Zhelavskaya, I., Vasile, R., Shprits, Y., Stolle, C., Matzka, J. (2019): Systematic Analysis of Machine Learning and Feature Selection Techniques for Prediction of the Kp Index. - *Space Weather*, *17*, 10, pp. 1461—1486.

DOI: http://doi.org/10.1029/2019SW002271

Space Weather

RESEARCH ARTICLE
10.1029/2019SW002271

Key Points:
- We tested a number of different machine learning algorithms for Kp prediction and validated them using K-fold cross validation
- Feature selection methods allow reducing the number of input variables by up to a factor of 16
- Information from solar wind measurements at L1 is not sufficient for long-term predictions of Kp

Correspondence to:
I. S. Zhelavskaya,
irina.zhelavskaya@gmail.com

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

Received 6 JUN 2019
Accepted 9 SEP 2019
Accepted article online 10 SEP 2019
Published online 28 OCT 2019

Systematic Analysis of Machine Learning and Feature Selection Techniques for Prediction of the Kp Index

I. S. Zhelavskaya[1,2], R. Vasile[1], Y. Y. Shprits[1,2,3], C. Stolle[1,4], and J. Matzka[1]

[1]GFZ Potsdam, Potsdam, Germany, [2]Institute of Physics and Astronomy, University of Potsdam, Potsdam, Germany, [3]Earth, Planetary and Space Sciences, University of California, Los Angeles, Los Angeles, CA, USA, [4]Institute of Earth and Environmental Science, University of Potsdam, Potsdam, Germany

Abstract The Kp index is a measure of the midlatitude global geomagnetic activity and represents short-term magnetic variations driven by solar wind plasma and interplanetary magnetic field. The Kp index is one of the most widely used indicators for space weather alerts and serves as input to various models, such as for the thermosphere and the radiation belts. It is therefore crucial to predict the Kp index accurately. Previous work in this area has mostly employed artificial neural networks to nowcast Kp, based their inferences on the recent history of Kp and on solar wind measurements at L1. In this study, we systematically test how different machine learning techniques perform on the task of nowcasting and forecasting Kp for prediction horizons of up to 12 hr. Additionally, we investigate different methods of machine learning and information theory for selecting the optimal inputs to a predictive model. We illustrate how these methods can be applied to select the most important inputs to a predictive model of Kp and to significantly reduce input dimensionality. We compare our best performing models based on a reduced set of optimal inputs with the existing models of Kp, using different test intervals, and show how this selection can affect model performance.

1. Introduction

The Kp index is one of the most widely used global measures of geomagnetic activity. It is used as an input to many scientific applications, including the parameterization of ionospheric ion outflow (Yau et al., 2011) and aurora particle precipitation (Emery et al., 2008) in the ionosphere, thermosphere (Bruinsma et al., 2018), hot plasma particle density (Denton et al., 2016; Korth et al., 1999), cold plasma density in the plasmasphere (Goldstein et al., 2014; Maynard & Chen, 1975; Pierrard et al., 2009; Zhelavskaya et al., 2017), plasmapause location (Carpenter & Anderson, 1992), and radiation belt models and wave parameterizations (Agapitov et al., 2015; Brautigam & Albert, 2000; Orlova et al., 2014; Ozeke et al., 2014; Shprits et al., 2007) in magnetospheric physics, among others. It is therefore important to predict the Kp index accurately in order to produce most reliable forecasts in the aforementioned areas.

A number of models for Kp index prediction have been developed in the past decades. All these models use solar wind parameters measured at L1 as an input, and the Kp index is their only output. Various methods were employed to develop these models. The first two models predicting Kp, Costello (1998) and Boberg et al. (2000), used feedforward neural networks (FNNs), a type of artificial neural networks often used for solving regression, classification, and clusterization problems. Wing et al. (2005) employed FNNs and recurrent neural networks to develop a predictive model of Kp and have shown that both types of networks have a similar performance. Recent studies by Bala and Reiff (2012) and Wintoft et al. (2017) have also employed FNNs for the Kp prediction. Tan et al. (2018) have used long-short-term memory, an artificial recurrent neural network architecture used in the field of deep learning that is powerful for processing sequential data such as sound, natural language, or other complex time series. Alternative methodologies, such as the nonlinear autoregressive moving average with exogenous inputs algorithm and support vector machines, were employed in (Balikhin et al., 2001; Boaghe et al., 2001; Ji et al., 2013); and Wang et al. (2015).

The models listed above were developed using different machine learning techniques. They were trained and tested using data from different time intervals (depending on the data availability but also on the choice of the training and testing time intervals made by the authors). The inputs to the models were also constructed differently in different studies. This makes it difficult to compare models and objectively evaluate progress.

Thus, it is also unclear whether another new modeling technique or a different way of constructing the inputs can improve the quality of the predictions. It has also not been systematically investigated whether it is the use of solar wind measurements at L1 as input that sets a limit to the prediction accuracy, since the single point observations around L1 cannot fully capture the complex solar wind-magnetosphere coupling.

In this study, we investigate what brings the most improvement to the model accuracy and whether there is a limit to the prediction accuracy set by using solar wind measurements at L1 as input to a model. We perform such an analysis by applying different machine learning modeling techniques to develop predictive models of Kp for different prediction horizons up to 12 hr and comparing their performance. We focus our analysis on three algorithms: Linear Regression (LR), artificial FNN (Bishop, 2006; Goodfellow et al., 2016), and Gradient Boosting (GB) (Friedman, 2001). We use LR as a benchmark for comparison with nonlinear (FNN) and ensemble-based (GB) models. We use the same validation technique and the same time intervals to train and validate the models and in doing so create an unbiased technique to validate and compare models. This analysis step helps determine to what extent the chosen modeling approach affects the accuracy of predictions for different prediction horizons.

We also compare how different approaches for constructing input variables to the models affect the accuracy of predictions. Additionally, we test different machine learning and information theoretical methods for optimal input selection. The motivation to explore methods for optimal input selection (also called *feature selection* in machine learning) is to identify the most important solar wind drivers to predict the Kp index. As the number of inputs to a model (or features) grows very quickly when their cadence is increased or when more time history is included, it becomes difficult to interpret the physical importance of each input variable and can also increase the training time. Feature selection methods allow us to find a subset of the most important inputs that contain a sufficient amount of information to model the target variable (here, Kp) and, at the same time, to achieve good accuracy of the predictions. We investigate feature selection procedures based on the following methods: Fast Function Extraction (FFX) (McConaghy, 2011), Random Forest (RF) (Ho, 1995), Mutual Information Maximization (MIM) (Bollacker & Ghosh, 1996), and Maximum Relevancy Minimum Redundancy (MRMR) (Ding & Peng, 2005; Peng et al., 2005). The RF algorithm is often used for feature selection and is implemented in many machine learning packages. The feature selection method based on the FFX algorithm makes use of the intrinsic feature selection of FFX and the K-fold cross-validation and is developed in this work. The last two methods are based on the concept of mutual information (MI).

Finally, we compare the best performing models to the previous predictive models of Kp using different time intervals for testing to illustrate the importance of choice of testing interval and how it can affect the model performance.

The structure of the paper is as follows: In section 2, we provide a brief description of the machine learning algorithms and feature selection methods used in this work. In section 3, we describe the data, the training and validation methodology, and the hyperparameter selection for all considered methods. In section 4, we present the comparison of the machine learning methods and results of the selection of input variables. We also present the comparison of our best performing models to the existing ones. Finally, we summarize the main results of the paper in section 5.

2. Machine Learning Background

This section provides a brief description of the algorithms used in this work. The description is intended to provide the reader with an overview of these methods, while more details of each method can be found in the references therein. Section 2.1 provides an overview of the algorithms used in this study to develop the predictive model of the Kp index, namely, GB, FNNs, and LR. Section 2.2 describes the procedures for optimal feature selection that we employ in this work based on FFX, RF, and information theoretical methods, namely, MIM and MRMR.

2.1. Machine Learning Algorithms for Model Development

FNN. FNNs are a type of artificial neural network inspired by the way biological neural networks in our brain process information (Bishop, 2006; Goodfellow et al., 2016). FNNs are used for solving regression, classification, and clusterization problems. In regression problems, they are used to find multivariate nonlinear relationships between the input and the output variables. An FNN consists of an input layer, an output layer,

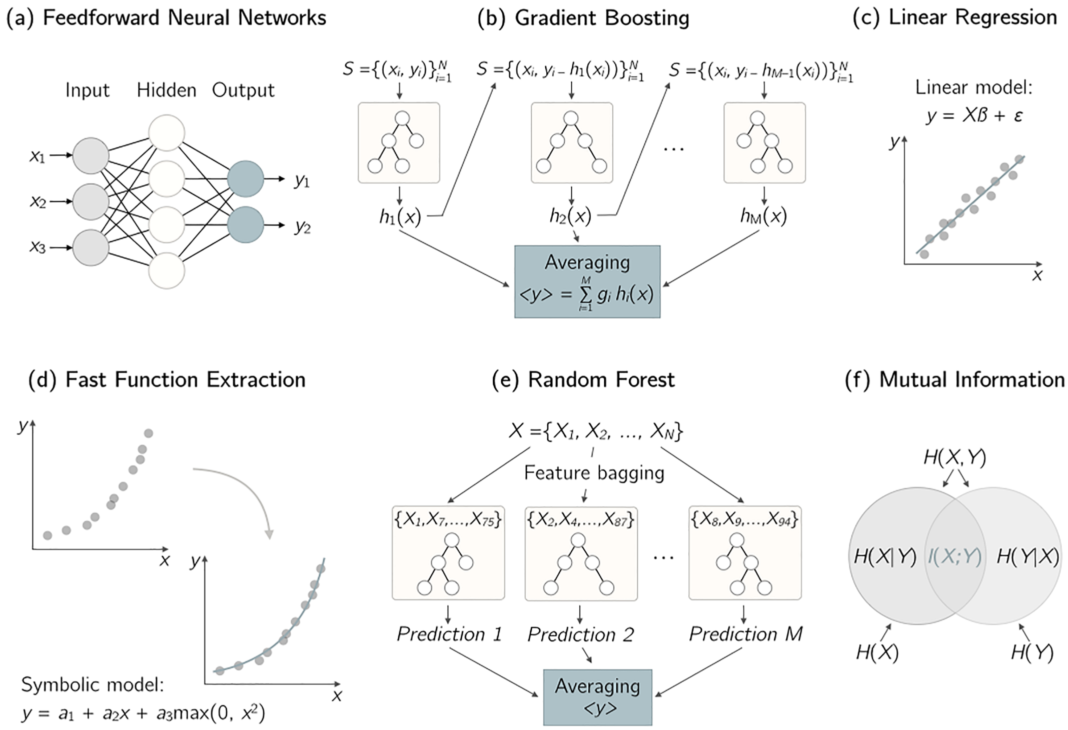
(a) Feedforward Neural Networks
Input
Hidden
Output
x_1
x_2
x_3
y_1
y_2
(b) Gradient Boosting
$S = \{(x_i, y_i)\}_{i=1}^{N}$
$S = \{(x_i, y_i - h_1(x_i))\}_{i=1}^{N}$
$S = \{(x_i, y_i - h_{M-1}(x_i))\}_{i=1}^{N}$
…
$h_1(x)$
$h_2(x)$
$h_M(x)$
Averaging
$<y> = \sum_{i=1}^{M} g_i\, h_i(x)$
(c) Linear Regression
Linear model:
$y = X\beta + \varepsilon$
y
x
(d) Fast Function Extraction
y
x
y
x
Symbolic model:
$y = a_1 + a_2x + a_3\max(0, x^2)$
(e) Random Forest
$X = \{X_1, X_2, \ldots, X_N\}$
Feature bagging
$\{X_1, X_7, \ldots, X_{75}\}$
$\{X_2, X_4, \ldots, X_{87}\}$
…
$\{X_8, X_9, \ldots, X_{94}\}$
Prediction 1
Prediction 2
Prediction M
Averaging
<y>
(f) Mutual Information
$H(X,Y)$
$H(X|Y)$
$I(X;Y)$
$H(Y|X)$
$H(X)$
$H(Y)$

Figure 1. A schematic representation of (a) Feedforward Neural Network, (b) Gradient Boosting, (c) Linear Regression, (d) Fast Function Extraction, (e) Random Forest, (f) Mutual Information methods for model construction and feature selection used in this work.

and a number of hidden layers. Its schematic representation is shown in Figure 1a for the case of a network with one hidden layer. Each node in the layer is a neuron, which can be thought of as the basic processing unit of a neural network. In the FNN, each neuron is connected to all neurons in the preceding and succeeding layer; neurons of the same layer are not connected. Each connection between two neurons in an FNN has a weight associated to it. The information in the FNN moves only forward, from the input to the output with no feedback connections or loops. An FNN is applied to solve a specific problem after it is trained on a set of data pertaining to this problem. Training is an optimization procedure, in which the weights (the internal parameters of the network) are tuned using the training set of data so that the difference between the network output and the actual target variable is minimal. A description of FNNs applied to space physics problems can be found in Chu, Bortnik, Li, Ma, Angelopoulos, et al. (2017), Chu, Bortnik, Li, Ma, Denton, et al. (2017), Zhelavskaya et al. (2017), and Zhelavskaya et al. (2018). In this work, we use the MATLAB Deep Learning Toolbox to train neural networks (https://mathworks.com/products/deep-learning.html).

GB. GB is an ensemble machine learning algorithm for solving classification and regression problems. It combines the outputs of many simple prediction models to obtain a more accurate prediction (Friedman, 2001). Boosting iteratively produces a hierarchy of these models, as shown in Figure 1b. These models are referred to as weak classifiers/regressors. In GB, each weak model is typically a shallow decision tree (Breiman et al., 1984). In a regression problem, the first model is trained to fit the actual output, and then every new model is trained to fit the residual between the actual target variable and the prediction value given by the previous model. The prediction is then given by a weighted linear combination of outputs of each weak model. The weights of that linear combination and internal parameters of each decision tree of the ensemble, such as the maximum tree depth, are determined during the training phase. In GB, training is performed using gradient descent minimization of the target cost function in a functional space. We use the python xgboost library to implement the GB algorithm (https://xgboost.readthedocs.io/en/latest/).

LR. LR is a linear approach to modeling the relationship between a scalar-dependent variable and one or more explanatory variables. The fit to the data is obtained using a maximum likelihood estimator (Bishop, 2006). LR algorithms may perform worse than other nonlinear methods (e.g., neural networks) in practice, since they are only able to model a linear relationship between input and output variables. On the other hand, especially in regression problems, they provide an analytic expression and allow for the interpretability of the result. We use the python sklearn library for LR (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html).

2.2. Feature Selection Procedures

Feature selection based on FFX. The FFE algorithm is a deterministic scalable algorithm for symbolic regression problems (McConaghy, 2011). Symbolic regression is a type of regression, in which the model is constructed by searching the space of mathematical expressions for the optimal combination of expressions, that is, the combination of mathematical blocks that best fits to a given data set in terms of accuracy and simplicity. These mathematical blocks are usually represented by a set of basis functions (e.g., polynomial functions, and exponential functions) that is used and combined iteratively to obtain more complex functions of the input variables. The FFX algorithm uses a deterministic procedure to build new regression functions in each iteration and allows for faster training times and prototyping, in comparison to other more general symbolic regression schemes, such as genetic programming (Koza, 1992). The choice of the basis functions determines the class of solutions for a specific problem. Typically, a polynomial basis augmented by nonlinear basis elements (such as max or min functions) is used to model quasi linear systems and still obtain much better performance than simpler LR schemes. In that sense, symbolic regression constitutes a natural nonlinear extension to LR models, by allowing an interpretation of the result and, at the same time, improving the accuracy by adding nonlinear elements to the solution.

