Received: 15 March 2022 | Revised: 21 August 2022 | Accepted: 20 September 2022

DOI: 10.1002/env.2772

RESEARCH ARTICLE

WILEY

An illustration of model agnostic explainability methods applied to environmental data

Christopher K. Wikle[1] | **Abhirup Datta**[2] | **Bhava Vyasa Hari**[3] | **Edward L. Boone**[4] | **Indranil Sahoo**[4] | **Indulekha Kavila**[5] | **Stefano Castruccio**[6] | **Susan J. Simmons**[7] | **Wesley S. Burr**[8] | **Won Chang**[9]

[1]Department of Statistics, University of Missouri, Columbia, Missouri, USA

[2]Department of Biostatistics, Johns Hopkins University, Baltimore, Maryland, USA

[3]Wipro Limited, Bengaluru, India

[4]Department of Statistical Sciences and Operations Research, Virginia Commonwealth University, Richmond, Virginia, USA

[5]School of Pure and Applied Physics, Mahatma Gandhi University, Athirampuzha, Kerala, India

[6]Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, Indiana, USA

[7]Institute for Advanced Analytics, North Carolina State University, Raleigh, North Carolina, USA

[8]Department of Mathematics, Trent University, Peterborough, Ontario, Canada

[9]Department of Mathematical Sciences, University of Cincinnati, Cincinnati, Ohio, USA

Correspondence
Christopher K. Wikle, Department of Statistics, University of Missouri, 146 Middlebush Hall, Columbia, MO 65203, USA.
Email: wiklec@missouri.edu

Funding information
National Institute of Environmental Health Sciences, Grant/Award Number: R01ES033739; National Science Foundation, Grant/Award Numbers: SES-1853096, DMS-1915803; Natural Sciences and Engineering Research Council of Canada, Grant/Award Number: 2017-04741

Abstract

Historically, two primary criticisms statisticians have of machine learning and deep neural models is their lack of uncertainty quantification and the inability to do inference (i.e., to explain what inputs are important). Explainable AI has developed in the last few years as a sub-discipline of computer science and machine learning to mitigate these concerns (as well as concerns of fairness and transparency in deep modeling). In this article, our focus is on explaining which inputs are important in models for predicting environmental data. In particular, we focus on three general methods for explainability that are model agnostic and thus applicable across a breadth of models without internal explainability: "feature shuffling", "interpretable local surrogates", and "occlusion analysis". We describe particular implementations of each of these and illustrate their use with a variety of models, all applied to the problem of long-lead forecasting monthly soil moisture in the North American corn belt given sea surface temperature anomalies in the Pacific Ocean.

KEYWORDS

explainable AI, feature shuffling, LIME, machine learning, Shapley values

1 | INTRODUCTION

A long-standing criticism of many algorithmic-motivated machine learning methods and modern deep neural network "artificial intelligence (AI)" methods has been the "black-box" nature of the models that prohibit uncertainty quantification and inference in the context of understanding which inputs are important for producing model predictions and/or classifications. In recent years, the sub-discipline of computer science known as *explainable AI* (sometimes referred to as XAI) has arisen to address these concerns. The literature on related topics has been expanding rapidly in recent years. Two recent papers describe modern approaches for uncertainty quantification (Abdar et al., 2021) and explainability (Samek et al., 2021). Comprehensive overviews are also given in Rudin et al. (2022) and Molnar (2022). Our focus in this work is on the explainability component, primarily because some of these explainability methods are not well-known in statistics, yet have broad application across a variety of models (i.e., they are "model agnostic"). In particular, we are interested in their application to predictive models used for environmental data.

Environmental statistics is concerned with a wide variety of problems such as: estimating ecological abundance, spatial and spatio-temporal prediction, climatological downscaling, long-lead prediction, and understanding health effects of environmental exposures, to name a few. Here, we are interested in problems for which a specified set of inputs are related to a specified set of outputs. In recent years, spatial and spatio-temporal prediction (interpolation) has been a big part of the environmental statistics literature. We are not particularly interested in those methods here because traditional best linear unbiased prediction approaches (e.g., kriging and its variants) provide a built-in estimate of the "weight" associated with each observation that is used to predict a given observation. Indeed, this is true of any predictive model that can be formulated as a linear model. Rather, here we are primarily interested in explaining models for which the response is modeled as a nonlinear function of the inputs. For example, we would like to know which geographic regions in the Pacific Ocean are most relevant to predicting soil moisture in the "corn belt" region of the midwest United States (US).

We illustrate three model agnostic explainability approaches in this manuscript. First, we consider simple feature shuffling methods where feature importance is determined by how much the prediction error of a model varies as individual features are shuffled, randomized, or perturbed (Section 2.1). The second approach we consider is the local interpretable model-agnostic explanations (LIME) method (Section 2.2), which is perhaps the best known local interpretable surrogate approach. Finally, we consider Shapley values (Section 2.3), a sophisticated occlusion analysis method that takes a game theoretic approach to explaining the output of any machine learning model. These and other related occlusion analysis methods are still underutilized in statistics.

The article is organized as follows. Section 2 gives an overview of the methods associated with implementation and interpretation of feature shuffling, LIME, and Shapley values. This is followed in Section 3 by an introduction to the long-lead forecasting environmental data example that is used to demonstrate the utility of these methods. Section 4 gives an overview of the machine learning algorithms and statistical models that will be used as frameworks to demonstrate the application of the explainability methods. Section 5 then demonstrates the use of these explainability approaches on a variety of models applied to the long-lead forecasting environmental data set. We conclude in Section 6 with a discussion, providing recommendations for the use of these algorithms in general, and an outline of future work needed for effective use of these methods and approaches in environmental statistics.

2 | MODEL AGNOSTIC EXPLAINABILITY METHODS

The essential problem of interest here is using general methods to distribute or explain the prediction score of a model output given known input features. It is then assumed that the attribution to these input features can be interpreted as the "importance" of the feature to the prediction. We treat all models as black boxes, even if we understand some aspects of their internal connections. The goal is that by understanding why a model makes a specific prediction, it will help the model user determine how much the model can be trusted and/or provide mechanistic insight and possibly causality.

As mentioned in Section 1, the methods we are concerned with here fall under the area of machine learning research known as explainable AI (XAI). This is distinct from the more recent area of research known as "Interpretable AI", which is focused on constructing models that are inherently interpretable by humans (see Rudin et al., 2022, for an overview and connections to XAI). In that work, the authors make the point that techniques such as those discussed below, tools for explaining black box models, are not needed for inherently interpretable models because models that are interpretable explicitly show what variables are being used, and how. The recent interest in building physical constraints into deep neural networks is an example of an interpretable AI model (e.g., Huang et al., 2021; Mohan

et al., 2020; Reichstein et al., 2019). Rudin et al. (2022) further make clear that Interpretable AI is not a subset of XAI, although the two are often used interchangeably. We agree that interpretable models are the "gold standard", but black box methods have proven quite useful for many prediction and classification tasks and they are not going away; thus, we should strive to explain them as much as possible.

Many statistical and machine learning methods have algorithm-specific approaches to facilitate explainability. For example, "attention mechanisms" or "pixel attribution saliency maps" in neural networks (Molnar, 2022). This is also true of tree-based methods such as random forests and boosting, for which there are well-established approaches to determine feature importance based on which variables are used in the splits associated with the tree construction (Breiman, 2001a). Our interest here is on flexible methods that are model agnostic, or at least largely so, and can be applied to a wide variety of models used to predict environmental data. In general, model-agnostic methods can be characterized as "global" or "local". Global approaches describe the importance or effect of features on model behavior on average. Examples of such methods include "partial dependence plots", "accumulated local effects plots", "functional decomposition", "permutation feature importance", "global surrogates", and others (see Chapter 8 of Molnar, 2022, for these and other approaches). For example, in our environmental example discussed in Section 3, we are interested in which predictors (locations) in the tropical Pacific Sea Surface Temperature Anomaly (SSTA) data set are most important, on average, for forecasting soil moisture in the US corn belt 3 months in the future across all years. Local approaches, as the name suggests, describes the importance of features on particular instances of the data. In our environmental example, this might be to identify SSTA locations that are important for predicting soil moisture in spring 2016 (at the end of strong El Niño event). There are several local model agnostic procedures such as "local surrogates" (e.g., LIME) and "occlusion analysis" (e.g., Shapley values) that have proven useful (see Chapter 9 of Molnar, 2022, for a comprehensive discussion). We note that in some cases it is possible to use a traditionally global procedure for local analysis (e.g., permutation feature importance) and local procedures (such as LIME and Shapley values) can be averaged across instances to give a global interpretation.

The remainder of this section describes the three model agnostic approaches we consider in our environmental forecasting application: feature shuffling, LIME, and Shapley values. We implement them from both a global and local perspective in our analysis in Section 5.

2.1 | Feature shuffling

Variable importance, or more specifically, permutation feature importance (also called "feature shuffling"), was initially developed in the context of machine learning algorithms for random forests by Breiman (2001a), under the impetus of understanding the interaction of variables providing classification accuracy. Similar work was done on neural networks and perturbation of inputs around the same time (e.g., Gevrey et al., 2003; Recknagel et al., 1997). More recently, Fisher et al. (2019) proposed a general framework suitable to any input-output model, which we briefly summarize here using the same notation as in the original work, and assuming, without loss of generality, only two covariates. In this framework, the authors focused on the problem of how much prediction models rely on specific covariates to achieve their accuracy. The authors emphasized that existing variable importance measures do not account for the fact that multiple prediction models can fit the data equally (or near-equally) well, a phenomenon labeled the "Rashomon Effect" (introduced into statistics by Breiman (2001b) as the multiplicity of good models that may consider the predictors in different ways).

Consider the random variable $Z = (Y, X_1, X_2)$ for outcome Y and two predictors, X_1, X_2. Now, consider two independent random variates from the same distribution as Z, $Z^{(a)} = (Y^{(a)}, X_1^{(a)}, X_2^{(a)})$ and $Z^{(b)} = (Y^{(b)}, X_1^{(b)}, X_2^{(b)})$, Let f be a fixed prediction model of interest. Then the question is how much f relies on covariate X_1 to predict Y. All prediction models are considered as measurable functions f from the predictor space $\mathcal{X}$ to the response space $\mathcal{Y}$, and we may evaluate model performance using non-negative loss functions, say, $\mathcal{L}$. So in this case of two observed random variates (realizations or instances), we are interested in assessing $\mathcal{L}$ for model f if random variate (b) is considered, but with the first covariate substituted with the random variate (a). In particular we are interested in assessing the expected value, $e_{\text{switch}}(f) = \mathrm{E}\left[\mathcal{L}\left(f, \left(Y^{(b)}, X_1^{(a)}, X_2^{(b)}\right)\right)\right]$, and comparing it with the expected value of the same loss without replacing the covariate: that is, $e_{\text{orig}}(f) = \mathrm{E}\left[\mathcal{L}\left(f, \left(Y^{(b)}, X_1^{(b)}, X_2^{(b)}\right)\right)\right] = \mathrm{E}\left[\mathcal{L}(f, Z)\right]$, as $Z^{(b)}$ and Z follow the same distribution. From these two quantities, the *model reliance (MR)* is then defined as

$$\mathrm{MR}(f) \equiv \frac{e_{\text{switch}}(f)}{e_{\text{orig}}(f)}, \tag{1}$$

although an alternative definition can be provided using the difference instead (Fisher et al., 2019, Appendix A.5). This MR has fairly intuitive interpretation: higher values of MR(f) signify greater reliance of f on the covariate swapped (X_1 in this case).

In terms of estimation, if we assume observations $\mathbf{Z}^{(i)} = (Y^{(i)}, X_1^{(i)}, X_2^{(i)}), i = 1, \ldots, n$, then $e_{\text{orig}}(f)$ can be inferred from a sample mean of the loss function across the observations,

$$\hat{e}_{\text{orig}}(f) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(f, \mathbf{Z}^{(i)}),$$

whereas $e_{\text{switch}}(f)$ can be estimated by considering the sample mean of the loss functions evaluated at all permutations where the first covariate is swapped with other observations:

$$\hat{e}_{\text{switch}}(f) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i}\mathcal{L}(f, (Y^{(j)}, X_1^{(i)}, X_2^{(j)})).$$

Then, MR(f) can be estimated by considering the ratio of these two estimates. As both estimates are U-statistics, unbiasedness and asymptotic normality can be retrieved with some mild conditions of finite moments (Fisher et al., 2019).

2.1.1 | Feature perturbation

Feature shuffling is useful, but can be difficult to implement in some cases (e.g., with time-dependent features). A related, but simpler, alternative is to consider perturbing features in a systematic way to determine their influence. This intuitive idea has long been used in modeling nonlinear systems (sensitivity analysis; e.g., Werbos, 1982) and in regression diagnostics (covariate perturbation; e.g., Cook & Weisberg, 1991). More recently, this approach has been used in deep neural modeling to increase explainability (feature perturbation; e.g., Hassan et al., 2021; Wang et al., 2020; Wickramasinghe et al., 2021).

The idea is quite simple. Our interest is in explaining the importance of particular features in the prediction of our response. That is, we consider the effect of input features, X, on a response Y given a fitted model, $\hat{f}(X, \theta)$, with parameters θ. One way to do this is simply to see how much the predicted response, $\hat{Y}$, changes as the fitted model is interrogated with perturbed features, where the features are perturbed by adding some value to one feature (or group of features) at time. The added value can be a random draw from an appropriate distribution, or may simply be an additive constant (e.g., one standard deviation). A prediction summary, such as mean square error (MSE), is calculated for the predictions from the perturbed input, say MSE_{pert}, and for the predictions from the unperturbed input, say MSE_{pred}. One can then calculate model reliance metrics as in Equation (1). It is often useful to plot these metrics. For example, differences or ratios of the prediction metrics can be plotted as function of the feature label to visualize the impact of each feature. In the context of spatial predictors, one would simply visualize the ratio of prediction metrics as a function of location (e.g., see Section 5.3).

