57.63 ± 1.76	18.63 ± 0.77	18.59 ± 0.77	35.44 ± 2.38	39.16 ± 3.92
(2, 2)	47.32 ± 8.43	90.87 ± 38.04	44.05 ± 1.05	81.61 ± 2.74	39.41 ± 1.49	39.45 ± 1.50	45.15 ± 2.44	40.34 ± 3.25
(2, 3)	30.09 ± 3.42	31.60 ± 3.84	53.88 ± 2.23	95.19 ± 4.25	48.38 ± 2.00	48.46 ± 1.99	49.37 ± 2.45	44.99 ± 3.65
(2, 4)	31.41 ± 3.97	37.90 ± 4.07	54.86 ± 1.92	99.64 ± 5.18	49.71 ± 1.90	49.76 ± 1.91	49.83 ± 1.96	49.46 ± 3.69
(3, 1)	47.56 ± 6.01	29.16 ± 2.87	21.13 ± 0.65	81.61 ± 2.74	18.71 ± 0.63	18.67 ± 0.62	34.60 ± 1.90	40.34 ± 3.25
(3, 2)	33.51 ± 3.20	36.66 ± 3.87	45.06 ± 1.14	95.19 ± 4.25	39.34 ± 1.41	39.31 ± 1.41	44.97 ± 2.29	44.99 ± 3.65
(3, 3)	43.29 ± 4.10	37.54 ± 7.33	53.09 ± 1.86	99.64 ± 5.18	47.47 ± 1.55	47.56 ± 1.57	48.89 ± 1.92	49.46 ± 3.69
(3, 4)	42.35 ± 5.76	31.52 ± 5.16	53.40 ± 2.06	101.18 ± 5.53	48.83 ± 1.79	48.73 ± 1.85	49.12 ± 1.90	50.96 ± 5.72
(4, 1)	31.16 ± 1.76	31.18 ± 3.36	22.25 ± 0.40	95.19 ± 4.25	18.92 ± 0.54	18.89 ± 0.54	35.33 ± 2.23	44.99 ± 3.65
(4, 2)	27.99 ± 3.35	29.59 ± 3.94	44.34 ± 0.80	99.64 ± 5.18	38.87 ± 1.24	38.86 ± 1.25	44.81 ± 1.93	49.46 ± 3.69
(4, 3)	36.98 ± 3.82	26.83 ± 2.20	51.47 ± 2.06	101.18 ± 5.53	46.73 ± 1.53	46.76 ± 1.62	48.36 ± 2.02	50.96 ± 5.72
(4, 4)	31.90 ± 3.38	32.58 ± 4.83	52.66 ± 2.21	103.16 ± 5.66	48.70 ± 1.81	48.62 ± 1.94	49.35 ± 2.06	52.29 ± ± 4.11

Note. The 27 day persistence gives a χ^2_{red} of 51.69 ± 9.14.

The WindNet performance on the error metrics, though, largely complements the correlation performance, and show WindNet has better performance than the benchmark models for delays larger than 1 day in most cases.

4.3. Activation Visualization

Using Grad-CAM activation maps (described in section 3.3.1) to visualize the activation, we analyze the activation of our models for various days of data. Figure 9 shows a sample Grad-CAM map from a fast and slow wind prediction using 211 Å data, and a similar map is shown from 193 Å prediction in Figure 10, for