The FFX algorithm performs an internal feature selection by building symbolic expressions using only a subset of input variables. As the input data set changes, different variables may appear in the expression of the final trained model. In the context of a K-fold cross-validation procedure (described in section 3.3), a number of different models are obtained by training on different training/validation partitions. Given this set of models, we develop the following feature selection procedure. We fix a threshold integer value $k \in [1 \ldots N]$, where N is the total number of trained model instances ($N = 5 \times 10$ for 5-fold CV with 10 repetitions). Only those input variables that appear in at least k of the trained model instances are extracted. As k approaches N, the set of extracted variables is reduced: For $k = N$, only those variables that appear in every trained model are selected. Thus, the FFX algorithm provides a definitive number of selected features for each threshold k. Contrary to the RF feature selection procedure described below, no ranking among the selected variables is obtained (they are considered to be equally important). The library used for this algorithm can be found at https://github.com/natekupp/ffx.

Feature selection based on RF. The forest of random trees, or more commonly, the RF algorithm, is an ensemble machine learning algorithm for classification and regression problems that can also be used for the input selection (Breiman, 2001; Ho, 1995). RF is based on decision trees (Breiman et al., 1984), similarly to GB, but in RF, each decision tree is fitted directly to the target variable; thus, all trees can be trained in parallel. Two points should be noted regarding the way the trees in the RF ensemble are constructed. First, each decision tree in the ensemble is built using the bootstrapped version of the initial training data set. Bootstrapping is an algorithm that produces replicas of a data set by performing random sampling with replacement. Therefore, each decision tree is built using a slightly modified version of the initial data set. Second, each decision tree is trained using a subset of inputs drawn randomly from the whole set of inputs, which makes the resulting trees uncorrelated to each other (Breiman, 2001). The output of the final model is the average of predictions made by all decision trees in the ensemble. Its schematic representation is shown in Figure 1e.

The RF algorithm can be used to rank the importance of variables in a regression or classification problem (Breiman, 2001) and perform optimal feature selection and therefore reduce the input dimensionality. The method is referred to as feature importance extraction and is based on the Gini importance or mean decrease impurity concept (Ho, 1995). The method returns an ordered list of input features according to the value of the mean decrease impurity, with the sum of the mean impurities over all variables being unity. To select a reduced number of input features a threshold value T, $0 < T < 1$, is fixed, and only those variables, the ordered cumulative sum of which is as close as possible to the threshold T, are selected. The selection of this threshold is usually performed empirically. We used the sklearn library to implement the RF regression algorithm (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html).

Feature selection based on MI. MI is a concept of information theory that can be used to study the relationships between different variables (usually input and target variables of a model). It is not a machine learning algorithm by itself but can be used for feature selection and, therefore, we present a brief description of it. MI between two variables X and Y is a measure that quantifies the amount of information obtained about one variable through the other variable. It is defined as

$$I(X,Y) = \int_X \int_Y p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy, \tag{1}$$

where $p(x,y)$ is the joint probability density function of X and Y and $p(x)$ and $p(y)$ are the marginal density functions. MI determines how similar the joint distribution $p(x,y)$ is to the product of the marginal distributions $p(x)$ and $p(y)$. If X and Y are independent, then $p(x,y)$ is equal to $p(x)p(y)$, and the integral in (1) is equal to zero. In practice, the probability distribution functions can be obtained by discretizing variables X and Y and using an alternative definition of MI utilizing the concept of Shannon entropy:

$$I(X,Y) = H(X) + H(Y) - H(X,Y), \tag{2}$$

where $H(X)$ and $H(Y)$ are entropies of X and Y, respectively, and $H(X,Y)$ is their joint entropy. Entropy is a measure of uncertainty of a variable, and entropies of discrete variables are defined as

$$\begin{aligned} H(X) &= -\sum p(x) \log p(x), \;\; H(Y) = -\sum p(y) \log p(y), \\ H(X,Y) &= -\sum \sum p(x,y) \log p(x,y). \end{aligned} \tag{3}$$

A more detailed description of MI and other concepts of information theory and their application in space physics can be found in Wing and Johnson (2019). Below, two methods based on the MI that are used in this work are described.

MIM. The MIM feature selection algorithm employs the concept of MI and selects input features that maximize the MI between them and the target variable. Formally, if $S_{t-1} = \left\{X_{f_1}, \ldots, X_{f_{t-1}}\right\}$ is the set of selected features at time step $t-1$, where f_i is the input feature selected at time step i, MIM selects the next input feature f_t by solving the following optimization problem:

$$f_t = \arg \max_{i \notin S_{t-1}} I(X_i, Y), \tag{4}$$

where X_i is the input variable that is not yet included in the set of selected features S_{t-1} and Y is the target variable. Simply stated, MIM selects input variables that have the largest MI with the target variable and ranks them according to their MI with the target variable. MIM makes the following assumptions:

- Assumption 1: The selected features X_S are independent and are also class-conditionally independent, given the unselected feature under consideration X_k (i.e., the knowledge of the unselected feature does not give additional knowledge of the selected features). Here, X_S is the reduced data set containing selected features.
- Assumption 2: All features are pairwise class-conditionally independent, that is, $p(X_i, X_j|Y) = p(X_i|Y)p(X_j|Y)$, meaning that $\sum I(X_j, X_k|Y)$ is zero.
- Assumption 3: All features are pairwise independent, that is, $p(X_i, X_j) = p(X_i)p(X_j)$, meaning that $\sum I(X_j, X_k)$ is zero.

These assumptions may not always hold in practice; therefore, MIM is not widely used as it can have a poor performance. Nevertheless, we include it for comparison and illustrative purposes.

Maximum Relevancy Minimum Redundancy (MRR). The MRMR algorithm also employs the concept of MI, similarly to MIM, but adds another term to the optimization problem in selecting the next input feature:

$$f_t = \arg \max_{i \notin S_{t-1}} I(X_i, Y) - \alpha \sum_{k=1}^{t-1} I(X_{f_k}, X_i), \quad \alpha = \frac{1}{(t-1)}, \tag{5}$$

where $I(X_{f_k}, X_i)$ denotes MI between already selected input variables X_{f_k} in S_{t-1} and candidates for the new input variable. Therefore, this method accounts not only for the “relevancy” of features, as in MIM, but also for the “redundancy” of information brought by new input features, that is expressed by the last term in (5). Even if MI of a new input variable with the target variable is large, it may be strongly correlated with already selected input variables in X_S and may therefore not bring in any new information. Therefore, other variables that minimize the redundancy factor with already selected inputs will be selected by MRMR. MRMR makes the assumptions 1 and 2 described above.

Both the MIM and MRMR feature selection algorithms provide a ranking of the the variables in the order of importance. The number of variables to be selected is usually chosen empirically.

Table 1
Considered Inputs for All Prediction Horizons

Solar wind and IMF parameter	$B, B_x, B_y, B_z, V_{sw}, nProt$
Aggregate functions	avg, min, max
Time windows, hr (here, current time is hr 0)	0–1, 1–2, …, 8–9
Other inputs	$\sin(2\pi T/24), \cos(2\pi T/24), \sin(2\pi D/365), \cos(2\pi D/365)$

3. Data and Methodology

3.1. Data

We use solar wind and interplanetary magnetic field (IMF) data from NASA's OMNIWeb data service during the time period of 1998–2017, to construct the input to our machine learning models. Specifically, we use 1-min resolution measurements of solar wind speed, proton density, total interplanetary magnetic field B, and interplanetary magnetic field components B_x, B_y, B_z in the GSM coordinate frame. Using solar wind data with 1-minute resolution allows better capturing spikes and minima in the solar wind parameter, compared to data with 5-min or 1-hr resolutions. The Kp index, that is, the target variable or the model output, is obtained from the GFZ Potsdam website (https://www.gfz-potsdam.de/en/kp-index/) and has a three-hr cadence.

3.2. Model Inputs

A variety of solar wind parameters and their derivatives, such as their time history or solar wind coupling functions (Newell et al., 2007), can be used as inputs to predictive models of Kp. Here, we consider a limited number of solar wind parameters and their time history as inputs. Specifically, we consider the solar wind speed (V_{sw}), proton density ($nProt$), IMF components B_x, B_y, B_z, and the total IMF magnitude B. We then construct minimum, maximum, and average values of these variables over 1-hr time windows (starting from 0-1 to 8-9 hr previous to the current time) and use them as inputs to our models. In some of the previous studies, 3-hr time windows were used to construct the inputs (e.g.,Bala & Reiff, 2012; Wintoft et al., 2017). We test how the window size affects the model performance by comparing the performance of models based on 1- and 3-hr inputs in section 4.2 (Figure 4).

To take into account possible seasonal variation, we also include indicators of day of the year and time of the day represented by $\sin(2\pi T/24), \cos(2\pi T/24), \sin(2\pi D/365), \cos(2\pi D/365)$, where T is the UT hour of the day and D is day of the year, following Wintoft et al. (2017).

In total, the considered variables comprise 166 input features. The output of the model is the Kp index with 3-hr cadence. One data sample corresponds to one Kp value in our data set (we do not interpolate Kp to 1-min cadence of the solar wind data); therefore, the number of data samples in the full data set for the 1998–2017 period is 58,439 and corresponds to the number of Kp values over this period. Please refer to Table 1 for a summary of the inputs.

3.3. Training and Validation Setup

We use a repeated K-fold cross-validation procedure with $K = 5$ folds and 10 repetitions in order to evaluate the performance of different models (i.e., obtain estimates of the mean and standard deviation of the training and validation errors) and compare them with each other.

In K-fold cross validation, the available data set is split into K folds, where one fold is used for validation and the remaining $K - 1$ folds are used for training (validation data are not shown to the model during training). This procedure is implemented K times with different validation fold each time. In a repeated K-fold cross-validation (CV), this procedure is repeated multiple times (here we choose 10), where in each repetition, the folds are split differently. After each repetition, model assessment metrics are computed (e.g., RMSE, and linear correlation coefficient), and then the scores from all repetitions are averaged to obtain the final model assessment score. In a fivefold CV with 10 repetitions, we train 50 model instances and compute/obtain 50 values of training and validation errors and use them to calculate the mean of model assessment scores. Such a repeated CV procedure produces a more robust assessment score than if CV is performed only once, and especially if only one hold-out test set is used instead of the CV procedure. A repeated CV also gives an idea about the variance of a model by examining the standard deviation of the model error. A high standard deviation indicates that the model produces different results when trained on different data splits and, therefore, has a high variance. Models with low standard deviation of error are desirable, since they perform similarly on different data splits and are therefore robust. Additionally, we withhold a separate test set comprising 10% of all data before the start of the CV procedure for the final model evaluation.

Since the data under consideration are a time series, the neighboring data points may be strongly correlated. Consequently, random splitting into training and validation sets may lead to correlations between these two sets. At the same time, if the data are split into CV folds sequentially, the distribution of the target variable in different validation folds can be significantly different, for example, it may occur that high Kp values are present only in the validation or only in the training data set. To avoid both of these unwanted scenarios, we implement an intermediate procedure. We first split all data into 35-day blocks sequential in time and then assign these 35-day blocks randomly to the CV folds. The reason for using a 35-day block length is to avoid the possible effect of 27-day recurrence caused by the rotation of the sun. Each block contains 280 measurements, and in total, we obtain 209 blocks using this procedure. Finally, the sizes of the training and validation sets for different CV splits comprise 41,918 and 10,640 data samples, respectively, and the test set 5880.

3.4. Hyperparameter Optimization

Hyperparameter optimization is an essential step of the optimal model selection. It is performed to find the model complexity that is appropriate for a specific regression or classification problem, so that the model does not underfit (its complexity is too low) or overfit (the complexity is too high) the training data. Each of the machine learning algorithms described in section 2 has a specific set of hyperparameters that can be tuned.

FNNs. We employ a single hidden layer neural network. One hidden layer is typically sufficient to approximate a continuous function (Cybenko, 1989). The number of neurons in the input layer is equal to the number of inputs, and only one neuron is present in the output layer, which outputs the predicted value of the Kp index. Therefore, the main hyperparameter to tune is the number of neurons in the hidden layer. The choice of the number of neurons is made using a grid search, and an optimal number of 19 neurons was determined when using all the 166 input variables for all horizons. Due to the shallowness of the network, we employ a second-order optimization method based on the Levenberg-Marquardt algorithm to train the networks (provided by the Matlab Deep Learning toolbox).

GB. As given by the xgboost python library (https://xgboost.readthedocs.io/en/latest/), the GB regression algorithm has 15 hyperparameters. Since the sensitivity of the results on many of them is negligible, we focus on tuning the three most important parameters: the number of estimators (i.e., the number of trees in the ensemble), the learning rate (regulates the step of the gradient descent), and the maximum depth of each tree (controls the complexity of the model and therefore affects overfitting). Using the grid search, we find that 100 estimators and a learning rate of 0.08 lead to the best performance on the validation set for all prediction horizons. Regarding the max depth of trees, we find that a max depth of 5 provides a good performance on the validation set and, at the same time, limits overfitting.

LR. The LR algorithm does not require hyperparameter optimization. The model complexity is low enough to avoid overfitting and, therefore, no regularization is necessary. The bias of the model is relatively high, however, due to the insufficiency of the model to perform well on the nonlinear task and cannot be mitigated unless opting for more complex, nonlinear models, or using more sophisticated input features constructed from the solar wind parameters (i.e., solar wind coupling functions).

FFX. The FFX algorithm has a number of hyperparameters to optimize in the FFX python package (five parameters corresponding to the choice of the basis functions that can be selected by a user, and eight other default parameters that were optimized by McConaghy, 2011). The most important is the choice of the basis functions. We obtained the best model performance by including single variables, the interaction terms that allow building low-order polynomials in the input variables (e.g., $x_1 \times x_2$ or x_1^2), and hinge functions (i.e., max and min functions) to introduce nonlinear thresholds.

RF. The hyperparameter search for the random forest algorithm is similar to that of GB. RF has 12 hyperparameters to optimize. We focus on the two main parameters that control the complexity of the model: the number of estimators and the maximum depth of each element of the ensemble. Using the grid search, we find that the optimal values are 30 estimators in the ensemble and a maximum depth of 7.

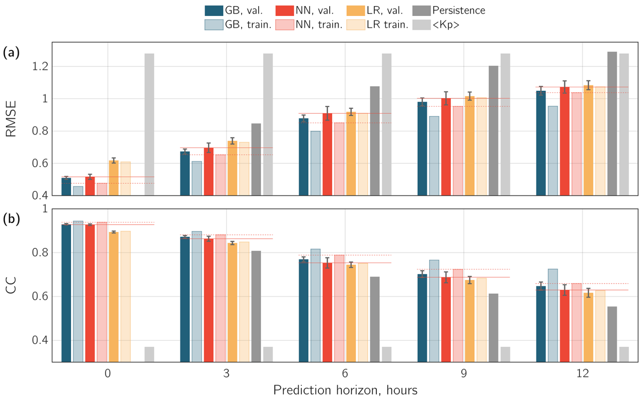

Figure 2. (a) Root mean square error (RMSE) of the ML methods used for model development as a function of prediction horizon in hours: for gradient boosting (blue), neural networks (red), linear regression (orange), persistence model (dark gray), and averaged Kp over the 15 previous days (light gray). The bars are arranged in pairs for the first three methods, and the solid (darker) colors show the error on the validation set, the faded (lighter) colors show the error on the training set. The error bars indicate the standard deviation of error obtained from fivefold cross validation with 10 repetitions. The horizontal red solid and dashed lines are help lines for a more convenient comparison between different methods (the solid lines correspond to the validation error of the NN-based models, the dashed lines to the training error). (b) Same as in the top panel, but using the Pearson correlation coefficient (CC) as a performance assessment metric.