2.2 | Local interpretable model-agnostic explanations (LIME)

The LIME approach to model explainability was developed by Ribeiro et al. (2016). The most important aspect of LIME is the surrogate model. Generally, a surrogate model is a model that emulates a complex black-box model. A local surrogate model is one that only needs to emulate the black-box model in local areas of the data's input/output space. LIME trains many local surrogate models in order to explain individual predictions rather than a more global surrogate model. For this to work, the local surrogate models must perform similarly to the black-box model in the neighborhood of each input output pair being explained. This is sometimes referred to as "local faithfulness." Thus, it is important how the surrogate models are trained. Similar to the notions of feature shuffling described in Section 2.1, the LIME procedure builds local surrogates by first perturbing the dataset to get the black-box predictions for the data points one is interested in explaining. These perturbed samples are weighted according to their proximity to the data point of interest. Then, one trains a weighted, interpretable model, such as a linear regression, on the perturbed data set, where the black-box predictions are used as the response. Finally, the prediction can be explained by interpreting this fitted local model. Molnar (2022) provides an accessible introduction to LIME.

Specifically, LIME taps the decision making process of the black-box model to generate a measure of the relative positive/negative importance assigned by the model to each of the features that enters its decision making process. Ribeiro et al. (2016) give the following examples distinguishing between features and interpretable data representations: the presence/absence of particular words (in contrast with the less easily humanly understandable features like word embeddings) in text classification problems, and the presence/absence of a contiguous patch of similar pixels (while the classifier may be representing the image as a tensor with three color channels per pixel) in image classification problems. LIME generates explanations by approximating the model ($f : \mathbb{R}^d \to \mathbb{R}$; e.g., a neural net model) locally with an interpretable model ($g : \mathbb{R}^{d'} \to \mathbb{R}$; e.g., a linear model) and repeating this for several individual representative predictions to provide a global understanding. It learns an interpretable model locally around the prediction. The method itself is summarized below using the same notation as in Ribeiro et al. (2016). The explanation provided by LIME is obtained by the following formulation:

$$\xi(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g). \tag{2}$$

Here, the class of potentially interpretable models G (e.g., the class of linear models such that $g(z') = \beta_g \cdot z'$), fidelity functions $\mathcal{L}$ (locality-aware loss, where π_x represents the locality function; see below), and complexity measures $\Omega(g)$ may be chosen suitably and the search may be conducted, using perturbations. For $x \in \mathbb{R}^d$ in the original representation of an instance being explained and $x' \in \{0,1\}^{d'}$ a binary vector denoting an interpretable representation of x, the domain of g is $\{0,1\}^{d'}$. That is, g acts over the presence/absence of interpretable components. Every $g \in G$ may not be simple enough to be interpretable and the complexity measure $\Omega(g)$ is a penalty for complexity (e.g., the time needed to compute $f(x)$ along with the number of samples examined, the number of non-zero weights in the case of linear models, etc.). An example for the locality-aware loss function $\mathcal{L}$ (a measure of how unfaithful g is in approximating f in the locality defined by $\pi_x(z)$) is

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)(f(z) - g(z'))^2,$$

with weight $\pi_x(z) = \exp(-D(x, z)^2/\sigma^2)$, where D is a suitable distance function between observations in data space, σ is a width measure, and z and z' are set as follows. For an instance x, instances around x' are sampled by drawing non-zero elements uniformly at random (with the number of such draws also uniformly sampled) to get a perturbed sample $z' \in \{0,1\}^{d'}$ from which the sample in the original representation $z \in \mathbb{R}^d$ is obtained and $f(z)$ determined. Note that $f(z)$ is called a label for the explanation model. Given the dataset $\mathcal{Z}$ of perturbed models and their associated labels, Equation (2) may be optimized to get an explanation $\xi(x)$. The coefficients of the linear model, β_g, are recovered by weighted linear regression.

In practice, N "perturbed" samples $z'_i, i = 1, 2, \ldots, N$ are drawn from x', the interpretable version of the instance x being explained. The "perturbed" instances are recovered in the original feature space as z. The labels $f(z)$ are generated and hence the explanation. Note that the choice of G, such as sparse linear models, means that if the underlying model is highly non-linear, even in the locality of the prediction, LIME may not return a faithful explanation.

LIME has only recently started to be used in environmental and ecological applications (e.g., Cha et al., 2021; Ryo et al., 2021; Taconet et al., 2021).

2.3 | Shapley values

The Shapley value, coined by Shapley (Roth, 1988; Shapley, 1953), is a metric from game theory for assigning credit to players in a fair manner depending on their contribution to the total payout from a game. In the game-theoretic context, "players" cooperate in a coalition and receive a certain profit from this cooperation. The use of Shapley values for features in models such as regression or machine learning is a more recent development, with the first connection being the work of Lipovetsky and Conklin (2001) (starting in 1998 and leading up to this article), the proposal for use in some machine learning models by Cohen et al. (2005), and a full generalized model development in the form used in current software packages by Štrumbelj and Kononenko (2014). This work led to the development of the `SHAP` package for Python (Lundberg & Lee, 2017), which also has an R wrapper (Maksymiuk et al., 2020), leading to widespread use. Work continues on adaptation and expansion of the method (e.g., Merrick & Taly, 2020) and the values themselves are widely used in applied machine learning areas such as: medicine (e.g., Ibrahim et al., 2020; Smith & Alvarez, 2021), chemistry and

pharmaceutical research (e.g., Rodríguez-Pérez & Bajorath, 2019, 2020), and economics (e.g., Antipov & Pokryshevskaya, 2020), among many others.

Shapley values were proposed for use in learning models for quantification of the contribution of each feature value to the prediction, as compared to a baseline of the average prediction. When considering classical linear models (i.e., linear or generalized linear), such quantification is simple: each feature is the product of the weight of the feature and the value of the feature, due to the linearity of the model. Black-box models such as neural networks are more complex (e.g., non-linear regression analogues), and this simple quantification does not work. To connect the game-theoretic context to machine learning predictions for interpretability, consider the "game" as a prediction task for a single instance of the dataset. The corresponding "gain" is the actual prediction for this instance minus the average prediction for all instances, and the "players" are the feature values of the instance that collaborate to receive the gain (i.e., predict a certain value).

As in Štrumbelj and Kononenko (2014), we may formally define the contribution of p different features relative to a model's average prediction conditional on a subset Q of feature values being known: $f_Q(x) = \mathrm{E}\left[f\middle|X_i = x_i, i \in Q\right]$, for $Q \subseteq S = \{1, 2, \ldots, p\}$ any subset of features and f the model. This then provides the contribution of a subset, namely: $\Delta_Q(x) = f_Q(x) - f_{\{\}}(x)$, the difference of the contribution of set Q as compared to the empty set. This is the change in prediction for a model connected to the observation of the values of a subset Q of features. Mapping back to the linear contribution approach valid for additive models requires mapping 2^p terms into p total contributions, one for each feature's value. Adding defined interactions, the authors then give the explicit definition (proved in Štrumbelj & Kononenko, 2010):

$$\phi_i(x) = \sum_{Q \subseteq S \setminus \{i\}} \frac{|Q|!\,(|S| - |Q| - 1)!}{|S|!} \left(\Delta_{Q \cup \{i\}}(x) - \Delta_Q(x)\right), \tag{3}$$

which is equivalent to the Shapley value (Shapley, 1953). Generally speaking, the p features have a "grand coalition" that has a certain "worth" (predictive power), Δ_S. We are (from above) able to determine how much each subset coalition is worth (Δ_Q). The goal is to distribute the prediction among the features in a fair way, taking into account all sub-coalitions (subsets of features). Shapley values are one such solution, satisfying certain desirable properties—they are a partition; the contributions are normalized; any feature with no impact on prediction is assigned a zero contribution; and local contributions are additive.

Practically, Equation (3) is exponential in complexity, making it impractical for use on interesting data sets of reasonable feature set size. Štrumbelj and Kononenko (2014) detail a computational approximation in a form of Monte Carlo integration, and also propose the use of quasi-random sampling (for the variance on the estimate of the Shapley value see Molnar, 2022). They further adapt the number of samples drawn for each feature relative to the feature's variance, keeping approximation error consistent across features despite differing population variances. These improvements are included in the Maksymiuk et al. (2020) and Lundberg and Lee (2017) implementations for R and Python. In model-building, Shapley values are useful for assigning weight (worth, as discussed above) to elements of a model, allowing an analyst to determine which elements, components, or aspects of a black-box model are most important to the predictive power of the model. Note that packages used for the estimation of Shapley values may use methods that are model-specific. For example, Aas et al. (2021) propose packages that take dependence into account for several models with various degrees of feature dependence. In the case of regression, the Shapley value and the Owen value reflect the average marginal contribution of individual regressor variables and individual groups of regressor variables to R^2, respectively (Hüttner & Sunder, 2011).

3 | ENVIRONMENTAL DATA EXAMPLE: LONG-LEAD FORECASTING OF SOIL MOISTURE

Soil moisture is an important driver of processes such as agricultural production, wildfire intensity, and hydrological runoff, to name a few. Thus, for management purposes, it can be useful to have skillful long-lead forecasts of anywhere from 3 months to a year in advance. Here, "skillful" refers to showing improvement over a baseline model (see Section 5.1 for a formal definition). Successful long-lead forecasts are usually tied to the longer-time scale dynamics of the ocean, and the long-distance atmospheric teleconnections induced by anomalous heating or cooling at the ocean surface. In the context of the Pacific Ocean, this is most represented by the El Niño—Southern Oscillation (ENSO) phenomenon, a quasi-periodic variation in anomalously warmer than normal ocean states in the central and eastern tropical Pacific Ocean

(El Niño) and colder than normal ocean states in the central tropical Pacific (La Niña). As has been known for quite some time, ENSO can serve as an effective predictor of atmospheric-derived conditions in North America and Oceania because of the associated teleconnection-based changes in the atmospheric circulation (e.g., Philander, 1990). Operationally, it has been demonstrated for nearly three decades that skillful long-lead forecasts based on statistical models that incorporate this relationship are as good as, and typically better than, deterministic models (e.g., Barnston et al., 1999; van Oldenborgh et al., 2005). Although linear models can be skillful in these settings (e.g., Penland & Magorian, 1993), nonlinear statistical methods often perform better than deterministic forecast models at least for some spatial regions and lead times (e.g., see the overview in McDermott and Wikle (2019) for references).

Here we consider long-lead soil moisture (SM) forecasts with lead times of 3 months, with a particular emphasis on predictions in the US corn belt in May given observations up to the previous February. We focus on SM forecasts because it is well-known that the amount of SM available to corn (and other crops) at certain phases of their phenology can significantly affect yield, and thus is of major interest to producers (e.g., Carleton et al., 2008). In addition, McDermott and Wikle (2016, 2019) showed that nonlinear spatio-temporal analog forecasts and deep echo state networks can yield skillful long-lead predictions for May across some regions of the corn belt (also, note that the US Climate Prediction Center produces seasonal outlooks (forecasts) for soil moisture[1]).

We use monthly SSTA in the Pacific Ocean region from 124E to 70W (every 2°) longitude and 30S to 60N (every 2°) latitude for the period January 1948–December 2021, where the anomalies are location-specific, based on deviations from the monthly climatology for the period 1971–2000. The data were obtained from the IRI/LDEO Climate Data Library[2]; specifically, the "NOAA NCDC ERSST version5: Extended Reconstructed SST" data as described in Huang et al. (2017)[3]. For SM, we extract monthly North America data but focus on the US corn belt region, here defined from 101.5W to 80.5W (every 0.5°) longitude and 35.5N to 48.5N (every 0.5°) latitude) for the period January 1948–December 2021. These data were also obtained from the global monthly high resolution soil moisture dataset originally produced by the Climate Prediction Center as described in Fan and Van Den Dool (2004)[4].

Summary statistics for the data are given in Table 1. As described in the Supporting Information section, all of the data used in the analyses presented below can be found on Zenodo (Simmons & Burr, 2022). The analysis code can be found on GitHub (Boone et al., 2022). This code provides details regarding normalization/standardization, preprocessing, input and output of the various models, and model architecture.

4 | MODEL-BUILDING METHODS

In this section, we describe briefly several models and model-building methods that we use to illustrate the explainability methods described above. We also describe some of the implementation decisions for these methods, but leave the details to the implementations in Section 5. First, we describe a spatial functional linear regression model to serve as a baseline, followed by the machine learning models that we will use to demonstrate the utility of the explainability approaches. There are a very large number of possible candidate models, so this set of models is a non-comprehensive convenience choice rather than an exhaustive coverage of all possible application areas, combinations of models, and explainability methods.

4.1 | Spatial functional linear regression

As a baseline, we consider a linear statistical model as a candidate that is rich enough to capture the main features of the data while retaining the simple interpretability of linear models. A naive linear model for predicting SM using the 3-month lagged SSTA data is given by

$$Y_t = \alpha + \sum_{s=1}^{S} \beta(s) X_{t-3}(s) + \epsilon_t, \tag{4}$$

[1] https://www.cpc.ncep.noaa.gov/soilmst/forecasts.shtml
[2] https://iridl.ldeo.columbia.edu/
[3] http://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCDC/.ERSST/.version5/.anom/
[4] https://iridl.ldeo.columbia.edu/SOURCES/.NOAA/.NCEP/.CPC/.GMSM/

TABLE 1 Summary statistics for the data sets used to demonstrate the explainable AI methods in this article: Sea Surface Temperature anomalies (SSTA) in the Pacific Ocean, and soil moisture (SM) in the central continental United States.

Summary statistic	Sea surface temperature Anomalies (SSTA) (°C)	Soil moisture (SM) (mm)
Total number of locations	3186	1125
Region	Pacific Ocean region	US corn belt
Latitude range	30S to 60N	35.5N to 48.5N
Longitude range	124E to 70W	101.5W to 80.5W
Resolution of data grid	2 deg × 2 deg	0.5 deg × 0.5 deg
Year range	1948–2021	1948–2021
Training data	1948–2013	1948–2013
Test data	2013–2021	2013–2021
Mean	0.00522	325.198
Median	−0.0116	328.431
Standard deviation	1.153	158.514
Skew	19.113	0.0411
Kurtosis	893.75	−0.885
Number of data-points	3186 × 888	1125 × 888

where Y_t is the SM at a land location at time t, $X_{t-3}(s)$ is the 3-month lagged SSTA data at sea location s, α is the intercept, $\beta(s)$ is the regression coefficient corresponding to lagged SSTA data at sea location s for predicting SM at the specified land location, and ϵ_t are errors.

