A Machine Learning Explainability Tutorial for Atmospheric Sciences

MONTGOMERY L. FLORA,[a,b,e] COREY K. POTVIN,[b,c,e] AMY MCGOVERN,[c,d,e] AND SHAWN HANDLER[a,b]

[a] *Cooperative Institute for Severe and High-Impact Weather Research and Operations, University of Oklahoma, Norman, Oklahoma*
[b] *NOAA/OAR/National Severe Storms Laboratory, Norman, Oklahoma*
[c] *School of Meteorology, University of Oklahoma, Norman, Oklahoma*
[d] *School of Computer Science, University of Oklahoma, Norman, Oklahoma*
[e] *NSF AI Institute for Research on Trustworthy AI in Weather, Climate, and Coastal Oceanography, Norman, Oklahoma*

(Manuscript received 23 February 2023, in final form 6 November 2023, accepted 7 November 2023)

ABSTRACT: With increasing interest in explaining machine learning (ML) models, this paper synthesizes many topics related to ML explainability. We distinguish explainability from interpretability, local from global explainability, and feature importance versus feature relevance. We demonstrate and visualize different explanation methods, how to interpret them, and provide a complete Python package (scikit-explain) to allow future researchers and model developers to explore these explainability methods. The explainability methods include Shapley additive explanations (SHAP), Shapley additive global explanation (SAGE), and accumulated local effects (ALE). Our focus is primarily on Shapley-based techniques, which serve as a unifying framework for various existing methods to enhance model explainability. For example, SHAP unifies methods like local interpretable model-agnostic explanations (LIME) and tree interpreter for local explainability, while SAGE unifies the different variations of permutation importance for global explainability. We provide a short tutorial for explaining ML models using three disparate datasets: a convection-allowing model dataset for severe weather prediction, a nowcasting dataset for subfreezing road surface prediction, and satellite-based data for lightning prediction. In addition, we showcase the adverse effects that correlated features can have on the explainability of a model. Finally, we demonstrate the notion of evaluating model impacts of feature groups instead of individual features. Evaluating the feature groups mitigates the impacts of feature correlations and can provide a more holistic understanding of the model. All code, models, and data used in this study are freely available to accelerate the adoption of machine learning explainability in the atmospheric and other environmental sciences.

KEYWORDS: Artificial intelligence; Classification; Data science; Machine learning; Model interpretation and visualization

1. Introduction

Machine learning algorithms (ML) are increasingly common in the atmospheric sciences and are being used for severe weather applications (e.g., Gagne et al. 2017; Lagerquist et al. 2017; Cintineo et al. 2020; Lagerquist et al. 2020; Flora et al. 2021; McGovern et al. 2023), ensemble postprocessing (e.g., Rasp and Lerch 2018), subfreezing road temperature prediction (Handler et al. 2020), model parameterization (e.g., Brenowitz et al. 2020), tropical cyclone prediction (e.g., Kumler-Bonfanti et al. 2020), and climate modeling (e.g., Hernández et al. 2020). A key advantage of ML models is their ability to leverage multiple input features and learn useful multivariate relationships for prediction, calibration, and postprocessing. However, many ML models are considered "black boxes" in that the end user cannot readily understand the internal workings of the model (McGovern et al. 2019). We may not need to understand ML systems in all circumstances, but in high-risk situations—like severe weather forecasting—decision-makers want to know why a model came to its prediction. To help build human forecasters trust in ML predictions, it is essential to explain the "why" of an ML model's output in understandable terms and to create real-time visualizations of these methods (Hoffman et al. 2017; Karstens et al. 2018; Jacovi et al. 2021). Moreover, understanding a model's inner workings can identify strengths and weaknesses and possibly lead to improvements in the model.

The atmospheric science community is beginning to adopt explainability methods (e.g., Lakshmanan et al. 2015; Minokhin et al. 2017; Herman and Schumacher 2018; Rasp and Lerch 2018; McGovern et al. 2019; Jergensen et al. 2020; Lagerquist et al. 2020; Gagne et al. 2019; Handler et al. 2020; Hamidi et al. 2020; Mecikalski et al. 2021; Loken et al. 2022; Shield and Houston 2022; Mamalakis et al. 2022, 2023). Given the increasing interest in model explainability, we synthesize recent research on multiple explainability methods using ML models developed for severe weather (Flora et al. 2021), subfreezing road surface temperature (Handler et al. 2020), and lightning (Chase et al. 2022, 2023). For example, we highlight the difference between *feature relevance* (expected contribution to the model's output) and *feature importance* (expected contribution to the model's quality, that is, correspondence between model output and the target; Murphy 1993); a distinction often neglected in the literature. We demonstrate how to use and interpret explainability methods with a code base developed by the authors (scikit-explain;[1] Flora and Handler 2022). Our contributions include highlighting the distinctions between *intepretability* and *explainability* (discussed below), local and global explainability, and feature importance and relevance. Similar to Chase et al. (2022, 2023), we

Handler's current affiliation: Verisk Analytics Incorporated, Jersey City, New Jersey.

Corresponding author: Montgomery Flora, monte.flora@noaa.gov

[1] https://github.com/monte-flora/scikit-explain.

DOI: 10.1175/AIES-D-23-0018.1 e230018

TABLE 1. A nonexhaustive list of definitions of interpretability and explainability provided in the literature. Many studies not included here do not define the terms and use them interchangeably. These are partial quotes from each source, but quotation marks are omitted for readability.

Source	Interpretability	Explainability
Kim et al. (2016)	A method is interpretable if a user can correctly and efficiently predict the method's results.	N/A (no distinction made)
Doshi-Velez and Kim (2017)	The ability to explain or to present [the model] in understandable terms to a human.	Explaining the model after it is trained with post hoc methods
Adadi and Berrada (2018)	Interpretable systems are explainable if their operations can be understood by human[s].	N/A (no explicit definition provided, but the terms are not treated interchangeably)
Rudin (2018)	An interpretable machine learning model is constrained in model form so that it is either useful to someone or obeys structural knowledge of the domain such as . . . the physical constraints that come from domain knowledge.	Where a second (post hoc) model is created to explain the black box model
Gilpin et al. (2018)	Describe the internals of a system in a way which is understandable to humans.	Models that are able summarize the reasons for [black box] behavior . . . or produce insights about the causes of their decisions
Murdoch et al. (2019)	The use of machine-learning models for the extraction of relevant knowledge about domain relationships contained in data.	N/A (no distinction made)
Miller (2019)	The degree to which a human can understand the cause of a decision.	N/A (no distinction is made)
Linardatos et al. (2020)	[Ability] to identify cause-and-effect relationships within the system's inputs and outputs.	Explainability … is associated with the internal logic and mechanics that are inside a machine learning system
Molnar (2020)	Adopts the definitions from Miller (2019) and Kim et al. (2016).	N/A (no distinction is made; instead distinguishes interpretability/explainability from *explanation* where explanation refers to explaining individual predictions)
Rudin et al. (2021)	An interpretable ML model obeys a domain-specific set of constraints to allow it to be more easily understood by humans. These constraints can differ dramatically depending on the domain.	Explaining a black box model with a simpler model

provide a tutorial approach to interpreting and demonstrating these explainability methods. This paper strives to provide a comprehensive understanding of model explainability, even for those who may be new to the concept. We assume familiarity with ML methods and terminology and recommend Chase et al. (2022, 2023) for novice readers. Chase et al. (2022, 2023) offers a great introduction to ML for operational meteorology and provides open-source code for training and developing ML models.

Interpretability versus explainability

Many methods have been developed to understand black box models better. In response, substantial research has emerged on topics such as *interpretable ML* and *explainable artificial intelligence* (XAI) (e.g., van Lent et al. 2004; Kim et al. 2016; Adadi and Berrada 2018; Rudin 2018; Gilpin et al. 2018; Miller 2019; Linardatos et al. 2020; Molnar et al. 2020a; Rudin et al. 2021). Given the nascency of these topics, the definitions of *explainability* and *interpretability* are inconsistent throughout the literature, and many articles treat them interchangeably (Table 1). In this paper, we define these terms as follows:

- *Interpretability* is the degree to which an entire model and its components can be understood without additional methods.
- *Explainability* is the degree to which any partially interpretable or uninterpretable model (i.e., black boxes) can be approximately understood through post hoc methods (e.g., verification, visualizations of important features, or learned relationships).

This distinction between interpretability and explainability is needed since some in the ML and statistics community favor producing interpretable models (i.e., restricting model complexity beforehand to impose interpretability; Rudin 2018;

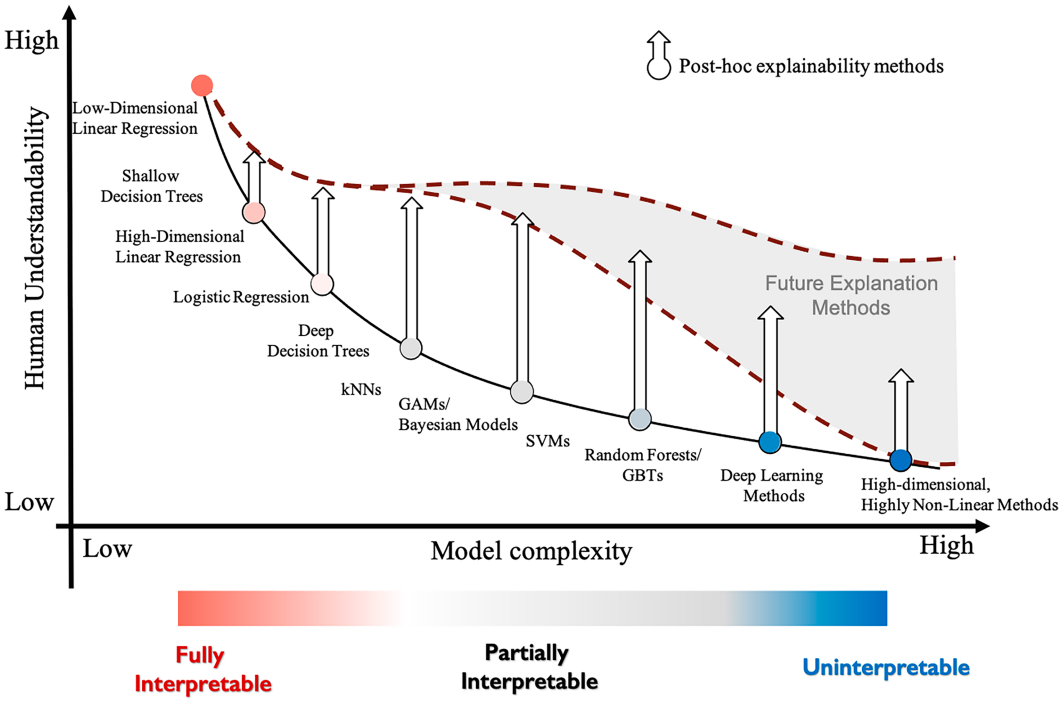

FIG. 1. Illustration of the relationship between understandability and model complexity. Fully interpretable models have high intrinsic understandability, while partially interpretable or simpler black box models have the most to gain from explainability methods. With increased dimensionality and nonlinearity, explainability methods can improve understanding. Still, there is considerable uncertainty about the ability of future explanation methods to improve the understandability of high-dimensional, highly nonlinear methods.

Rudin et al. 2021), while the general trend in the ML community is to continue developing partially interpretable and black box models and implementing post hoc methods to explain them. Lipton (2016) defines a fully interpretable model as one that has *simulatability* (the entire model can be considered at once), *decomposability* (each component of the model is human understandable) and *algorithmic transparency* (one can understand how the model was trained). A partially interpretable model may only meet one of these criteria. Explainability can be further subdivided into *model-specific explainability* (where the components of the model can be used for the explanation) and *model-agnostic explainability* (where no components of the model itself are used and no assumption is made about the model structure).

Figure 1 provides an illustration of interpretability and explainability. Fully interpretable models do not require post hoc explainability methods to improve understanding, while uninterpretable models have the most to gain from additional explanation methods. For example, low-dimensional linear regression is fully interpretable, and a shallow decision tree is partially interpretable. In contrast, a deep neural network (DNN) or a dense random forest is uninterpretable but can be approximately understood through external explanation methods. Explanation methods can only approximate model behavior, as they would otherwise be as incomprehensible as the black-box model itself. We do not view this as a limitation of explanation methods, as suggested by other studies (e.g., Rudin 2018; Rudin et al. 2021), since abstracting a complex model is required for human understanding. For example, it is common in the weather community to replace the full Navier–Stokes equations with conceptual models that are more understandable (e.g., quasigeostrophic theory). However, the model complexity controls the degree of explainability (Molnar et al. 2019). As the number of features increases or their interactions become more complex, the explanations for the behavior of the ML model will become similarly complex and possibly less accurate. It is uncertain how much progress can be made in comprehending complex, high-dimensional models through existing and future explanation techniques (Fig. 1).

2. Data

The following sections briefly describe the three datasets used in this study. They describe the feature engineering process, the target variable, and the classification task.

a. *Severe wind dataset*

The severe wind dataset is derived from the output of the 2017–19 Warn-on-Forecast System (WoFS), which is an experimental 3-km ensemble that produces rapidly updating forecast guidance at 0–6-h lead times. Additional details of the WoFS are found in Wheatley et al. (2015), Jones et al. (2016, 2020). The ML dataset contains features derived from intrastorm and environmental variables extracted from within 30-min ensemble storm tracks (Flora et al. 2019, 2021; Table 2). Environmental features are spatial averages (within a track) of

TABLE 2. Modified from Flora et al. (2021). Input variables from the WoFS. The asterisk (*) refers to negatively oriented variables. CAPE is convective available potential energy, CIN is convective inhibition, and LCL is the lifting condensation level. The midlevel lapse rate is computed over the 500–700-hPa layer, and the low-level lapse rate is computed over the 0–3-km layer. HAILCAST refers to the maximum hail diameter from WRF-HAILCAST (Adams-Selin and Ziegler 2016; Adams-Selin et al. 2019). The near-surface buoyancy (B) is defined as $B = g(\theta'_{e,z=0}/\overline{\theta}_{e,z=0})$ where g is the acceleration due to gravity, $\overline{\theta}_{e,z=0}$ is the lowest model level average equivalent potential temperature, and $\theta'_{e,z=0}(=\theta_{e,z=0} - \overline{\theta}_{e,z=0})$ is the perturbation equivalent potential temperature of the lowest model level. Values in the parentheses indicate those variables are extracted from different vertical levels or layers; 1 mb = 1 hPa.