MIM and MRMR. These methods do not have hyperparameters as the methods described above, but there is one factor that may influence how MI is calculated and that is how the variables are discretized. There are several ways in which this can be done, and this is currently an active area of research (Ali et al., 2015; Gao et al., 2017; Jiang & Wang, 2016, e.g.). One way is to bin variables uniformly using bins of a predefined size for each variable (e.g.,Wing et al., 2016; Wing & Johnson, 2019). Sturges (1926) proposed that the optimal bin size for a normal distribution is $n_b = \log_2(n) + 1$ and bin width $w = range/n_b$, where n is the total number of measurements in the data set and range is the maximum-minimum value of a variable. We have explored a number of different bin sizes and found that $n_b = 20$ is optimal for our task.

4. Results

4.1. Comparison of the ML Methods for Model Development

In this section, we compare the performance of the GB, neural network (NN), and LR-based models on the task of predicting the Kp index for prediction horizons from 0 to 12 hr ahead. All models have the same inputs described in section 3.2 and are trained and validated using the repeated K-fold cross validation procedure described in section 3.3.

Figures 2a and 2b show the cross-validation root mean square error (RMSE) and correlation coefficient (CC) of the different models as a function of prediction horizon intervals, respectively. For reference, errors of the persistence model and of the averaged Kp model are shown as well. The persistence model is a model that always predicts the most recent known value of Kp, and we defined the averaged Kp model as a model that predicts the mean value of the Kp index over the previous 15 days. The values of RMSE and CC for all the methods are also listed in Table A1 for reference. As can be seen from Figure 2, the averaged Kp model, $< \mathrm{Kp} >$, has the highest RMSE and the lowest CC (shown in light gray), as expected. The persistence model (dark gray) has the second highest error and lowest CC, and its performance decreases for the longer prediction horizons. For the LR, GB, and NN methods, the bars in the plot are arranged in pairs, where the color corresponds to a particular method. Solid (darker) colors show the mean validation error and faded (lighter) colors show the mean training error, both obtained in the cross validation procedure. It can be seen that the error of the LR model (shown in yellow) is much lower than that of the persistence and $< \mathrm{Kp} >$ models but is higher compared to the NN- and GB-based models, especially for prediction horizons 0 and 3 hr ahead. For longer prediction horizons, the errors of the LR-, NN-, and GB-based models are comparable. This confirms the existence of a nonlinear component in the Kp prediction problem for the short-term prediction horizons that can be modeled only with nonlinear methods. The GB-based (blue) and NN-based (red) models yield similar validation errors for prediction horizons 0 and 3 hr ahead. The validation error of the GB model is slightly smaller for the longer prediction horizons. The standard deviation of error (shown with

Table 2
Number of Features Selected Using the Feature Selection Algorithm Based on FFX with Threshold $k = 50$

Prediction horizon, hr	0	3	6	9	12
Number of selected inputs (out of 166 available inputs)	52	14	13	12	10

error bars) is less than 0.05 for both the GB- and the NN-based models, which is significantly less than the discretization of Kp levels, indicating that the models are quite robust. It should be also noted that the difference between the training (faded blue) and the validation (solid blue) errors of the GB model is larger than the one of the NN model, which may indicate that the GB-based model is overfit. It is therefore difficult to make a definitive conclusion about which model is suited better for the problem of Kp prediction since the difference between their performance is small. Due to the fact that the overfitting behavior is not desirable for a model, we choose the NN-based model as a benchmark model for further use and comparison to the previous studies. More information on overfitting and why it is not desirable can be found in Zhelavskaya et al. (2017).

4.2. Comparison of Feature Selection Methods

In this section, we compare the performance of the input selection algorithms described in section 2.2 for the same prediction horizons (0 to 12 hr ahead). We also test how the size of a time window used for constructing the input variables to the models affect the accuracy of the predictions.

A feature selection procedure provides a list of the most important input variables, and a model based on these input variables needs to be constructed and trained in order to assess the quality of the selected inputs (and hence, the feature selection method). In the previous section, we have selected the NN-based model as a benchmark model for further use and comparison to the previous studies. Hence, we also use neural networks as a benchmark algorithm to test and compare different feature selection procedures. For each prediction horizon, we train and compare neural networks with five different configurations of input variables: (1) all input variables, as in section 4.1 (166 variables listed in Table 1), (2) inputs selected by the FFX algorithm, (3) inputs selected by the RF algorithm, and inputs selected by (4) the MIM and (5) the MRMR feature selection algorithms. Since only FFX provides the definitive number of selected inputs, as discussed in section 2.2, the number of selected inputs for the RF, MIM, and MRMR algorithms is chosen to be equal to the number of inputs selected by the FFX algorithm. This allows an objective comparison between the feature selection methods, as the number of inputs is the same for all methods. The number of selected inputs for different prediction horizons is listed in Table 2. The full list of optimal input variables selected by different algorithms is provided in Appendix B.

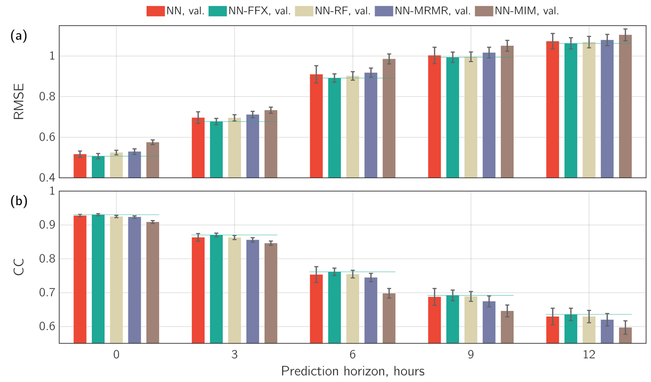

Figure 3. (a) Root mean square error (RMSE) and (b) correlation coefficient (CC) of the trained models used for input selection as a function of prediction horizon for neural networks trained on the whole set of inputs replotted from Figure 2 (red), neural networks trained on inputs selected using FFX with threshold $k = 50$ (green), neural network trained on inputs selected using random forest (light yellow), and neural networks trained on inputs selected using MRMR (violet) and MIM (brown); The bars show the average validation (a) RMSE and (b) correlation coefficient. The error on the training set is not shown in this figure. The error bars indicate the standard deviation of the error obtained from fivefold cross validation with 10 repetitions. The horizontal green lines are help lines for a more convenient comparison between different methods (they correspond to the validation error of the NN-FFX models).

Figures 3a and 3b show the RMSE and correlation coefficients of the neural networks with the inputs selected by different feature selection algorithms versus the prediction horizon. The neural network containing all 166 input variables is denoted as NN (shown with the red bars) and is the same as in Figure 2; all other models are denoted as NN-X, where X corresponds to the algorithm used to select the optimal inputs (FFX, RF, MIM, or MRMR). Only the error on the validation set is shown, since the training error is very close to it for all the methods. The values of RMSE and CC for all the methods are also listed in Table A2 for reference, for both validation and training sets. Neural networks having the inputs selected by the MIM algorithm show the poorest performance, while all other neural networks have comparable validation errors. This confirms that the MI of input variables with the target variable used in isolation from other input variables is not a good indicator of the importance of features when the input variables are correlated with each other. It should therefore not be used in such cases. Models with the inputs selected by other methods (FFX, RF, and MRMR) have small differences in the resulting errors and perform similarly to the model containing all 166 inputs. Compared to all other models, NN-FFX shows the best performance for

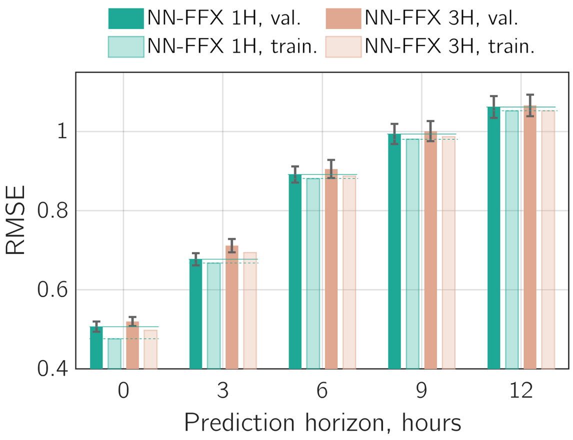

Figure 4. Comparison between 1- and 3-hr cadence inputs. Format is the same as in Figure 2. The horizontal green solid and dashed lines are help lines for a more convenient comparison between the models (the solid lines correspond to the validation error of the NN-FFX 1H models, the dashed lines to the training error).

all horizons as well as a reduced variance, that is, lower standard deviation of the validation error. Despite having fewer inputs, it also shows a slightly better performance compared to the neural network containing all 166 inputs. This result indicates that using these methods, particularly FFX, we can significantly reduce the number of input parameters and select the optimal ones containing a sufficient amount of information to model the target variable, that is, the Kp index. We can use the obtained set of optimal input variables to gain a better understanding of what solar wind drivers are most significant to predict the Kp index for different horizons.

To identify how the construction of inputs affects the performance of the models, we perform the same analysis for the inputs constructed using 3-hr time windows (for computing averages, min, and max of solar wind parameters). The performance of different feature selection algorithms is similar to the results shown in Figure 3, and for brevity, we only present the results of the best performing models. Figure 4 shows the RMSE of the NN-FFX model with inputs constructed using 1-hr time windows (shown with green) and with inputs constructed using 3-hr time windows (pink) versus prediction horizon. The values of RMSE and CC for all the methods are also listed in Table A3 for reference. The model with the inputs constructed using 1-hr time windows shows a slightly lower validation error than the model based on 3-hr inputs; however, the improvement is marginal, and the differences in the model errors are very small. This potentially indicates that a further increase of the cadence of inputs may not lead to significant improvements, when using solar wind measurements at L1 as input to the predictive model of Kp.

4.3. Resulting Models

Based on the results obtained in the previous sections, we select the optimal models for each prediction horizon, that is, the ones that do not overfit show the lowest validation error and lowest standard deviation of error, for further use and comparison with existing models. These are the neural network-based models with the input variables selected by the FFX feature selection algorithm (NN-FFX).

In our further analysis, we apply these models to all data combined (training, validation, and test sets, described in section 3.3) for all prediction horizons, to produce the normalized occurrence maps (presented in this section, Figure 6) and to compute the accuracy metrics (section 4.4, Table 3). We use the whole data set to do that in order to maximize the coverage and the number of measurements in the comparison (the test set provides only 5,880 measurements). Also, the results for the training, validation, and test sets separately are similar to the ones produced using the full data set (please see Appendix C for the accuracy metrics computed on the training, validation, and test sets separately). The normalized distribution of the training, validation, and test sets is shown in Figure 5, in blue, red, and yellow, respectively. Due to the use of cross validation procedure, the splitting into training and validation sets was performed for 10 × 5 times (fivefold CV with 10 repetitions). Therefore, we compute the mean value and the standard deviation of the

Table 3
Fit Performance Statistics of the NN-FFX Models for Different Prediction Horizons Computed on All Data

Prediction horizon, hr	0	3	6	9	12
Number of values in comparison	58,439	58,439	58,439	58,439	58,439
Intercept of the linear fit	0.2624	0.4444	0.7724	0.9653	1.1073
Slope of the linear fit	0.8712	0.7641	0.5888	0.4894	0.4152
Pearson correlation coefficient (R)	0.9361	0.8742	0.7668	0.7002	0.6430
Root mean square error (RMSE)	0.4865	0.6698	0.8853	0.9856	1.0565
Mean absolute error (MAE)	0.3786	0.5081	0.6734	0.7495	0.8042
Mean error (ME, or bias)	0.0179	−7.26e-04	−0.0030	−0.0016	0.0078
Prediction efficiency (PE)	0.8761	0.7643	0.5880	0.4903	0.4134

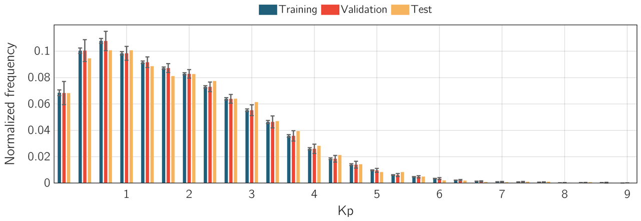

Figure 5. Histogram of normalized distribution of training, validation and test sets. The gray error bars show the spread of the normalized number of measurements in the training and validation sets in different Kp bins as per different splits produced in the cross validation procedure.

normalized frequency of measurements in the training and validation sets over different splits. The standard deviation of the normalized frequency is shown with the error bars. No error bars are associated with the test set, since it is the same for all splits. It can be seen that the distributions of measurements in the training, validation, and test sets are similar to each other (the coverage of measurements in different sets is similar for different Kp bins), which supports the use of the combined data set (i.e., training, validation, and test) for further analysis.

Figure 6 shows the normalized occurrence of the observed versus predicted Kp. The occurrence is normalized by the number of measurements of the observed Kp, that is, the color of each bin denotes the number of measurements in that bin divided by the total number of measurements in that bin of observed Kp. Bins containing four or fewer measurements are not taken into account. The gray dashed lines show the ideal fit to observations and the blue dashed lines show the obtained linear fits. The CC and the RMSE of the models for each prediction horizon are indicated as well. It can be seen that the models for prediction horizons 0 (nowcast) and 3 hr ahead perform well: Most of the measurements are clustered close to the diagonal and the spread of the observed versus predicted Kp values is quite small. However, the performance decreases for longer prediction horizons as the models tend to underestimate the Kp index for higher Kp values, that is, the bias of the models for high Kp values increases as the prediction horizon increases. This indicates that either the information from the solar wind measurements at L1 is not sufficient to predict the elevated geomagnetic activity for the longer prediction horizons or that it cannot be predicted accurately due to the lack of observations of high Kp in our training data set. The second option is less likely, since influence of the lack of observations of high Kp values would manifest in all the models. Indeed, the models for predicting Kp for 0 and 3 hr ahead are capable of predicting those events (Figures 6a, 6b, 7a, and 7b). This suggests that such a decrease in performance is due to the lack of information in the solar wind measurements at L1. This behavior of predictive models based on the solar wind measurements at L1 is also noted by Shprits et al. (2019) and is discussed and analyzed there in more detail.

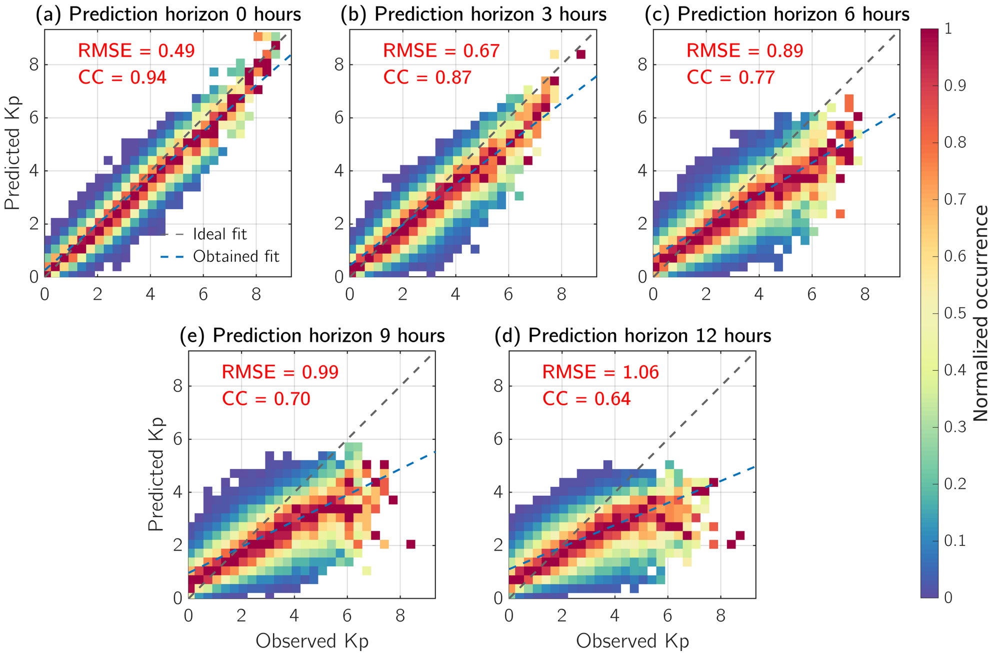

Figure 6. Correlation between the observed and predicted Kp values by the neural network for all data (combined training, validation, and test sets) for prediction horizons (a) 0 (nowcast), (b) 3, (c) 6, (d) 9, and (e) 12 hr ahead. The gray dashed lines indicate the perfect fit and the blue dashed lines indicate the obtained fit.