The naive model in Equation (4) has multiple shortcomings. Most importantly, it does not account for the spatial structure in the covariates $\{X_{t-3}(s) : s = 1, \dots, S\}$ by treating them as separate independent predictors as opposed to treating them as one predictor surface. Since SSTA is observed at $S = 3186$ locations, this leads to an over-parametrized model with $S = 3186$ unknown parameters. Since there are on the order of 800 training data-points (66 years of monthly data in the training period) for each land location and $S = 3186$ parameters, the linear model becomes unidentifiable and requires some dimension reduction or regularization.

Machine learning methods like CNNs (see Section 4.4) explicitly accommodate this dimension reduction by accounting for the spatial nature of the covariates. That is, CNNs achieve this by treating $\{X_{t-3}(s) : s = 1, \dots, S\}$ as an image and subsequently uses block convolution with shared weights and pooling to reduce dimensionality of the predictor and parameter space. The key observation for such dimension reduction in CNNs is that the functional relationship connecting the SSTA $X_{t-3}(s)$ to SM Y_t should be similar for nearby locations s because SSTA exhibits spatial dependence. Thus, we consider a functional regression model that uses the same principles within the linear setting by assuming that the $\beta(s)$ for nearby locations s should be similar. This ensures dimensional reduction while retaining the interpretability of the linear model. Specifically, we treat $\{X_{t-3}(s) : s = 1, \dots, S\}$ as a single spatial surface (a function in two-dimensional Euclidean space) instead of as 3186 separate predictors, and consider the traditional functional linear model

$$Y_t = \alpha + \rho Y_{t-3} + \sum_{m=2}^{12} \eta_m + \int \beta(s) X_{t-3}(s) ds + \epsilon_t, \tag{5}$$

where $\beta(s)$ is now a coefficient function on the ocean locations connecting the functional SSTA data to the SM outcome. This is analogous to the transition operator in integro-difference equation (IDE) models for spatio-temporal processes (e.g., see the review in Wikle et al., 2019) and integral projection models used in ecology (e.g., Merow et al., 2014). Additionally, the 3-month lagged SM Y_{t-3} is included in (5) to account for auto-correlation, while the η_m are introduced as month-specific intercept parameters to account for periodicity.

As is common in functional regression and IDE models, dimension reduction in Equation (5) is accomplished by expanding the coefficient function $\beta(s)$ in terms of basis function coefficients as $\beta(s) = \sum_{k=1}^{K} \gamma_k \phi_k(s)$, where $\{\phi_k(\cdot) | k =$

$1, \ldots, K\}$ is a set of basis functions appropriate for the analysis. Because $K \ll S = 3186$, this achieves considerable dimension reduction as the model can now be written as a linear model in terms of the unknown γ_k's

$$Y_t = \alpha + \rho Y_{t-3} + \sum_{m=2}^{12} \eta_m + \sum_{k=1}^{K} \gamma_k \int \phi_k(s) X_{t-3}(s) ds + \epsilon_t = \alpha + \rho Y_{t-3} + \sum_{m=2}^{12} \eta_m + \sum_{k=1}^{K} \gamma_k X^*_{t-3,k} + \epsilon_t, \tag{6}$$

where for the given choice of bases $\phi_k(s)$, $X^*_{t-3,k}$ can be pre-computed as $\int \phi_k(s) X_{t-3}(s) ds \approx \frac{1}{S} \sum_{s=1}^{S} \phi_k(s) X_{t-3}(s)$. Thus, the functional linear regression in Equation (6) becomes the traditional multiple linear regression in the parameters $\alpha, \{\eta_m | m = 2, \ldots, 12\}$, and $\{\gamma_k | k = 1, \ldots, K\}$. This facilitates easy implementation of the model while accommodating the important aspects of the data (e.g., spatially structured covariates, autocorrelation, and periodicity). Interpretability is also straightforward in the linear paradigm. The effect of the lagged sea surface temperature $X_{t-3}(s)$ on the soil moisture data can be assessed by simply looking at the estimate $\widehat{\beta}(s) = \sum_{k=1}^{K} \widehat{\gamma}_k \phi_k(s)$. This is demonstrated in Section 5.1.

While the functional linear model offers direct interpretability on the importance of SSTA at each ocean location s via the coefficient surface $\widehat{\beta}(s)$, we can also compute more model-free measures of feature importance. We calculated the model reliance (MR) of the functional linear model using Equation (1) in Section 2.1 as follows. Let z_t denote the set of all covariates used to predict SM Y_t at hold-out time t. For the linear model, z_t includes an autoregressive term, month indicators and 3-month lagged SSTA. Let f denote the fitted linear model that takes the covariates z_t and predicts Y_t. Let $Z_{\text{out}} = (z'_1, z'_2, \ldots, z'_{T_{\text{out}}})'$ denote the full matrix of features for the hold-out dataset and $Y_{\text{out}} = (Y_1, \ldots, Y_{T_{\text{out}}})'$ denote the hold-out SM data where $1, \ldots, T_{\text{out}}$ denotes the time-points in the hold-out data. The denominator e_{orig} in MR is calculated as

$$e_{\text{orig}} = \frac{1}{T_{\text{out}}} \sum_{t=1}^{T_{\text{out}}} (Y_t - f(z_t))^2 = \frac{1}{T_{\text{out}}} \| Y_{\text{out}} - f(Z_{\text{out}}) \|_2^2.$$

This yields e_{orig} for one land location in the corn-belt. Repeating this process on the SM data for each of the land locations in the corn-belt and averaging yields the overall e_{orig}. This is simply the MSPE over the entire hold-out time-period and over all the locations in the corn-belt.

To calculate e_{switch} we use eq. (3.3) of Fisher et al. (2019). Due to computational demands, instead of calculating MR for each SSTA location s, we calculate MR for sets of SSTA locations that are in the same cluster. For a cluster c, let $z_{c,t}$ denote the subset of z_t that corresponds to the SSTA locations s in cluster c and $z_{-c,t}$ denote the rest of the covariates for that time. Without loss of generality, we write $z'_t = (z'_{c,t}, z'_{-c,t})$. For hold-out time t_0 and cluster c, we then create a shuffled covariate set $z'_{\text{switch},t}(c, t_0) = (z'_{c,t_0}, z'_{-c,t})$. Stacking $z'_{\text{switch},t}(c, t_0)$ for $t = 1, 2, \ldots, t_0 - 1, t_0 + 1, \ldots, T_{\text{out}}$ we have our shuffled covariate matrix $Z_{\text{switch}}(c, t_0)$ (note that $t = t_0$ is excluded). Now, e_{switch} for cluster c and time point t_0 can be calculated as

$$e_{\text{switch}}(c, t_0) = \frac{1}{T_{\text{out}} - 1} \sum_{t \in \{1, \ldots, T_{\text{out}}\}, t \neq t_0} (Y_t - f(z_{\text{switch},t}(c, t_0)))^2 = \frac{1}{T_{\text{out}} - 1} \left\| Y_{\text{out},-t_0} - f(Z_{\text{switch}}(c, t_0)) \right\|_2^2,$$

where $Y_{\text{out},-t_0}$ denotes the vector Y_{out} with the data point for time t_0 removed. We repeat and average over all hold-out time points to obtain

$$e_{\text{switch}}(c) = \frac{1}{T_{\text{out}}} \sum_{t_0=1}^{T_{\text{out}}} e_{\text{switch}}(c, t_0) = \frac{1}{T_{\text{out}}} \frac{1}{(T_{\text{out}} - 1)} \sum_{t_0=1}^{T_{\text{out}}} \left\| Y_{\text{out},-t_0} - f(Z_{\text{switch}}(c, t_0)) \right\|_2^2.$$

Finally, we repeat this over all the land locations in the corn-belt and average to get the overall $e_{\text{switch}}(c)$. The MR for cluster c is then calculated as the ratio of $e_{\text{switch}}(c)$ and e_{orig}.

4.2 | Extreme gradient boosting

We consider machine learning boosting trees to demonstrate the Shapely value explainability method in Section 5.2. Boosted trees, originally developed by Freund and Schapire (1996) and expanded upon by Friedman (2001), are

TABLE 2 Some of the important parameters that are controllable by the analyst in the XGBoost algorithm implementation in Python and R.

Parameter	Description
`booster`	The booster defines the overall structure of the model. The `gbtree` and `dart` boosters are tree-based models and the `gblinear` booster uses linear functions; the default is `gbtree`.
`Eta` or `learning_rate`	This specifies the rate at which the boosting model is allowed to learn (i.e., it is the shrinkage imposed on tree's nodes as they are added to the model). The value for the learning rate is between 0 and 1, where 1 does not impose any penalty. Usually, the learning rate is less than 0.3; the default is 0.3.
`max_depth`	This specifies the maximum depth each tree in the ensemble is allowed to grow. Allowing trees to go too deep will overfit the data. This parameter is usually between 4 and 6, with the default at 6.
`nrounds`	This specifies the number of trees in the ensemble of trees.
`objective`	There are various objective or loss functions that can be specified; the default is squared error loss.
`Gamma`	This parameter prevents overfitting by setting a minimum loss reduction threshold that must be met in order to further partition a leaf node. The larger Gamma is, the more conservative the algorithm is. Gamma ranges from 0 to infinity; default is 0.
`Lambda`	The L_2 regularization term on weights. Larger values will make algorithm more conservative; default is 1.
`min_child_weight`	This is the minimum sum allowed in order for a tree to be further partitioned. The algorithm will not allow a partition to occur if the sum of the instance weights are lower than this threshold; default is 1.
`Sampling`	The XGBoost algorithm allows various ways to sample from the data to train the model. For example, `subsample` allows one to sample observations in the creation of the trees, while `colsample_bynode` is one of the options that samples features.

ubiquitous in modern data analysis. The idea is to create a committee of weak learners in the form of shallow trees to learn the signal in the data. The trees are sequentially created such that each new tree created is based on residuals from previous learners. Each new tree builds upon the information from the previous trees and slowly improves the prediction. The success of the algorithm is dependent upon identifying features that appropriately capture the information in the data and the hyperparameters defining the algorithm. One of the most popular implementations of gradient boosted trees was developed by Chen and Guestrin (2016) in their eXtreme Gradient Boosting (XGBoost) algorithm. This algorithm has been shown to be fast and efficient with good predictability (for example, XGBoost was used by most of the top 10 teams in the 2015 KDDCup competition) and has been implemented in several environmental studies (e.g., Liu et al., 2022; Ma et al., 2021; Zhang et al., 2020). Although there are many hyperparameters that can be tuned (see Xgboost Developers, 2021, for a complete list), we list a few of the most impactful ones in Table 2.

4.3 | Artificial neural networks

Artificial neural networks (ANNs) date back to the development of the perceptron (Rosenblatt, 1958) and multiple-layer networks to Ivakhnenko and Lapa (1967), although their use stagnated until computer architectures developed sufficiently to allow them to be fit in practice. Since the availability of GPU processing (circa 2009), ANNs have been increasingly used for predicting or classifying complex datasets such as images, text and other phenomena (e.g., Zou et al., 2008). These models are often what one thinks of when hearing the phrase "artificial intelligence" as they are associated with deep neural networks, deep learning and so forth. ANNs are nonlinear mathematical models inspired by the structure of the human brain with a network of layered hidden units (neurons) that transform information at each layer and pass it onto the next layer (e.g., see the statistics-friendly review in Fan et al., 2021). This structure allows for data with complex relationships to be fit and predicted. The biggest criticism of these models is that the parameters (weights) that govern the model are not readily interpretable given there are so many of them and they are not uniquely identifiable.

ANNs have been used extensively in environmental research. For example, Rahimikhoob (2010) considered estimating solar radiation in semi-arid environments, Amaratunga et al. (2020) and Guo et al. (2021) used ANNs to predict

rice paddy production based on climate data, Dogan et al. (2008) used ANNs to predict groundwater level using climate data, and many authors have used ANNs for air pollution modeling and monitoring (e.g., Alimissis et al., 2018; Araujo et al., 2020; Cabaneros et al., 2019; Pawul & liwka, 2016).

To specify an ANN for the examples presented here we use the following notation:

$$\text{ANN}\left(\text{Input Size, Nodes Layer } 1_{af_1}, \text{Nodes Layer } 2_{af_2}, \ \ldots\ , \text{Nodes Layer } \ k_{af_k}, \text{Output Size}\right),$$

where af_i corresponds to the activation (transfer) functions employed in all nodes for layer i. There are many choices for activation functions (e.g., Abadi et al., 2015; Zou et al., 2008). The models used in this article use hyperbolic tangent (`tanh`) and linear (`lin`) activation functions. All ANNs were fit using the `Keras` package (Chollet, 2015) for `Tensorflow` (version 2.4.0, Abadi et al. (2015)) with a mean square error objective function and the `Adam`, optimizer using Python 3.8.8. For smaller models, 1000 epochs[5] were used for training, and for larger models, 2000 epochs were used. We demonstrate the feature perturbation explainability approach (Section 2.1.1) with ANNs in Section 5.3. Training details and application-specific details are given there.

4.4 | Convolutional neural networks

Convolutional neural networks (CNNs) are regularized ANNs, originally designed to analyze visual imagery or spatial data. Their first appearance in the literature can be dated back to 1980s and early 1990s (Fukushima & Miyake, 1982; Waibel et al., 1989; Zhang et al., 1988; Zhang et al., 1990). These earlier forms of CNNs were largely inspired by the physiological work of Hubel and Wiesel (1968). CNNs experienced a similar stagnation as ANNs due to computational challenges with their implementation, including the large computational complexity and vanishing gradients (Goodfellow et al., 2016). These problems have been solved by the use of ReLU activation functions (Krizhevsky et al., 2017) and availability of advanced GPUs (Steinkraus et al., 2005). Since the early 2010s, CNNs have been used for a wide range of image classification (e.g., Ciregan et al., 2012) and video analysis (e.g., Ji et al., 2012) problems. CNNs are frequently used in environmental research because of their ability to extract informative features from spatial data, which are common in environmental sciences (e.g., Anderson & Radio, 2022; Kattenborn et al., 2021; Pan et al., 2019; Wang et al., 2021). In addition, CNNs have been employed to predict ground water potential maps from ground water conditioning factors by Panahi et al. (2020), and in real-time prediction of particulate matter in the atmosphere (Chae et al., 2021).