Table 7
HSE Threat Score Comparison

(*H*, *D*)	WindNet 193	WindNet 211	XGBoost	Persistence	SVM Linear	SVM RBF	SVM Polynomial	Naive mean
(1, 1)	0.081 ± 0.023	0.112 ± 0.040	0.776 ± 0.037	1.000 ± 0.000	0.748 ± 0.038	0.748 ± 0.038	0.263 ± 0.026	0.0
(1, 2)	0.042 ± 0.022	0.150 ± 0.042	0.329 ± 0.012	0.858 ± 0.037	0.288 ± 0.027	0.271 ± 0.028	0.162 ± 0.035	0.0
(1, 3)	0.167 ± 0.008	0.140 ± 0.041	0.061 ± 0.015	0.351 ± 0.021	0.0	0.0	0.029 ± 0.013	0.0
(1, 4)	0.212 ± 0.042	0.206 ± 0.047	0.036 ± 0.018	0.199 ± 0.024	0.0	0.0	0.0	0.0
(2, 1)	0.203 ± 0.064	0.227 ± 0.029	0.711 ± 0.036	0.858 ± 0.037	0.850 ± 0.022	0.845 ± 0.021	0.292 ± 0.027	0.0
(2, 2)	0.293 ± 0.022	0.297 ± 0.040	0.423 ± 0.096	0.351 ± 0.021	0.461 ± 0.017	0.449 ± 0.017	0.148 ± 0.033	0.0
(2, 3)	0.225 ± 0.036	0.198 ± 0.037	0.030 ± 0.027	0.199 ± 0.024	0.0	0.0	0.020 ± 0.011	0.0
(2, 4)	0.239 ± 0.045	0.282 ± 0.044	0.043 ± 0.038	0.215 ± 0.022	0.0	0.0	0.0	0.0
(3, 1)	0.310 ± 0.051	0.259 ± 0.031	0.753 ± 0.024	0.351 ± 0.021	0.850 ± 0.026	0.844 ± 0.026	0.323 ± 0.026	0.0
(3, 2)	0.237 ± 0.048	0.292 ± 0.029	0.472 ± 0.107	0.199 ± 0.024	0.426 ± 0.018	0.408 ± 0.024	0.113 ± 0.029	0.0
(3, 3)	0.328 ± 0.038	0.287 ± 0.017	0.116 ± 0.047	0.215 ± 0.022	0.0	0.0	0.0	0.0
(3, 4)	0.357 ± 0.031	0.294 ± 0.026	0.024 ± 0.022	0.236 ± 0.026	0.0	0.0	0.0	0.0
(4, 1)	0.286 ± 0.037	0.309 ± 0.027	0.737 ± 0.034	0.199 ± 0.024	0.849 ± 0.032	0.845 ± 0.034	0.292 ± 0.031	0.0
(4, 2)	0.298 ± 0.040	0.200 ± 0.039	0.428 ± 0.083	0.215 ± 0.022	0.431 ± 0.024	0.428 ± 0.025	0.157 ± 0.037	0.0
(4, 3)	0.289 ± 0.070	0.200 ± 0.056	0.115 ± 0.044	0.236 ± 0.026	0.0	0.015 ± 0.009	0.011 ± 0.010	0.0
(4, 4)	0.251 ± 0.049	0.314 ± 0.080	0.035 ± 0.022	0.307 ± 0.033	0.0	0.0	0.007 ± 0.006	0.0

Note. The 27-day persistence model gives a TS of 0.506 ± 0.029. Cases with TS 0.0, imply a value less than 1e−3.