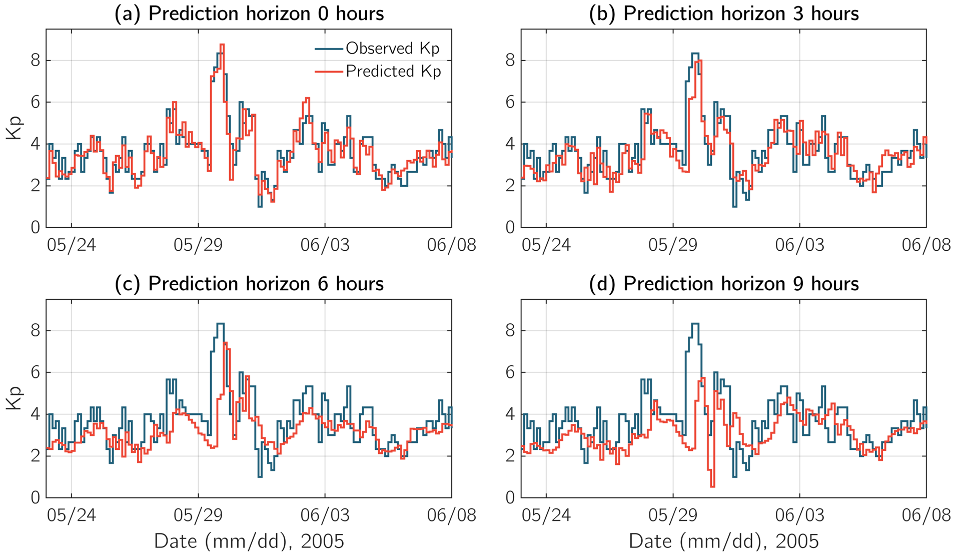

Figure 7. Examples of Kp prediction for prediction horizons (a) 0, (b) 3, (c) 6, and (d) 9 hr ahead on the event from the test set. The blue lines show the observed Kp index, the red lines show the predicted Kp index.

Figure 7 shows examples of the Kp index prediction for different prediction horizons during the May–June 2005 period, that is, an event from the test set (not used in the training). Again, the models for prediction horizons for 0 and 3 hr ahead perform better than models for longer prediction horizons: The latter do not capture the arrival of the storm nor its magnitude. Also, the models underestimate high Kp values for longer prediction horizons (the bias of the models for high Kp values increases). Again, this potentially indicates that the information in the solar wind measurements at L1 is not sufficient for long-term predictions, especially for predicting the elevated geomagnetic activity.

4.4. Benchmarking

Following Liemohn et al. (2018), we calculate the standard assessment metrics for comparison with previous and future studies. The standard assessment metrics proposed in Liemohn et al. (2018) are listed in Table 3 and are the following: linear fit intercept and slope, Pearson correlation coefficient, RMSE, mean absolute error, mean error (ME), and prediction efficiency. We calculate these metrics using the whole data set, as discussed in the previous section (Figure 5). It can be seen that RMSE increases (and correlation coefficient decreases) for the longer prediction horizons, as shown and discussed before. Mean absolute error and prediction efficiency have a similar behavior, indicating that the model performance is better for short-term predictions than for long-term. It is interesting to note that ME, or bias, is close to zero for all prediction horizons. The slope of the linear fit decreases with prediction horizon, however, indicating that the model underpredict Kp as the prediction horizon increases. But, as seen in Figure 6, the model tends to underpredict high Kp values and overpredict low Kp values (which is also reflected in the values of the intercept of the linear fit in Table 3). Since there are many more low Kp values in our data set than high Kp values, the differences tend to cancel out, and as a result, ME is close to zero for all horizons.

Table 4
Comparison With Existing Kp Predictive Models for Prediction Horizons 0, 3, and 6 hr Ahead

Prediction horizon	Model	RMSE	CC	RMSE NN-FFX model	CC NN-FFX model	Test period
0 (nowcast)	Wintoft et al. (2017)	0.55	0.92	**0.51**	**0.93**	2001, 2011
3 hr	Boberg et al. (2000)	0.98	0.77	n/a	n/a	1986–1996
	Bala and Reiff (2012)	0.65	0.86	**0.62**	**0.88**	2001/04, 2006/01–2007/12
	Tan et al. (2018)	**0.64**	**0.81**	0.65	0.80	2013/12–2014/9
6 hr	Bala and Reiff (2012)	0.85	0.76	**0.82**	**0.78**	2001/04, 2006/01–2007/12

Note. Numbers in bold indicate the best performance within one row (one model for one prediction horizon).

We also compare our best performing models (NN-FFX) to the existing predictive models of Kp. The results of this comparison are presented in Table 4. We only consider models that can be compared to our models in terms of prediction horizons. We have therefore not included the following studies into this comparison, as their prediction horizons cannot be directly compared to the ones used in this study (nowcast, 3, 6, 9, and 12 hr ahead): 1 hr ahead in Balikhin et al. (2001), Boaghe et al. (2001), Costello (1998), and Ji et al. (2013); 1 and 4 hr ahead in Wing et al. (2005); results for 1-hr predictions by Bala and Reiff (2012). To allow direct comparison between the models, we use the same training, validation, and test time periods as in those studies. We use the input variables to the models that are found by the FFX feature selection algorithm. The comparison, when done in such a way, also helps illustrate that the training interval, as well as validation and test intervals, can affect the performance of the resulting model. The accuracy metrics in Table 4 are computed on the test set of the corresponding study. Overall, the resulting accuracy of our models is comparable or slightly better than that of the listed studies. There are no published studies for prediction horizons of 9 and 12 hr ahead to compare. The table also reflects that model errors change depending on the chosen training and test sets (e.g., prediction for 3 hr ahead) and also differ from those listed in Table 3. This means that selecting only a specific time interval for testing or validation can affect the resulting model performance. It also demonstrates that assessing a model error using just a specific time interval may not reflect the actual performance of a model. The cross validation procedure described in this study attempts to overcome these issues by including an element of randomness and ensuring that the distributions of the training, validation, and test sets are representative and similar to each other.

5. Discussion

The results obtained in the previous sections demonstrate that machine learning-based models driven by the solar wind measurements at L1 can produce accurate short-term Kp predictions, but the accuracy is reduced for long-term prediction horizons (> 3 hr ahead). This is observed for all machine learning methods considered in this study and for the previous studies as well (Table 4). The models cannot capture the storm onset times for long-term horizons accurately and tend to underpredict high Kp and overpredict low Kp values. This indicates that the information contained in solar wind measurements at L1 is not sufficient for accurate long-term predictions of Kp (> 3 hr ahead). Other information sources, such as images of the Sun or features derived from them, should be incorporated into the model to produce accurate long-term forecasts.

Nonlinear machine learning methods, such as GB and neural networks, perform significantly better than LR for short-term prediction horizons, but their performance becomes comparable as the prediction horizon increases (Figure 2). This implies that the relation between Kp and solar wind measurements at L1 is nonlinear for short-term prediction horizons, but there is little to no gain in using nonlinear methods for longer prediction horizons (> 3 hr ahead) when using solar wind measurements at L1 as input to the model.

It should also be noted that all models listed in Table 4, as well as ours, have a similar performance (for the same horizon) independent of the inputs or the modeling techniques/methods used. This implies that the usage of another new method/ML technique or a different way of constructing the inputs will probably not bring much more improvement in the performance of a model that uses solar wind measurements at L1 as an input. Moreover, it should be noted that the average RMSE of the nowcast models considered here is ~ 0.5 (during disturbed times, $\mathrm{Kp} > 4$, the average RMSE is ~ 0.8 Shprits et al., 2019), which is higher than the cadence of Kp, that is $1/3$. Ideally, if the solar wind measurements at L1 contained sufficient information for the prediction of Kp at the current moment, the maximum RMSE of a model would be equal $1/3$, since a model could be wrong only by one Kp bin (due to the discretization of Kp; the average RMSE of such a model would be much lower than $1/3$, but the RMSE during storms or onset of storms could be larger). However, the fact that the average RMSE of the models is greater than $1/3$, independent of the way the inputs are constructed or the method used to develop a model, implies that there is a stochastic component of the magnetosphere system that is not captured in the solar wind measurements and cannot be modeled properly, assuming that our models optimally utilize all the data. Since Kp reflects the geomagnetic disturbance at the Earth's surface due to electric currents in the ionosphere and magnetosphere, there is an uncertainty in the direct relation between solar wind and Kp associated with the coupling between ionosphere and magnetosphere, which is a complex stochastic process. It is also possible that the magnetosphere operates in different regimes depending on the type of incoming solar wind and this, in turn, affects the Kp index. This can be further investigated by defining different types of solar wind and training models separately for different types of solar wind. Types of solar wind can be defined depending on the charge state composition of the solar wind, solar wind speed, proton temperature, proton density, etc. (for example, as done in Heidrich-Meisner & Wimmer-Schweingruber, 2018 or Xu & Borovsky, 2015).

Feature selection methods, such as FFX, RF, or MRMR, the use of which was demonstrated in this paper, showed their capabilities to select the most important/significant inputs to the model. Using these methods, the number of inputs to the model was reduced from 166 to 52 for nowcast and from 166 to 10 for prediction 12 hr ahead (Table 2). At the same time, the performance of the models based on the reduced input set remained the same or even slightly improved (when using the FFX feature selection algorithm) compared to the model based on all 166 inputs.

In addition to the significant reduction of input dimensionality, the selected input variables can be analyzed to understand the main drivers for the Kp predictive models. Different methods select slightly different variables as the optimal ones, but we can consider the best performing model, which is based on the variables selected by the FFX method. For the nowcast of Kp, minimum of B_z over the previous 7 hr and maximum of solar wind speed V_{sw} over the previous hour are ones of the selected variables, as well as average and minumum of V_{sw}, average and maximum B_z, average, minimum, and maximum of B (their time history over the varying time intervals). It is interesting to note that the FFX algorithm also selects the minimum and maximum of B_y and maximum of B_x components, as well as minimum and maximum of proton density over the previous 4 hr and the indicators of seasonal variability. For the 3 hr ahead prediction, the dimensionality is reduced even more, down to 14 input variables. The subset of the same variables is selected (min and avg B_z; max and min V_{sw}; max and min B, max B_y, max and min of proton density, and seasonal indicators), except that they are selected for only the several previous hours. At the same time, the performance of the model containing only these 14 inputs is even slightly better compared to the model containing 166 variables, indicating that 14 variables are sufficient and encompass the necessary information to predict Kp 3 hr ahead. The selected input variables for all prediction horizons and all feature selection methods considered are displayed in Appendix B.

6. Conclusions

In this study, we explore how different machine learning algorithms, namely GB, FNNs, and LR, perform on the task of predicting the Kp index for prediction horizons of 0, 3, 6, 9, and 12 hr ahead using solar wind measurements at L1 as an input. We also illustrate how different feature selection methods can be applied to select the optimal inputs to the predictive model of the Kp index. In particular, we assess the performance of four feature selection procedures based on the FFX, RF, MIM, and MRMR algorithms. We have found that

1) The models trained using neural networks (NNs) and GB notably outperform the models constructed using LR for the short-term prediction horizons. This implies the existence of a nonlinear component in the Kp prediction problem for short-term predictions that cannot be modeled using linear methods alone.
2) The performance of all considered methods decreases as the prediction horizon increases. This likely means that the information in the solar wind measurements at L1 is not sufficient to produce accurate long-term predictions (e.g., $>$ 3 hr), especially for high Kp values, and is not sufficient to accurately capture the arrival time of geomagnetic storms for the long-term prediction horizons.
3) The proposed FFX feature selection algorithm (i.e., a procedure for finding optimal input variables to a model) outperforms other feature selection algorithms considered in this study. It provides a significant reduction of the number of input variables sufficient to model the Kp index (starting from more than threefold for the nowcast to more than 16-fold for the prediction of 12 hr ahead).
4) Despite having fewer inputs, the models based on the reduced set of input variables obtained with the FFX algorithm have a slightly better performance than the models based on the full input set. This implies that using the FFX feature selection algorithm, we can significantly reduce the input dimensionality, obtain a set of the most significant input variables sufficient for predicting the Kp index for different prediction horizons, and, at the same time, improve the model performance.

The obtained sets of optimal input variables can be used to gain an understanding of what inputs are the most important and physically meaningful for predicting the Kp index. Moreover, the models can be trained faster and have less tendency to overfit using such a reduced set of inputs. The feature selection methods described in this work can also be applied to other problems in space physics in order to significantly reduce the input dimensionality and identify the most important inputs that contain sufficient information to produce accurate predictions.

Appendix A: RMSE and CC of all methods considered in the paper

Tables below contain values of RMSE and CC on the validation and training sets for all the methods considered in this study. Table A1 contains RMSE and CC of the methods presented in Figure 2: Gradient Boosting (GB), Neural Networks (NN), Linear Regression (LR), Persistence, and <Kp>. Table A2 contains RMSE and CC of the methods presented in Figure 3: Neural Networks with all 166 inputs (NN), NN with inputs selected by Fast Function Extraction (NN-FFX), Random Forest (NN-RF), Maximum Relevancy Minimum Redundancy (NN-MRMR), and Mutual Information Maximization (NN-MIM) feature selection procedures. Table A3 contains RMSE and CC of the methods presented in Figure 4: Neural Networks with inputs selected by Fast Function Extraction feature selection procedure constructed using 1-hr intervals (NN-FFX 1H) and 3-hr intervals (NN-FFX 3H).

Appendix B: Optimal Inputs Selected by Feature Selection Algorithms

Tables below contain reduced sets of input variables selected by the feature selection algorithms described in this paper (section 2.2). Results for all prediction horizons considered in this work (from nowcast to 12 hr ahead) are presented for both 1- and 3-hr-based inputs. The format of the inputs is the following: The subscript denotes the operation which is performed to obtain the input (taking max, min, or averaging), and the numbers in the brackets (e.g., $(-3;-2)$ in $B_{z_{\min(-3;-2)}}$) denote the time interval over which max, min, or average is taken. T is the UT hour of the day, and D is day of the year.