A typical CNN consists of two parts: (i) a feature extraction part composed of alternating convolutional and pooling layers; and (ii) a nonlinear regression part with the standard ANN layers (see the overview in Fan et al., 2021). A convolutional layer takes images or spatial data as input and generates multiple feature maps as output through convolution using filters (weights). Unlike classical approaches, the filters used for convolution are estimated based on training data, rather than pre-specified as one of the commonly used basis functions such as the radial basis functions. The generated feature maps are then supplied to pooling layers, which conduct downsampling by partitioning the feature map into sub-regions and finding a summary value (such as the average or the maximum value) within each partition. At the end of the final feature extraction part, all the feature maps are collected and vectorized into one long vector and then supplied to the final ANN layers.

In this article, a CNN model was trained with scaled input and output data for the SM forecast problem. LIME (Section 2.2) and Shapley values (Section 2.3) were used to evaluate the impact of each input location of SSTA on the SM output locations with respect to the trained model. As described above, this problem considers a 3 month lead-time prediction. SSTA data are mapped onto a 46×84 grid, with zero padding for locations that do not have associated data (e.g., land locations for SSTA data). The data are then flattened (vectorized) before being input into the model and reshaped back internally in the model. This is to aid the LIME and Shapley value calculations. All hidden layers in the model use a `tanh` for the activation function. A `sigmoid` is used as the activation function for the output layer as the output (soil moisture) values are normalized between 0 and 1. A dropout layer is included before each dense layer (used only during training). The full network structure is given in Table 3. The model is compiled with mean square error as the loss function and with the `Adam` optimizer (Kingma & Ba, 2015). We also consider a CNN model with both SSTA and lagged SM as inputs. The structure of this model is shown in Figure 1.

[5]An *epoch* is a training iteration in `TensorFlow` language.

TABLE 3 Network structure corresponding to type of input layer, shape of the output from each layer, and the number of parameters associated with each layer of the CNN model which takes SSTA as input.

Layer (type)	Output shape	Parameters
reshape (Reshape)	(None, 46, 84)	0
lambda (Lambda)	(None, 46, 84, 1)	0
conv2d (Conv2D)	(None, 46, 84, 64)	320
average_pooling2d (AveragePooling2D)	(None, 23, 42, 64)	0
conv2d_1 (Conv2D)	(None, 23, 42, 16)	4112
average_pooling2d_1 (AveragePooling2D)	(None, 11, 21, 16)	0
flatten (Flatten)	(None, 3696)	0
dropout (Dropout)	(None, 3696)	0
dense (Dense)	(None, 2250)	8,318,250
dropout_1 (Dropout)	(None, 2250)	0
dense_1 (Dense)	(None, 1125)	2,532,375

4.5 | Reference approaches and model comparison metrics

In addition to the statistical and machine learning approaches described above, we also consider two simple approaches for prediction that serve as baselines for evaluating the performances of the more sophisticated algorithms. As the most basic prediction approach, we consider the SM prediction at a location simply to be the SM value at that location 3 months before. This is known as a "persistence" forecast and is a common reference forecast in climate forecasting (e.g., Wilks, 2011). The second reference method we consider is simply predicting the SM at each location by the monthly average of SM for that location based on the entire training data. This type of forecast is referred to as "climatology".

We compare all the models based on the hold-out performance of the model forecasts for two test sets—the entire hold-out data, and more specifically, for the month of May in all of the years of the holdout period as described in Section 3. We present both mean square prediction error (MSPE) and test R^2 (i.e., the R^2 when the predicted SM is regressed on the true SM). Additionally, we also look at the *skill score* that compares a predictive metric for a model of interest to that from a reference model (see Wikle et al., 2019, for a general formulation). The skill score compares the MSPE between the reference model (MSPE_{ref}) and the statistical or machine learning model M of interest (MSPE_{M}) as $\text{Skill Score} = (\text{MSPE}_{\text{ref}} - \text{MSPE}_{\text{M}})/\text{MSPE}_{\text{ref}}$. We will use the persistence predictions as the reference for the skill score to assess the predictive performance of the other statistical and machine learning models applied in this article. Thus, models that include SSTA compared to persistence will help us understand the incremental contribution of SSTA to SM predictions relative to the lagged SM reference forecast. In this case, skill scores above zero show the model of interest has forecast skill relative to the reference, and skill scores less than zero indicate that forecasts from the model of interest have less skill than the reference.

5 | RESULTS

The following sections show the results of applying the models and explainability methods discussed above to the problem of long-lead forecasting SM in the US corn belt given 3-month lagged SSTA in the Pacific Ocean. Table 4 summarizes the input, output, and explainability methods used for each example. Reproducible code for all examples can be found on GitHub (Boone et al., 2022).

As discussed in Section 3, we consider monthly predictions of SM data from the US corn belt between 35.5N and 48.5N latitude and between 101.5W and 80.5W longitude. Figure 2 (left) shows the 1125 distinct spatial locations with SM data available within this region. Data from 1948 to 2013 are used for training while data from 2014 to 2021 are used to assess model performance. Due to the use of 3-month lagged predictor variables in the models, the first 3 months of data from 1948 are not used.

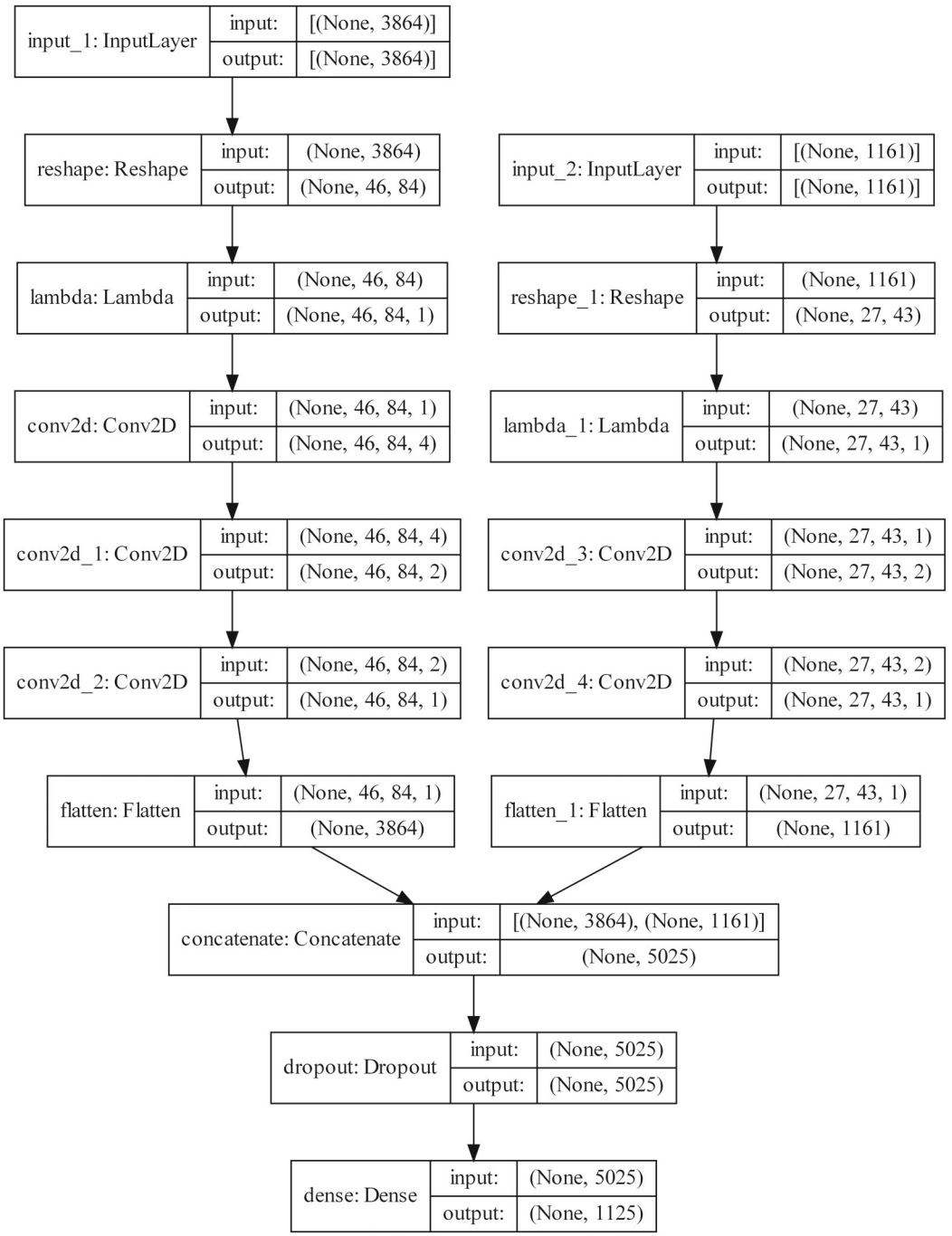

FIGURE 1 Network structure of the CNN model which takes SSTA and SM as inputs.

TABLE 4 Metadata for the spatial functional linear model (FLM), XGBoost, ANN, and CNN models: $t = 1, 2, \ldots, 12$ corresponds to the index for the month for which the values are quoted; SHAP stands for Shapley value and LIME for the LIME value.

Model	Input	Output	Explainability approach
Spatial FLM	$SSTA_{t-3}$, SM_{t-3}	SM_t	$\widehat{\beta}_t(s) = \sum_k \gamma_k(t)\phi_k(s,t), \widehat{MR}_t(c) = \hat{e}_{\text{switch}}(f)/\hat{e}_{\text{orig}}(f)$
XGBoost	$SSTA_{t-3}$, SM_{t-3}	SM_t	$SHAP_t(PCA_{75})$
ANN	$SSTA_{t-3}$	SM_t	$\overline{MSE}_t(s) = \langle SE_{pert}/SE_{unpert}\rangle_l$
CNN	$SSTA_{t-3}$	SM_t	$\langle SHAP_t\rangle_l(s), \langle LIME_t\rangle_l(s)$
CNN	$SSTA_{t-3}$, SM_{t-3}	SM_t	$\langle SHAP_t\rangle_l(s), \langle SHAP_t\rangle_l(l)$

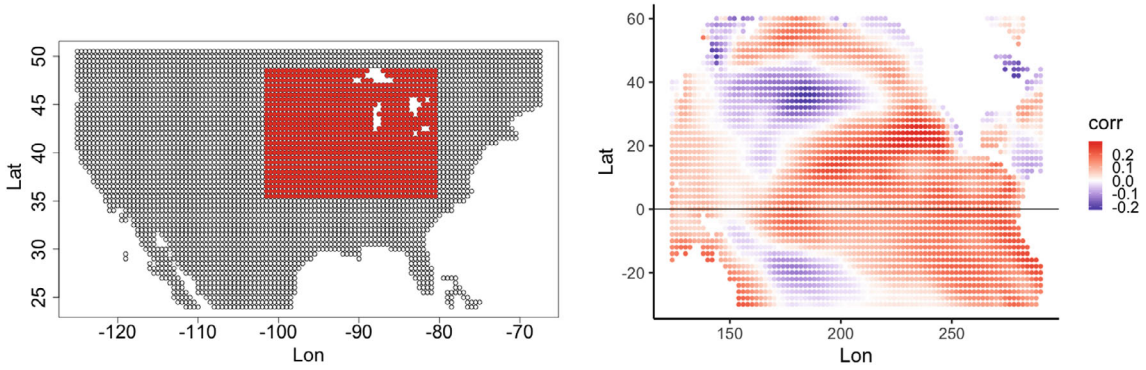

FIGURE 2 (Left) Locations (red) for the soil moisture (SM) data in the US corn belt. The white regions in the red box correspond to the Great Lakes of North America (note, the other white regions correspond to ocean locations). (Right) Average marginal correlation of 3-month lagged sea surface temperature anomalies (SSTA) at each ocean location with all of the SM data in the corn belt. The map runs from Australia in the southwest to Greenland in the northeast.

TABLE 5 Performance metrics (MSPE, test R^2, and skill score relative to the persistence forecast; see text) for the climatology, spatial functional linear model (FLM), XGBoost, ANN (Raw), ANN (PCA_{60}), ANN (Wide Raw), and ANN (Wide PCA_{60}), CNN (SSTA), CNN (SSTA, SM) forecasts of SM in the US corn belt for all months in test period (2014–2021) and for just the May months in the test period.

	All months			May months		
	MSPE	**Test R^2**	**Skill score**	**MSPE**	**Test R^2**	**Skill score**
Persistence	3979	0.65	–	2562	0.83	–
Climatology	5116	0.54	−29%	4945	0.63	−93%
Spatial FLM	2252	0.79	43%	1880	0.86	27%
XGBoost	2642	0.74	34%	2519	0.82	2%
ANN (Raw)	5695	0.53	−43%	4995	0.66	−94%
ANN (PCA_{60})	5779	0.53	−45%	4690	0.69	−83%
ANN (Wide Raw)	6535	0.47	−64%	5856	0.34	−128%
ANN (Wide PCA_{60})	6299	0.45	−58%	5163	0.32	−101%
CNN (SSTA)	4862	0.55	−22%	4323	0.68	−69%
CNN (SSTA, SM)	3824	0.63	4%	3352	0.74	−31%

Note: See Section 4 for additional details about each forecast model.

For the predictors we consider 3-month lagged SSTA. Figure 2 (right) shows the map of average marginal correlation of the 3-month lagged SSTA at each location with all of the SM data in the corn belt. We observe some regional patches of positive and negative correlations with the SM data, with highest positive correlations in the central Pacific and Northern Pacific, likely tied to the ENSO and Pacific decadal oscillations, respectively.