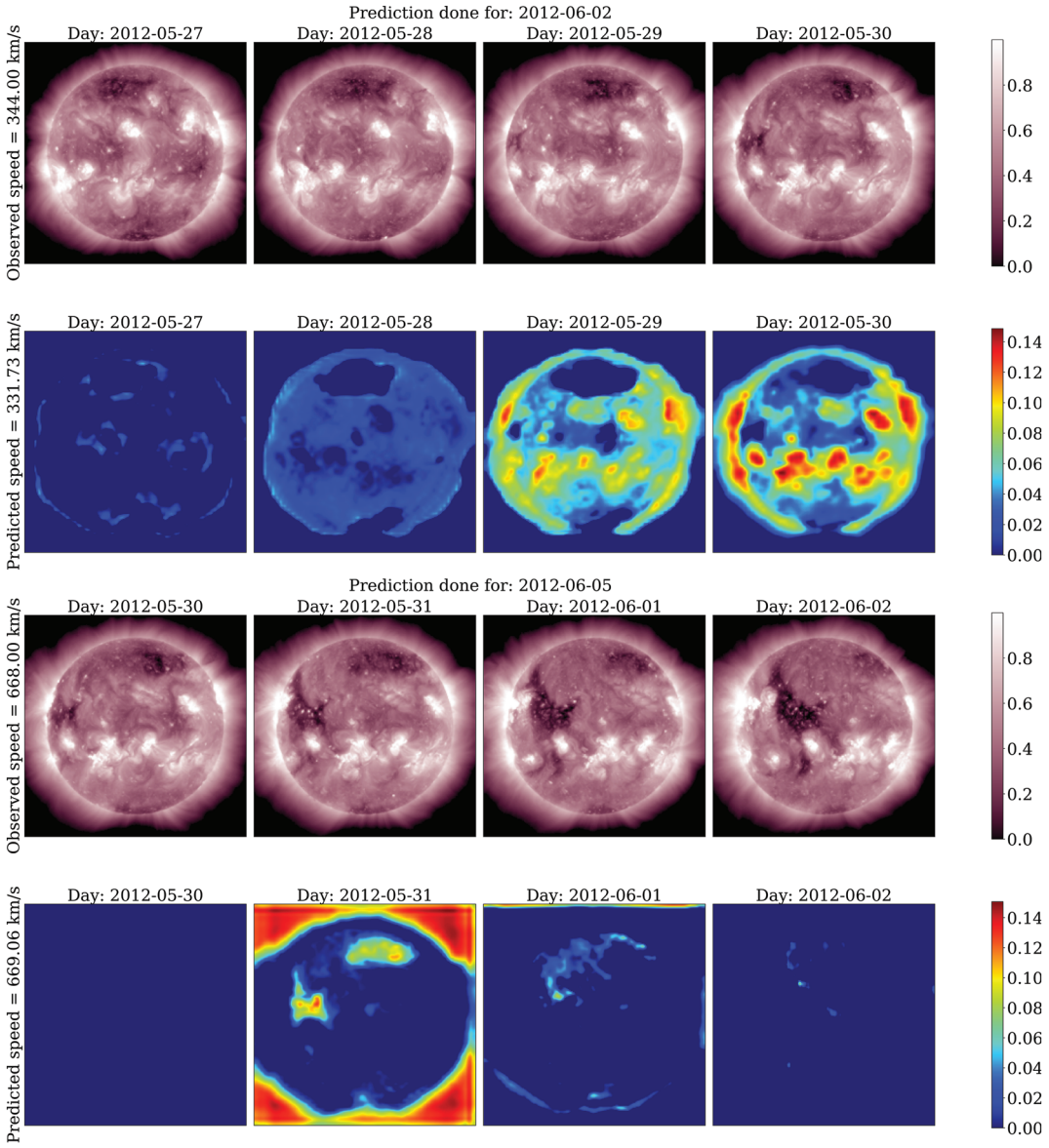

Figure 9. GC activation maps for a fast (top) and slow wind (above) prediction using 211 Å data, with the color map corresponding to each row given on the right. The activation maps and the images have been rescaled between 0 and 1 rowwise for ease of comparison. For the fast wind prediction, note how the maximum activation occurs at the CH, 3 to 4 days prior to prediction, which seems to match with the correlations obtained in the literature (Vršnak et al., 2007). The slow wind, on the other hand activates the AR closer to the prediction, with no activation at the CH.

comparison. We see that the CHs are activated for the prediction of the fast wind, and the ARs are predominantly activated for the slow wind prediction. The CH peak activation for fast wind occurs 3 to 4 days prior to prediction, which seems to corroborate with the correlations independently obtained (Vršnak et al., 2007). The slow wind activation is peaked at the AR close to the day of prediction (and also at the earliest day prior to prediction for 211 Å data), with activation at other regions of the Sun further away from prediction. We hypothesize this might be due to bias of the LSTM to the most recent input to the network—but this is still a hypothesis.