Intrastorm	Environment
Updraft helicity (0–2 km, 2–5 km)	Storm-relative helicity (0–1, 0–3 km)
Cloud-top temp*	75-mb mixed-layer CAPE
0–2-km avg vertical vorticity	75-mb mixed-layer CIN
Composite reflectivity	75-mb mixed-layer LCL
1–3-km max reflectivity	75-mb mixed-layer equivalent potential temperature
3–5-km max reflectivity	U shear (0–6 km, 0–1 km)
80-m wind speed	V shear (0–6 km, 0–1 km)
10–500-m bulk wind shear	10-m U
10-m divergence*	10-m V
Column-max updraft	Midlevel lapse rate
Column-min downdraft*	Low-level lapse rate
Low-level updraft (1 km AGL)	Temp (850, 700, 500 mb)
HAILCAST max hail diameter	Dewpoint temp (850, 700, 500 mb)
Near-surface buoyancy*	Geopotential height (850, 700, 500 mb)

the ensemble mean and standard deviation fields valid at the beginning of the 30-min forecast period. Intrastorm features include both the spatial average values of ensemble mean and standard deviation fields (similar to environmental features) from time-composited fields and the ensemble mean and standard deviation of spatial 90th percentile of each ensemble member within a storm track (meant to capture storm intensity). The target variable is whether a severe wind report occurs within an ensemble storm track. Although Flora et al. (2021) developed ML models for all three severe hazards (wind, hail, tornadoes), we use only the severe wind dataset in the present study since the severe wind model was the most skillful of the three. The final dataset has 91 features, 510 000 examples, and a 3.6% base rate.

b. *Road surface dataset*

The road surface dataset from Handler et al. (2020) spans two cool seasons: 1 October 2016–31 March 2017 and 1 October 2017–31 March 2018. Thirty features were used for training, including near-surface variables from the High-Resolution Rapid Refresh (HRRR) model, as well as derived features (Handler et al. 2020, their Table 3). The variables were informed by previous research that identified variables relevant for modulating surface road temperatures (e.g., Crevier and Delage 2001). Hourly road surface temperature observations from the Road Weather Information System (RWIS) sites were used as the target variable for training. Each example was labeled below or above freezing based on the temperature reported by the RWIS site. The RWIS sites used are shown in Fig. 1a of Handler et al. (2020). The final dataset has 30 features, 1 million examples, and a 39.7% base rate.

c. *Lightning dataset*

The Storm Event Imagery (SEVIR; Veillette et al. 2020) database is a spatiotemporal dataset curated on 10 000 storm events. The SEVIR dataset has four *Geostationary Environmental Satellite System–16* (*GOES-16*) variables: visible reflectance (VIS), midtropospheric water vapor brightness temperature (WV), infrared brightness temperature (IR), and Geostationary Lightning Mapper (GLM) flashes. The dataset also contains Next-Generation Radar (NEXRAD) vertically integrated liquid (VIL). The SEVIR images are largely located over thunderstorms and general convective activity. Using these weather-centered imageries, Chase et al. (2022, 2023) lowered the spatial resolution and then extracted spatial percentiles (0, 1, 10, 25, 50, 75, 90, 99, 100) from the satellite variables (excluding lightning flashes) and the radar-based VIL. The target is a binary variable indicating whether at least one GLM flash is present in the image. Additional details on the dataset can be found in Chase et al. (2022, 2023). The final dataset has 36 features, 60 000 examples, and a 50% base rate. It is worth noting that the SEVIR dataset is tailored and not representative of lightning climatology.

3. Machine learning algorithms

This study uses classification logistic regression, random forests, and neural network models available in the Python sci-kit learn package (Pedregosa et al. 2011). Consistent with Flora et al. (2021), Handler et al. (2020), and Chase et al. (2023), we use logistic regression to predict whether a storm track will be associated with a severe wind report, a random forest for the road surface dataset to predict whether a road will freeze, and a neural network for predicting lightning flashes.

a. *Logistic regression with elastic nets*

A logistic regression model is a linear regression model designed for classification tasks. Given a binary outcome

variable y (1 or 0), we can estimate the probability that y belongs to a particular class [e.g., $P(y = 1|X)$] as

$$P(y = 1|X) = \frac{1}{1 + \exp\left(-\beta_0 - \sum_{i=1}^{N} \beta_i x_i\right)}, \quad (1)$$

where β_i are the learned weights, x_i are the features and β_0 is the bias term. Although logistic regression is based on a linear model, the predicted probability is not linearly related to the input features. Unless the features are binary, binary classification is inherently nonlinear due to the nonlinear transformation of continuous features into binary target variables. Given that the summation in Eq. (1) is an exponent, a multiplicative interaction exists between all N features. Therefore, the logistic regression model has questionable interpretability in probability space, especially as the number of features increases. Regularizations, both L1 and L2, are used for training. L1 regularization acts as a feature selection method by zeroing coefficients for less useful features, while L2 regularization encourages smaller weights, thereby discouraging the model from heavily favoring a small subset of features.

b. *Random forest*

The random forest (Breiman 2001) is an increasingly popular ML algorithm. A classification random forest is comprised of multiple decision trees, each partitioning the feature space into subregions of increasing "purity" (homogeneity of the target variable). To improve the predictive accuracy of the random forest, each tree is trained on a bootstrapped resampled version of the data, and for each split, only a small random subset of features is considered. For each tree, the prediction is the proportion of positive class examples (the number of positive class examples divided by the total number of examples in the leaf node). The final prediction of the forest is the ensemble average of the separate tree predictions.

c. *Feed-forward neural network*

A standard feed-forward neural network consists of multiple layers of interconnected nodes, or neurons, organized sequentially. Information flows in one direction, from the input layer to the hidden layers and then to the output layer. Each neuron in the network receives inputs from the previous layer and applies a weighted sum and an activation function (e.g., ReLU) to produce an output. A binary classification model passes the final output through a sigmoid function to produce a single probability.

4. Explainability methods

In line with Lipton (2016), Molnar (2020) identifies five scopes of ML explainability, which can be summarized into three main categories:

- *Algorithmic transparency*: How does the algorithm create the model?
- *Global explainability*: How does the trained model as a whole make predictions? How do components of the model affect the predictions?
- *Local explainability*: Why did the model make a certain prediction for a specific set of examples?

Model explainability typically refers to global or local explainability, as algorithmic transparency does not refer to a specific model or prediction. Both global and local explainability methods can be summarized as measuring and visualizing:

- *Feature relevance* and *feature importance*: The ranking of features or sets of features by how much they contribute to a model's output or its quality (e.g., Breiman 2001; Lakshmanan et al. 2015; Greenwell et al. 2018; Lundberg and Lee 2017; Covert et al. 2020b).
- *Feature effects*: The expected functional relationship between a feature (or set of features) and an ML model's output (e.g., Friedman 2001; Apley and Zhu 2016; Greenwell et al. 2018; Lundberg and Lee 2017).
- *Feature interactions*: How a given feature's effect is dependent on other features and the strength of that effect (e.g., Friedman and Popescu 2008; Greenwell et al. 2018; Molnar et al. 2019; Oh 2019; Kuhn and Johnson 2019).

Global approaches attempt to decompose the model into parts that can be understood individually (Murdoch et al. 2019; Molnar et al. 2020a). Local approaches explain individual predictions. Local methods can include but are not limited to decomposing a prediction into the contribution of each feature (e.g., Saabas 2014; Ribeiro et al. 2016; Lundberg and Lee 2017) or developing counterfactual explanations to form what-if scenarios (Molnar et al. 2020a; Molnar 2020). Combining global and local explainability approaches can provide a holistic understanding of the model's behavior. A summary of the methods discussed in the following section is given in Fig. 2. Though several disparate explainability methods exist, we will discuss how many of them can be subsumed by one or two approaches.

a. *Feature importance versus relevance*

Ranking features within a dataset based on their contribution to the model is a crucial component of model interpretability and explainability. In the literature, feature ranking methods tend to measure one of three quantities:

1) strength of univariate relationship with the target variable,
2) expected contribution to the model's output, or
3) expected contribution to the model's quality.

The first category does not involve the model (e.g., is model agnostic) and reflects data characteristics, such as correlations with the target variable or the Kullback–Leibler J measure (Lakshmanan et al. 2015). Regression coefficients, feature attribution methods [e.g., Shapley additive explanations (SHAP), tree interpreter, local interpretable model-agnostic explanations (LIME)], and partial dependence/accumulated local effect variance (Greenwell et al. 2018) are examples of the second category, while the third category includes different variations of permutation importance (Breiman 2001; Strobl et al. 2008; Lakshmanan et al. 2015; Au et al. 2021; König et al. 2020; Covert et al. 2020b), Gini impurity importance (Breiman 2001; McGovern et al. 2019), Shapley additive global importance

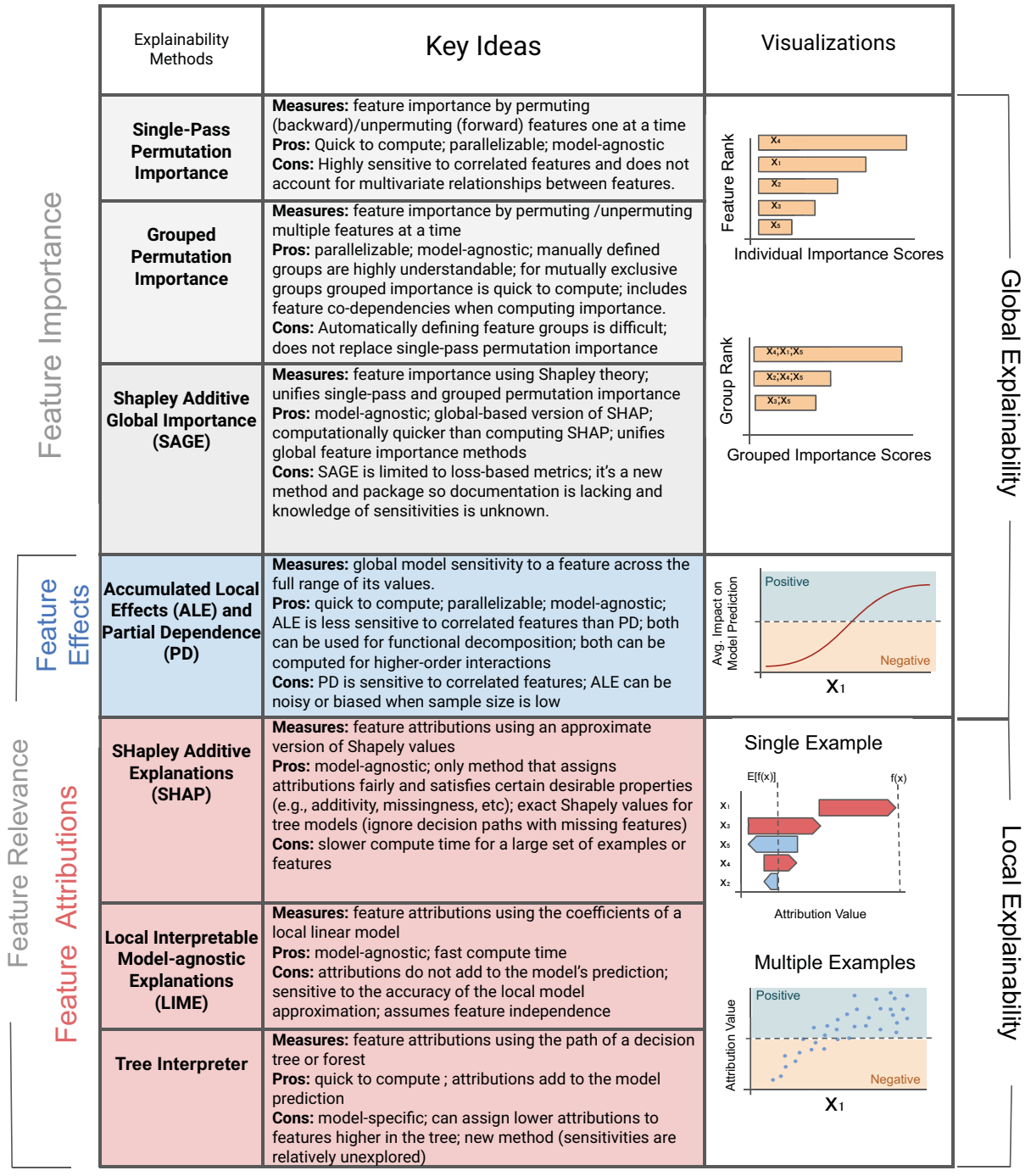

FIG. 2. Explainability methods discussed in this study, their key ideas, and typical visualizations. Methods shaded in gray are unified by SAGE, and those shaded in red are unified by SHAP.

(SAGE; Covert et al. 2020b), and sequential feature selection (McGovern et al. 2019).