We first present the results of the persistence model as the forecast baseline in Table 5. We see that persistence generally offers considerably improved prediction when only restricted to May. This is not surprising given that there are years where there is little change in SM over a three-month span in the northern hemisphere spring. We also present the performance of the simple climatology forecasts in Table 5. We see that for both test sets the climatology forecasts have considerable negative skill scores with respect to persistence suggesting that seasonality alone is not sufficient in explaining the variation in the SM data.

5.1 | Spatial functional linear model

As discussed in Section 4.1, a functional linear model makes a nice comparison model for the black-box machine learning models considered here because linear models are inherently interpretable, while still being more

complex than simple statistical models. The model was implemented using PCA (empirical orthogonal function, EOF) basis functions (see Wikle et al., 2019, for an overview of spatial principal component/EOF analysis). The first 21 PC time series were used as inputs since they explain 80% of the variation in the SST data. From the 22nd PC onwards, every PC contributes less than 1% of the total variance explained. If considering a cutoff much higher than 80% one would need to include many more PCs. For example, 95% variance is explained by 47 PCs, leading to 60 total parameters. With only a training sample size of 790 we decided against such over parameterization. Note that other unsupervised dimension reduction procedures could be considered here as well. Indeed, machine learning approaches have been demonstrated to be superior to PCA in some environmental applications (e.g., Raza et al., 2019; Zhong et al., 2021). We focus on PCA because of their ubiquitous use in previous SSTA forecasting applications.

Figure 3 presents the scatterplot of true SM values and predictions from the functional linear model (Equation (5), Section 4.1) for the two test sets. We note that the predictions are generally better for May than for the test set spanning all months. There is no noticeable trend in the fits as a function of the year.

Table 5 presents the summary metrics for the predictions. We see a 43% skill score (improvement in MSPE over that of persistence) for the functional linear model for the test set spanning all months, and the skill score for test set that only includes data from May months is 27%. The results suggest consistent improvement in the model fits from the functional linear model compared to the persistence predictions. Both the functional linear model and persistence fits are better for the months of May than when considering all months in the test set. Consequently the improvement of the functional linear model over persistence is relatively less for May as persistence predictions for May already offer a high test R^2 of 83% (Table 5).

Figure 4 shows maps of the MSPE from the functional linear model for each SM data location in the US corn belt. We note that when considering average MSPE over all months (left figure) the lowest MSPEs (best forecasts) are in the areas south and east of the Great Lakes and in the north west portion of the domain. When predicting only for May (right) we

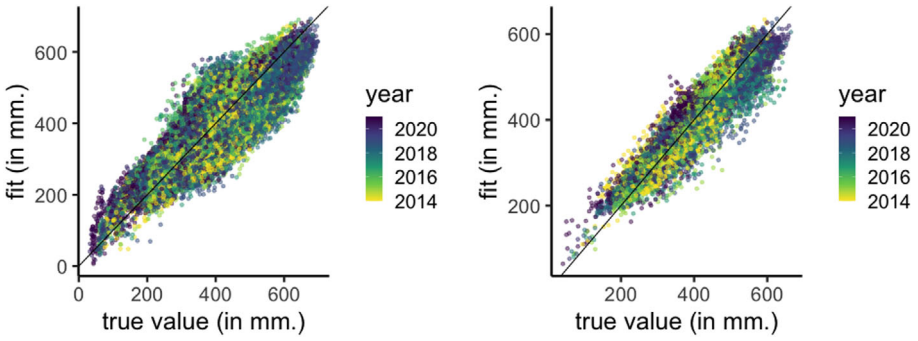

FIGURE 3 Scatterplot of true SM values and predictions from the functional linear model for the two test sets—all months (left) and May months (right).

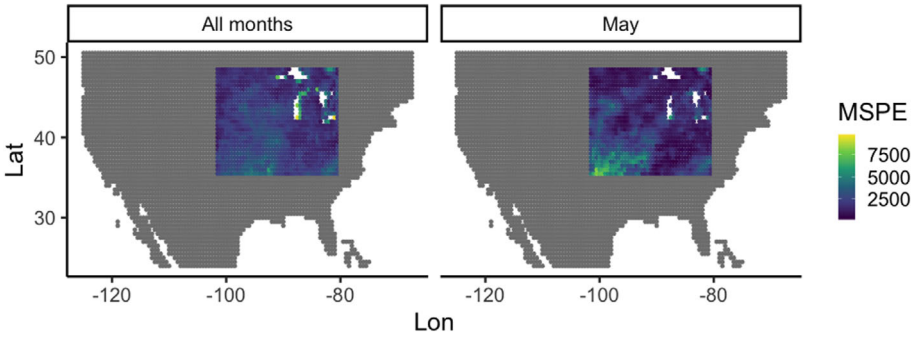

FIGURE 4 Map of MSPE for SM data from the functional linear model at each location in the US corn belt for all months (left) and only May months (right). Note the agreement (low error) is quite good in large swaths of belt for all months, and similarly good in all but the southwest region for the May months.

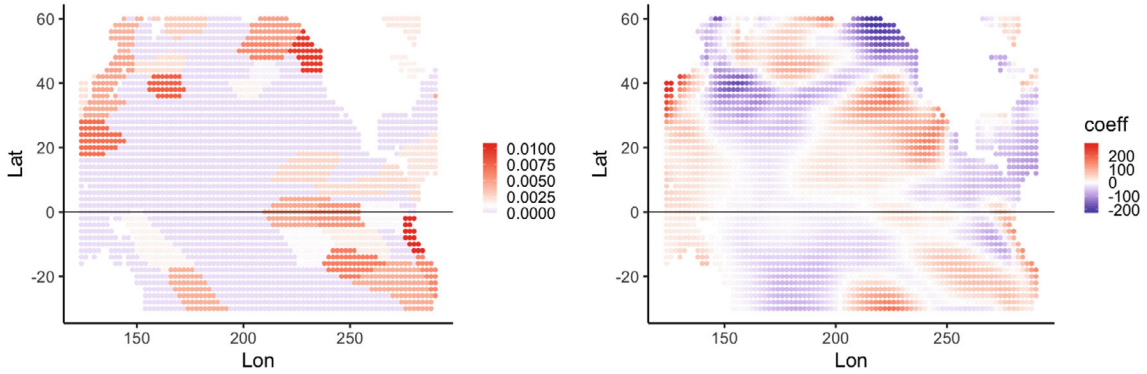

FIGURE 5 Variable importance maps for the spatial functional linear model: (Left) Map of the model reliance metric in log-scale for predicting SM in the months of May (Section 2.1); (Right) Map of the functional regression coefficient $\widehat{\beta}(s)$ for the functional linear model at a sea surface location s, averaged for all SM locations in the corn belt.

see again that the best forecasts are in the southeast and northwest portions of the domain, but note that MSPE is much higher for the southwest corner of the corn belt than when considering all months.

The purpose of this analysis is to examine the contribution of lagged SSTA at a particular ocean location s on the SM predictions from the functional linear model. Figure 5 shows maps of two different variable important measures for the functional linear model. The left figure gives the maps of the model reliance (MR) metric (Section 2.1) on the log-scale for each sea location. We observe that a large majority of the SSTA locations have $\log(\text{MR}) \leq 0$ (i.e., $\text{MR} \leq 1$) implying that they had little importance in the final model. Notable areas with positive log(MR) include patches off the west coast of North and South America and a patch between 30N–40N latitude and 150E–175E longitude.

While the MR metrics allow comparison of the linear model to black-box machine learning models, the linear model also offers a more direct measure of interpretability through the regression coefficient $\beta(s)$. Figure 5 (right) provides a plot of the average $\widehat{\beta}(s)$ for all the SM locations for each SSTA location s. There are some agreements with the marginal correlation map in Figure 2 (right), for example, a large patch of negative coefficients between 20N–40N latitude and 150E–200E longitude, and large patch of positive coefficients off the coast of California. There are also notable differences (much of the equatorial region and south of it). This is expected because, unlike the marginal correlations in Figure 2 (right), $\beta(s)$ is essentially the partial correlation of SSTA at an ocean location with SM after adjusting for SSTA among the nearby ocean locations. The interpretation of a partial correlation is contingent upon the set of explanatory variables used (e.g., the lagged SM covariates) and hence partial correlations for different choices of covariates may not be directly comparable without standardization. Note also that the model reliance metric and spatially-varying coefficients show some areas of agreement—particularly, the near coast of South America up through the central tropical Pacific, the northern coast of North America and the far eastern coast of Asia. Differences include a region in the southern ocean for the reliance metric that does not show up in the spatially-varying parameter plot, and an area from the central Pacific to coastal North America in the parameter plot that does not show up in the reliance metric plot. Note that an advantage of the spatially-varying parameter plot is that one can see the sign of the impact (i.e., negative and positive parameter values).

5.2 | Extreme gradient boosting

As mentioned in Section 4.2, it can be helpful to manually create features that better capture the signal in the data or provide dimension reduction or decorrelation. Thus, instead of using the raw SSTA data, we created clusters by first reducing the dimensionality through spatial PCA (EOFs) and then using a "partition around medoids" (PAM) clustering algorithm (e.g., Van der Laan et al., 2003). This identified 75 clusters for the SSTA data. Using the clustered data, the following features were created using 3-month lagged information: (1) means within the 3-month lagged clusters; (2) variances within the 3-month lagged clusters; (3) difference from last year's mean (mean of the most recent 3 month period minus the mean of the previous year's 3 month period at the same time of year); and (4) ratio of current variance to last year's variance (ratio of the most recent 3 month variance to the previous year's 3 month variance at the same time of year). This gave a total of 300 (4×75) features. In addition to these features, the monthly information was one hot encoded

(converted to dummy variables), and the 3 month lagged SM data was also used as a predictor. This brought the total number of features to 315. The response being predicted is the SM 3 months into the future. The XGBoost algorithm was trained on the hyperparameters listed in Section 4.2. The tuning produced a learning rate of 0.05 with 70% of observations sampled in the creation of each tree and 80% of the features randomly selected. The maximum depth of each tree was set to 6 with a Gamma value of 2, the number of trees was set to 400 and the minimum sum allowed for a tree to split was set to 1. The default squared error loss function was used.

Table 5 shows that the XGBoost model performed well, but did not do as well as the spatial functional linear model with regards to its skill, yet it does show a similar pattern in that there is less skill for the May predictions relative to persistence than for all months combined.

The XGBoost algorithm has several internal measures of feature importance. One commonly used measure is the gain of each feature, which is the relative contribution of that feature to the model across all trees. Contribution is measured by the amount of information gained when that variable is used to split the data in a tree. This information is then averaged across all trees. Based upon the gain for all features, the means within the 3-month lagged clusters showed the most influence across all created features and was used to compare to the Shapley values from the same model. The gain for these features can be seen in the top left panel of Figure 6. All gains in this plot are positive, with larger values (darker red) indicating more influential areas. We compare this information to the corresponding mean absolute Shapley values averaged for all periods as shown in the top right panel of Figure 6. The scaled colors vary from white to dark red with the darker red values indicating a larger affect of the SSTA data. The Shapley value shows very similar patterns, especially the very influential areas in the south central Pacific.

An advantage of the Shapley value is its localness, that is, its ability to examine the influence for a particular instance. To illustrate this, the bottom panels in Figure 6 show the Shapley values for predicting May SM in 2016 (left) and 2021

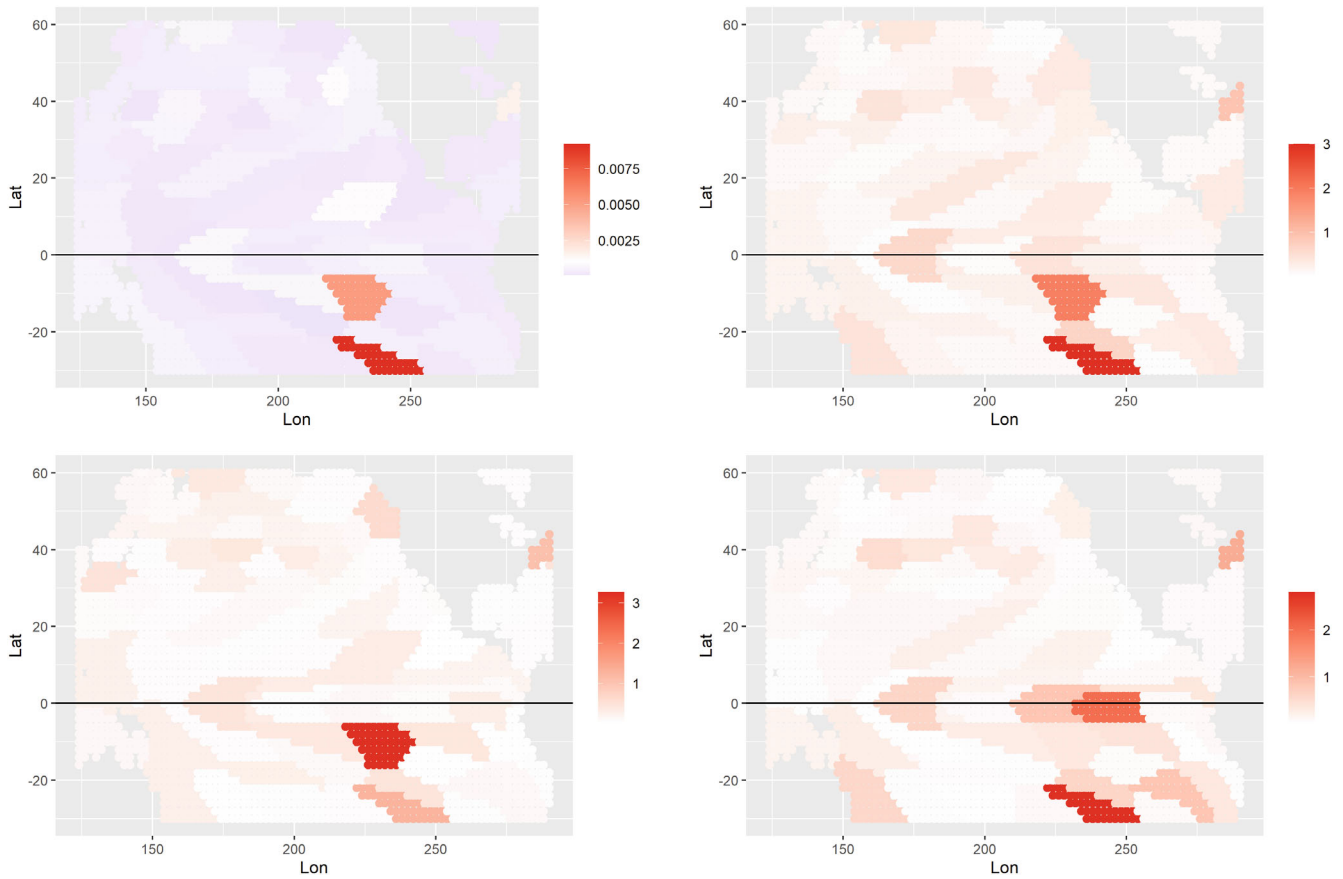

FIGURE 6 Interpretability plots for the XGBoost model for 3-month long lead forecasts of SM in the corn belt: (Top left) the overall gain for SSTA features in the XGBoost model; (Top right) mean absolute Shapley values over all months; (Bottom) absolute values of the Shapley values for the XGboost model for 2016 (left) and 2021 (right), with darker red indicating higher SSTA increasing the SM in the corn belt. These values are averaged over all locations in the corn belt. The scale bars vary with instance to facilitate interpretation.