To understand the statistics of activation given to CHs and ARs, we look at the mean activation value (as described previously), and plot it for "fast-wind" and "slow-wind" predictions from the model. While each cross validation model set will have its own activation plot, we present the plot for both 193 and 211 Å models using 4 day history of data with 3 days of delay. We also plot the activation for the models using 4 day history

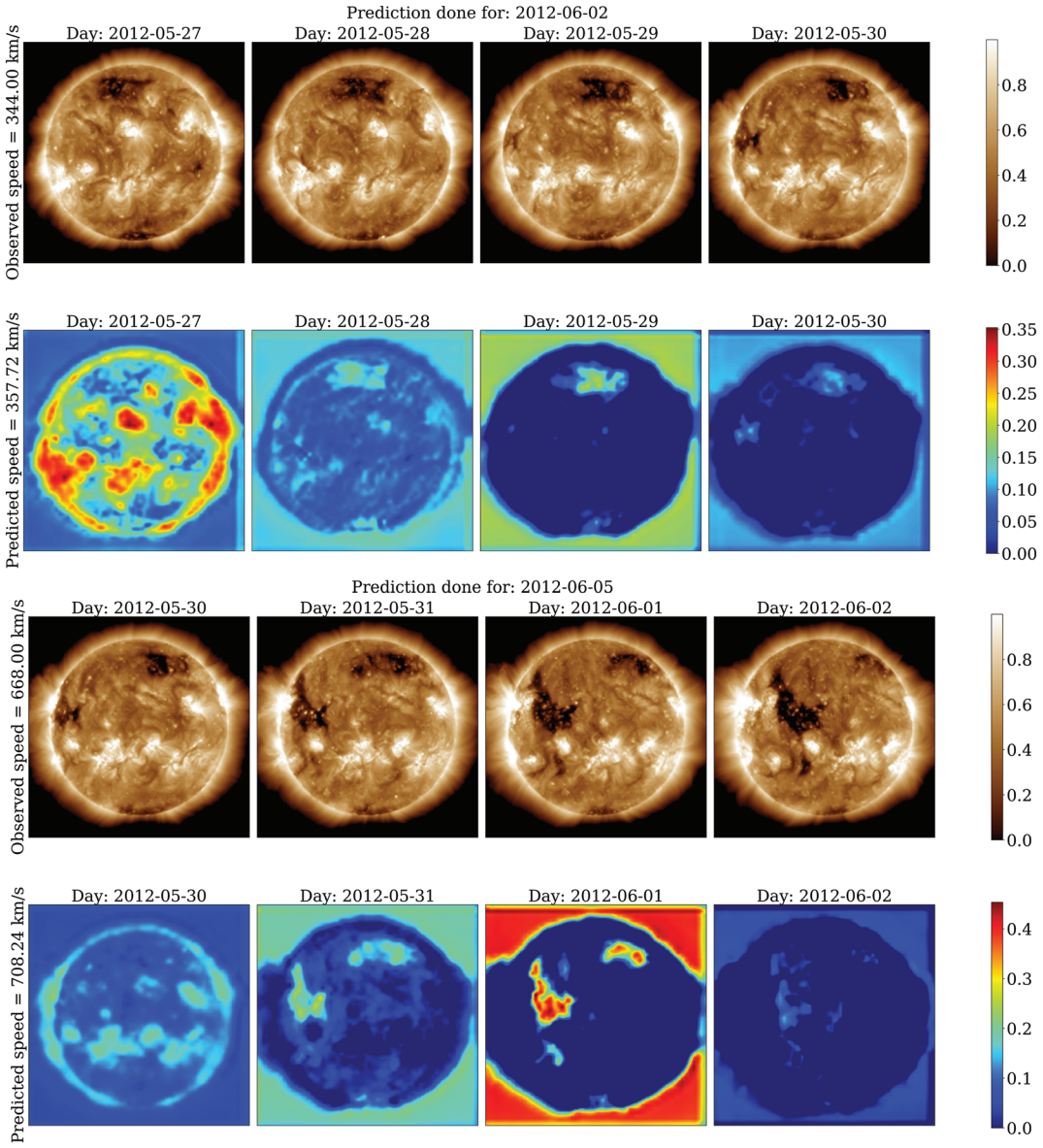

Figure 10. GC activation maps for a fast (top) and slow wind (above) prediction using 193 Å data, with the color map corresponding to each row given on the right. The activation maps and the images have been rescaled between 0 and 1 rowwise for ease of comparison. For the fast wind prediction, note how the maximum activation occurs at the CH, 3 to 4 days prior to prediction, which seems to match with the correlations obtained in the literature (Vršnak et al., 2007). The slow wind, on the other hand activates the AR both closer and further away from prediction, and activated at the small CH on the closest day to prediction. However, other regions of the quiet Sun show a higher activation further away from the day of prediction. The slow wind activation is quite mixed and unclear when compared with the fast wind activation.

of data with 2 days of delay, since it shows consistent (and good) performance using both 193 and 211 Å data. These trends are shown in Figures 11 and 12, respectively.