Appendix C: Fit performance statistics

Table A1
RMSE and CC of the Methods Presented in Figure 2

		RMSE		CC	
Prediction horizon, hr	Method	Validation	Training	Validation	Training
0	GB	0.5098	0.4573	0.9291	0.9437
	NN	0.5167	0.4769	0.9274	0.9386
	LR	0.6181	0.6090	0.8941	0.8974
	Persistence	-		-	
	<Kp>	1.2800		0.3700	
3	GB	0.6736	0.6118	0.8721	0.8966
	NN	0.6964	0.6530	0.8632	0.8813
	LR	0.7392	0.7303	0.8439	0.8484
	Persistence	0.8470		0.8080	
	<Kp>	1.2800		0.3700	
6	GB	0.8789	0.7990	0.7696	0.8163
	NN	0.9097	0.8505	0.7532	0.7886
	LR	0.9189	0.9102	0.7448	0.7513
	Persistence	1.0770		0.6900	
	<Kp>	1.2800		0.3700	
9	GB	0.9807	0.8906	0.7020	0.7658
	NN	1.0029	0.9532	0.6875	0.7239
	LR	1.0163	1.0070	0.6751	0.6836
	Persistence	1.2040		0.6130	
	<Kp>	1.2800		0.3700	
12	GB	1.0492	0.9539	0.6473	0.7252
	NN	1.0726	1.0374	0.6295	0.6588
	LR	1.0839	1.0742	0.6168	0.6271
	Persistence	1.2910		0.5540	
	<Kp>	1.2800		0.3700	

Table A2
RMSE and CC of the Methods Presented in Figure 3

		RMSE		CC	
Prediction horizon, hr	Method	Validation	Training	Validation	Training
0	NN	0.5167	0.4769	0.9274	0.9386
	NN-FFX	0.5068	0.4764	0.9301	0.9386
	NN-RF	0.5247	0.5004	0.9248	0.9320
	NN-MRMR	0.5295	0.4994	0.9234	0.9324
	NN-MIM	0.5755	0.5642	0.9087	0.9127
3	NN	0.6964	0.6530	0.8632	0.8813
	NN-FFX	0.6774	0.6678	0.8703	0.8747
	NN-RF	0.6956	0.6892	0.8627	0.8658
	NN-MRMR	0.7117	0.7033	0.8559	0.8602
	NN-MIM	0.7334	0.7263	0.8460	0.8497
6	NN	0.9097	0.8505	0.7532	0.7886
	NN-FFX	0.8913	0.8814	0.7615	0.7685
	NN-RF	0.9017	0.8953	0.7547	0.7596
	NN-MRMR	0.9178	0.9091	0.7446	0.7514
	NN-MIM	0.9854	0.9788	0.6979	0.7039
9	NN	1.0029	0.9532	0.6875	0.7239
	NN-FFX	0.9936	0.9804	0.6919	0.7031
	NN-RF	0.9963	0.9857	0.6887	0.6979
	NN-MRMR	1.0164	1.0008	0.6744	0.6879
	NN-MIM	1.0504	1.0430	0.6459	0.6535
12	NN	1.0726	1.0374	0.6295	0.6588
	NN-FFX	1.0618	1.0523	0.6357	0.6453
	NN-RF	1.0683	1.0573	0.6294	0.6404
	NN-MRMR	1.0786	1.0700	0.6201	0.6296
	NN-MIM	1.1039	1.0993	0.5970	0.6029

Table A3
RMSE and CC of the Methods Presented in Figure 4

		RMSE		CC	
Prediction horizon, hr	Method	Validation	Training	Validation	Training
0	NN-FFX 1H	0.5068	0.4764	0.9301	0.9386
	NN-FFX 3H	0.5201	0.4978	0.9254	0.9320
3	NN-FFX 1H	0.6774	0.6678	0.8703	0.8747
	NN-FFX 3H	0.7118	0.6944	0.8551	0.8631
6	NN-FFX 1H	0.8913	0.8814	0.7615	0.7685
	NN-FFX 3H	0.9051	0.8878	0.7523	0.7640
9	NN-FFX 1H	0.9936	0.9804	0.6919	0.7031
	NN-FFX 3H	1.0007	0.9870	0.6850	0.6967
12	NN-FFX 1H	1.0618	1.0523	0.6357	0.6453
	NN-FFX 3H	1.0659	1.0521	0.6307	0.6442

Table B1
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 0$ (nowcast) With a 1-hr Time Window Used to Construct Input Features

$h = 0$ (nowcast)			
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{z_{\mathrm{min}(-3;-2)}}$	$B_{\mathrm{avg}(-1;0)}$	$B_{z_{\mathrm{min}(-3;-2)}}$	$B_{z_{\mathrm{min}(-3;-2)}}$
$B_{z_{\mathrm{min}(-4;-3)}}$	$B_{z_{\mathrm{min}(-1;\ 0)}}$	$V_{sw_{\mathrm{max}(-1;\ 0)}}$	$V_{sw_{\mathrm{max}(-1;\ 0)}}$
$B_{z_{\mathrm{min}(-5;-4)}}$	$B_{z_{\mathrm{min}(-2;-1)}}$	$V_{sw_{\mathrm{max}(-2;-1)}}$	$B_{z_{\mathrm{min}(-4;-3)}}$
$B_{z_{\mathrm{min}(-7;-6)}}$	$B_{z_{\mathrm{min}(-3;-2)}}$	$V_{sw_{\mathrm{avg}(-2;-1)}}$	$B_{z_{\mathrm{min}(-2;-1)}}$
$V_{sw_{\mathrm{avg}(-2;-1)}}$	$B_{z_{\mathrm{min}(-4;-3)}}$	$V_{sw_{\mathrm{min}(-1;\ 0)}}$	$V_{sw_{\mathrm{avg}(-1;\ 0)}}$
$V_{sw_{\mathrm{max}(-1;\ 0)}}$	$B_{z_{\mathrm{min}(-5;-4)}}$	$B_{z_{\mathrm{min}(-1;\ 0)}}$	$V_{sw_{\mathrm{max}(-2;-1)}}$
$V_{sw_{\mathrm{min}(-1;\ 0)}}$	$B_{z_{\mathrm{min}(-6;-5)}}$	$B_{z_{\mathrm{min}(-5;-4)}}$	$V_{sw_{\mathrm{avg}(-2;-1)}}$
$V_{sw_{\mathrm{min}(-3;-2)}}$	$B_{\mathrm{avg}(-7;-6)}$	$B_{\mathrm{max}(-3;-2)}$	$V_{sw_{\mathrm{min}(-1;\ 0)}}$
$B_{\mathrm{avg}(-9;-8)}$	$V_{sw_{\mathrm{avg}(-2;-1)}}$	$B_{z_{\mathrm{min}(-4;-3)}}$	$V_{sw_{\mathrm{max}(-3;-2)}}$
$V_{sw_{\mathrm{min}(-5;-4)}}$	$V_{sw_{\mathrm{avg}(-3;-2)}}$	$B_{z_{\mathrm{min}(-2;-1)}}$	$V_{sw_{\mathrm{min}(-2;-1)}}$
$V_{sw_{\mathrm{min}(-9;-8)}}$	$V_{sw_{\mathrm{avg}(-4;-3)}}$	$B_{\mathrm{max}(-9;-8)}$	$V_{sw_{\mathrm{avg}(-3;-2)}}$
$B_{\mathrm{max}(-1;0)}$	$V_{sw_{\mathrm{avg}(-5;-4)}}$	$V_{sw_{\mathrm{max}(-3;-2)}}$	$V_{sw_{\mathrm{max}(-4;-3)}}$
$nProt_{\mathrm{max}(-1;0)}$	$V_{sw_{\mathrm{avg}(-6;-5)}}$	$B_{z_{\mathrm{min}(-7;-6)}}$	$B_{\mathrm{max}(-3;-2)}$
$nProt_{\mathrm{max}(-2;-1)}$	$V_{sw_{\mathrm{avg}(-7;-6)}}$	$B_{z_{\mathrm{min}(-6;-5)}}$	$B_{z_{\mathrm{min}(-1;\ 0)}}$
$nProt_{\mathrm{max}(-3;-2)}$	$V_{sw_{\mathrm{max}(-1;\ 0)}}$	$B_{\mathrm{max}(-1;0)}$	$V_{sw_{\mathrm{min}(-3;-2)}}$
$nProt_{\mathrm{max}(-4;-3)}$	$V_{sw_{\mathrm{max}(-2;-1)}}$	$B_{z_{\mathrm{avg}(-3;-2)}}$	$B_{\mathrm{max}(-4;-3)}$
$B_{\mathrm{max}(-2;-1)}$	$V_{sw_{\mathrm{max}(-3;-2)}}$	$B_{z_{\mathrm{min}(-9;-8)}}$	$B_{\mathrm{max}(-2;-1)}$
$nProt_{\mathrm{min}(-2;-1)}$	$V_{sw_{\mathrm{max}(-4;-3)}}$	$V_{sw_{\mathrm{max}(-5;-4)}}$	$V_{sw_{\mathrm{avg}(-4;-3)}}$
$nProt_{\mathrm{min}(-3;-2)}$	$V_{sw_{\mathrm{max}(-5;-4)}}$	$B_{\mathrm{max}(-2;-1)}$	$B_{z_{\mathrm{min}(-5;-4)}}$
$\sin(2\pi T/24)$	$V_{sw_{\mathrm{max}(-6;-5)}}$	$B_{z_{\mathrm{min}(-8;-7)}}$	$B_{\mathrm{max}(-1;0)}$
$\cos(2\pi T/24)$	$V_{sw_{\mathrm{max}(-7;-6)}}$	$B_{z_{\mathrm{avg}(-2;-1)}}$	$V_{sw_{\mathrm{max}(-5;-4)}}$
$\sin(2\pi D/365)$	$V_{sw_{\mathrm{min}(-1;\ 0)}}$	$B_{x_{\mathrm{min}(-7;-6)}}$	$B_{\mathrm{max}(-5;-4)}$
$\cos(2\pi D/365)$	$V_{sw_{\mathrm{min}(-2;-1)}}$	$B_{z_{\mathrm{avg}(-4;-3)}}$	$B_{\mathrm{max}(-6;-5)}$
$B_{\mathrm{max}(-6;-5)}$	$V_{sw_{\mathrm{min}(-3;-2)}}$	$\cos(2\pi D/365)$	$B_{\mathrm{avg}(-3;-2)}$
$B_{\mathrm{max}(-9;-8)}$	$V_{sw_{\mathrm{min}(-4;-3)}}$	$B_{\mathrm{max}(-4;-3)}$	$V_{sw_{\mathrm{min}(-4;-3)}}$
$B_{\mathrm{min}(-6;-5)}$	$V_{sw_{\mathrm{min}(-5;-4)}}$	$V_{sw_{\mathrm{min}(-2;-1)}}$	$B_{\mathrm{avg}(-4;-3)}$
$B_{z_{\mathrm{avg}(-1;\ 0)}}$	$V_{sw_{\mathrm{min}(-6;-5)}}$	$B_{y_{\mathrm{max}(-3;-2)}}$	$V_{sw_{\mathrm{avg}(-5;-4)}}$
$V_{sw_{\mathrm{avg}(-1;\ 0)}}$	$V_{sw_{\mathrm{min}(-7;-6)}}$	$B_{z_{\mathrm{avg}(-1;\ 0)}}$	$B_{\mathrm{avg}(-2;-1)}$
$B_{x_{\mathrm{max}(-2;-1)}}$	$V_{sw_{\mathrm{min}(-8;-7)}}$	$B_{\mathrm{max}(-7;-6)}$	$V_{sw_{\mathrm{max}(-6;-5)}}$
$B_{x_{\mathrm{min}(-1;\ 0)}}$	$V_{sw_{\mathrm{min}(-9;-8)}}$	$\sin(2\pi T/24)$	$B_{\mathrm{max}(-7;-6)}$
$B_{x_{\mathrm{min}(-5;-4)}}$	$B_{\mathrm{max}(-1;0)}$	$B_{x_{\mathrm{min}(-1;\ 0)}}$	$B_{\mathrm{avg}(-5;-4)}$
$B_{y_{\mathrm{max}(-1;\ 0)}}$	$nProt_{\mathrm{max}(-1;0)}$	$B_{y_{\mathrm{min}(-1;\ 0)}}$	$B_{\mathrm{avg}(-1;0)}$
$B_{y_{\mathrm{max}(-2;-1)}}$	$nProt_{\mathrm{max}(-2;-1)}$	$V_{sw_{\mathrm{max}(-4;-3)}}$	$B_{\mathrm{avg}(-6;-5)}$
$B_{y_{\mathrm{max}(-3;-2)}}$	$nProt_{\mathrm{max}(-3;-2)}$	$B_{\mathrm{max}(-5;-4)}$	$V_{sw_{\mathrm{min}(-5;-4)}}$
$B_{y_{\mathrm{max}(-4;-3)}}$	$nProt_{\mathrm{max}(-4;-3)}$	$B_{z_{\mathrm{avg}(-5;-4)}}$	$V_{sw_{\mathrm{avg}(-6;-5)}}$
$B_{y_{\mathrm{max}(-5;-4)}}$	$B_{\mathrm{max}(-2;-1)}$	$B_{\mathrm{max}(-8;-7)}$	$B_{\mathrm{max}(-8;-7)}$
$B_{y_{\mathrm{min}(-1;\ 0)}}$	$B_{\mathrm{max}(-3;-2)}$	$B_{x_{\mathrm{min}(-9;-8)}}$	$B_{\mathrm{max}(-9;-8)}$
$B_{y_{\mathrm{min}(-2;-1)}}$	$\sin(2\pi D/365)$	$B_{z_{\mathrm{max}(-9;-8)}}$	$V_{sw_{\mathrm{max}(-7;-6)}}$
$B_{y_{\mathrm{min}(-3;-2)}}$	$\cos(2\pi D/365)$	$B_{y_{\mathrm{max}(-1;\ 0)}}$	$B_{\mathrm{avg}(-7;-6)}$

Table B1 (*continued*)

$h = 0$ (nowcast)			
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{y_{\min(-4;-3)}}$	$B_{\max(-4;-3)}$	$V_{sw_{\max(-9;-8)}}$	$V_{sw_{\min(-6;-5)}}$
$B_{z_{\text{avg}(-2;-1)}}$	$B_{\max(-5;-4)}$	$B_{\text{avg}(-1;0)}$	$B_{\text{avg}(-8;-7)}$
$B_{z_{\text{avg}(-3;-2)}}$	$B_{\max(-6;-5)}$	$B_{z_{\text{avg}(-6;-5)}}$	$V_{sw_{\text{avg}(-7;-6)}}$
$B_{z_{\text{avg}(-4;-3)}}$	$B_{\max(-7;-6)}$	$B_{y_{\max(-9;-8)}}$	$B_{z_{\min(-6;-5)}}$
$B_{z_{\text{avg}(-5;-4)}}$	$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{x_{\max(-3;-2)}}$	$V_{sw_{\max(-8;-7)}}$
$B_{z_{\text{avg}(-6;-5)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$	$V_{sw_{\text{avg}(-3;-2)}}$	$B_{\text{avg}(-9;-8)}$
$B_{z_{\text{avg}(-7;-6)}}$	$nProt_{\text{avg}(-1;0)}$	$B_{\max(-6;-5)}$	$V_{sw_{\min(-7;-6)}}$
$B_{z_{\text{avg}(-8;-7)}}$	$B_{y_{\max(-5;-4)}}$	$B_{z_{\max(-8;-7)}}$	$V_{sw_{\text{avg}(-8;-7)}}$
$B_{z_{\text{avg}(-9;-8)}}$	$B_{z_{\text{avg}(-2;-1)}}$	$B_{z_{\max(-3;-2)}}$	$V_{sw_{\max(-9;-8)}}$
$B_{z_{\max(-3;-2)}}$	$B_{z_{\text{avg}(-3;-2)}}$	$B_{y_{\min(-4;-3)}}$	$V_{sw_{\min(-8;-7)}}$
$B_{z_{\max(-4;-3)}}$	$B_{z_{\text{avg}(-4;-3)}}$	$\cos(2\pi T/24)$	$V_{sw_{\text{avg}(-9;-8)}}$
$B_{z_{\max(-5;-4)}}$	$B_{z_{\text{avg}(-5;-4)}}$	$B_{\text{avg}(-2;-1)}$	$B_{z_{\min(-7;-6)}}$
$B_{z_{\max(-6;-5)}}$	$B_{z_{\text{avg}(-7;-6)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$	$V_{sw_{\min(-9;-8)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B2
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 0$ (nowcast) With a 3-hr Time Window Used to Construct Input Features