(right). Note the similarity in the 2016 plot with the two average plots in the top panels of the figure. Given that 2016 was a strong El Niño year, this suggests that the overall influence of SSTA is driven mostly by ENSO events, as one would expect.

Thus, the Shapley plots may be helpful for climate scientists who are interested in understanding teleconnections between SSTA and SM on a year-to-year basis. Certain regions show consistent importance across the test years, including the central tropical Pacific and coastal South America regions that are typical for ENSO events. One might get additional insight by comparing these plots to the various climate indices associated with the Pacific region (note: NOAA makes available updated monthly values for a large number of climate indices[6]).

5.3 | Artificial neural networks

We consider four models here that predict standardized SM observations using Z-score transformations at each spatial location. Note that the specification of the number of layers and the number of nodes in each hidden layer is chosen by trial and error, as is the case for most deep model implementations. We did evaluate the sensitivity to different choices of the number of hidden layers and nodes per layer, and found that the predictive performance was not monotonic with increasing depth, as expected. Models for each prediction month are fit separately using 1000 epochs.

The first model is a naive approach that uses the data from all 3186 SSTA locations in February as the input, and predicts all 1224 corn belt SM locations the following May; we call this the "Raw" model and specify it by ANN (3186_{tanh}, 2000_{tanh}, 2000_{tanh}, 1224_{lin}, 1224). The second model is the same as the first except it uses 6 months of lagged SSTA as predictors (i.e., September through February) to predict SM the following May. This model is given by: ANN(19,116, $19, 116_{\text{tanh}}$, 7000_{tanh}, 5000_{tanh}, 1224_{lin}, 1224); we call this the "Wide Raw" model. The third and fourth models considered here are based on dimension reduced input features. While the naive Raw model approach is viable, it is computationally intensive due to the high dimension of the parameter space. To reduce the high dimensionality associated with ANNs and hence, the computational burden, Migenda et al. (2021) suggest employing PCA to pre-process the input data. In essence, this forms an additional layer to the ANN with a linear activation function. However, this layer is "pretrained" (i.e., the PCA weights are unsupervised) and does not get updated during the training of the ANN. Specifically, to obtain the PCA projection, the SSTA data from the beginning of the dataset to November 2012 were used and 60 components were selected (note that 60 components account for 95.7% of the variation in the SSTA data for this period). We use ANN(60, 100_{tanh}, 100_{tanh}, 500_{tanh}, 1224_{lin}, 1224) to fit the data and refer to this model as the "PCA_{60}" model. The fourth model is a wide version of this the PCA model but using PCA input features from September through February. To implement this model we use: ANN(360, 420_{tanh}, 400_{tanh}, 1224_{lin}, 1224), and call this the "Wide PCA_{60}" model. As shown in Table 5, these ANN models are among the poorest performers in terms of MSPE, test R^2 and skill score relative to persistence.

We illustrate the random perturbation explainability approach (Section 2.1.1) here using the ANN predictions. This involves perturbing the SSTA input data and examining the influence on the model predictions via MSE_{pred}. In particular, for each SSTA location the corresponding sample standard deviation for each site and month (calculated over time) is then added to the observed SSTA value for that site, leaving all other SSTA values fixed. These perturbed data are then used to predict SM and the MSPE is calculated. This is done for every SSTA site and the ratio of the perturbed to unperturbed MSPE is recorded; recall, this is the model reliance metric in feature shuffling methods (Equation (1) in Section 2.1). For the PCA models, the PCA transformation was applied to the perturbed data.

Figure 7 shows maps of the model reliance ratios for 2014 for the Raw and PCA_{60} models, and Figure 8 shows the model reliance ratios for three input months (December 2013–February 2014) for the Wide Raw and Wide PCA_{60} models when applied to forecast SM in May 2014. The model reliance results in Figure 7 show that the equatorial "warm pool" region off the west coast of South America in the Pacific Ocean appears to influence the MSPE ratio for the Raw model more than the PCA_{60} model, as evidenced by the red region. This is suggestive of an El Niño signal in SSTA, but this is interesting since February and May of 2014 do not correspond to an ENSO year. We do note, however, that the Pacific North America (PNA) statistic for those times show a strong negative influence (see the PNA statistic

[6] https://psl.noaa.gov/data/climateindices/list/

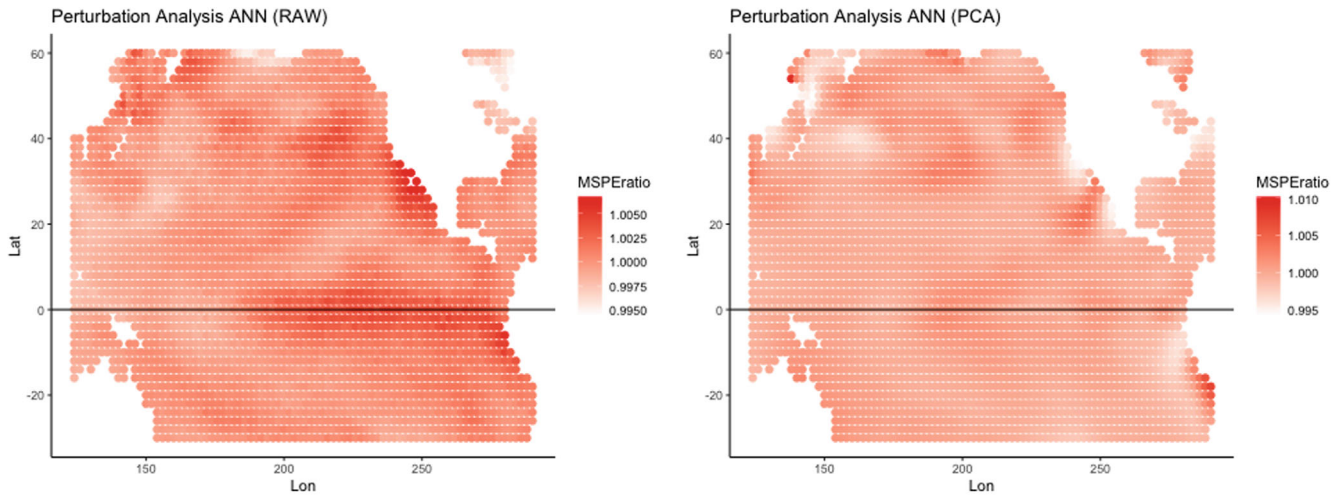

FIGURE 7 Model reliance perturbation results for the naïve approach using: (Left) Raw; and (Right) PCA_{60} transformed SSTA data with 60 components using February alone to predict May 2014. Note that the scales differ between plots to facilitate interpretation.

from the NOAA website[7]. Both models show an area of influence of the western coast of Mexico. In general, the Raw and PCA_{60} models suggest somewhat different areas of influence, and there are even larger differences between years (not shown here).

The plots for the Wide models shown in Figure 8 suggest a somewhat different story. The pattern for each lag for the Wide Raw model is indicative of an ENSO pattern in the central tropical Pacific, and includes an extension through coastal North America. However, the plots for the Wide PCA_{60} do not exhibit this structure and vary from across each lag.

5.4 | Convolutional neural networks

A CNN model was implemented to predict SM with SSTA data as input along with Shapley values and LIME as explainers. Table 5 gives the prediction summary metrics (test R^2, MSPE, and skill scores relative to persistence) for the forecasts corresponding to all months and for May months. Although the summary statistics show that the forecasts from the CNN model for this example are not as skillful as persistence (excepting CNN (SSTA, SM) for predictions for all months alone), the spatial functional linear model, and XGBoost, the CNN approach does yield useful interpretations when paired with LIME and Shapley values as described below.

Figure 9 shows heatmaps at SSTA locations for the absolute value of the LIME summaries for May averaged over the years 2014–2021 (left) and for May 2016 (right), corresponding to the mean of all SM prediction locations. The top panels of Figure 10 show the corresponding heatmap plots for the average of the absolute Shapley value summaries for SSTA over the years 2014–2021 (left) and the absolute Shapley value summaries for SSTA for May 2016 (right). The middle panels of Figure 10 show heatmaps of the absolute Shapley values for CNN model trained with both SSTA and SM as input; the left panel shows the SSTA contribution and the right panel shows the SM contribution for the year 2016. The bottom panels of Figure 10 show heatmaps of the absolute Shapley values for CNN model trained with both SSTA and SM as input; the left panel shows the average of the absolute Shapley value summaries for SSTA over the years 2014–2021 and the right panel shows the average of the absolute Shapley value summaries for SM over the years 2014–2021. The LIME results are not as clear as for the XGBoost results, but do suggest that regions off of Oceania, the central tropical Pacific, and the near coastal waters off of North and South America may be important predictors for SM. Shapley values are easier to interpret with respect to the CNN predictions given they exhibit substantially more spatial smoothness. They show many of the same regions being important as presented earlier, but notably, show a very strong indication that the ENSO region from tropical South America through the central Pacific is quite important for the prediction of SM in May 2016. Recall that ocean conditions in February 2016 suggested a strong El Niño.

[7]https://psl.noaa.gov/data/climateindices/list/

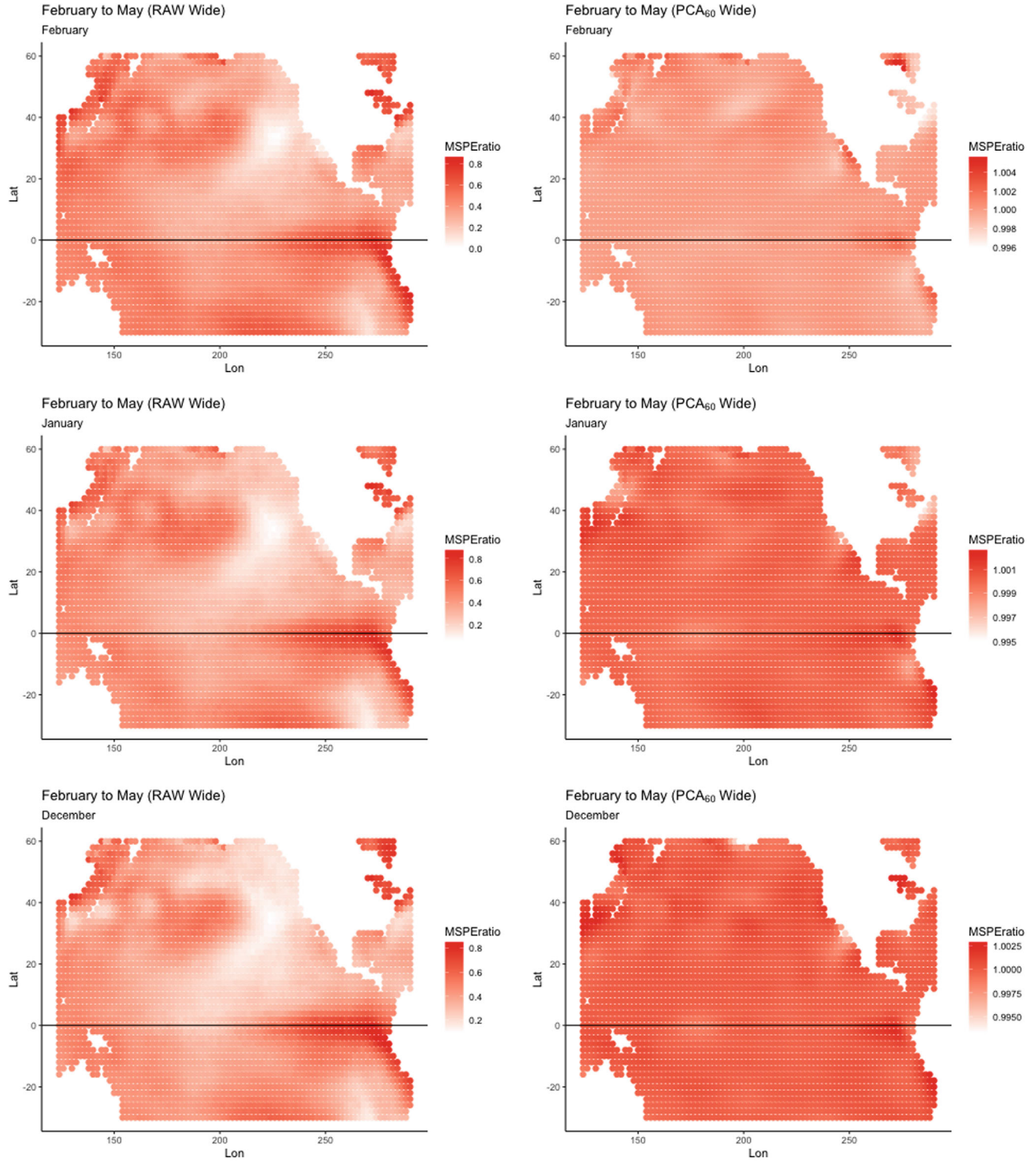

FIGURE 8 Model reliance perturbation results for the wide approach using Wide Raw (left) and Wide PCA_{60} (right) for transformed SSTA data with 60 PCA components. Here, May 2014 is predicted using September 2013 to February 2014 data (note, the plots for September, October, and November are omitted). The scales on the plots vary to facilitate interpretation.