It can be seen from these plots that the fast SW induces greater activation at the CHs closer to the day of prediction, and the activation (at CH) decreases as we go farther into the past—however, for the 211 Å model, the activation shows slight increase. The fast wind also seems to activate the AR at much further times—for both 193 and 211 Å. Note, however, that the peak CH activation is larger than the peak AR activation for 193 Å—for 211 Å data, they are consistent within the errors in Figure 11. For the same parameters in Figure 12, CH peaks at 3 days prior to prediction for both the channels and then goes down. Interestingly, however, the AR also seems to be activated to a similar level, but much further away from prediction time.

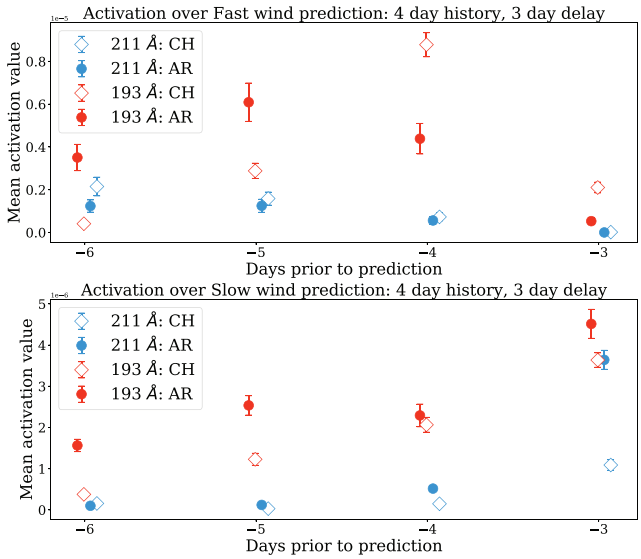

Figure 11. Variation of mean activation for the 4 day history and 2 day delay model, for a fast and slow SW prediction respectively. The activation is shown for models using 193 and 211 Å data, respectively. The activation is shown over CH and AR alone. The error bars indicate the standard error on the mean value, estimated from the standard deviation of the sample of activations. Please note that the error bars here represent 3*S*, that is, thrice the standard error to make sure they are visible. Those activations with seemingly no error bars have very small errors.

For the slow wind, activation for ARs remains high for much longer duration than the CHs—however, the peak occurs closer to the day of prediction, rather than further away from prediction. This trend is seen in both Figures 11 and 12.

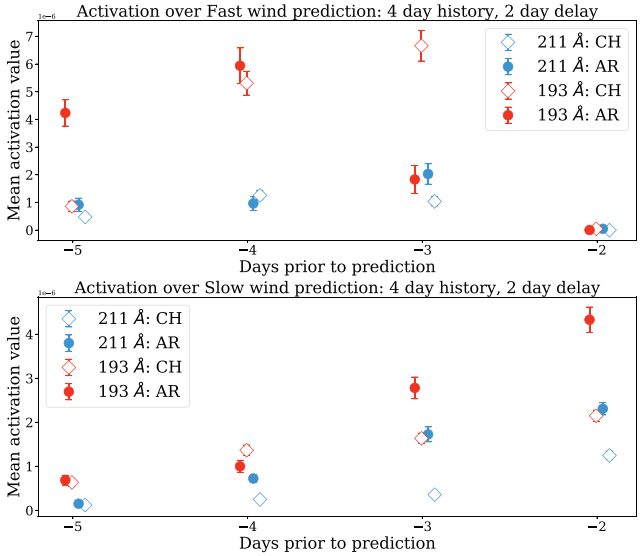

Figure 12. Variation of mean activation for the 4 day history and 3 day delay model, for fast and slow SW prediction, respectively. The activation is shown for models using 193 and 211 Å data, respectively. The activation is shown over CH and AR alone. The error bars indicate the standard error on the mean value, estimated from the standard deviation of the sample of activations. Please note that the error bars here represent 3*S*, that is, thrice the standard error to make sure they are visible. Those activations with seemingly no error bars have very small errors.