In general, the first two categories can be defined as measures of *feature relevance*, while *feature importance* is formally defined with respect to model quality [van der Laan 2006; Covert et al. 2020b; Hooker et al. 2021; *quality* is defined as the correspondence between the model's output and the target variable (Murphy 1993)]. We can further separate the notion of feature importance into *model-specific feature importance* and *model-agnostic feature importance*. Model-specific importance quantifies how much a set of features contributes to the performance of a given model. In contrast, model-agnostic importance quantifies the hypothetical range of contributions in which any well-performing model may rely on a set of

features for model performance (Fisher et al. 2018; Covert et al. 2020b).[2] For example, sequential feature selection measures the importance of a feature by removing or adding a feature and retraining the model. Situations can arise where, due to compensatory effects, one seemingly important variable is removed, and the model adjusts using the remaining variables (Kuhn and Johnson 2019). Therefore, sequential feature selection approximates model-agnostic feature importance. There are alternative variations on the "remove and retrain" approach where the feature is marginally or conditionally permuted and the model retrained to determine importance (Hooker et al. 2021), but these approaches are often computationally expensive. The most common way to measure model-specific feature importance is the permutation importance method, which evaluates the change in model performance after permuting a feature's values. Permuting a feature maintains its marginal distribution but breaks up the relationship with the target variable. The general drawback to this approach is that marginally permuting a feature's values alters the conditional joint distribution among features. Breaking up conditional distributions can cause the importance scores to be heavily impacted by out-of-distribution or unphysical samples (Hooker et al. 2021). However, permutation importance methods that attempt to maintain conditional distributions are either restricted to specific ML algorithms (e.g., random forests; Strobl et al. 2008) or computationally restrictive (Hooker et al. 2021) and inevitably impact the interpretation of the results (Molnar et al. 2020b).

b. *Shapley-based methods and the unification of explainability methods*

A promising approach for model explainability is to think of the model's output or quality as a sum of contributions (ϕ) from each feature (N features) (Lundberg and Lee 2017; Covert et al. 2020b):

$$\text{model output or quality} = \text{bias} + \phi_1 + \phi_2 + \cdots + \phi_N. \quad (2)$$

The idea is that the bias (e.g., the base rate for feature relevance or the accuracy of a climatological prediction for feature importance) is a starting point, and each feature contributes positively or negatively until the final score (output or quality) is achieved. The simplest way to compute ϕ for a given feature is computing the difference in model output after "removing" the feature (i.e., marginalizing it out by replacing its value with a random value from the training distribution). This approach, however, is not "fair" to a given feature as it does not account for feature interactions. To ensure fairness, we would need to compute the difference in output when a feature is included and not included in a feature subset for all possible feature subsets (Fig. 3). The theoretical idea behind this approach comes from game theory and scores known as Shapley values (Shapley 1953). For contributions to model output, ϕ_i in Eq. (2) is known as the SHAP value, while for model quality, it is known as the SAGE value. As a reminder, SHAP is computed for a single example (local explainability), while SAGE is computed over a dataset (global explainability).

What does it mean to say Shapley values are "fair"? For fair values, they must satisfy the following axioms:

1) Local accuracy (additivity): The sum of the contributions of each feature plus the bias must equal the final outcome.
2) Consistency (monotonicity): If an ML model changes so that the marginal contribution of a feature increases or stays the same, the feature attribution must also increase or remain the same, respectively.
3) Missingness: Missing features (e.g., features that have been marginalized out) must have a zero contribution to the model.

The Shapley values are the only method that satisfies all three of these properties (Shapley 1953; Young 1985; Lundberg et al. 2018, 2020).

Except for tree-based methods and low-dimensional datasets, computing exact Shapley values is intractable as it requires creating $N!$ possible feature subsets. Another issue is appropriately accounting for missing features. Typically, their values are replaced with samples from their marginal distribution to approximate missing features. This approach is repeated multiple times for different samples to improve the estimation of their "missingness." For this study, we use the default permutation-based method in the SHAP package. This method creates many feature order permutations and then, for each feature ordering, iterates completely through the features in both the forward and reverse directions. Though approximate, this approach guarantees local accuracy (additivity) and allows for feature clustering (using SHAP's partition masker), which improves the Shapley value estimates when features are correlated/colinear.

We can group the features into coalitions (feature groupings) and compute an extension of the Shapley values known as Owen values (Owen 1977; López and Saboya 2009). To compute the Owen value for x_j, we compute the weighted average change in prediction when x_j is included and not included in all possible feature subsets, but such that the subsets exclude features from one grouping. We repeat the calculation with each feature grouping excluded and average those values. For example, let us consider the road surface dataset, which has multiple temperature- and radiation-based features. In one iteration, we might create a feature subset of just temperature variables and measure the impact of including and excluding surface temperature. In the next iteration, we would exclude the temperature variables, include all the radiation variables, and again measure the effect of including and excluding surface temperature. Though we are creating distinct feature groupings, the outcome is a unique contribution from each feature. Another key benefit of the Owen values is that the number of feature subsets to evaluate is significantly reduced. For the remainder of the paper, we will refer to the Owen values as SHAP values. We compute the SHAP values

[2] The idea of model-specific importance and model-agnostic importance is referred to as model-based predictive power and universal predictive power in Covert et al. (2020b). Fisher et al. (2018) loosely refers to the notion of model-agnostic importance with the idea of *model class reliance*: the highest and lowest degree to which any well-performing model with a given class may rely on a predictor for prediction accuracy.

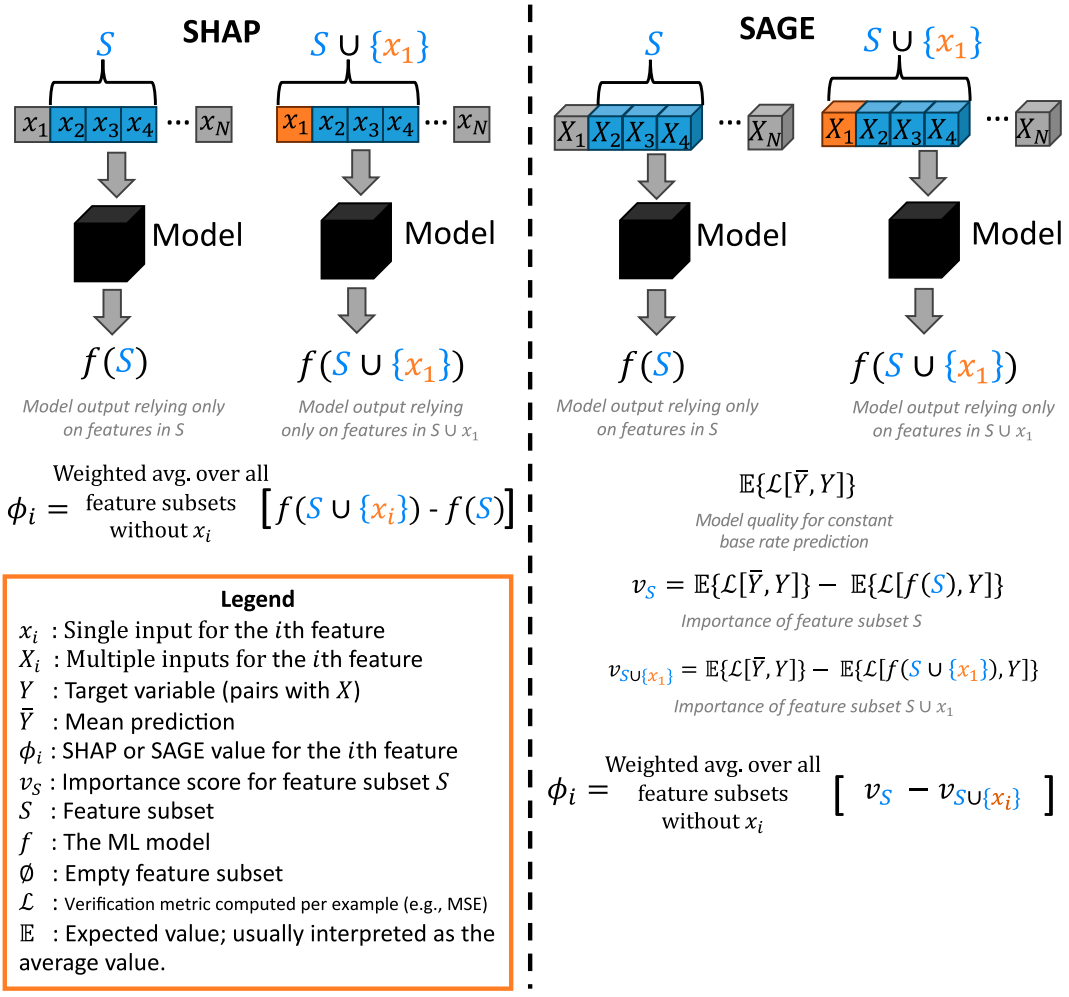

FIG. 3. Annotated illustration of the (left) SHAP and (right) SAGE computation.

on 2500 random samples from each training dataset to capture the global aspects of each model.

Though several disparate explainability methods exist in the literature, many can be unified using the Shapley-based approach (Lundberg and Lee 2017; Covert et al. 2020b,a, their Fig. 2). Many explainability approaches are based on simulating the effect of feature removal, which Covert et al. (2020a) demonstrated is implicitly tied to the cooperative game theory that Shapley methods are based on. For example, the LIME (Ribeiro et al. 2016) method fits a linear model on perturbations of the dataset around the example to be explained and uses the model coefficients to generate the individual feature contributions in Eq. (2). The "width" of the local area where the perturbations are generated is dictated by a kernel function (often exponential). Using the kernel method outlined in Lundberg and Lee (2017), the LIME values become approximate SHAP values. The LIME method, however, cannot guarantee local accuracy, is prone to providing misleading explanations when features are correlated, and is subject to the accuracy of the local linear model. A less well-known method, tree interpreter (Saabas 2014; Loken et al. 2022), can provide feature contributions for tree-based methods. Still, it only considers a single feature ordering and has consistency issues as features near the root can incorrectly be given less weight (Lundberg et al. 2020). Ultimately, it is unnecessary to compute tree interpreter values as exact Shapley values can be computed for tree-based models, and the computation times for tree interpreter and the tree-based SHAP method are comparable (Lundberg et al. 2020).

SAGE (Covert et al. 2020b) also unifies the existing permutation importance methods for measuring feature importance. Permutation importance is one of the most popular methods for assessing feature importance. It was first introduced in Breiman (2001) but was later expanded in Lakshmanan et al. (2015) and generalized in König et al. (2020); Au et al. (2021). The main goal of permutation importance is to measure the expected model quality when the values of a single feature are permuted. Permuting a feature's values renders it uninformative of the target variable but maintains the marginal distribution so as not to introduce output bias. The feature is considered unimportant if the expected model quality is relatively unchanged after the feature values are shuffled. When repeated for each

feature, this method is known as the *single-pass* permutation importance (McGovern et al. 2019). Generally, one could permute any number of features to evaluate their importance to the model, known as grouped permutation importance (Au et al. 2021). By permuting multiple features, we better account for feature codependencies (Lakshmanan et al. 2015; Gregorutti et al. 2015). For example, numerous studies have found that when features are correlated, their single-pass permutation importance scores can be reduced (Strobl et al. 2007, 2008; Gregorutti et al. 2015, 2017). Furthermore, when permuting more than one feature, the total importance is not equal to the sum of their individual importances, as it also depends on the codependencies between the features (Gregorutti et al. 2015). Though evaluating the output of the different permutation importance methods can be useful, SAGE unifies all preexisting permutation importance-based methods by systematically assessing the impact of withholding multiple feature subsets (Fig. 2) and using a Shapley-style equation (Fig. 3).

c. Accumulated local effects

Computing SHAP values for all training set samples is often unduly computationally expensive. To complement SHAP, we measure global feature effects using accumulated local effects (ALE; Apley and Zhu 2016), which is an alternative to partial dependence (PD; Friedman 2001) that properly accounts for feature codependencies. The ALE for the feature x_j is

$$\mathrm{ALE}_j(x_j) = \int_{\min(z_j)}^{x_j} \mathbb{E}\left[\frac{\partial f(\mathbf{X})}{\partial X_j} | X_j = z_j\right] dz_j - c, \tag{3}$$

where f is the ML model, $\mathbf{X}$ is the set of all features, z_j are the values of x_j, and c is the integration constant. The constant c is the mean of $\mathrm{ALE}(x_j)$, so the mean feature effect is zero.

ALE computes the expected change in prediction over a series of conditional distributions for a given feature and then accumulates (integrates) them to return the feature effect. By computing the average change in prediction over a series of small windows, ALE isolates the impact of the feature from the effects of all other features and avoids the pitfall of PD, which can suffer from unlikely or nonphysical combinations of feature values. More details on the ALE calculation are provided in Molnar (2020) and Flora (2020). For this study, we compute the ALE on the same 50 000 random samples from each training dataset used for the SAGE computation.

To aid in interpreting the feature effects, we compute the conditional base rate per feature [i.e., $p(y = 1|x_i)$] using a Bayesian histogram method (Python package bayeshist; Hafner 2022). This method assumes a beta distributed prior [$p(y = 1) \approx \mathcal{B}(\alpha, \beta)$ where α and β are shape parameters] and a similar distribution for the posterior [$p(y = 1|n_i^+, n_i^-) \approx \mathcal{B}(\alpha + n_i^+, \beta + n_i^-)$] where n_i^+, n_i^- are the number of positive and negative samples in the ith bin, respectively. By binning feature x_i's values, we can compute $\mathcal{B}$ in a series of quantiles. The method compares and combines each pair of neighboring bins if they are likely from the same event rate sample. The final output is a conditional base rate distribution for each bin, which allows us to show the median value and 95% confidence intervals.

d. Training or testing dataset for explainability?