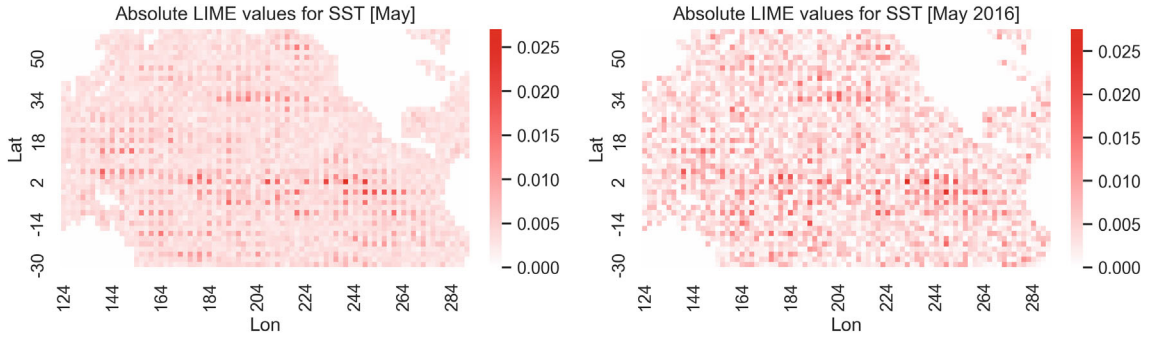

FIGURE 9 Absolute LIME values averaged across the years 2014–2021 (left) and absolute LIME values for the year 2016 (right). The CNN model used here takes SSTA data as input.

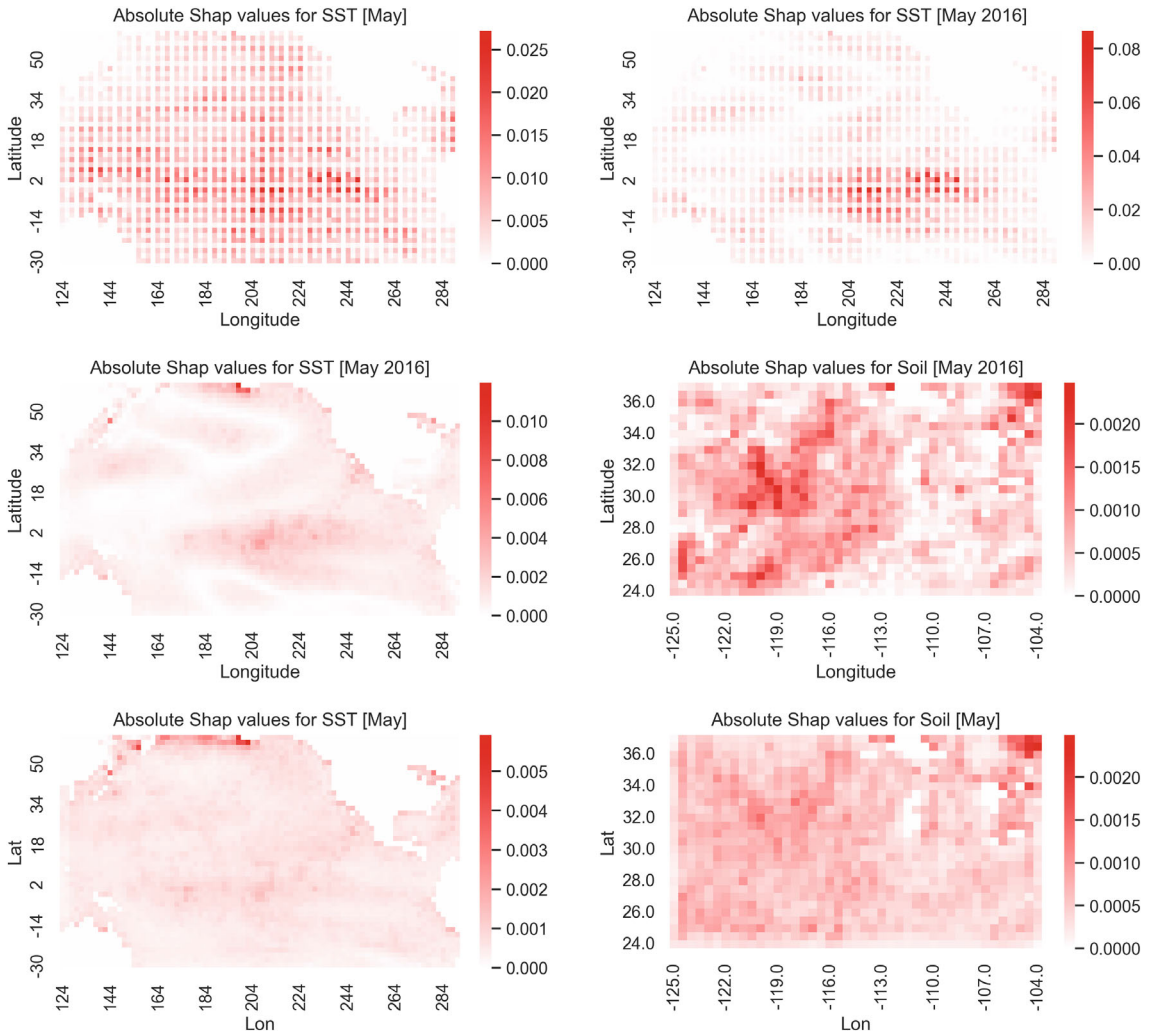

FIGURE 10 Upper panels: (Left) Absolute Shapley values averaged across the years 2014–2021; and (Right) the absolute Shapley values for the year 2016 for the CNN model for the month of May with SSTA as input. Middle panels: (Left) Absolute Shapley values, SSTA part; and (Right) SM part, for the year 2016, for the CNN model for the month of May trained with both SSTA and SM as input. Bottom panels: (Left) Absolute Shapley values, SSTA part; and (Right) SM part, averaged over the years 2014–2021, for the CNN model for the month of May trained with both SSTA and SM as input. The scales on the plots vary to facilitate interpretation.

6 | DISCUSSION

We have illustrated how one can use three model-agnostic explainability approaches on various models applied to an environmental prediction problem. In particular, we consider feature shuffling, LIME, and Shapley values as explainability approaches and consider spatial functional regression, XGboost, ANNs, and CNNs to perform prediction of SM over the US corn belt region given SSTA predictors lagged at least 3 months.

The functional linear model gave the best predictions in terms of MSPE relative to a persistence forecast, followed by the XGboost model; the CNN (SSTA, SM) model performed marginally better for predictions for all months alone. We note that all three of these models considered 3 month lagged SM in addition to the lagged SSTA data as predictors. The network neural models, excepting CNN (SSTA, SM) for predictions for all months alone, all performed substantially worse than persistence, which would imply that they would not be reasonable forecasts for operational consideration. There was considerable spatial and year-to-year variability in the forecast quality, which has also been demonstrated in previous studies (e.g., McDermott & Wikle, 2016). It is not surprising that the linear model and the XGBoost models were skillful. Linear models for long lead prediction have been used operationally for quite some time, and Figure 2 shows that there is a strong linear association between Pacific SSTA and corn belt SM. Further, McDermott and Wikle (2019) showed that for some regions of the corn belt nonlinear models performed well, and for other regions they did not. We still find it surprising that the neural models did not perform better in general in the examples presented here. We recognize that given there is a wide range of potential architectures and modeling choices for such models, it is quite possible that other neural models might provide skillful forecasts. We conjecture that the primary limitation here was the relatively small training sample—it is well known that deep neural models often require a great deal of data, or pre-training to be successful.

In terms of explainability, each model considered one or two explainability approaches for the out-of-sample predictions. Although these methods tended to find coherent regions in the SSTA features that were important for the predictions, they were not always in agreement. There are many possible reasons for this, including the presence of additional predictors that were confounding the interpretation, as well as inherent differences in the explainability procedures themselves (e.g., LIME relies on the quality of its local surrogate models). Nevertheless, the explainability metrics did provide suggestions for particular regions in the SSTA feature space that were important for individual instances (forecast years) and could be investigated by climate scientists for potential physical mechanisms (e.g., ENSO or PNA patterns).

We note that there are several challenges with the approaches we present here. First, the explainability methods all have difficulty when there is dependence in the predictors. As we illustrate, this can be mitigated to some extent by pre-clustering and/or dimension reduction (e.g., PCA/EOF). In addition, deep models such as the ANN and CNN models, require the user to select many different hyperparameters with respect to model architecture (e.g., number of hidden layers, number of units per layer, activation functions, etc.) and whether input or output features should be reduced in dimension *a priori*. This is always a challenge with such models, but it is not clear how much this affects explainability. For example, we showed with the ANN implementation that feature perturbation results differed depending on whether we first consider a PCA dimension reduction on the input features. In addition, although the CNN models were not uniformly skillful, they did capture strong ENSO SSTA patterns of importance.

We also note that the implementation of these explainability methods is very computationally expensive for models where there are many input features and multivariate responses. Furthermore, although one of the strengths of these procedures is that they can be applied to explain individual instances, it can be quite labor intensive to manually evaluate each instance and summarize overall effects.

Finally, the predictive models considered here do not explicitly account for the temporal dynamics of the underlying SSTA and SM processes. Models such as spatio-temporal dynamic models (e.g., Cressie & Wikle, 2011) or combinations of CNNs and RNNs (e.g., Khaki et al., 2020; Zhu et al., 2021) could be used for this purpose and it would be interesting future research to investigate their use here, as well as their interpretability. Furthermore, future research may include exploring situations where the explainability approaches agree and disagree and to gain an understanding of why they are performing this way (i.e., "explain the explainer"). Finally, more research may be devoted to implementing and interpreting these explainability methods more efficiently with high-dimensional input and output spaces.

ACKNOWLEDGMENTS

The authors would like to thank Lance Waller for providing extensive editorial suggestions. We also wish to thank the guest editors for the opportunity to participate in the special issue, and the reviewers for their excellent

suggestions on the initial submission. Christopher K. Wikle was partially supported by the U.S. National Science Foundation (NSF) Grant SES-1853096. Abhirup Datta was partially supported by National Science Foundation (NSF) Grant DMS-1915803 and National Institute of Environmental Health Sciences (NIEHS) Grant R01ES033739. Wesley S. Burr was partially supported by a Natural Sciences and Engineering Council (NSERC) Discovery Grant 2017-04741.

CONFLICT OF INTEREST

The authors declare no potential conflict of interests.

ORCID

Christopher K. Wikle https://orcid.org/0000-0002-0655-2696
Edward L. Boone https://orcid.org/0000-0003-0755-6899
Indranil Sahoo https://orcid.org/0000-0002-4960-7984
Indulekha Kavila https://orcid.org/0000-0001-6637-576X
Stefano Castruccio https://orcid.org/0000-0002-6728-965X
Wesley S. Burr https://orcid.org/0000-0002-2058-1899
Won Chang https://orcid.org/0000-0003-3556-1249

REFERENCES

Aas, K., Jullum, M., & Løland, A. (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence*, *298*, 103502–103525.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Zheng X, (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., & Nahavandia, S. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, *76*, 243–297.

Alimissis, A., Philippopoulos, K., Tzanis, C., & Deligiorgi, D. (2018). Spatial estimation of urban air pollution with the use of artificial neural network models. *Atmospheric Environment*, *191*, 205–213.

Amaratunga, V., Wickramasinghe, L., Perera, A., Jayasinghe, J., & Rathnayake, U. (2020). Artificial neural network to estimate the paddy yield prediction using climatic data. *Mathematical Problems in Engineering*, *2020*, 8627824. https://doi.org/10.1155/2020/8627824

Anderson, S., & Radio, V. (2022). Evaluation and interpretation of convolutional long short-term memory networks for regional hydrological modelling. *Hydrology and Earth System Sciences*, *26*(3), 795–825.

Antipov, E. A., & Pokryshevskaya, E. B. (2020). Interpretable machine learning for demand modeling with high-dimensional data using gradient boosting machines and shapley values. *Journal of Revenue and Pricing Management*, *19*(5), 355–364.

Araujo, L. N., Belotti, J. T., Alves, T. A., de Souza Tadano, Y., & Siqueira, H. (2020). Ensemble method based on artificial neural networks to estimate air pollution health risks. *Environmental Modelling & Software*, *123*, 104567.

Barnston, A. G., Glantz, M. H., & He, Y. (1999). Predictive skill of statistical and dynamical climate models in SST forecasts during the 1997–98 El Nino episode and the 1998 La Nina onset. *Bulletin of the American Meteorological Society*, *80*(2), 217–244.

Boone, E. L., Simmons, S. J., Hari, B. V., Chang, W., & Burr, W. S. (2022, March). Code for models for Wikle et al. (2022): Version 0.9.0. Zenodo. https://doi.org/10.5281/zenodo.6353636

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

Breiman, L. (2001b). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, *16*(3), 199–231.

Cabaneros, S. M., Calautit, J. K., & Hughes, B. R. (2019). A review of artificial neural network models for ambient air pollution prediction. *Environmental Modelling & Software*, *119*, 285–304.

Carleton, A. M., Arnold, D. L., Travis, D. J., Curran, S., & Adegoke, J. O. (2008). Synoptic circulation and land surface influences on convection in the Midwest US "Corn Belt" during the summers of 1999 and 2000. PartI: Composite synoptic environments. *Journal of Climate*, *21*(14), 3389–3415.

Cha, Y., Shin, J., Go, B., Lee, D.-S., Kim, Y., Kim, T., & Park, Y.-S. (2021). An interpretable machine learning method for supporting ecosystem management: Application to species distribution models of freshwater macroinvertebrates. *Journal of Environmental Management*, *291*, 112719.