5. Discussion

The problem of SW prediction can be approached in two ways—one, through purely theoretical modeling of the mechanism and fine tuning parameters to fit observations, and two, through purely empirical modeling and attempting to extract the physics. We propose the WindNet to empirically model SW speed using AIA imagery data alone. We are able to predict the SW speed better with the 211 Å data and obtain a correlation of 0.55 with the observed wind speed in the cross validation. The best performing models using 193 Å and 211 Å both outperform most of the larger delay benchmark models, and the 27 day persistence model. The χ^2_{red}, which accounts for uncertainty in the measurement itself, indicates that our best models outperform the 27 day persistence and are only slightly worse off than an autoregressive model with a single day delay—more so for lead time predictions of 3–4 days. The activation plots seem to suggest the trained WindNet model pays attention to certain solar features consistent with heuristic expectations from solar wind theory, for example,the peak in activation at the CH 3-4 days prior to prediction. However, the significance of interpretation of activation values depend on a couple of other factors:

- Fitting error of our model: We still have a maximum correlation of 0.51(0.55) for the 193 Å(211 Å) data.
- Visualization: The Grad-CAM used in this work gives a very coarse localization of activation, and thus may not point to precise origin of the particular kinds of wind.
- Segmentation: Defining a region as CH accurately is difficult with intensity values alone—ideally, one would require extrapolated magnetic field lines to check for CHs. Thus, an accurate definition of CHs based on intensity is required. In this work, we attempted to automate both the CH and AR definitions using histogram analysis automatically; thus, there is bound to be some form of uncertainty. Hence, better segmentation methods may accurately capture the entire activation within a CH or an AR and give a much better estimate of activation per unit area.

At first glance, our WindNet may appear to not outperform existing models in terms of the metrics used. However, comparing our model to existing models (like the regressive models of Rotter et al., 2015, or Wang & Sheeley, 1990) would be an apples-to-oranges comparison, since

- we perform predictions over multiple Carrington rotations on the whole 8 year data set;
- our prediction target is the daily averaged SW speed, and as such must be compared to daily averaged predictions by other models;
- we perform fivefold cross validation on this data set. However, due to a lack of confidence intervals in the previous results, we are unable to check if our results are statistically different from the existing models.
- Our model is built entirely using open source software, and the codebase may be used by the community at large to improve upon the results.

Thus, any benchmarking of our model must be done with models undergoing the same data preparation procedure, the same span of data and at the same cadence. We thus do not compare our results with the existing aforementioned models.

To overcome this limitation, we propose empirical benchmark models not unlike the existing empirical solar wind prediction models. In this regard, WindNet shows reasonable performance vis-a-vis the benchmark models—however, there are numerous improvements possible.

- Data preparation: As $H + D$ increases, more samples are discarded (as explained in section 2.4). This may be made more efficient by performing the cross validation (CV) first, and then splitting into folds later with the downside of high memory consumption.
- ICME mitigation: Our random assignment of fivefold cross validation is to ensure the ICMEs are distributed uniformly across all the folds, thereby influencing all the CVs equally. Due to inadequate number of ICME samples, we do not characterize them.
- Network architecture: Better architectures may be designed to improve the prediction vis-a-vis the observations, or more novel ML methods may be employed for a direct prediction.
- Visualization: Visualization of ML models is an hot area of research in the ML community—thus, more accurate visualization techniques may be expected to emerge in coming years.
- TS evaluation: As seen in section 4, the HSE capturing algorithm misses many potential enhancements due to the speed increases not satisfying the absolute speed change criteria. Hence, the TS evaluation should be taken with caution.