According to Molnar (2020, their section 5.5.2), favoring the training or testing dataset for feature importance or relevance remains an open question. Lakshmanan et al. (2015) argued for only using the training dataset. The goal of measuring feature importance is quantifying how the model relies on each feature and not how well the model generalizes to unseen data. Generally, the testing dataset's conditional distribution is unlikely to fully represent the training conditional distribution. If the ML model learned a pattern in the training dataset that is underrepresented in the testing dataset, then evaluating feature importance on the testing dataset can bias our understanding of how the model works. For example, consider an imaginary scenario where the training dataset has temperature ranges from −15° to 10°C, whereas the testing dataset range is from −5° to 5°C. If the ML model learned to rely heavily upon temperatures < -10°C to predict freezing road surfaces, we would fail to determine that using the testing dataset. One could evaluate the feature importance on training and testing data to identify any discrepancies. Still, it would be necessary to ascertain whether the differences are due to poor sampling or overfitting. To avoid these difficulties, feature importance in this tutorial is evaluated using the training dataset.

5. Demonstration of explainability methods

a. General approach to model explainability

This section will outline a general strategy for explaining an ML model. Our first step is to analyze the most relevant and important features. By examining the discrepancies (or agreements) between these two rankings and the learned relationships, we can improve our understanding of the model and identify strengths or weaknesses. We limit this section to the road surface dataset as it has the fewest features and is the most intuitive prediction task of the three datasets.

The SHAP and SAGE feature rankings for the road surface dataset are shown in Fig. 4, and the learned effects of the top six most important features are shown in Fig. 5. Figure 4a displays the features ranked by their mean absolute SHAP value. Each feature is represented by a scatterplot color coded by the normalized feature value, and density can be approximated by vertical spread. This plot allows a more comprehensive interpretation of feature rank and effect. For example, T_{sfc} SHAP values display a bimodal distribution, with negative values for higher temperatures and positive values for moderate to low temperatures, with a sparse density in between. We interpret SHAP values as an increase or decrease in the model prediction compared to the global average model prediction. For example, a cold surface temperature can increase the probability of a freezing road surface by 10%–15% given the static base rate of 40%. The SHAP values may not be as helpful for predictions near the base rate. We interpret the SAGE values similarly: the decrease or increase in model performance compared to simply predicting the base rate. Suppose the model is near-optimal (i.e., a Bayesian classifier). In that case,

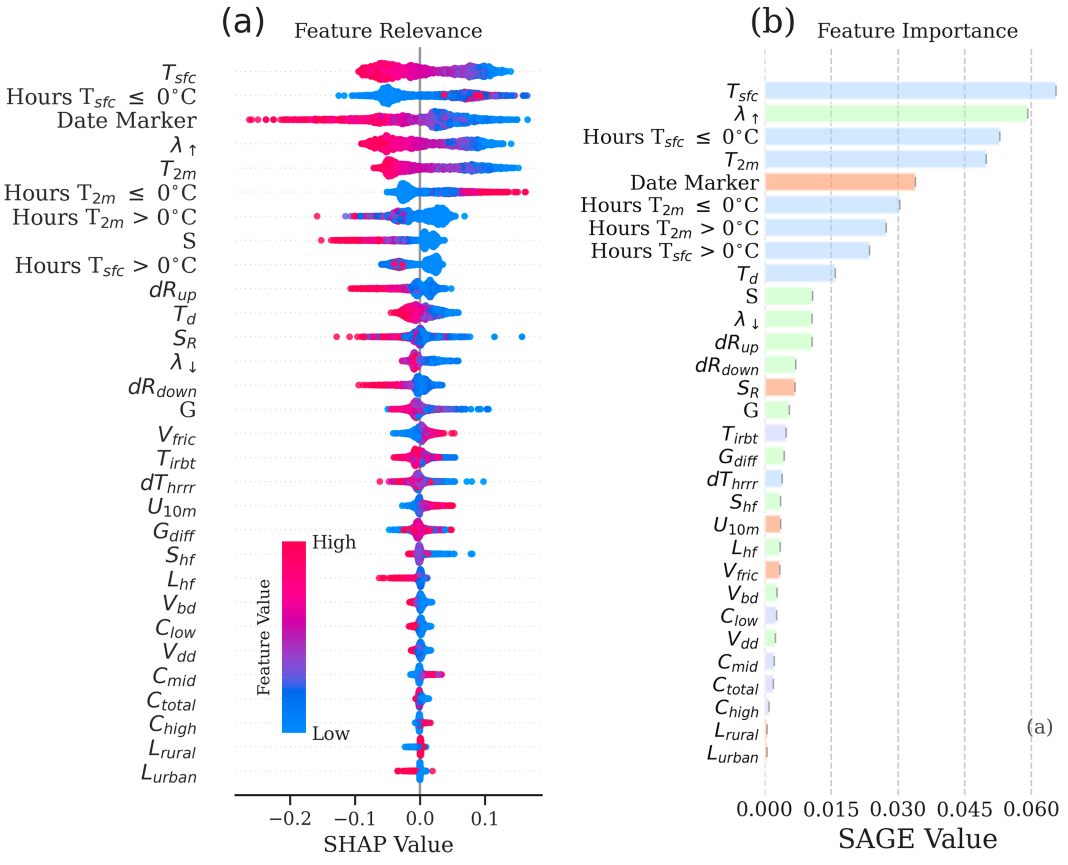

FIG. 4. (a) SHAP and (b) SAGE feature rankings for the road surface dataset. For (a), scatter points are SHAP values, while the color coding indicates the minimum-maximum normalized feature value (0–1). To approximate a violin-style plot, vertical spread is applied to dense regions. Features for the SAGE rankings are color coded by type: temperatures in blue, radiation in green, cloud coverage in purple, and remaining orange.

we can also interpret SAGE values as conditional mutual information or how informative a feature is of the target variable given the other features in the dataset. When analyzing importance, it is crucial to remember that features can have lower importance due to their information already being included in other features, which may be more informative of the target variable.

The learned effect plots in Fig. 5 are dense but include multiple useful details:

- Black curve: Average first-order effect—the direct contribution of a single feature independent of the other features—as measured by the ALE method.
- Scatter points: SHAP values as a function of a feature's value.
- Dashed red curve: Conditional base rate measured by the Bayesian histogram method described in section 4c.
- Rug plot on the bottom: The approximate distribution of the feature values (higher density equates to more samples).

By combining the ALE curve with SHAP dependence plots, we can better understand how changes in a feature's value affect the model predictions on average and for specific examples. For instance, the ALE curve highlights the average first-order effect, while the vertical spread of SHAP values for a specific feature value reveals the impact of higher-order effects. The scatter points are color coded based on the values of one of the most important features, which helps identify potential feature interactions. When exploring relevant higher-order effects, we use the strong heredity principle from Kuhn and Johnson (2019): "interaction terms may only be considered if the [first order] terms preceding the interaction are effective at explaining [the target variable]." The idea is that if a feature has a weak first-order effect, it is unlikely to be included in a meaningful second-order or higher effect. To estimate the strongest interaction with the most important feature, we bin the SHAP values of the most important feature by the feature values of all other features and use the feature with the highest linear correlation coefficient. Potential interaction effects should be interpreted cautiously, as some may be spurious compensation effects (e.g., probabilities cannot exceed 1) or arise from feature correlations, as shown below.

The most relevant and important features are physically plausible for predicting freezing road surfaces (Fig. 4). For example, the surface temperature is the most relevant and important feature (T_{sfc}) as cooler surface temperatures, especially $<-2.5°C$, are highly likely to be associated with freezing road surfaces (Figs. 4a and 5a). The duration of freezing

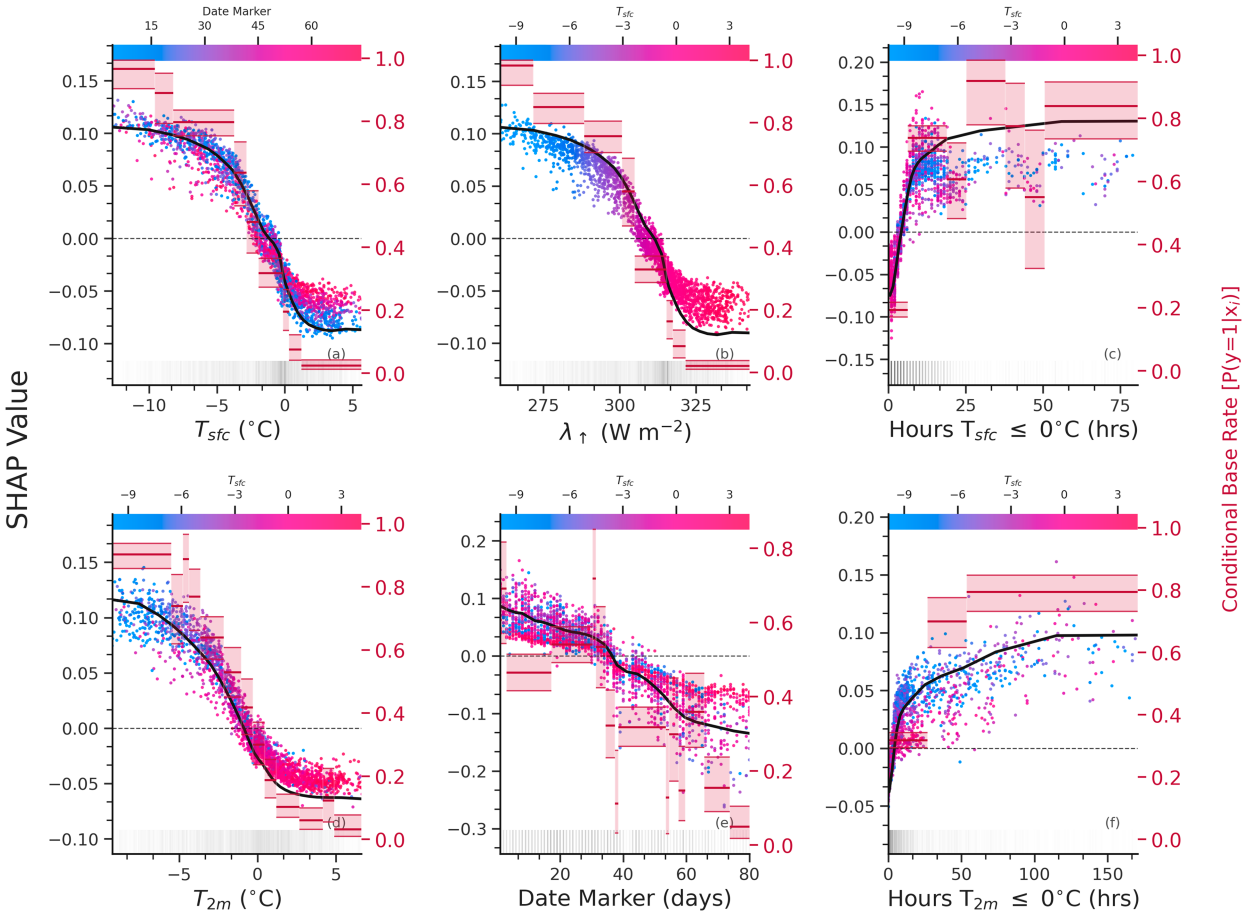

FIG. 5. (a)–(f) Feature effects plots for the top six most important road surface dataset features. The ALE curve is black, while the scatter points are SHAP values. The red histogram plateaus indicate the conditional base rate, and the 95% uncertainty region with its values are on the right *y* axis. The rug plot on the bottom of each panel indicates the approximate distribution. Scatter points are color coded by the surface temperature values, except for the surface temperature in (a), which is color coded by the feature with the strongest approximate interaction. Vertical dispersion of SHAP values indicates feature interactions.

temperatures and date marker (absolute distance from 10 January) are also intuitive features for predicting freezing road surfaces. The learned relationships could be treated as forecasting rules of thumb. For example, the likelihood of freezing road surfaces greatly increases if surface temperatures have been below 0°C for at least 12.5 h (Fig. 5c). However, freezing temperatures for longer than 12.5 h do little to increase the odds of a frozen road surface as freezing temperatures for longer than 12.5 h were associated with an average surface temperature of −6°C (21.2°F). Road surfaces in the dataset were always frozen at those temperatures, so freeze duration adds little information. Last, there is possibly an erroneous interaction between surface temperature and freezing duration (Fig. 5c). There is a dichotomous effect where warmer temperatures can either increase or decrease the impact of freezing durations between 12 and 24 h. This result may point to a model deficiency deserving further attention or the influence of a third variable that could be identified and explored by feature interaction methods like SHAP interaction values (Lundberg et al. 2018).

Though the most relevant features are generally also the most important, some differences between the rankings provide insights into the model and the physical processes involved. For example, the date marker is tied for the most relevant feature (Fig. 4a), but its importance is less than half that of the most important feature. Based on Figs. 4a and 5e, being further from 10 January significantly decreases the probability of a frozen road surface, which is physically reasonable as temperatures are climatologically warmer throughout the CONUS during the Fall/Spring season and unlikely to produce frozen roads. Conversely, frozen road surfaces as we approach the middle of winter become almost certain for parts of the CONUS. While the date does provide a general indication of the season and, thus, broad climatological patterns, it does not account for the variability of specific weather conditions that lead to road freezing, such as sudden cold fronts or snowfall. Thus, a date maker is an excellent feature for predicting nonfrozen road surfaces but less useful for frozen roads.

It may be surprising that many of the radiation-based variables have low/near-zero importance (Fig. 4b) as energy fluxes

at the surface are the primary driver of road surface conditions (Crevier and Delage 2001). For example, at nighttime with near clear skies, radiation away from the surface can promote rapid cooling (Crevier and Delage 2001). Unfortunately, capturing these higher-order effects is difficult for traditional ML models, especially random forests, which select from a random set of features for each split, decreasing the odds of detecting higher-order effects (Wright et al. 2016). Furthermore, using individual radiation terms instead of the total radiation budget could reduce importance and relevance. Finally, recall that the road surface dataset is a 1-h nowcasting dataset composed of numerical weather prediction (NWP)-derived variables. Thus, the surface and 2-m temperature values reflect radiation-based changes due to the surface and radiation parameterization schemes. Moreover, the radiation features also contain information about cloud coverage, explaining why those features have lower importance and relevance. Thus, the surface temperature variable incorporates much of the information from the cloud coverage and radiation variables and is more strongly correlated with the target variable (i.e., whether the road is frozen). Nevertheless, the ML model can still derive nonzero importance from the radiation terms, as the temperature and radiation features are only partially redundant due to the influence of other variables and imperfections in the radiation parameterization scheme.

b. *Impact of correlated features on model explainability*

It is a well-known issue that correlated features can significantly impact model explainability methods, especially those that use permutation-based approaches (Strobl et al. 2007, 2008; Gregorutti et al. 2015, 2017; Hooker et al. 2021; Molnar et al. 2021). In this section, we will use the lightning dataset to demonstrate how correlated features can impact feature importance and relevance scores and provide a complementary approach where feature importance/relevance is assessed based on feature groupings rather than individually.