$h = 0$ (nowcast)			
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$B_{\text{avg}(-3;0)}$	$B_{\text{avg}(-3;0)}$	$B_{z_{\min(-3;\ 0)}}$	$B_{z_{\min(-3;\ 0)}}$
$B_{x_{\text{avg}(-3;\ 0)}}$	$B_{\max(-9;-6)}$	$V_{sw_{\max(-3;\ 0)}}$	$B_{z_{\min(-6;-3)}}$
$B_{\min(-9;-6)}$	$B_{\min(-3;0)}$	$B_{z_{\min(-6;-3)}}$	$V_{sw_{\max(-3;\ 0)}}$
$B_{x_{\text{avg}(-6;-3)}}$	$B_{y_{\max(-3;\ 0)}}$	$B_{\max(-9;-6)}$	$V_{sw_{\text{avg}(-3;\ 0)}}$
$B_{x_{\max(-3;\ 0)}}$	$B_{y_{\max(-6;-3)}}$	$\cos(2\pi D/365)$	$B_{\max(-3;0)}$
$B_{x_{\max(-9;-6)}}$	$B_{y_{\min(-9;-6)}}$	$B_{y_{\max(-3;\ 0)}}$	$B_{\max(-6;-3)}$
$B_{y_{\text{avg}(-3;\ 0)}}$	$B_{z_{\text{avg}(-3;\ 0)}}$	$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\min(-3;\ 0)}}$
$B_{y_{\text{avg}(-6;-3)}}$	$B_{z_{\text{avg}(-6;-3)}}$	$B_{z_{\min(-9;-6)}}$	$V_{sw_{\max(-6;-3)}}$
$B_{y_{\max(-3;\ 0)}}$	$B_{z_{\text{avg}(-9;-6)}}$	$B_{x_{\max(-3;\ 0)}}$	$B_{\text{avg}(-3;0)}$
$B_{y_{\max(-6;-3)}}$	$B_{z_{\max(-3;\ 0)}}$	$V_{sw_{\min(-3;\ 0)}}$	$B_{\text{avg}(-6;-3)}$
$B_{y_{\min(-3;\ 0)}}$	$B_{z_{\min(-3;\ 0)}}$	$B_{\max(-3;0)}$	$V_{sw_{\text{avg}(-6;-3)}}$
$B_{y_{\min(-6;-3)}}$	$B_{z_{\min(-6;-3)}}$	$B_{z_{\text{avg}(-6;-3)}}$	$B_{\max(-9;-6)}$
$B_{z_{\text{avg}(-3;\ 0)}}$	$B_{z_{\min(-9;-6)}}$	$\sin(2\pi T/24)$	$B_{\text{avg}(-9;-6)}$
$B_{z_{\text{avg}(-6;-3)}}$	$V_{sw_{\text{avg}(-6;-3)}}$	$B_{x_{\min(-9;-6)}}$	$V_{sw_{\min(-6;-3)}}$
$B_{z_{\text{avg}(-9;-6)}}$	$V_{sw_{\text{avg}(-9;-6)}}$	$B_{y_{\min(-3;\ 0)}}$	$B_{z_{\min(-9;-6)}}$
$B_{z_{\max(-3;\ 0)}}$	$V_{sw_{\text{avg}(-3;\ 0)}}$	$B_{\max(-6;-3)}$	$V_{sw_{\max(-9;-6)}}$
$B_{z_{\max(-6;-3)}}$	$V_{sw_{\max(-3;\ 0)}}$	$V_{sw_{\max(-9;-6)}}$	$V_{sw_{\text{avg}(-9;-6)}}$
$B_{z_{\max(-9;-6)}}$	$V_{sw_{\max(-6;-3)}}$	$B_{x_{\min(-3;\ 0)}}$	$V_{sw_{\min(-9;-6)}}$
$B_{z_{\min(-3;\ 0)}}$	$V_{sw_{\max(-9;-6)}}$	$B_{z_{\max(-9;-6)}}$	$B_{z_{\text{avg}(-3;\ 0)}}$

Table B2 (*continued*)

	$h = 0$ (nowcast)		
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$V_{sw_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\text{min}(-3;\ 0)}}$	$B_{\text{avg}(-3;0)}$	$B_{y_{\text{max}(-3;\ 0)}}$
$V_{sw_{\text{max}(-3;\ 0)}}$	$V_{sw_{\text{min}(-6;-3)}}$	$B_{z_{\text{avg}(-9;-6)}}$	$B_{y_{\text{max}(-6;-3)}}$
$V_{sw_{\text{min}(-3;\ 0)}}$	$V_{sw_{\text{min}(-9;-6)}}$	$B_{y_{\text{min}(-9;-6)}}$	$B_{\text{min}(-6;-3)}$
$V_{sw_{\text{min}(-6;-3)}}$	$nProt_{\text{avg}(-6;-3)}$	$V_{sw_{\text{avg}(-3;\ 0)}}$	$B_{\text{min}(-9;-6)}$
$V_{sw_{\text{min}(-9;-6)}}$	$nProt_{\text{max}(-3;0)}$	$B_{y_{\text{max}(-6;-3)}}$	$B_{\text{min}(-3;0)}$
$nProt_{\text{max}(-3;0)}$	$nProt_{\text{max}(-6;-3)}$	$B_{z_{\text{max}(-3;\ 0)}}$	$B_{z_{\text{avg}(-6;-3)}}$
$nProt_{\text{max}(-6;-3)}$	$nProt_{\text{avg}(-3;0)}$	$B_{x_{\text{max}(-9;-6)}}$	$B_{y_{\text{max}(-9;-6)}}$
$nProt_{\text{min}(-3;0)}$	$nProt_{\text{max}(-9;-6)}$	$B_{\text{avg}(-9;-6)}$	$B_{y_{\text{min}(-3;\ 0)}}$
$nProt_{\text{min}(-6;-3)}$	$nProt_{\text{min}(-3;0)}$	$B_{y_{\text{max}(-9;-6)}}$	$B_{x_{\text{min}(-3;\ 0)}}$
$nProt_{\text{min}(-9;-6)}$	$\sin(2\pi D/365)$	$V_{sw_{\text{max}(-6;-3)}}$	$B_{y_{\text{min}(-6;-3)}}$
$\sin(2\pi T/24)$	$\cos(2\pi D/365)$	$B_{x_{\text{min}(-6;-3)}}$	$B_{x_{\text{min}(-6;-3)}}$
$\cos(2\pi T/24)$	$B_{\text{avg}(-6;-3)}$	$nProt_{\text{max}(-3;0)}$	$B_{z_{\text{max}(-9;-6)}}$
$\sin(2\pi D/365)$	$B_{\text{avg}(-9;-6)}$	$B_{y_{\text{min}(-6;-3)}}$	$B_{z_{\text{max}(-6;-3)}}$
$\cos(2\pi D/365)$	$B_{\text{max}(-3;0)}$	$B_{z_{\text{max}(-6;-3)}}$	$B_{x_{\text{min}(-9;-6)}}$
$B_{\text{max}(-3;0)}$	$B_{\text{max}(-6;-3)}$	$\cos(2\pi T/24)$	$B_{z_{\text{max}(-3;\ 0)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B3

Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 3$ hr ahead With a 1-hr Time Window Used to Construct Input Features

	$h = 3$ hr ahead		
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{z_{\text{min}(-1;\ 0)}}$	$B_{z_{\text{min}(-1;\ 0)}}$	$B_{z_{\text{min}(-1;\ 0)}}$	$B_{z_{\text{min}(-1;\ 0)}}$
$B_{z_{\text{min}(-2;-1)}}$	$B_{z_{\text{min}(-2;-1)}}$	$V_{sw_{\text{max}(-1;\ 0)}}$	$V_{sw_{\text{max}(-1;\ 0)}}$
$V_{sw_{\text{max}(-1;\ 0)}}$	$B_{z_{\text{min}(-3;-2)}}$	$B_{\text{max}(-3;-2)}$	$B_{\text{max}(-1;0)}$
$V_{sw_{\text{min}(-1;\ 0)}}$	$B_{z_{\text{min}(-4;-3)}}$	$B_{z_{\text{min}(-9;-8)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$
$B_{\text{max}(-1;0)}$	$V_{sw_{\text{avg}(-2;-1)}}$	$B_{z_{\text{min}(-2;-1)}}$	$B_{z_{\text{min}(-2;-1)}}$
$nProt_{\text{max}(-1;0)}$	$V_{sw_{\text{max}(-1;\ 0)}}$	$\cos(2\pi D/365)$	$V_{sw_{\text{max}(-2;-1)}}$
$nProt_{\text{min}(-1;0)}$	$V_{sw_{\text{max}(-2;-1)}}$	$B_{z_{\text{min}(-4;-3)}}$	$B_{\text{max}(-2;-1)}$
$\sin(2\pi T/24)$	$V_{sw_{\text{min}(-1;\ 0)}}$	$B_{x_{\text{min}(-9;-8)}}$	$B_{\text{max}(-3;-2)}$
$\cos(2\pi D/365)$	$V_{sw_{\text{min}(-2;-1)}}$	$B_{\text{max}(-1;0)}$	$V_{sw_{\text{min}(-1;\ 0)}}$
$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{\text{max}(-1;0)}$	$B_{z_{\text{min}(-6;-5)}}$	$B_{\text{avg}(-1;0)}$
$B_{\text{min}(-9;-8)}$	$nProt_{\text{max}(-1;0)}$	$V_{sw_{\text{max}(-3;-2)}}$	$V_{sw_{\text{avg}(-2;-1)}}$
$B_{y_{\text{max}(-1;\ 0)}}$	$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{z_{\text{min}(-3;-2)}}$	$V_{sw_{\text{max}(-3;-2)}}$
$B_{z_{\text{avg}(-2;-1)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$	$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{\text{max}(-4;-3)}$
$B_{z_{\text{avg}(-5;-4)}}$	$nProt_{\text{avg}(-1;0)}$	$\sin(2\pi T/24)$	$B_{\text{avg}(-2;-1)}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B4
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 3$ hr Ahead With a 3-hr Time Window Used to Construct Input Features

$h = 3$ hr ahead			
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$B_{\mathrm{avg}(-3;0)}$	$B_{\mathrm{avg}(-3;0)}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$
$B_{x_{\mathrm{avg}(-3;\ 0)}}$	$B_{\mathrm{min}(-3;0)}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$B_{\mathrm{max}(-3;0)}$
$B_{\mathrm{min}(-3;0)}$	$B_{x_{\mathrm{min}(-3;\ 0)}}$	$B_{\mathrm{max}(-3;0)}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$
$B_{\mathrm{min}(-9;-6)}$	$B_{x_{\mathrm{min}(-6;-3)}}$	$\cos(2\pi D/365)$	$B_{\mathrm{avg}(-3;0)}$
$B_{x_{\mathrm{max}(-9;-6)}}$	$B_{y_{\mathrm{max}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-9;-6)}}$	$V_{sw_{\mathrm{avg}(-3;\ 0)}}$
$B_{y_{\mathrm{avg}(-3;\ 0)}}$	$B_{y_{\mathrm{min}(-3;\ 0)}}$	$B_{z_{\mathrm{avg}(-3;\ 0)}}$	$B_{\mathrm{max}(-6;-3)}$
$B_{y_{\mathrm{max}(-3;\ 0)}}$	$B_{z_{\mathrm{avg}(-3;\ 0)}}$	$B_{x_{\mathrm{min}(-3;\ 0)}}$	$B_{\mathrm{avg}(-6;-3)}$
$B_{y_{\mathrm{max}(-9;-6)}}$	$B_{z_{\mathrm{max}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-6;-3)}}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$
$B_{y_{\mathrm{min}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$	$\sin(2\pi T/24)$	$B_{z_{\mathrm{min}(-6;-3)}}$
$B_{y_{\mathrm{min}(-9;-6)}}$	$B_{z_{\mathrm{min}(-6;-3)}}$	$B_{y_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-6;-3)}}$
$B_{z_{\mathrm{avg}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-9;-6)}}$	$B_{\mathrm{max}(-9;-6)}$	$B_{\mathrm{max}(-9;-6)}$
$B_{z_{\mathrm{avg}(-9;-6)}}$	$V_{sw_{\mathrm{avg}(-6;-3)}}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{avg}(-6;-3)}}$
$B_{z_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{avg}(-9;-6)}}$	$B_{y_{\mathrm{max}(-3;\ 0)}}$	$B_{\mathrm{avg}(-9;-6)}$
$B_{z_{\mathrm{max}(-6;-3)}}$	$V_{sw_{\mathrm{avg}(-3;\ 0)}}$	$B_{z_{\mathrm{max}(-9;-6)}}$	$V_{sw_{\mathrm{min}(-6;-3)}}$
$B_{z_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$B_{x_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-9;-6)}}$
$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-6;-3)}}$	$B_{\mathrm{avg}(-3;0)}$	$B_{z_{\mathrm{min}(-9;-6)}}$
$V_{sw_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-9;-6)}}$	$B_{x_{\mathrm{min}(-9;-6)}}$	$V_{sw_{\mathrm{avg}(-9;-6)}}$
$nProt_{\mathrm{max}(-3;0)}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$	$B_{z_{\mathrm{avg}(-6;-3)}}$	$V_{sw_{\mathrm{min}(-9;-6)}}$
$nProt_{\mathrm{max}(-6;-3)}$	$V_{sw_{\mathrm{min}(-6;-3)}}$	$V_{sw_{\mathrm{max}(-9;-6)}}$	$B_{y_{\mathrm{max}(-3;\ 0)}}$
$nProt_{\mathrm{avg}(-3;0)}$	$V_{sw_{\mathrm{min}(-9;-6)}}$	$B_{y_{\mathrm{min}(-9;-6)}}$	$B_{\mathrm{min}(-3;0)}$
$nProt_{\mathrm{max}(-9;-6)}$	$nProt_{\mathrm{avg}(-6;-3)}$	$B_{z_{\mathrm{max}(-3;\ 0)}}$	$B_{\mathrm{min}(-6;-3)}$
$nProt_{\mathrm{min}(-3;0)}$	$nProt_{\mathrm{max}(-3;0)}$	$B_{y_{\mathrm{max}(-6;-3)}}$	$B_{z_{\mathrm{avg}(-3;\ 0)}}$
$nProt_{\mathrm{min}(-6;-3)}$	$nProt_{\mathrm{max}(-6;-3)}$	$B_{z_{\mathrm{avg}(-9;-6)}}$	$B_{y_{\mathrm{max}(-6;-3)}}$
$\sin(2\pi T/24)$	$nProt_{\mathrm{avg}(-3;0)}$	$nProt_{\mathrm{max}(-3;0)}$	$B_{\mathrm{min}(-9;-6)}$
$\cos(2\pi T/24)$	$nProt_{\mathrm{min}(-3;0)}$	$B_{\mathrm{max}(-6;-3)}$	$B_{y_{\mathrm{min}(-3;\ 0)}}$
$\sin(2\pi D/365)$	$\sin(2\pi D/365)$	$B_{x_{\mathrm{max}(-9;-6)}}$	$B_{y_{\mathrm{max}(-9;-6)}}$
$\cos(2\pi D/365)$	$\cos(2\pi D/365)$	$V_{sw_{\mathrm{avg}(-3;\ 0)}}$	$B_{x_{\mathrm{min}(-3;\ 0)}}$
$B_{\mathrm{avg}(-9;-6)}$	$B_{\mathrm{avg}(-6;-3)}$	$B_{z_{\mathrm{max}(-6;-3)}}$	$B_{z_{\mathrm{max}(-6;-3)}}$
$B_{\mathrm{max}(-3;0)}$	$B_{\mathrm{max}(-3;0)}$	$B_{x_{\mathrm{min}(-6;-3)}}$	$B_{z_{\mathrm{max}(-3;\ 0)}}$
$B_{\mathrm{max}(-6;-3)}$	$B_{\mathrm{max}(-6;-3)}$	$B_{y_{\mathrm{max}(-9;-6)}}$	$B_{x_{\mathrm{min}(-6;-3)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B5
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 6$ hr Ahead With a 1-hr Time Window Used to Construct Input Features