This work is a first step toward training and testing various ML models for predicting other SW target parameters, such as proton density, temperature, and magnetic field (specifically, B_z). The code and data used in this work is open sourced (model may be found on github: https://github.com/Vishal-Upendran/WindNet). Our publicly released source code promotes reproducible research by allowing others to reproduce the results presented here. This includes data partitioning, cross-validation, model training, and evaluation. This code base can be built upon by other researchers to further improve the performance of solar wind prediction models. Furthermore, with the ever-increasing research on Interpretable AI, this codebase may be used by researchers to come up with various methods of visualizations to quantify regions of solar wind emergence.

Acknowledgments
We acknowledge use of NASA/GSFC's Space Physics Data Facility's OMNIWeb service, and OMNI data, and the AIA data is available on the Stanford Digital repository. U. V. would like to thank Alex Varghese and Mahendra Khened (Medical image Reconstruction Laboratory, Department of Engineering Design, IIT Madras) for long discussions on activation visualization and Interpretable ML. U. V would also like to thank Dattaraj Dhuri (Department of Astronomy and Astrophysics, TIFR-Mumbai) for independent verification of codes and results. U. V. would like to thank Durgesh Tripathi (Inter University Centre for Astronomy and Astrophysics, Pune, India) for providing computing facility, and insightful comments on the statistics of results. The authors would also like to acknowledge the two anonymous referees, whose comments helped substantially improve the manuscript. S. H. and U. V acknowledge support from the Max-Planck Partner Group Program and the Ramanujan Fellowship SB/S2/RJN-73. M. C. M. C. acknowledges support from NASA's SDO/AIA (NNG04EA00C) contract to LMSAL. AIA is an instrument onboard SDO, a mission for NASA's Living With a Star program.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Altschuler, M. D., & Newkirk, G. (1969). Magnetic fields and the structure of the solar corona. I: Methods of calculating coronal fields. *Solar Physics*, *9*, 131–149. https://doi.org/10.1007/BF00145734

Bradski, G. (2000). The OpenCV library. Dr. Dobb's Journal of Software Tools.

Bu, X., Luo, B., Shen, C., Liu, S., Gong, J., Cao, Y., & Wang, H. (2019). Forecasting High-Speed solar wind streams based on solar extreme ultraviolet images. *Space Weather*, *17*, 1040–1058. https://doi.org/10.1029/2019SW002186

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY: ACM. https://doi.org/10.1145/2939672.2939785

Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Corey, D. M., Dunlap, W. P., & Burke, M. J. (1998). Averaging correlations: Expected values and bias in combined Pearson rs and Fisher's z transformations. *The Journal of General Psychology*, *125*(3), 245–261. https://doi.org/10.1080/00221309809595548

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *CVPR09*.

Deng, L., & Yu, D. (2014). *Deep learning: Methods and applications*: Now Foundations and Trends. https://doi.org/10.1561/2000000039

Galvez, R., Fouhey, D. F., Jin, M., Szenicer, A., Muñoz-Jaramillo, A., Cheung, M. C. M., et al. (2019). A machine-learning data set prepared from the NASA solar dynamics observatory mission. *The Astrophysical Journal Supplement*, *242*(1), 7. https://doi.org/10.3847/1538-4365/ab1005

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT Press. https://www.deeplearningbook.org

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

Jian, L. K., MacNeice, P. J., Taktakishvili, A., Odstrcil, D., Jackson, B., Yu, H.-S., et al. (2015). Validation for solar wind prediction at Earth: Comparison of coronal and heliospheric models installed at the CCMC. *Space Weather*, *13*, 316–338. https://doi.org/10.1002/2015SW001174

Krieger, A. S., Timothy, A. F., & Roelof, E. C. (1973). A coronal hole and its identification as the source of a high velocity solar wind stream. *Solar Physics*, *29*(2), 505–525. https://doi.org/10.1007/BF00150828

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436.