Correlated features can impact both feature importance and feature relevance. It is well known that for models like logistic regression or neural networks, the model coefficients are nonunique when two features or more are linearly dependent (Gregorich et al. 2021). In the case of two highly correlated features, the model can either learn to favor one or the other (through regularization) or keep both, but with opposite signs as a compensation effect. To demonstrate this effect, Fig. 6 shows the SHAP and SAGE feature rankings for the lightning dataset. One discrepancy is that WV_{1st} is a highly relevant feature (ranked 8th) but much less important (lowest rank). The learned effect for WV_{1st} opposes the base rate and can be a strong effect for higher WV temperatures (Fig. 7f). Recall that different feature sets are permuted for the SAGE calculation; some of these sets exclude most of the features. In these situations, compensating features like WV_{1st} will be heavily penalized (lower importance) as their benefit to the model is contingent on one or more other features. Thus, the negative SAGE score, but higher mean absolute SHAP values is indicative of a compensating effect. Without the other features, the learned relationship WV_{1st} worsens the model as it does not reflect the underlying dataset (the same argument can be made for WV_{10th}).

Feature correlations impact model explainability because correlated features can lead to shared relevance, which can obscure the individual impact of each feature on the model's output or quality. Therefore, a useful alternative to analyzing individual features is to group them to avoid misinterpreting the model's behavior or the true feature–target relationship. By doing so, we can capture collective effects rather than isolating the influence of individual feature variations on the model's output. Figure 8 revisits the analysis of the road surface dataset and shows the grouped SHAP and SAGE rankings, which are obtained by summing together the SHAP/SAGE values for features in each group. To normalize the feature values for each group, we scale the features using minimum–maximum normalization and then compute the average scaled feature value.

Based on the grouped SHAP and SAGE rankings, the freezing duration features (see Table 3) are the most relevant and important, while radiation variables have much less relevance and importance. This is not inconsistent with top individual features in Fig. 4 but provides a concise picture of the model's behavior. For example, when grouped together, the radiation features are important but produce lower SHAP values than the temperature variables. We can similarly summarize the feature groups in the lightning dataset (Fig. 9). Though VIL_{max} was the single most important feature (Fig. 6b), the IR features are the most relevant and important overall, and the importance of WV and VIS are nonzero but minimal. The lightning probability is more sensitive to VIL than IR, as can be seen with the strong bimodal distribution among the SHAP values and the sharp slope of the ALE curve for VIL (cf. Figs. 7a,c). Spatially, VIL contains highly localized variables (high values associated with storms); therefore, it is reasonable that it would have a highly dichotomous impact on the model. However, the lightning dataset has an even class balance, and the IR_{min} substantially lowers the lightning probabilities, increasing its importance. If the dataset had a more representative lightning climatology (including more nonevents), VIL would likely have higher importance than IR.

c. *Using explainability to diagnose model weaknesses*

In the sections above, we have found that the most relevant and important features of the road surface and lightning ML models in this study are physically plausible. Likewise, one would be hard pressed to find issues with the top features of the severe wind dataset (e.g., 80-m wind speed, composite reflectivity, and cloud-top temperature) as each represents convection intensity and low-level wind flow (Fig. 10). However, are the least important features also reasonable? In this case, many of the negative importance features are likely to have high model-agnostic feature importance but are negatively impacted by correlated features [Fig. 10b; e.g., 0–1-km storm-relative helicity (SRH) vs 0–3-km SRH, updraft vs hail, composite reflectivity vs 3–5-km maximum reflectivity]. One exception is downdraft, which is strongly correlated with other

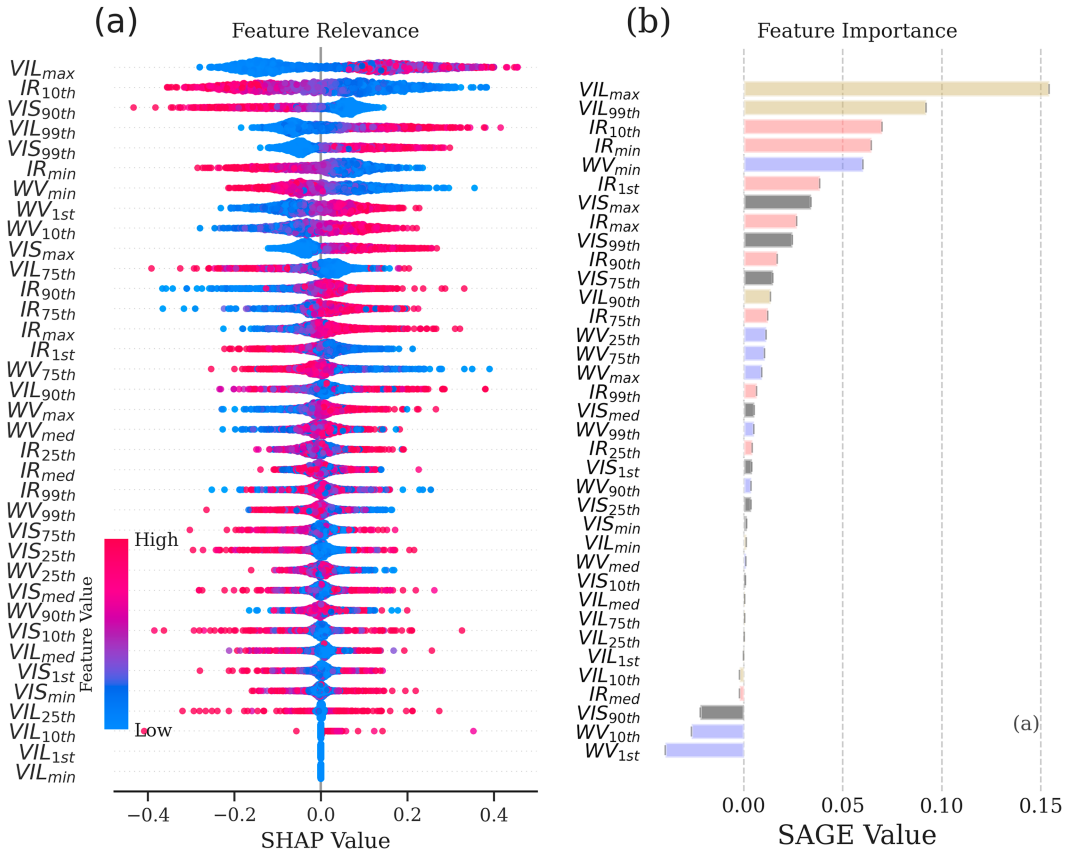

FIG. 6. As in Fig. 4, but for the lightning dataset. VIL features are in gold, IR in red, WV in blue, and VIS in black.

features (e.g., 10-m divergence), but none of those features are ranked highly. Thus, the negative importance of downdraft speed does not appear to arise from compensatory features. One primary generation mechanism for severe convective winds is a strong downdraft (either a downburst or a supercell rear-flank downdraft), so one might expect this feature to be highly important. Figure 11 shows the learned relationships for the downdraft and 80-m wind speeds. For both features, the learned relationship matches the base rate trend, and for 80-m wind speeds between 30 and 40 kt (1 kt $\approx$ 0.51 m s^{-1}) and downdraft speeds between -7 and -6 m s^{-1}, the base rates are similar, but strong downdrafts (<-7 m s^{-1}) do have a lower base rate. We know that strong downdrafts are more common over the Great Plains (Romanic et al. 2022) and, unfortunately, we are more likely to miss severe wind reports in that region as well (Trapp et al. 2006). While we acknowledge that there are other mechanisms for generating severe near-surface winds, we hypothesize that missing reports may have lowered the base rate for stronger downdrafts and negatively impacted this feature's importance. Our analysis highlights the importance of understanding model/data deficiencies to avoid misinterpreting feature importance. Other studies such as Clare et al. (2022) already recognize the need to rely on intuition from physical theory when evaluating the trustworthiness of XAI methods. Being aware of the model deficiencies can identify areas for model improvements or could be presented to the end user. For example, we could inform forecasters that strong downdrafts do not explicitly translate to higher severe wind probabilities with the current model.

d. Using explainability to monitor an ML model

Until this point, we discussed how to use explainability to understand ML models that are known to perform well. However, we can also use explainability to debug models during development or monitor a model in operations. For example, imagine a scenario where the road surface model runs during winter. Let us assume the temperatures are near freezing over the northern CONUS and have been for a while, so frozen road surface probabilities should be near 100%. However, we find that output for northern Michigan is closer to 40%. Analyzing a SHAP waterfall plot shows how each feature contributes to a single prediction (Fig. 12). This plot displays the base rate, the final prediction, and how each feature "forces" the prediction away from the base rate (either positively or negatively). In this toy example case, the surface temperature lowers the probability of a frozen road surface by eight percentage points, which is unexpected. On further inspection, an alert user would see that the surface temperature is rather warm at 29°C (84°F); the issue is likely wrong units! During model development, it is possible to introduce unit mismatches;

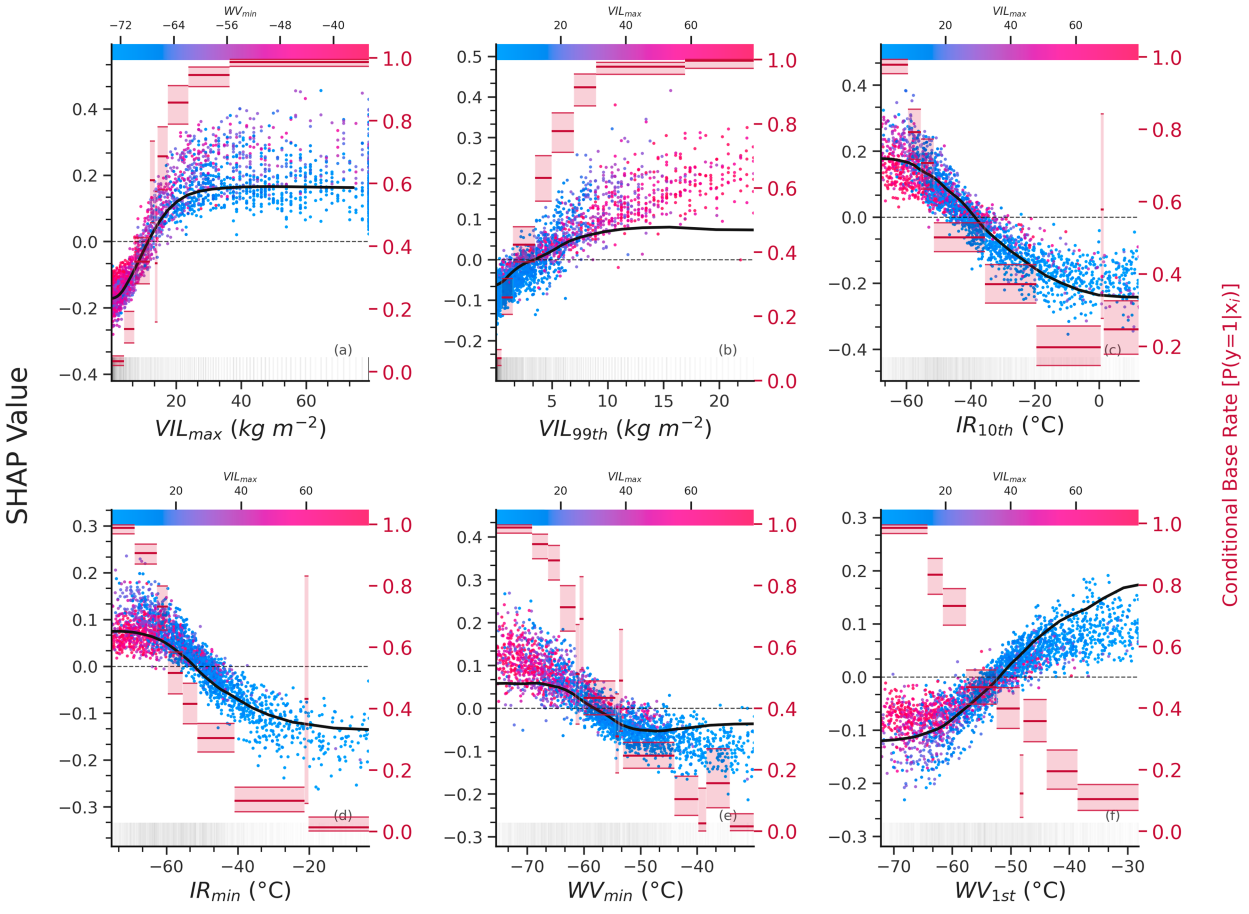

FIG. 7. As in Fig. 5, but for the lightning dataset.

the authors have used waterfall plots to diagnose unit errors for the quasi-operational ML models used in the WoFS. We can similarly use SAGE rankings to diagnose dataset errors (Fig. 13). The temperature variables (T_d, T_{sfc}, and T_{2m}) have near-zero importance, which is physically implausible and is also associated with the wrong units. This type of analysis is useful during the model development period to identify potential issues with the dataset.