$h = 6$ hr ahead			
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{z_{\text{min}(-1;\ 0)}}$	$B_{\text{avg}(-1;0)}$	$B_{\text{max}(-1;0)}$	$B_{\text{max}(-1;0)}$
$B_{z_{\text{min}(-4;-3)}}$	$B_{z_{\text{min}(-1;\ 0)}}$	$V_{sw_{\text{max}(-1;\ 0)}}$	$B_{\text{max}(-2;-1)}$
$V_{sw_{\text{max}(-1;\ 0)}}$	$B_{z_{\text{min}(-2;-1)}}$	$B_{z_{\text{min}(-1;\ 0)}}$	$B_{\text{max}(-3;-2)}$
$V_{sw_{\text{min}(-1;\ 0)}}$	$V_{sw_{\text{max}(-1;\ 0)}}$	$\cos(2\pi D/365)$	$V_{sw_{\text{max}(-1;\ 0)}}$
$B_{\text{max}(-1;0)}$	$V_{sw_{\text{max}(-2;-1)}}$	$B_{z_{\text{min}(-8;-7)}}$	$B_{\text{avg}(-1;0)}$
$nProt_{\text{max}(-1;0)}$	$V_{sw_{\text{min}(-1;\ 0)}}$	$B_{z_{\text{min}(-3;-2)}}$	$B_{\text{avg}(-2;-1)}$
$nProt_{\text{min}(-1;0)}$	$V_{sw_{\text{min}(-2;-1)}}$	$B_{x_{\text{min}(-7;-6)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$
$\sin(2\pi T/24)$	$B_{\text{max}(-1;0)}$	$\sin(2\pi T/24)$	$B_{\text{max}(-4;-3)}$
$\cos(2\pi D/365)$	$nProt_{\text{max}(-1;0)}$	$B_{z_{\text{min}(-5;-4)}}$	$V_{sw_{\text{max}(-2;-1)}}$
$B_{\text{min}(-1;0)}$	$B_{\text{max}(-2;-1)}$	$B_{\text{max}(-9;-8)}$	$B_{\text{avg}(-3;-2)}$
$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{z_{\text{min}(-2;-1)}}$	$V_{sw_{\text{min}(-1;\ 0)}}$
$B_{\text{min}(-9;-8)}$	$V_{sw_{\text{avg}(-1;\ 0)}}$	$V_{sw_{\text{max}(-7;-6)}}$	$V_{sw_{\text{avg}(-2;-1)}}$
$B_{z_{\text{avg}(-5;-4)}}$	$nProt_{\text{avg}(-1;0)}$	$B_{y_{\text{max}(-1;\ 0)}}$	$B_{\text{max}(-5;-4)}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B6
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 6$ hr Ahead With a 3-hr Time Window Used to Construct Input Features

$h = 6$ hr ahead			
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$B_{\text{avg}(-3;0)}$	$B_{\text{avg}(-3;0)}$	$B_{\text{max}(-3;0)}$	$B_{\text{max}(-3;0)}$
$B_{\text{min}(-3;0)}$	$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\text{max}(-3;\ 0)}}$	$B_{\text{avg}(-3;0)}$
$B_{\text{min}(-9;-6)}$	$B_{z_{\text{max}(-3;\ 0)}}$	$B_{z_{\text{min}(-3;\ 0)}}$	$B_{z_{\text{min}(-3;\ 0)}}$
$B_{x_{\text{min}(-3;\ 0)}}$	$B_{z_{\text{min}(-3;\ 0)}}$	$\cos(2\pi D/365)$	$V_{sw_{\text{max}(-3;\ 0)}}$
$B_{y_{\text{max}(-3;\ 0)}}$	$B_{z_{\text{min}(-9;-6)}}$	$B_{z_{\text{min}(-9;-6)}}$	$B_{\text{max}(-6;-3)}$
$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\text{avg}(-6;-3)}}$	$\sin(2\pi T/24)$	$V_{sw_{\text{avg}(-3;\ 0)}}$
$B_{z_{\text{avg}(-6;-3)}}$	$V_{sw_{\text{avg}(-9;-6)}}$	$B_{x_{\text{min}(-3;\ 0)}}$	$B_{\text{avg}(-6;-3)}$
$B_{z_{\text{avg}(-9;-6)}}$	$V_{sw_{\text{avg}(-3;\ 0)}}$	$B_{z_{\text{min}(-6;-3)}}$	$B_{\text{max}(-9;-6)}$
$B_{z_{\text{min}(-3;\ 0)}}$	$V_{sw_{\text{max}(-3;\ 0)}}$	$B_{y_{\text{min}(-3;\ 0)}}$	$V_{sw_{\text{min}(-3;\ 0)}}$
$B_{z_{\text{min}(-9;-6)}}$	$V_{sw_{\text{max}(-6;-3)}}$	$B_{z_{\text{max}(-9;-6)}}$	$V_{sw_{\text{max}(-6;-3)}}$
$V_{sw_{\text{max}(-3;\ 0)}}$	$V_{sw_{\text{min}(-3;\ 0)}}$	$B_{z_{\text{max}(-3;\ 0)}}$	$B_{\text{avg}(-9;-6)}$
$V_{sw_{\text{min}(-3;\ 0)}}$	$V_{sw_{\text{min}(-6;-3)}}$	$V_{sw_{\text{max}(-9;-6)}}$	$B_{z_{\text{min}(-6;-3)}}$
$V_{sw_{\text{min}(-9;-6)}}$	$V_{sw_{\text{min}(-9;-6)}}$	$B_{x_{\text{min}(-9;-6)}}$	$V_{sw_{\text{avg}(-6;-3)}}$
$nProt_{\text{max}(-3;0)}$	$nProt_{\text{max}(-3;0)}$	$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\text{min}(-6;-3)}}$
$nProt_{\text{avg}(-3;0)}$	$nProt_{\text{avg}(-3;0)}$	$B_{y_{\text{max}(-3;\ 0)}}$	$V_{sw_{\text{max}(-9;-6)}}$
$nProt_{\text{min}(-3;0)}$	$nProt_{\text{min}(-3;0)}$	$B_{x_{\text{max}(-3;\ 0)}}$	$V_{sw_{\text{avg}(-9;-6)}}$
$\sin(2\pi T/24)$	$\sin(2\pi D/365)$	$B_{\text{max}(-9;-6)}$	$B_{z_{\text{min}(-9;-6)}}$
$\cos(2\pi D/365)$	$\cos(2\pi D/365)$	$nProt_{\text{min}(-3;0)}$	$B_{\text{min}(-3;0)}$
$B_{\text{max}(-3;0)}$	$B_{\text{max}(-3;0)}$	$B_{\text{avg}(-3;0)}$	$V_{sw_{\text{min}(-9;-6)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B7
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 9$ hr Ahead With a 1-hr Time Window Used to Construct Input Features

$h = 9$ hr ahead			
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{z_{\min(-1;\ 0)}}$	$B_{\text{avg}(-1;0)}$	$B_{\max(-1;0)}$	$B_{\max(-1;0)}$
$B_{z_{\min(-4;-3)}}$	$B_{z_{\min(-1;\ 0)}}$	$V_{sw_{\max(-2;-1)}}$	$B_{\max(-2;-1)}$
$B_{z_{\min(-7;-6)}}$	$B_{z_{\min(-2;-1)}}$	$\cos(2\pi D/365)$	$B_{\max(-3;-2)}$
$B_{z_{\min(-9;-8)}}$	$V_{sw_{\max(-1;\ 0)}}$	$B_{z_{\min(-3;-2)}}$	$B_{\text{avg}(-1;0)}$
$V_{sw_{\max(-1;\ 0)}}$	$V_{sw_{\max(-2;-1)}}$	$B_{z_{\min(-9;-8)}}$	$B_{\text{avg}(-2;-1)}$
$V_{sw_{\min(-9;-8)}}$	$V_{sw_{\min(-1;\ 0)}}$	$\sin(2\pi T/24)$	$B_{\max(-4;-3)}$
$B_{\max(-1;0)}$	$V_{sw_{\min(-2;-1)}}$	$B_{z_{\min(-1;\ 0)}}$	$B_{\text{avg}(-3;-2)}$
$nProt_{\min(-1;0)}$	$B_{\max(-1;0)}$	$B_{x_{\min(-7;-6)}}$	$V_{sw_{\max(-1;\ 0)}}$
$\sin(2\pi T/24)$	$nProt_{\max(-1;0)}$	$B_{z_{\min(-6;-5)}}$	$B_{\max(-5;-4)}$
$\cos(2\pi D/365)$	$B_{\max(-2;-1)}$	$B_{y_{\min(-1;\ 0)}}$	$V_{sw_{\text{avg}(-1;\ 0)}}$
$B_{\min(-1;0)}$	$nProt_{\min(-1;0)}$	$B_{x_{\min(-1;\ 0)}}$	$B_{\text{avg}(-4;-3)}$
$B_{z_{\text{avg}(-1;\ 0)}}$	$\cos(2\pi D/365)$	$B_{z_{\max(-9;-8)}}$	$V_{sw_{\max(-2;-1)}}$
$B_{\min(-8;-7)}$	$B_{z_{\text{avg}(-1;\ 0)}}$	$B_{z_{\min(-2;-1)}}$	$B_{\max(-6;-5)}$
$B_{\min(-9;-8)}$	$V_{sw_{\text{avg}(-1;\ 0)}}$	$V_{sw_{\max(-7;-6)}}$	$B_{\text{avg}(-5;-4)}$
$B_{z_{\text{avg}(-5;-4)}}$	$nProt_{\text{avg}(-1;0)}$	$B_{\max(-5;-4)}$	$V_{sw_{\text{avg}(-2;-1)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B8
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon $h = 9$ hr Ahead With a 3-hr Time Window Used to Construct Input Features

$h = 9$ hr ahead			
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$B_{\text{avg}(-3;0)}$	$B_{\text{avg}(-3;0)}$	$B_{\max(-3;0)}$	$B_{\max(-3;0)}$
$B_{\min(-3;0)}$	$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\max(-3;\ 0)}}$	$B_{\text{avg}(-3;0)}$
$B_{\min(-9;-6)}$	$B_{z_{\min(-3;\ 0)}}$	$\cos(2\pi D/365)$	$B_{\max(-6;-3)}$
$B_{y_{\max(-3;\ 0)}}$	$V_{sw_{\text{avg}(-9;-6)}}$	$B_{z_{\min(-3;\ 0)}}$	$V_{sw_{\max(-3;\ 0)}}$
$B_{z_{\text{avg}(-3;\ 0)}}$	$V_{sw_{\text{avg}(-3;\ 0)}}$	$\sin(2\pi T/24)$	$B_{\text{avg}(-6;-3)}$
$B_{z_{\text{avg}(-6;-3)}}$	$V_{sw_{\max(-3;\ 0)}}$	$B_{z_{\min(-9;-6)}}$	$B_{z_{\min(-3;\ 0)}}$
$B_{z_{\text{avg}(-9;-6)}}$	$V_{sw_{\max(-6;-3)}}$	$B_{x_{\min(-3;\ 0)}}$	$V_{sw_{\text{avg}(-3;\ 0)}}$
$B_{z_{\min(-3;\ 0)}}$	$V_{sw_{\min(-3;\ 0)}}$	$B_{z_{\max(-9;-6)}}$	$B_{\max(-9;-6)}$
$B_{z_{\min(-9;-6)}}$	$V_{sw_{\min(-6;-3)}}$	$B_{y_{\min(-3;\ 0)}}$	$B_{\text{avg}(-9;-6)}$
$V_{sw_{\max(-3;\ 0)}}$	$V_{sw_{\min(-9;-6)}}$	$B_{z_{\max(-3;\ 0)}}$	$V_{sw_{\min(-3;\ 0)}}$
$V_{sw_{\min(-9;-6)}}$	$nProt_{\max(-3;0)}$	$B_{z_{\min(-6;-3)}}$	$V_{sw_{\max(-6;-3)}}$
$nProt_{\max(-3;0)}$	$nProt_{\text{avg}(-3;0)}$	$B_{x_{\min(-9;-6)}}$	$V_{sw_{\text{avg}(-6;-3)}}$
$nProt_{\min(-3;0)}$	$nProt_{\min(-3;0)}$	$nProt_{\min(-3;0)}$	$B_{z_{\min(-6;-3)}}$
$\sin(2\pi T/24)$	$\sin(2\pi D/365)$	$B_{x_{\max(-9;-6)}}$	$V_{sw_{\max(-9;-6)}}$
$\cos(2\pi D/365)$	$\cos(2\pi D/365)$	$B_{y_{\max(-3;\ 0)}}$	$V_{sw_{\min(-6;-3)}}$
$B_{\max(-3;0)}$	$B_{\max(-3;0)}$	$V_{sw_{\max(-9;-6)}}$	$B_{\min(-3;0)}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B9
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon h = 12 hr Ahead With a 1-hr Time Window Used to Construct Input Features

$h = 12$ hr ahead			
FFX (1H)	RF (1H)	MRMR (1H)	MIM (1H)
$B_{\mathrm{avg}(-1;0)}$	$B_{\mathrm{avg}(-1;0)}$	$B_{\mathrm{max}(-1;0)}$	$B_{\mathrm{max}(-1;0)}$
$B_{z_{\mathrm{min}(-1;\ 0)}}$	$B_{z_{\mathrm{min}(-1;\ 0)}}$	$V_{sw_{\mathrm{max}(-2;-1)}}$	$B_{\mathrm{max}(-2;-1)}$
$B_{z_{\mathrm{min}(-4;-3)}}$	$V_{sw_{\mathrm{max}(-1;\ 0)}}$	$\cos(2\pi D/365)$	$B_{\mathrm{avg}(-1;0)}$
$B_{z_{\mathrm{min}(-9;-8)}}$	$V_{sw_{\mathrm{max}(-2;-1)}}$	$B_{z_{\mathrm{min}(-9;-8)}}$	$B_{\mathrm{max}(-3;-2)}$
$V_{sw_{\mathrm{max}(-1;\ 0)}}$	$V_{sw_{\mathrm{min}(-1;\ 0)}}$	$\sin(2\pi T/24)$	$B_{\mathrm{avg}(-2;-1)}$
$B_{\mathrm{max}(-1;0)}$	$B_{\mathrm{max}(-1;0)}$	$B_{z_{\mathrm{min}(-1;\ 0)}}$	$B_{\mathrm{avg}(-3;-2)}$
$nProt_{\mathrm{min}(-1;0)}$	$B_{\mathrm{max}(-2;-1)}$	$B_{x_{\mathrm{avg}(-9;-8)}}$	$B_{\mathrm{max}(-4;-3)}$
$\sin(2\pi T/24)$	$nProt_{\mathrm{min}(-1;0)}$	$B_{z_{\mathrm{min}(-5;-4)}}$	$B_{\mathrm{max}(-5;-4)}$
$\cos(2\pi D/365)$	$\sin(2\pi D/365)$	$B_{x_{\mathrm{min}(-1;\ 0)}}$	$B_{\mathrm{avg}(-4;-3)}$
$B_{\mathrm{min}(-1;0)}$	$\cos(2\pi D/365)$	$B_{z_{\mathrm{min}(-3;-2)}}$	$V_{sw_{\mathrm{max}(-1;\ 0)}}$
$B_{z_{\mathrm{avg}(-5;-4)}}$	$V_{sw_{\mathrm{avg}(-1;\ 0)}}$	$B_{z_{\mathrm{max}(-9;-8)}}$	$B_{\mathrm{max}(-6;-5)}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table B10
Features Selected Using FFX, RF, MRMR, and MIM Feature Selection Algorithms for the Prediction Horizon h = 12 hr Ahead With a 3-hr Time Window Used to Construct Input Features