Lemen, J. R., Title, A. M., Akin, D. J., Boerner, P. F., Chou, C., Drake, J. F., et al. (2012). The Atmospheric Imaging Assembly (AIA) on the solar dynamics observatory (SDO). *Solar Physics*, *275*(1), 17–40. https://doi.org/10.1007/s11207-011-9776-8

Linker, J. A., Miki, Z., Biesecker, D. A., Forsyth, R. J., Gibson, S. E., Lazarus, A. J., et al. (1999). Magnetohydrodynamic modeling of the solar corona during whole sun month. *Journal of Geophysical Research*, *104*(A5), 9809–9830. https://doi.org/10.1029/1998JA900159

NASA (2017). Space weather. https://www.nasa.gov/mission_pages/rbsp/science/rbsp-spaceweather.html

Otsu, N. (1979). A threshold selection method from Gray-Level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, *9*(1), 62–66. https://doi.org/10.1109/TSMC.1979.4310076

Owens, M. J., Arge, C. N., Spence, H. E., & Pembroke, A. (2005). An event-based approach to validating solar wind speed predictions: High-speed enhancements in the Wang-Sheeley-Arge model. *Journal of Geophysical Research*, *110*, A12105. https://doi.org/10.1029/2005JA011343

Owens, M. J., Challen, R., Methven, J., Henley, E., & Jackson, D. R. (2013). A 27 day persistence model of near-Earth solar wind conditions: A long lead-time forecast and a benchmark for dynamical models. *Space Weather*, *11*, 225–236. https://doi.org/10.1002/swe.20040

Owens, M. J., Spence, H. E., McGregor, S., Hughes, W. J., Quinn, J. M., Arge, C. N., et al. (2008). Metrics for solar wind prediction models: Comparison of empirical, hybrid, and physics-based schemes with 8 years of L1 observations. *Space Weather*, *6*, S08001. https://doi.org/10.1029/2007SW000380

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

Pesnell, W. D., Thompson, B. J., & Chamberlin, P. C. (2012). The solar dynamics observatory (SDO). *Solar Physics*, *275*, 3–15. https://doi.org/10.1007/s11207-011-9841-3

Reiss, M. A., Temmer, M., Veronig, A. M., Nikolic, L., Vennerstrom, S., Schngassner, F., & Hofmeister, S. J. (2016). Verification of high-speed solar wind stream forecasts using operational solar wind models. *Space Weather*, *14*, 495–510. https://doi.org/10.1002/2016SW001390

Riley, P., Linker, J. A., Mikić, Z., Lionello, R., Ledvina, S. A., & Luhmann, J. G. (2006). A comparison between global solar magnetohydrodynamic and potential field source surface model results. *The Astrophysical Journal*, *653*, 1510–1516. https://doi.org/10.1086/508565

Rotter, T., Veronig, A. M., Temmer, M., & Vršnak, B. (2012). Relation between coronal hole areas on the Sun and the solar wind parameters at 1 AU. *Solar Physics*, *281*, 793–813. https://doi.org/10.1007/s11207-012-0101-y

Rotter, T., Veronig, A. M., Temmer, M., & Vršnak, B. (2015). Real-time solar wind prediction based on SDO/AIA coronal hole data. *Solar Physics*, *290*(5), 1355–1370. https://doi.org/10.1007/s11207-015-0680-5

Schwenn, R. (2006). Space weather: The solar perspective. *Living Reviews in Solar Physics*, *3*(1), 2. https://doi.org/10.12942/lrsp-2006-2

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via Gradient-Based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 618–626). https://doi.org/10.1109/ICCV.2017.74

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems* (pp. 3104–3112).

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*.

Temmer, M., Hinterreiter, J., & Reiss, M. A. (2018). Coronal hole evolution from multi-viewpoint data as input for a STEREO solar wind speed persistence model. *Journal of Space Weather and Space Climate*, *8*, A18. https://doi.org/10.1051/swsc/2018007

Vršnak, B., Temmer, M., & Veronig, A. M. (2007). Coronal holes and solar wind high-speed Streams: I. Forecasting the solar wind parameters. *Solar Physics*, *240*, 315–330. https://doi.org/10.1007/s11207-007-0285-8

Walt, S. A., Colbert, S. C., & Varoquaux, G. A. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Wang, Y.-M., & Sheeley, N. R. Jr. (1990). Solar wind speed and coronal flux-tube expansion. *The Astrophysical Journal*, *355*, 726–732.

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

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (pp. 3320–3328).