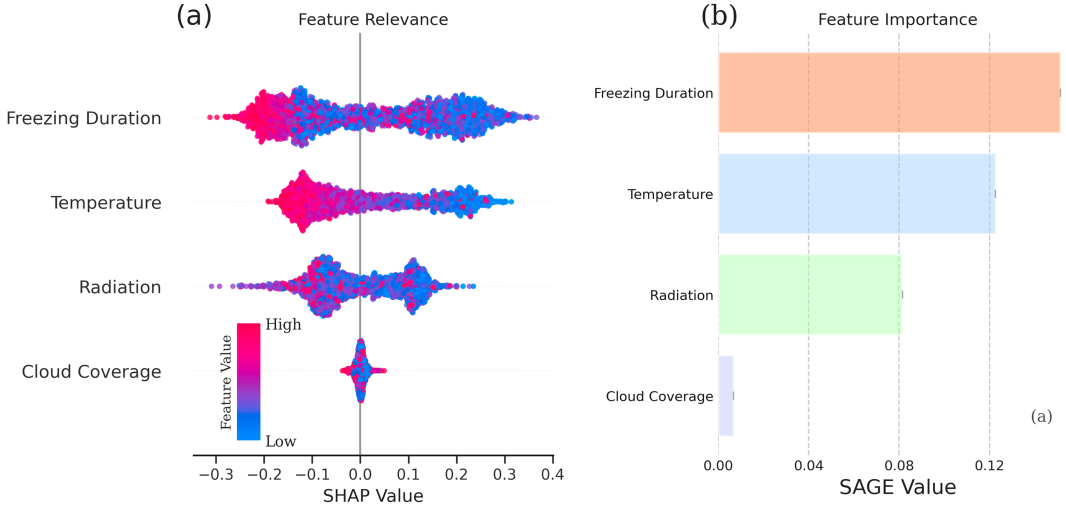

FIG. 8. As in Fig. 4, but for feature groups in the road surface dataset.

TABLE 3. Modified from Handler et al. (2020). Input features to the random forest for the road surface dataset. Terms are listed as follows: surface (SFC), radiation flux (RF), solar flux (SF), heat flux (HF), longwave (LW), shortwave (SW), and cloud coverage (CC).

Temperature	Radiation	CC	Freezing duration	Other
SFC (T_{sfc})	Incoming SW RF (S)	Total (C_{total})	Hours $T_{2\text{m}} \leq 0$°C	SFC friction velocity (V_{fric})
2-m ($T_{2\text{m}}$)	Visible downward SF (V_{bd})	Low (C_{low})	Hours $T_{2\text{m}} \geq 0$°C	SFC roughness (S_R)
2-m dewpoint (T_d)	Upward LW RF ($\lambda_{\uparrow}$)	Mid (C_{mid})	Hours $T_{\text{sfc}} \leq 0$°C	10-m wind speed ($U_{10\text{m}}$)
$\Delta T_{2\text{m}} - T_{\text{sfc}}$ (HRRR$_{dT}$)	SFC latent HF (L_{hf})	High (C_{mid})	Hours $T_{\text{sfc}} \geq 0$°C	Urban HRRR land classification
	SFC sensible HF (S_{hf})		Absolute distance from 10 Jan	Rural HRRR land classification
	Visible diffuse downward SF ($\lambda_{\downarrow}$)			
	Ground flux (G)			
	Simulated brightness temp (T_{irbt})			
	$\Delta S - \lambda_{\uparrow}$			
	$\Delta S - \lambda_{\downarrow}$			
	$\Delta G - S_{\text{hf}}$			
	$\Delta G - S_{\text{hf}}$			

6. Summary

Motivated by the increasing interest in explaining machine learning models, this study synthesizes recent research on explainability methods for traditional ML models. Our goal is to provide a tutorial for using these methods to accelerate the adoption of explainability methods within atmospheric and other environmental sciences. This includes distinguishing explainability from interpretability (Fig. 1), local versus global explainability, and feature importance versus feature relevance. We demonstrate visualizations of the different explainability methods, how to interpret them, and provide a comprehensive Python package (Flora and Handler 2022) to enable other researchers to use these methods. The explainability methods covered in this tutorial are largely Shapely based as these methods unify many preexisting methods. In fact, all removal-based explanation methods are implicitly tied to cooperative game theory, the foundation of Shapley values. Local attributions methods like LIME (Ribeiro et al. 2016) and tree interpreter (Saabas 2014) are unified by SHAP, while global feature importance methods like single-pass and multipass permutation importance (McGovern et al. 2019) and grouped and relative feature importance (Au et al. 2021; König et al. 2020) are unified by SAGE.

To demonstrate the SHAP and SAGE methods, we applied them to three disparate datasets: a convection-allowing model dataset for severe weather prediction, a nowcasting dataset for subfreezing road surface prediction, and satellite-based data for lightning prediction. We demonstrated a general approach to explaining an ML model. This process includes

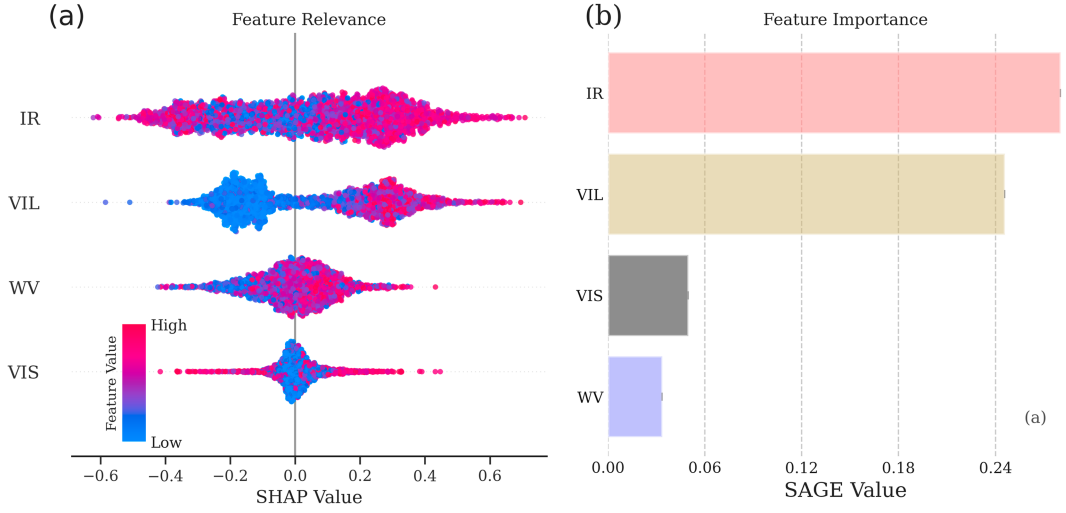

FIG. 9. As in Fig. 4, but for feature groups in the lightning dataset.

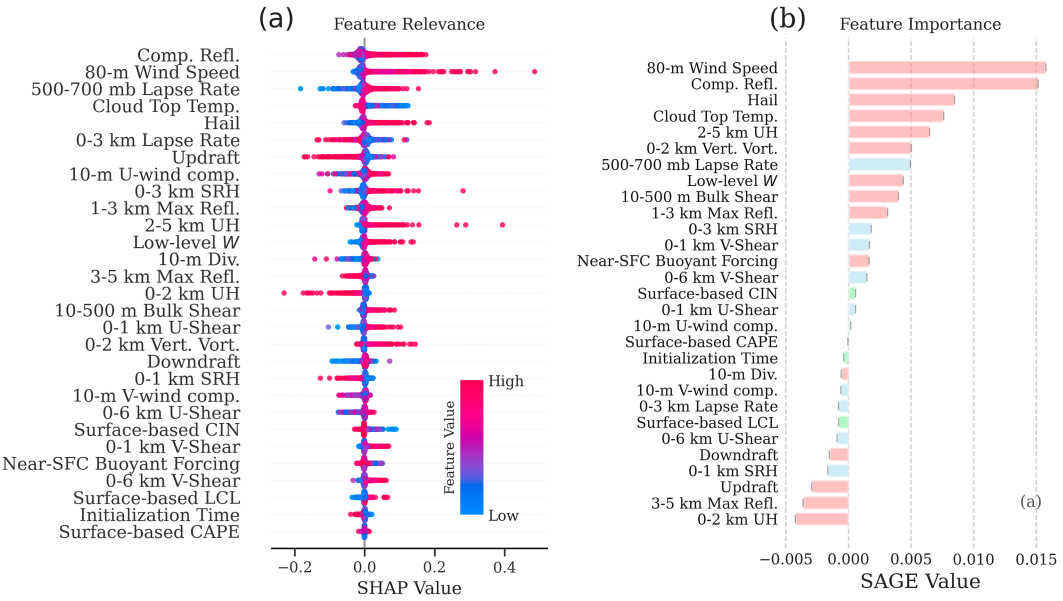

FIG. 10. As in Fig. 4, but the severe wind dataset.

examining discrepancies (or agreements) between the most relevant and important features and the learned relationships. Feature relevance measures the contribution of a feature to the model's prediction, while feature importance measures the contribution to the model's performance. By exploring both, we can identify the strengths and weaknesses of the model, which is a first step toward building trust in the model.

Next, we demonstrated how feature correlations can negatively impact feature relevance and importance. For example, feature correlations can result in learned feature–target relationships having the wrong sign due to a compensating effect for models like logistic regression and neural networks. When strongly correlated features are present, SAGE assigns negative importance to these features as their learned relationship

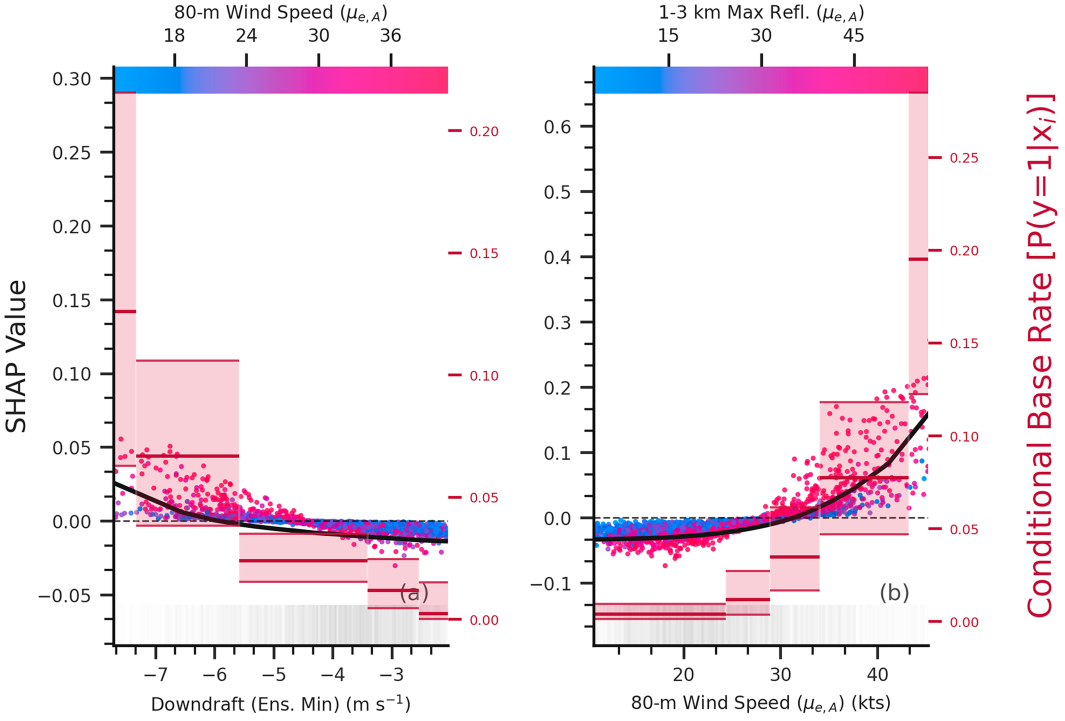

FIG. 11. As in Fig. 5, but the severe wind dataset.

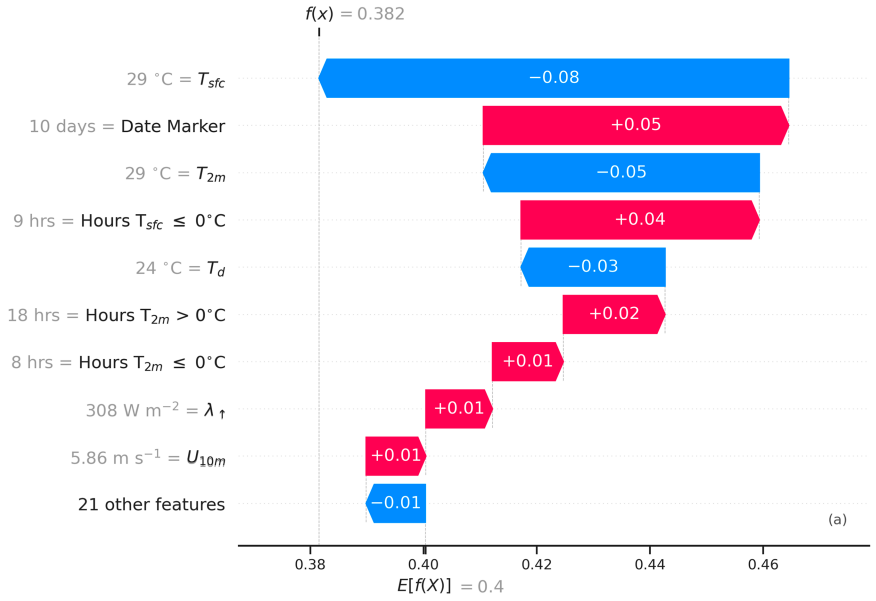

FIG. 12. A waterfall plot where red arrows indicate positive SHAP values and blue arrows indicate negative SHAP values. Contributions are ranked by their absolute magnitude, and feature values are provided on the left-hand side. The $E[f(x)]$ is the bias term from Eq. (2) and will equal the base rate.

degrades the model performance if the features they are correlated with are absent.