$h = 12$ hr ahead			
FFX (3H)	RF (3H)	MRMR (3H)	MIM (3H)
$B_{\mathrm{avg}(-3;0)}$	$B_{\mathrm{avg}(-3;0)}$	$B_{\mathrm{max}(-3;0)}$	$B_{\mathrm{max}(-3;0)}$
$B_{\mathrm{min}(-3;0)}$	$B_{z_{\mathrm{avg}(-3;\ 0)}}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$	$B_{\mathrm{avg}(-3;0)}$
$B_{\mathrm{min}(-9;-6)}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$	$\cos(2\pi D/365)$	$B_{\mathrm{max}(-6;-3)}$
$B_{y_{\mathrm{max}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-6;-3)}}$	$\sin(2\pi T/24)$	$B_{\mathrm{avg}(-6;-3)}$
$B_{z_{\mathrm{avg}(-3;\ 0)}}$	$V_{sw_{\mathrm{avg}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$
$B_{z_{\mathrm{avg}(-6;-3)}}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$B_{x_{\mathrm{min}(-9;-6)}}$	$B_{\mathrm{max}(-9;-6)}$
$B_{z_{\mathrm{avg}(-9;-6)}}$	$V_{sw_{\mathrm{max}(-6;-3)}}$	$B_{z_{\mathrm{min}(-9;-6)}}$	$B_{\mathrm{avg}(-9;-6)}$
$B_{z_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$	$B_{z_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{avg}(-3;\ 0)}}$
$B_{z_{\mathrm{min}(-9;-6)}}$	$V_{sw_{\mathrm{min}(-6;-3)}}$	$B_{x_{\mathrm{max}(-3;\ 0)}}$	$B_{z_{\mathrm{min}(-3;\ 0)}}$
$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{min}(-9;-6)}}$	$B_{z_{\mathrm{max}(-9;-6)}}$	$V_{sw_{\mathrm{max}(-6;-3)}}$
$V_{sw_{\mathrm{min}(-9;-6)}}$	$nProt_{\mathrm{max}(-3;0)}$	$nProt_{\mathrm{min}(-3;0)}$	$V_{sw_{\mathrm{min}(-3;\ 0)}}$
$nProt_{\mathrm{max}(-3;0)}$	$nProt_{\mathrm{avg}(-3;0)}$	$B_{y_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{avg}(-6;-3)}}$
$nProt_{\mathrm{min}(-3;0)}$	$nProt_{\mathrm{min}(-3;0)}$	$B_{z_{\mathrm{min}(-6;-3)}}$	$B_{z_{\mathrm{min}(-6;-3)}}$
$\sin(2\pi T/24)$	$\sin(2\pi D/365)$	$B_{y_{\mathrm{min}(-9;-6)}}$	$B_{\mathrm{min}(-3;0)}$
$\cos(2\pi D/365)$	$\cos(2\pi D/365)$	$B_{x_{\mathrm{min}(-3;\ 0)}}$	$V_{sw_{\mathrm{max}(-9;-6)}}$
$B_{\mathrm{max}(-3;0)}$	$B_{\mathrm{max}(-3;0)}$	$V_{sw_{\mathrm{max}(-3;\ 0)}}$	$V_{sw_{\mathrm{min}(-6;-3)}}$

Note. RF, MRMR, and MIM provide the ordered list of variables, with the most important variable at the top of the list. FFX does not provide the feature importance ranking.

Table C1
Fit Performance Statistics of the NN-FFX Models for Different Prediction Horizons Computed on the Training Set

Prediction horizon, hr	0	3	6	9	12
Number of values in comparison	42479	42479	42479	42479	42479
Intercept of the linear fit	0.2597	0.4363	0.7578	0.9467	1.0866
Slope of the linear fit	0.8743	0.7671	0.5935	0.4951	0.4225
Pearson correlation coefficient (R)	0.9376	0.8758	0.7696	0.7048	0.6493
Root mean square error (RMSE)	0.4822	0.6668	0.8827	0.9817	1.0511
Mean absolute error (MAE)	0.3758	0.5062	0.6714	0.7473	0.8007
Mean error (ME, or bias)	0.0238	0.0020	−9.0616e-05	0.0040	0.0126
Prediction efficiency (PE)	0.8787	0.7670	0.5923	0.4967	0.4216

Table C2
Fit Performance Statistics of the NN-FFX Models for Different Prediction Horizons Computed on the Validation Set

Prediction horizon, hr	0	3	6	9	12
Number of values in comparison	10080	10080	10080	10080	10080
Intercept of the linear fit	0.2780	0.4870	0.8496	1.0579	1.2035
Slope of the linear fit	0.8618	0.7503	0.5670	0.4659	0.3872
Pearson correlation coefficient (R)	0.9303	0.8666	0.7518	0.6758	0.6105
Root mean square error (RMSE)	0.4987	0.6792	0.8936	0.9990	1.0755
Mean absolute error (MAE)	0.3853	0.5150	0.6835	0.7596	0.8197
Mean error (ME, or bias)	0.0062	−0.0012	0.0054	0.0180	0.0141
Prediction efficiency (PE)	0.8654	0.7509	0.5652	0.4563	0.3721

Table C3
Fit Performance Statistics of the NN-FFX Models for Different Prediction Horizons Computed on the Test Set

Prediction horizon, hr	0	3	6	9	12
Number of values in comparison	5880	5880	5880	5880	5880
Intercept of the linear fit	0.2570	0.4359	0.7569	0.9542	1.1052
Slope of the linear fit	0.8653	0.7642	0.5884	0.4836	0.4061
Pearson correlation coefficient (R)	0.9350	0.8745	0.7695	0.7056	0.6477
Root mean square error (RMSE)	0.4964	0.6754	0.8898	0.9909	1.0632
Mean absolute error (MAE)	0.3872	0.5101	0.6710	0.7480	0.8029
Mean error (ME, or bias)	−0.0059	−0.0199	−0.0387	−0.0436	−0.0374
Prediction efficiency (PE)	0.8740	0.7646	0.5914	0.4965	0.4184

Acknowledgments
We acknowledge support by the European Unions Horizon 2020 research and innovation program under grant agreement No 776287 SWAMI. IZ was supported by Geo.X, the Research Network for Geosciences in Berlin and Potsdam, under Grant No SO_087_GeoX. Solar wind data and geomagnetic indices were obtained from http://omniweb.gsfc.nasa.gov/form/dx1.html. Kp index of geomagnetic activity was obtained from the GFZ Potsdam website (https://www.gfz-potsdam.de/en/kp-index/). IZ thanks Nikita Aseev, Alexander Drozdov, Hayley Allison, and Dominika Boneberg for helpful discussions.

References

Agapitov, O., Artemyev, A., Mourenas, D., Mozer, F., & Krasnoselskikh, V. (2015). Empirical model of lower band chorus wave distribution in the outer radiation belt. *Journal of Geophysical Research: Space Physics*, *120*, 10–425. https://doi.org/10.1002/2015JA021829

Ali, R., Siddiqi, M. H., & Lee, S. (2015). Rough set-based approaches for discretization: A compact review. *Artificial Intelligence Review*, *44*(2), 235–263.

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

Balikhin, M., Boaghe, O., Billings, S., & Alleyne, H. C. (2001). Terrestrial magnetosphere as a nonlinear resonator. *Geophysical Research Letters*, *28*(6), 1123–1126.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin, Heidelberg: Springer-Verlag.

Boaghe, O., Balikhin, M., Billings, S., & Alleyne, H. (2001). Identification of nonlinear processes in the magnetospheric dynamics and forecasting of DST index. *Journal of Geophysical Research*, *106*(A12), 30,047–30,066.

Boberg, F., Wintoft, P., & Lundstedt, H. (2000). Real time Kp predictions from solar wind data using neural networks. *Physics and Chemistry of the Earth, Part C: Solar, Terrestrial & Planetary Science*, *25*(4), 275–280.

Bollacker, K. D., & Ghosh, J. (1996). Linear feature extractors based on mutual information. In *Proceedings of 13th International Conference on Pattern Recognition*, *2*, pp. 720–724.

Brautigam, D., & Albert, J. (2000). Radial diffusion analysis of outer radiation belt electrons during the October 9, 1990, magnetic storm. *Journal of Geophysical Research*, *105*(A1), 291–309.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. Wadsworth Int. *Group*, *37*(15), 237–251.

Bruinsma, S., Sutton, E., Solomon, S., Fuller-Rowell, T., & Fedrizzi, M. (2018). Space weather modeling capabilities assessment: Neutral density for orbit determination at low earth orbit. *Space Weather*, *16*, 1806–1816. https://doi.org/10.1029/2018SW002027

Carpenter, D. L., & Anderson, R. R. (1992). An ISEE/whistler model of equatorial electron density in the magnetosphere. *Journal of Geophysical Research*, *97*(A2), 1097–1108. https://doi.org/10.1029/91JA01548

Chu, X. N., Bortnik, J., Li, W., Ma, Q., Angelopoulos, V., & Thorne, R. M. (2017). Erosion and refilling of the plasmasphere during a geomagnetic storm modeled by a neural network. *Journal of Geophysical Research: Space Physics*, *122*, 7118–7129. https://doi.org/10.1002/2017JA023948

Chu, X., Bortnik, J., Li, W., Ma, Q., Denton, R., & Yue, C. (2017). A neural network model of three-dimensional dynamic electron density in the inner magnetosphere. *Journal of Geophysical Research: Space Physics*, *122*, 9183–9197. https://doi.org/10.1002/2017JA024464

Costello, K. A. (1998). *Moving the Rice MSFM into a real-time forecast mode using solar wind driven forecast modules*: Doctoral dissertation, Rice University. Retrieved from https://scholarship.rice.edu/bitstream/handle/1911/19251/9827384.PDF?sequence=1&isAllowed=y

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Denton, M., Henderson, M. G., Jordanova, V. K., Thomsen, M. F., Borovsky, J. E., Woodroffe, J., & Pitchford, D. (2016). An improved empirical model of electron and ion fluxes at geosynchronous orbit based on upstream solar wind conditions. *Space Weather*, *14*, 511–523. https://doi.org/10.1002/2016SW001409

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, *3*(02), 185–205.

Emery, B. A., Coumans, V., Evans, D. S., Germany, G. A., Greer, M. S., Holeman, E., & Xu, W. (2008). Seasonal, Kp, solar wind, and solar flux variations in long-term single-pass satellite estimates of electron and ion auroral hemispheric power. *Journal of Geophysical Research*, *113*, A06311. https://doi.org/10.1029/2007JA012866

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of statistics*, *29*, 1189–1232.

Gao, W., Kannan, S., Oh, S., & Viswanath, P. (2017). Estimating mutual information for discrete-continuous mixtures, *Advances in Neural Information Processing Systems* (pp. 5986–5997).

Goldstein, J., Pascuale, S. D., Kletzing, C., Kurth, W., Genestreti, K., Skoug, R., & Spence, H. (2014). Simulation of Van Allen Probes plasmapause encounters. *Journal of Geophysical Research: Space Physics*, *119*, 7464–7484. https://doi.org/10.1002/2014JA020252

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*: MIT press.

Heidrich-Meisner, V., & Wimmer-Schweingruber, R. F. (2018). Solar wind classification via k-means clustering algorithm. In E. Camporeale, S. Wing, & J. R. Johnson (Eds.), *Machine Learning Techniques for Space Weather* (pp. 397–424): Elsevier.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, *1*, pp. 278–282.

Ji, E. Y., Moon, Y. J., Park, J., Lee, J. Y., & Lee, D. H. (2013). Comparison of neural network and support vector machine methods for Kp forecasting. *Journal of Geophysical Research: Space Physics*, *118*, 5109–5117. https://doi.org/10.1002/jgra.50500

Jiang, S. Y., & Wang, L. X. (2016). Efficient feature selection based on correlation measure between continuous and discrete features. *Information Processing Letters*, *116*(2), 203–215.

Korth, H., Thomsen, M. F., Borovsky, J. E., & McComas, D. J. (1999). Plasma sheet access to geosynchronous orbit. *Journal of Geophysical Research*, *104*, 25,047–25,061. https://doi.org/10.1029/1999JA900292

Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press.

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

Maynard, N., & Chen, A. (1975). Isolated cold plasma regions: Observations and their relation to possible production mechanisms. *Journal of Geophysical Research*, *80*(7), 1009–1013.

McConaghy, T. (2011). FFX: Fast, scalable, deterministic symbolic regression technology. In R. Riolo, E. Vladislavleva, & J. H. Moore (Eds.), *Genetic Programming Theory and Practice IX* (pp. 235–260). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-1770-5_13

Newell, P., Sotirelis, T., Liou, K., Meng, C. I., & Rich, F. (2007). A nearly universal solar wind-magnetosphere coupling function inferred from 10 magnetospheric state variables. *Journal of Geophysical Research*, *112*, A01206. https://doi.org/10.1029/2006JA012015

Orlova, K., Spasojević, M., & Shprits, Y. (2014). Activity-dependent global model of electron loss inside the plasmasphere. *Geophysical Research Letters*, *41*, 3744–3751. https://doi.org/10.1002/2014GL060100

Ozeke, L. G., Mann, I. R., Murphy, K. R., Jonathan Rae, I., & Milling, D. K. (2014). Analytic expressions for ULF wave radiation belt radial diffusion coefficients. *Journal of Geophysical Research: Space Physics*, *119*, 1587–1605. https://doi.org/10.1002/2013JA019204

Peng, L. F., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

Pierrard, V., Goldstein, J., André, N., Jordanova, V. K., Kotova, G. A., Lemaire, J. F., et al. (2009). Recent progress in physics-based models of the plasmasphere. *Space Science Reviews*, *145*(1), 193–229. https://doi.org/10.1007/s11214-008-9480-7

Shprits, Y. Y., Meredith, N. P., & Thorne, R. M. (2007). Parameterization of radiation belt electron loss timescales due to interactions with chorus waves. *Geophysical Research Letters*, *34*, L11110. https://doi.org/10.1029/2006GL029050

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

Sturges, H. A. (1926). The choice of a class interval. *Journal of the American Statistical Association*, *21*(153), 65–66. https://doi.org/10.1080/01621459.1926.10502161

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

Wang, J., Zhong, Q., Liu, S., Miao, J., Liu, F., Li, Z., & Tang, W. (2015). Statistical analysis and verification of 3-hourly geomagnetic activity probability predictions. *Space Weather*, *13*, 831–852. https://doi.org/10.1002/2015SW001251

Wing, S., & Johnson, J. R. (2019). Applications of information theory in solar and space physics. *Entropy*, *21*(2), 140. https://doi.org/10.3390/e21020140

Wing, S., Johnson, J. R., Camporeale, E., & Reeves, G. D. (2016). Information theoretical approach to discovering solar wind drivers of the outer radiation belt. *Journal of Geophysical Research: Space Physics*, *121*, 9378–9399. https://doi.org/10.1002/2016JA022711

Wing, S., Johnson, J. R., Jen, J., Meng, C. I., Sibeck, D. G., Bechtold, K., & Takahashi, K. (2005). Kp forecast models. *Journal of Geophysical Research*, *110*, A04203. https://doi.org/10.1029/2004JA010500

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

Xu, F., & Borovsky, J. E. (2015). A new four-plasma categorization scheme for the solar wind. *Journal of Geophysical Research: Space Physics*, *120*, 70–100. https://doi.org/10.1002/2014JA020412

Yau, A. W., Peterson, W., & Abe, T. (2011). Influences of the ionosphere, thermosphere and magnetosphere on ion outflows. In W. Liu, & M. Fujimoto (Eds.), *The Dynamic Magnetosphere* (pp. 283–314). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-007-0501-2_16

Zhelavskaya, I. S., Shprits, Y. Y., & Spasojević, M. (2017). Empirical modeling of the plasmasphere dynamics using neural networks. *Journal of Geophysical Research: Space Physics*, *122*, 11,227–11,244. https://doi.org/10.1002/2017JA024406

Zhelavskaya, I. S., Shprits, Y. Y., & Spasojević, M. (2018). Chapter 12—Reconstruction of plasma electron density from satellite measurements via artificial neural networks. In E. Camporeale, S. Wing, & J. R. Johnson (Eds.), *Machine Learning Techniques for Space Weather* (pp. 301–327): Elsevier. Retrieved from http://www.sciencedirect.com/science/article/pii/B9780128117880000123