Chae, S., Shin, J., Kwon, S., Lee, S., Kang, S., & Lee, D. (2021). PM10 and PM2.5 real-time prediction models using an interpolated convolutional neural network. *Scientific Reports*, *11*, 11952. https://doi.org/10.1038/s41598-021-91253-9

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

Chollet, F., (2015). *Keras*. . https://github.com/fchollet/keras

Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (pp. 3642-3649). IEEE. https://doi.org/10.1109/CVPR.2012.6248110

Cohen, S., Dror, G., & Ruppin, E. (2005, August). *Playing the game of feature selection*. Proceedings of the 19th International Joint Conference on Artificial Intelligence (pp. 1-8). https://www.researchgate.net/profile/Eytan-Ruppin/publication/228966610_Playing_the_game_of_feature_selection/links/0fcfd505c00fce8801000000/Playing-the-game-of-feature-selection.pdf.

Cook, R. D., & Weisberg, S. (1991). *Dynamic graphics and regresion diagnostics using xlisp-stat* (Technical Report). University of Minnesota.

Cressie, N., & Wikle, C. K. (2011). *Statistics for spatio-temporal data*. John Wiley & Sons.

Dogan, A., Demirpence, H., & Cobaner, M. (2008). Prediction of groundwater levels from lake levels and climate data using ANN approach. *Water SA*, *34*(2), 199–208.

Fan, J., Ma, C., & Zhong, Y. (2021). A selective overview of deep learning. *Statistical Science*, *36*(2), 264–290.

Fan, Y., & Van Den Dool, H. (2004). Climate prediction center global monthly soil moisture data set at 0.5 resolution for 1948 to present. *Journal of Geophysical Research: Atmospheres*, *109*(D10), D10102. https://doi.org/10.1029/2003JD004345.

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1–81.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. Proceedings of the 13th International Conference Machine Learning (pp. 148—156).

Friedman, J. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Fukushima, K., & Miyake, S. (1982). *Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition*. In *Competition and cooperation in neural nets* (pp. 267–285). Springer.

Gevrey, M., Dimopoulos, I., & Lek, S. (2003). Review and comparison of methods to study the contribution of variables in artificial neural network models. *Ecological Modelling*, *160*(3), 249–264.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Guo, Y., Xiang, H., Li, Z., Ma, F., & Du, C. (2021). Prediction of rice yield in East China based on climate and agronomic traits data using artificial neural networks and partial least squares regression. *Agronomy*, *11*(2), 282.

Hassan, M. D., Nasret, A. N., Baker, M. R., & Mahmood, Z. S. (2021). Enhancement automatic speech recognition by deep neural networks. *Periodicals of Engineering and Natural Sciences*, *9*(4), 921–927.

Huang, B., Thorne, P. W., Banzon, V. F., Boyer, T., Chepurin, G., Lawrimore, J. H., & Zhang, H. M. (2017). Extended reconstructed sea surface temperature, version 5 (ERSSTv5): Upgrades, validations, and intercomparisons. *Journal of Climate*, *30*(20), 8179–8205.

Huang, Y., Li, J., Shi, M., Zhuang, H., Zhu, X., Chérubin, L., Tang, Y. (2021). *ST-PCNN: Spatio-Temporal Physics-Coupled Neural Networks for Dynamics Forecasting*. arXiv preprint arXiv:2108.05940.

Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, *195*(1), 215–243.

Hüttner, F., & Sunder, M. (2011). Decomposing R2 with the owenvalue. Working paper; 100.

Ibrahim, L., Mesinovic, M., Yang, K.-W., & Eid, M. A. (2020). Explainable prediction of acute myocardial infarction using machine learning and Shapley values. *IEEE Access*, *8*, 210410–210417.

Ivakhnenko, A. G., & Lapa, V. G. (1967). *Cybernetics and forecasting techniques* (Vol. *8*). American Elsevier Publishing Company.

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(1), 221–231.

Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on convolutional neural networks (CNN) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *173*, 24–49.

Khaki, S., Wang, L., & Archontoulis, S. V. (2020). A CNN-RNN framework for crop yield prediction. *Frontiers in Plant Science*, *10*, 1750. https://doi.org/10.3389/fpls.2019.01750

Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization*. In Y. Bengio (Ed.), *Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* http://arxiv.org/abs/1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

Lipovetsky, S., & Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, *17*(4), 319–330.

Liu, X., Liu, T., & Feng, P. (2022). Long-term performance prediction framework based on XGBoost decision tree for pultruded FRP composites exposed to water, humidity and alkaline solution. *Composite Structures*, *284*, 115184. https://doi.org/10.1016/j.compstruct.2022.115184

Lundberg, S. M., & Lee, S.-I. (2017). *A unified approach to interpreting model predictions*. In I. Guyon (Ed.), *Advances in neural information processing systems* (Vol. *30*, pp. 4765–4774). Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Ma, M., Zhao, G., He, B., Li, Q., Dong, H., Wang, S., & Wang, Z. (2021). XGBoost-based method for flash flood risk assessment. *Journal of Hydrology*, *598*, 126382. https://doi.org/10.1016/j.jhydrol.2021.126382

Maksymiuk, S., Gosiewska, A., & Biecek, P. (2020). shapper: wrapper of python library 'shap' [Computer software manual]. R package version 0.1.3. https://CRAN.R-project.org/package=shapper.

McDermott, P. L., & Wikle, C. K. (2016). A model-based approach for analog spatio-temporal dynamic forecasting. *Environmetrics*, *27*(2), 70–82.

McDermott, P. L., & Wikle, C. K. (2019). Deep echo state networks with uncertainty quantification for spatio-temporal forecasting. *Environmetrics*, *30*(3), e2553.

Merow, C., Dahlgren, J. P., Metcalf, C. J. E., Childs, D. Z., Evans, M. E., Jongejans, E., & McMahon, S. M. (2014). Advancing population ecology with integral projection models: A practical guide. *Methods in Ecology and Evolution*, *5*(2), 99–110.

Merrick, L., & Taly, A. (2020). The explanation game: Explaining machine learning models using Shapley values. Proceedings of the International Cross-Domain Conference for Machine Learning and Knowledge Extraction (pp. 17–38).

Migenda, N., Ralf, M., & Wolfram, S. (2021). Adaptive dimensionality reduction for neural network-based online principal component analysis. *PLoS One*, *16*(3), e0248896. https://doi.org/10.1371/journal.pone.0248896

Mohan, A. T., Lubbers, N., Livescu, D., & Chertkov, M. (2020). Embedding hard physical constraints in convolutional neural networks for 3D turbulence. *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable*. Independently Published. https://christophm.github.io/interpretable-ml-book/

Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S. (2019). Improving precipitation estimation using convolutional neural network. *Water Resources Research*, *55*(3), 2301–2321.

Panahi, M., Sadhasivam, N., Pourghasemi, H. R., Rezaie, F., & Lee, S. (2020). Spatial prediction of groundwater potential mapping based on convolutional neural network (CNN) and support vector regression (SVR). *Journal of Hydrology*, *588*, 125033. https://www.sciencedirect.com/science/article/pii/S0022169420304935

Pawul, M., & liwka, M. (2016). Application of artificial neural networks for prediction of air pollution levels in environmental monitoring. *Journal of Ecological Engineering*, *17*(4), 190–196.

Penland, C., & Magorian, T. (1993). Prediction of Niño 3 sea surface temperatures using linear inverse modeling. *Journal of Climate*, *6*(6), 1067–1076.

Philander, S. (1990). *El Niño, La Niña, and the southern oscillation*. Academic Press.

Rahimikhoob, A. (2010). Estimating global solar radiation using artificial neural network and air temperature data in a semi-arid environment. *Renewable Energy*, *35*(9), 2131–2135.

Raza, A., Bardhan, S., Xu, L., Yamijala, S. S., Lian, C., Kwon, H., & Wong, B. M. (2019). A machine learning approach for predicting defluorination of per-and polyfluoroalkyl substances (PFAS) for their efficient treatment and removal. *Environmental Science & Technology Letters*, *6*(10), 624–629.

Recknagel, F., French, M., Harkonen, P., & Yabunaka, K.-I. (1997). Artificial neural network approach for modelling and prediction of algal blooms. *Ecological Modelling*, *96*(1-3), 11–28.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, *566*(7743), 195–204.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you? Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135– 1144).

Rodríguez-Pérez, R., & Bajorath, J. (2019). Interpretation of compound activity predictions from complex machine learning models using local approximations and Shapley values. *Journal of Medicinal Chemistry*, *63*(16), 8761–8777.

Rodríguez-Pérez, R., & Bajorath, J. (2020). Interpretation of machine learning models using Shapley values: Application to compound potency and multi-target activity predictions. *Journal of Computer-Aided Molecular Design*, *34*(10), 1013–1026.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386.

Roth, A. E. (1988). *The Shapley value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics survey*, *16*, 1–85. https://doi.org/10.1214/21-SS133

Ryo, M., Angelov, B., Mammola, S., Kass, J. M., Benito, B. M., & Hartig, F. (2021). Explainable artificial intelligence enhances the ecological interpretability of black-box species distribution models. *Ecography*, *44*(2), 199–205.

Samek, W., Montavon, G., Lapuschkin, S., Anders, C. J., & Müller, K.-R. (2021). Explaining deep neural networks and beyond: A review of methods and applications. *Proceedings of IEEE*, *109*(3), 247–278.

Shapley, L. S. (1953). *A value for n-person games*. In H. Kuhn & A. Tucker (Eds.), *Contributions to the theory of games The Annals of Mathematics* (Vol. *28*, pp. 307–317). Princeton University Press.

Simmons, S. J., & Burr, W. S. (2022, March). *Soil moisture and sea surface temperature data for Wikle et al. (2022)*. Zenodo. https://doi.org/10.5281/zenodo.6353971

Smith, M., & Alvarez, F. (2021). Identifying mortality factors from Machine Learning using Shapley values–A case of COVID19. *Expert Systems with Applications*, *176*, 114832.

Steinkraus, D., Buck, I., & Simard, P. (2005). *Using GPUs for machine learning algorithms*. Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR'05) (pp. 1115–1120).

Štrumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, *11*, 1–18.

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, *41*(3), 647–665.

Taconet, P., Porciani, A., Soma, D. D., Mouline, K., Simard, F., Koffi, A. A., & Moiroux, N. (2021). Data-driven and interpretable machine-learning modeling to explore the fine-scale environmental determinants of malaria vectors biting rates in rural Burkina Faso. *Parasites & Vectors*, *14*(1), 1–23.

Van der Laan, M., Pollard, K., & Bryan, J. (2003). A new partitioning around Medoids algorithm. *Journal of Statistical Computation and Simulation*, *73*(8), 575–584.

van Oldenborgh, G. J., Balmaseda, M. A., Ferranti, L., Stockdale, T. N., & Anderson, D. L. (2005). Did the ECMWF seasonal forecast model outperform statistical ENSO forecast models over the last 15 years? *Journal of Climate*, *18*(16), 3240–3249.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *37*(3), 328–339.

Wang, J., Liu, Z., Foster, I., Chang, W., Kettimuthu, R., & Kotamarthi, V. R. (2021). Fast and accurate learned multiresolution dynamical downscaling for precipitation. *Geoscientific Model Development*, *14*(10), 6355–6372.

Wang, Z., Zhao, L., Chen, H., Qiu, L., Mo, Q., Lin, S., & Lu D. (2020). *Diversified arbitrary style transfer via deep feature perturbation*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 7789–7798).

Werbos, P. J. (1982). *Applications of advances in nonlinear sensitivity analysis*. In *System modeling and optimization* (pp. 762–770). Springer.

Wickramasinghe, C. S., Amarasinghe, K., Marino, D. L., Rieger, C., & Manic, M. (2021). Explainable unsupervised machine learning for cyber-physical systems. *IEEE Access*, *9*, 131824–131843.

Wikle, C. K., Zammit-Mangion, A., & Cressie, N. (2019). *Spatio-temporal statistics with R*. Chapman & Hall/CRC Press.

Wilks, D. S. (2011). *Statistical methods in the atmospheric sciences. XGBoost manual* (Vol. *100*). Academic Press.

Xgboost Developers. (2021). https://xgboost.readthedocs.io/en/stable/index.html

Zhang, W., Itoh, K., Tanida, J., & Ichioka, Y. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, *29*(2), 4790–4797.

Zhang, W., Tanida, J., Itoh, K., & Ichioka, Y. (1988). *Shift-invariant pattern recognition neural network and its optical architecture*. Proceedings of Annual Conference of the Japan Society of Applied Physics. The Japan Society of Applied Physics (JSAP).

Zhang, X., Nguyen, H., Bui, X.-N., Tran, Q.-H., Nguyen, D.-A., Bui, D. T., & Moayedi, H. (2020). Novel soft computing model for predicting blast-induced ground vibration in open-pit mines based on particle swarm optimization and XGBoost. *Natural Resources Research*, *29*, 711–721.

Zhong, S., Zhang, K., Bagheri, M., Burken, J. G., Gu, A., & Li, B. (2021). Machine learning: new ideas and tools in environmental science and engineering. *Environmental Science & Technology*, *55*(19), 12741–12754.

Zhu, T., Guo, Y., Li, Z., & Wang, C. (2021). Solar radiation prediction based on convolution neural network and long short-term memory. *Energies*, *14*(24), 8498. https://doi.org/10.3390/en14248498

Zou, J., Han, Y., & So, S.-S. (2008). Overview of artificial neural networks. In D. J. Lvingstone (Ed.), *Artificial Neural Networks* (pp. 14–22). Humana Press, a part of Springer Science + Business Media, LLC.

How to cite this article: Wikle, C. K., Datta, A., Hari, B. V., Boone, E. L., Sahoo, I., Kavila, I., Castruccio, S., Simmons, S. J., Burr, W. S., & Chang, W. (2023). An illustration of model agnostic explainability methods applied to environmental data. *Environmetrics*, *34*(1), e2772. https://doi.org/10.1002/env.2772