We demonstrate grouped feature relevance and feature importance as methods to mitigate the impacts of correlated features. Rather than analyzing features individually, we can evaluate their collective impact on the model prediction or performance. We showed how to use explainability methods to diagnose model weaknesses. We found that downdraft speed had low importance in the severe wind model despite strong downdrafts being associated with a higher likelihood of severe winds. The learned relationship was consistent with the severe wind base rates for strong downdraft speeds. However, we suspect these base rates are underestimated as strong downdrafts are more common over the Great Plains, where underreporting of severe wind is more likely. Last, we demonstrated how explainability methods can be used to monitor models in real time to identify spurious predictions (e.g., due to incorrect feature units).

There are some caveats to the ML model explainability approach demonstrated in this paper. First, though Shapely values are a powerful and theoretically sound method, they have limitations. As model complexity increases, feature contributions to the model prediction and performance may be nonadditive and, therefore, could provide a poor explanation of model behavior (Gosiewska and Biecek 2019; Kumar et al. 2020). Second, though Shapley values can provide critical insights into model behavior, translating the values into an end-user explanation is not straightforward (Kumar et al. 2020). Kumar et al. (2020) provides multiple possible approaches to translate Shapley-based results into a human-centric explanation, but further work is needed in this area. Third, in addition to the important features and their first-order effects, it is crucial to consider feature interactions. Feature interactions describe how two or more predictors work together to influence the performance of the ML model. However, measuring feature interactions can be challenging, and more work must be done to evaluate them accurately. One comprehensive set of metrics for assessing feature interactions, developed by Friedman and Popescu (2008), includes various statistics that describe the departure of second-order effects from the additive effect between two predictors. There are also SHAP-based methods for computing interaction values (Lundberg et al. 2018). However, feature interaction explainability methods can be computationally intractable for more than two features, and distinguishing the significance of second or higher-order interactions from noise or correlations can be prohibitively difficult.

Explaining an ML model is an involved process. Properly explaining any ML model requires a solid understanding of the underlying algorithm, a thorough knowledge of the data, and the limitations of traditional supervised ML approaches. Understanding the dataset's features, distribution, and relationships with the target variable is crucial. Furthermore, appreciating the limits of a supervised ML approach is indispensable. Supervised learning assumes a good signal-to-noise ratio with minimal label uncertainties, which is unrealistic for atmospheric science datasets. And the relevant and important features of the models generated from these data may or may not reflect the actual data-generating process. Explainability methods allow us to determine if the ML model learned meaningful relationships and identify possible model deficiencies. By recognizing these deficiencies, we can correct them or, at the very least, present them to the end user as a means to establish trust. By thoroughly explaining the model, including its strengths and weaknesses, we can help users understand

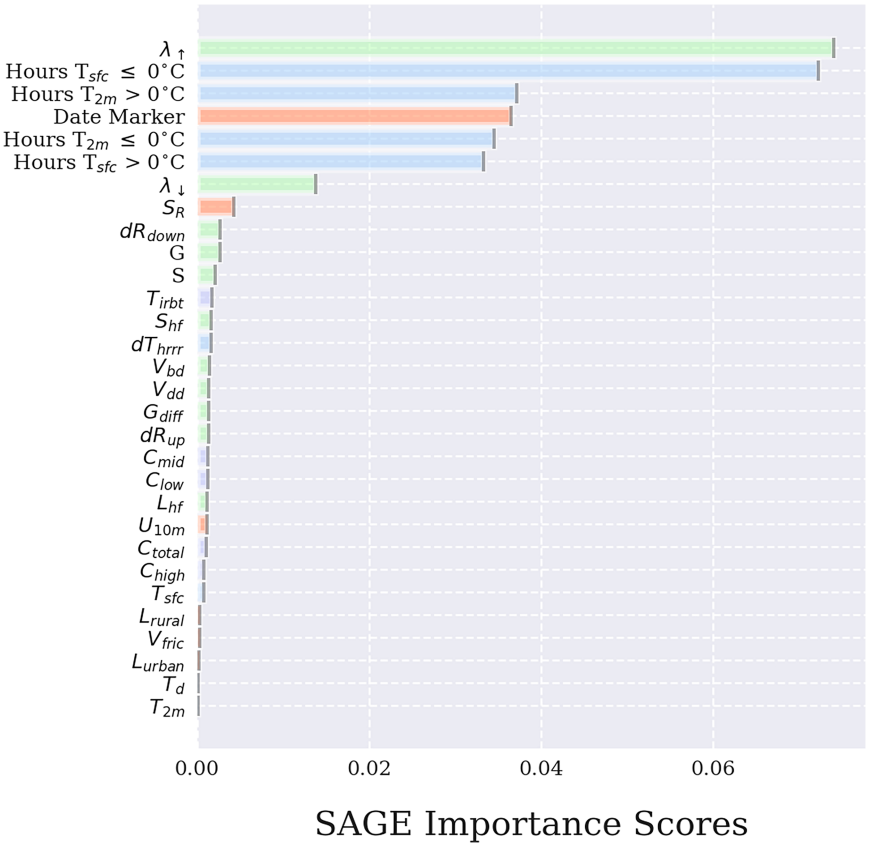

FIG. 13. SAGE-based feature rankings for bad road surface data.

when and why they should trust the model's predictions, fostering a more meaningful and productive relationship between the users and the AI tools they use.

Acknowledgments. Funding was provided by NOAA/Office of Oceanic and Atmospheric Research under NOAA–University of Oklahoma Cooperative Agreement NA21OAR4320204, U.S. Department of Commerce. The authors thank Eric Loken for informally reviewing an early version of the manuscript. We also acknowledge the team members responsible for generating the experimental WoFS output, which includes Kent Knopfmeier, Brian Matilla, Thomas Jones, Patrick Skinner, Brett Roberts, and Nusrat Yussouf. This material is also based upon work supported by the National Science Foundation under Grant ICER-2019758. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Data availability statement. To accelerate the adoption of machine learning explainability within meteorology, we created a repository with a code tutorial for the topics discussed in this paper. The latest version of the code is found at https://github.com/monte-flora/explain_tutorial. We have also created the scikit-explain Python package (Flora and Handler 2022), which computes and visualizes several explainability methods (including methods not discussed in this paper). All three datasets and ML models used in this study are available at https://doi.org/10.5281/zenodo.8184201. Code for easily downloading these data is available in the GitHub repository.

REFERENCES

Adadi, A., and M. Berrada, 2018: Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, **6**, 52 138–52 160, https://doi.org/10.1109/ACCESS.2018.2870052.

Adams-Selin, R. D., and C. L. Ziegler, 2016: Forecasting hail using a one-dimensional hail growth model within WRF. *Mon. Wea. Rev.*, **144**, 4919–4939, https://doi.org/10.1175/MWR-D-16-0027.1.

——, A. J. Clark, C. J. Melick, S. R. Dembek, I. L. Jirak, and C. L. Ziegler, 2019: Evolution of WRF-HAILCAST during the 2014–16 NOAA/hazardous weather testbed spring forecasting experiments. *Wea. Forecasting*, **34**, 61–79, https://doi.org/10.1175/WAF-D-18-0024.1.

Apley, D. W., and J. Zhu, 2016: Visualizing the effects of predictor variables in black box supervised learning models. arXiv, 1612.08468v2, https://doi.org/10.48550/arXiv.1612.08468.

Au, Q., J. Herbinger, C. Stachl, B. Bischl, and G. Casalicchio, 2021: Grouped feature importance and combined features

effect plot. arXiv, 2104.11688v1, https://doi.org/10.1007/s10618-022-00840-5.

Breiman, L., 2001: Random forests. *Mach. Learn.*, **45**, 5–32, https://doi.org/10.1023/A:1010933404324.

Brenowitz, N. D., T. Beucler, M. Pritchard, and C. S. Bretherton, 2020: Interpreting and stabilizing machine-learning parametrizations of convection. *J. Atmos. Sci.*, **77**, 4357–4375, https://doi.org/10.1175/JAS-D-20-0082.1.

Chase, R. J., D. R. Harrison, A. Burke, G. M. Lackmann, and A. McGovern, 2022: A machine learning tutorial for operational meteorology. Part I: Traditional machine learning. *Wea. Forecasting*, **37**, 1509–1529, https://doi.org/10.1175/WAF-D-22-0070.1.

——, ——, G. M. Lackmann, and A. McGovern, 2023: A machine learning tutorial for operational meteorology. Part II: Neural networks and deep learning. *Wea. Forecasting*, **38**, 1271–1293, https://doi.org/10.1175/WAF-D-22-0187.1.

Cintineo, J. L., M. J. Pavolonis, J. M. Sieglaff, L. Cronce, and J. Brunner, 2020: NOAA ProbSevere v2.0—ProbHail, ProbWind, and ProbTor. *Wea. Forecasting*, **35**, 1523–1543, https://doi.org/10.1175/WAF-D-19-0242.1.

Clare, M. C. A., M. Sonnewald, R. Lguensat, J. Deshayes, and V. Balaji, 2022: Explainable artificial intelligence for Bayesian neural networks: Toward trustworthy predictions of ocean dynamics. *J. Adv. Model. Earth Syst.*, **14**, e2022MS003162, https://doi.org/10.1029/2022MS003162.

Covert, I., S. Lundberg, and S.-I. Lee, 2020a: Explaining by removing: A unified framework for model explanation. arXiv, 2011.14878v2, https://doi.org/10.48550/arxiv.2011.14878.

——, ——, and ——, 2020b: Understanding global feature contributions with additive importance measures. arXiv, 2004.00668v2, https://doi.org/10.48550/arXiv.2004.00668.

Crevier, L.-P., and Y. Delage, 2001: METRo: A new model for road-condition forecasting in Canada. *J. Appl. Meteor.*, **40**, 2026–2037, https://doi.org/10.1175/1520-0450(2001)040<2026:MANMFR>2.0.CO;2.

Doshi-Velez, F., and B. Kim, 2017: Towards a rigorous science of interpretable machine learning. arXiv, 1702.08608v2, https://doi.org/10.48550/arXiv.1702.08608.

Fisher, A., C. Rudin, and F. Dominici, 2018: All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. arXiv, 1801.01489v5, https://doi.org/10.48550/arXiv.1801.01489.

Flora, M. L., 2020: Storm-scale ensemble-based severe weather guidance: Development of an object-based verification framework and applications of machine learning. Ph.D. thesis, University of Oklahoma, 193 pp., https://shareok.org/handle/11244/326598.

——, and S. Handler, 2022: Scikit-explain. GitHub, https://github.com/monte-flora/scikit-explain.

——, P. S. Skinner, C. K. Potvin, A. E. Reinhart, T. A. Jones, N. Yussouf, and K. H. Knopfmeier, 2019: Object-based verification of short-term, storm-scale probabilistic mesocyclone guidance from an experimental Warn-on-Forecast system. *Wea. Forecasting*, **34**, 1721–1739, https://doi.org/10.1175/WAF-D-19-0094.1.

——, C. K. Potvin, P. S. Skinner, S. Handler, and A. McGovern, 2021: Using machine learning to generate storm-scale probabilistic guidance of severe weather hazards in the Warn-on-Forecast system. *Mon. Wea. Rev.*, **149**, 1535–1557, https://doi.org/10.1175/MWR-D-20-0194.1.

Friedman, J. H., 2001: Greedy function approximation: A gradient boosting machine. *Ann. Stat.*, **29**, 1189–1232, https://doi.org/10.1214/aos/1013203451.

——, and B. E. Popescu, 2008: Predictive learning via rule ensembles. *Ann. Appl. Stat.*, **2**, 916–954, https://doi.org/10.1214/07-AOAS148.

Gagne, D. J., II, A. McGovern, S. E. Haupt, R. A. Sobash, J. K. Williams, and M. Xue, 2017: Storm-based probabilistic hail forecasting with machine learning applied to convection-allowing ensembles. *Wea. Forecasting*, **32**, 1819–1840, https://doi.org/10.1175/WAF-D-17-0010.1.

——, S. E. Haupt, D. W. Nychka, and G. Thompson, 2019: Interpretable deep learning for spatial analysis of severe hailstorms. *Mon. Wea. Rev.*, **147**, 2827–2845, https://doi.org/10.1175/MWR-D-18-0316.1.

Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, 2018: Explaining explanations: An overview of interpretability of machine learning. arXiv, 1806.00069v3, https://doi.org/10.48550/arXiv.1806.00069.

Gosiewska, A., and P. Biecek, 2019: iBreakDown: Uncertainty of model explanations for non-additive predictive models. arXiv, 1903.11420, https://arxiv.org/abs/1903.11420v1.

Greenwell, B. M., B. C. Boehmke, and A. J. McCarthy, 2018: A simple and effective model-based variable importance measure. arXiv, 1805.04755v1, https://doi.org/10.48550/arXiv.1805.04755.

Gregorich, M., S. Strohmaier, D. Dunkler, and G. Heinze, 2021: Regression with highly correlated predictors: Variable omission is not the solution. *Int. J. Environ. Res. Public Health*, **18**, 4259, https://doi.org/10.3390/ijerph18084259.

Gregorutti, B., B. Michel, and P. Saint-Pierre, 2015: Grouped variable importance with random forests and application to multiple functional data analysis. *Comput. Stat. Data Anal.*, **90**, 15–35, https://doi.org/10.1016/j.csda.2015.04.002.

——, ——, and ——, 2017: Correlation and variable importance in random forests. *Stat. Comput.*, **27**, 659–678, https://doi.org/10.1007/s11222-016-9646-1.

Hafner, D., 2022: Bayesian-histograms. GitHub, https://github.com/dionhaefner/bayesian-histograms.

Hamidi, Y., L. Raynaud, L. Rottner, and P. Arbogast, 2020: Texture-based classification of high-resolution precipitation forecasts with machine-learning methods. *Quart. J. Roy. Meteor. Soc.*, **146**, 3014–3028, https://doi.org/10.1002/qj.3823.

Handler, S. L., H. D. Reeves, and A. McGovern, 2020: Development of a probabilistic subfreezing road temperature nowcast and forecast using machine learning. *Wea. Forecasting*, **35**, 1845–1863, https://doi.org/10.1175/WAF-D-19-0159.1.

Herman, G. R., and R. S. Schumacher, 2018: "Dendrology" in numerical weather prediction: What random forests and logistic regression tell us about forecasting extreme precipitation. *Mon. Wea. Rev.*, **146**, 1785–1812, https://doi.org/10.1175/MWR-D-17-0307.1.

Hernández, J. D. R., Ó. J. Mesa, and U. Lall, 2020: ENSO dynamics, trends, and prediction using machine learning. *Wea. Forecasting*, **35**, 2061–2081, https://doi.org/10.1175/WAF-D-20-0031.1.

Hoffman, R. R., D. S. LaDue, H. M. Mogil, P. J. Roebber, and J. G. Trafton, 2017: *Minding the Weather: How Expert Forecasters Think*. MIT Press, 488 pp.

Hooker, G., L. Mentch, and S. Zhou, 2021: Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. *Stat. Comput.*, **31**, 82, https://doi.org/10.1007/s11222-021-10057-z.

Jacovi, A., A. Marasović, T. Miller, and Y. Goldberg, 2021: Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in AI. *FAccT'21 Proc. 2021 ACM Conf. on Fairness, Accountability, and Transparency*, Online, Association for Computing Machinery, 624–635, https://doi.org/10.1145/3442188.3445923.

Jergensen, G. E., A. McGovern, R. Lagerquist, and T. Smith, 2020: Classifying convective storms using machine learning. *Wea. Forecasting*, **35**, 537–559, https://doi.org/10.1175/WAF-D-19-0170.1.

Jones, T. A., K. Knopfmeier, D. Wheatley, G. Creager, P. Minnis, and R. Palikonda, 2016: Storm-scale data assimilation and ensemble forecasting with the NSSL experimental Warn-on-Forecast system. Part II: Combined radar and satellite data experiments. *Wea. Forecasting*, **31**, 297–327, https://doi.org/10.1175/WAF-D-15-0107.1.

——, and Coauthors, 2020: Assimilation of *GOES-16* radiances and retrievals into the Warn-on-Forecast system. *Mon. Wea. Rev.*, **148**, 1829–1859, https://doi.org/10.1175/MWR-D-19-0379.1.

Karstens, C. D., and Coauthors, 2018: Development of a human–machine mix for forecasting severe convective events. *Wea. Forecasting*, **33**, 715–737, https://doi.org/10.1175/WAF-D-17-0188.1.

Kim, B., R. Khanna, and O. O. Koyejo, 2016: Examples are not enough, learn to criticize! Criticism for interpretability. *NIPS'16: Proc. 30th Int. Conf. on Neural Information Processing Systems*, Barcelona, Spain, Association for Computing Machinery, 2288–2296, https://dl.acm.org/doi/10.5555/3157096.3157352.

König, G., C. Molnar, B. Bischl, and M. Grosse-Wentrup, 2020: Relative feature importance. arXiv, 2007.08283v1, https://doi.org/10.48550/arXiv.2007.08283.

Kuhn, M., and K. Johnson, 2019: *Feature Engineering and Selection: A Practical Approach for Predictive Models*. 1st ed. Chapman and Hall/CRC, 310 pp., https://doi.org/10.1201/9781315108230.

Kumar, I. E., S. Venkatasubramanian, C. Scheidegger, and S. A. Friedler, 2020: Problems with Shapley-value-based explanations as feature importance measures. arXiv, 2002.11097v2, https://doi.org/10.48550/arXiv.2002.11097.

Kumler-Bonfanti, C., J. Stewart, D. Hall, and M. Govett, 2020: Tropical and extratropical cyclone detection using deep learning. *J. Appl. Meteor. Climatol.*, **59**, 1971–1985, https://doi.org/10.1175/JAMC-D-20-0117.1.

Lagerquist, R., A. McGovern, and T. Smith, 2017: Machine learning for real-time prediction of damaging straight-line convective wind. *Wea. Forecasting*, **32**, 2175–2193, https://doi.org/10.1175/WAF-D-17-0038.1.

——, ——, C. R. Homeyer, D. J. Gagne II, and T. Smith, 2020: Deep learning on three-dimensional multiscale data for next-hour tornado prediction. *Mon. Wea. Rev.*, **148**, 2837–2861, https://doi.org/10.1175/MWR-D-19-0372.1.

Lakshmanan, V., C. Karstens, J. Krause, K. Elmore, A. Ryzhkov, and S. Berkseth, 2015: Which polarimetric variables are important for weather/no-weather discrimination? *J. Atmos. Oceanic Technol.*, **32**, 1209–1223, https://doi.org/10.1175/JTECH-D-13-00205.1.

Linardatos, P., V. Papastefanopoulos, and S. Kotsiantis, 2020: Explainable AI: A review of machine learning interpretability methods. *Entropy*, **23**, 18, https://doi.org/10.3390/e23010018.

Lipton, Z. C., 2016: The mythos of model interpretability. arXiv, 1606.03490v3, https://doi.org/10.48550/arXiv.1606.03490.

Loken, E. D., A. J. Clark, and A. McGovern, 2022: Comparing and interpreting differently designed random forests for next-day severe weather hazard prediction. *Wea. Forecasting*, **37**, 871–899, https://doi.org/10.1175/WAF-D-21-0138.1.

López, S., and M. Saboya, 2009: On the relationship between Shapley and Owen values. *Cent. Eur. J. Oper. Res.*, **17**, 415–423, https://doi.org/10.1007/s10100-009-0100-8.

Lundberg, S. M., and S.-I. Lee, 2017: A unified approach to interpreting model predictions. *NIPS'17: Proc. 31st Int. Conf. on Neural Information Processing Systems*, Long Beach, CA, Association for Computing Machinery, 4768–4777, https://dl.acm.org/doi/10.5555/3295222.3295230.

——, G. G. Erion, and S.-I. Lee, 2018: Consistent individualized feature attribution for tree ensembles. arXiv, 1802.03888v3, https://doi.org/10.48550/arXiv.1802.03888.

——, and Coauthors, 2020: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, **2**, 56–67, https://doi.org/doi:10.1038/s42256-019-0138-9.

Mamalakis, A., E. A. Barnes, and I. Ebert-Uphoff, 2022: Investigating the fidelity of explainable artificial intelligence methods for applications of convolutional neural networks in geoscience. *Artif. Intell. Earth Syst.*, **1**, e220012, https://doi.org/10.1175/AIES-D-22-0012.1.

——, ——, and ——, 2023: Carefully choose the baseline: Lessons learned from applying XAI attribution methods for regression tasks in geoscience. *Artif. Intell. Earth Syst.*, **2**, e220058, https://doi.org/10.1175/AIES-D-22-0058.1.

McGovern, A., R. Lagerquist, D. J. Gagne II, G. E. Jergensen, K. L. Elmore, C. R. Homeyer, and T. Smith, 2019: Making the black box more transparent: Understanding the physical implications of machine learning. *Bull. Amer. Meteor. Soc.*, **100**, 2175–2199, https://doi.org/10.1175/BAMS-D-18-0195.1.

——, R. J. Chase, M. Flora, D. J. Gagne II, R. Lagerquist, C. K. Potvin, N. Snook, and E. Loken, 2023: A review of machine learning for convective weather. *Artif. Intell. Earth Syst.*, **2**, e220077, https://doi.org/10.1175/AIES-D-22-0077.1.

Mecikalski, J. R., T. N. Sandmæl, E. M. Murillo, C. R. Homeyer, K. M. Bedka, J. M. Apke, and C. P. Jewett, 2021: A random-forest model to assess predictor importance and nowcast severe storms using high-resolution radar–GOES satellite–lightning observations. *Mon. Wea. Rev.*, **149**, 1725–1746, https://doi.org/10.1175/MWR-D-19-0274.1.

Miller, T., 2019: Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, **267**, 1–38, https://doi.org/10.1016/j.artint.2018.07.007.

Minokhin, I., C. G. Fletcher, and A. Brenning, 2017: Forecasting northern polar stratospheric variability with competing statistical learning models. *Quart. J. Roy. Meteor. Soc.*, **143**, 1816–1827, https://doi.org/10.1002/qj.3043.

Molnar, C., 2020: *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Bookdown, 320 pp.

——, G. Casalicchio, and B. Bischl, 2019: Quantifying model complexity via functional decomposition for better post-hoc interpretability. arXiv, 1904.03867v2, https://doi.org/10.48550/arXiv.1904.03867.

——, ——, and ——, 2020a: Interpretable machine learning—A brief history, state-of-the-art and challenges. arXiv, 2010.09337v1, https://doi.org/10.48550/arXiv.2010.09337.

——, and Coauthors, 2020b: *Limitations of Interpretable Machine Learning Methods*. Bookdown, https://slds-lmu.github.io/iml_methods_limitations/.

——, and Coauthors, 2021: General pitfalls of model-agnostic interpretation methods for machine learning models. arXiv, 2007.04131v2, https://doi.org/10.48550/arXiv.2007.04131.

Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, 2019: Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci. USA*, **116**, 22 071–22 080, https://doi.org/10.1073/pnas.1900654116.

Murphy, A. H., 1993: What is a good forecast? An essay on the nature of goodness in weather forecasting. *Wea. Forecasting*, **8**, 281–293, https://doi.org/10.1175/1520-0434(1993)008<0281:WIAGFA>2.0.CO;2.

Oh, S., 2019: Feature interaction in terms of prediction performance. *Appl. Sci.*, **9**, 5191, https://doi.org/10.3390/app9235191.

Owen, G., 1977: Values of games with a priori unions. *Mathematical Economics and Game Theory*, R. Henn and O. Moeschlin, Eds., Lecture Notes in Economics and Mathematical Systems, Vol. 141, Springer, 76–88.

Pedregosa, F., and Coauthors, 2011: Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Rasp, S., and S. Lerch, 2018: Neural networks for postprocessing ensemble weather forecasts. *Mon. Wea. Rev.*, **146**, 3885–3900, https://doi.org/10.1175/MWR-D-18-0187.1.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: Model-agnostic interpretability of machine learning. arXiv, 1606.05386v1, https://doi.org/10.48550/arXiv.1606.05386.

Romanic, D., M. Taszarek, and H. Brooks, 2022: Convective environments leading to microburst, macroburst and downburst events across the United States. *Wea. Climate Extremes*, **37**, 100474, https://doi.org/10.1016/j.wace.2022.100474.

Rudin, C., 2018: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. arXiv, 1811.10154v3, https://doi.org/10.48550/arXiv.1811.10154.

——, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, 2021: Interpretable machine learning: Fundamental principles and 10 grand challenges. arXiv, 2103.11251v2, https://doi.org/10.48550/arXiv.2103.11251.

Saabas, A., 2014: Interpreting random forests. Diving into Data, https://blog.datadive.net/interpreting-random-forests/.

Shapley, L. S., 1953: A value for n-person games. *Contributions to the Theory of Games*, H. Kuhn and A. Tucker, Eds., Princeton University Press, 307–317, https://doi.org/10.1515/9781400881970-018.

Shield, S. A., and A. L. Houston, 2022: Diagnosing supercell environments: A machine learning approach. *Wea. Forecasting*, **37**, 771–785, https://doi.org/10.1175/WAF-D-21-0098.1.

Strobl, C., A.-L. Boulesteix, A. Zeileis, and T. Hothorn, 2007: Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinf.*, **8**, 25, https://doi.org/10.1186/1471-2105-8-25.

——, ——, T. Kneib, T. Augustin, and A. Zeileis, 2008: Conditional variable importance for random forests. *BMC Bioinf.*, **9**, 307, https://doi.org/10.1186/1471-2105-9-307.

Trapp, R. J., D. M. Wheatley, N. T. Atkins, R. W. Przybylinski, and R. Wolf, 2006: Buyer beware: Some words of caution on the use of severe wind reports in postevent assessment and research. *Wea. Forecasting*, **21**, 408–415, https://doi.org/10.1175/WAF925.1.

van der Laan, M. J., 2006: Statistical inference for variable importance. *Int. J. Biostat.*, **2**, 2, https://doi.org/10.2202/1557-4679.1008.

van Lent, M., W. Fisher, and M. Mancuso, 2004: An explainable artificial intelligence system for small-unit tactical behavior. *IAAI'04 Proc. 16th Conf. on Innovative Applications of Artificial Intelligence*, San Jose, CA, Association for Computing Machinery, 900–907, https://dl.acm.org/doi/10.5555/1597321.1597342.

Veillette, M. S., S. Samsi, C. J. Mattioli, and H. Larochelle, 2020: SEVIR: A storm event imagery dataset for deep learning applications in radar and satellite meteorology. *NIPS'20: Proc. 34th Int. Conf. on Neural Information Processing Systems*, Vancouver, BC, Canada, Association for Computing Machinery, 22 009–22 019, https://dl.acm.org/doi/10.5555/3495724.3497570.

Wheatley, D. M., K. H. Knopfmeier, T. A. Jones, and G. J. Creager, 2015: Storm-scale data assimilation and ensemble forecasting with the NSSL experimental Warn-on-Forecast system. Part I: Radar data experiments. *Wea. Forecasting*, **30**, 1795–1817, https://doi.org/10.1175/WAF-D-15-0043.1.

Wright, M. N., A. Ziegler, and I. R. König, 2016: Do little interactions get lost in dark random forests? *BMC Bioinf.*, **17**, 145, https://doi.org/10.1186/s12859-016-0995-8.

Young, H. P., 1985: Monotonic solutions of cooperative games. *Int. J. Game Theory*, **14**, 65–72, https://doi.org/10.1007/BF01769885.