Pattern Recognition 160 (2025) 111171

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Explaining predictive uncertainty by exposing second-order effects

Florian Bley [a,b], Sebastian Lapuschkin [c], Wojciech Samek [a,b,c], Grégoire Montavon [d,b,*]

[a] Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Marchstr. 23, Berlin 10587, Germany
[b] BIFOLD – Berlin Institute for the Foundations of Learning and Data, Ernst-Reuter Platz 7, Berlin 10587, Germany
[c] Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Salzufer 15/16, Berlin 10587, Germany
[d] Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 14, Berlin 14195, Germany

ARTICLE INFO

Keywords:
Explainable AI
Predictive uncertainty
Ensemble models
Second-order attribution

ABSTRACT

Explainable AI has brought transparency to complex ML black boxes, enabling us, in particular, to identify which features these models use to make predictions. So far, the question of how to explain predictive uncertainty, i.e., why a model 'doubts', has been scarcely studied. Our investigation reveals that predictive uncertainty is dominated by *second-order effects*, involving single features or product interactions between them. We contribute a new method for explaining predictive uncertainty based on these second-order effects. Computationally, our method reduces to a simple covariance computation over a collection of first-order explanations. Our method is generally applicable, allowing for turning common attribution techniques (LRP, Gradient × Input, etc.) into powerful second-order uncertainty explainers, which we call CovLRP, CovGI, etc. The accuracy of the explanations our method produces is demonstrated through systematic quantitative evaluations, and the overall usefulness of our method is demonstrated through two practical showcases.

1. Introduction

As deep learning methods make decisions in increasingly critical scenarios, measuring the degree of certainty in a prediction becomes important to avoid costly mistakes made by AI systems. In the context of autonomous driving, a model anticipating high uncertainty may choose a safer route [1] or prompt the human driver to take control in dangerous situations. When performing reinforcement learning to train a steering agent, uncertainty estimates can allow the model to reduce speed in unfamiliar situations to avoid collisions [2]. In the context of diagnosing diseases (e.g. [3,4]), predictive uncertainty can be used to identify out-of-distribution tissue images [5] or help indicate in which cases consultation with an expert clinician is necessary [6]. In the field of finance, accurate real-time uncertainty predictions of portfolio valuations are vital in identifying risk-optimal investing strategies [7].

High predictive uncertainty often occurs in the context of complex machine learning tasks [8], where data scarcity prevents the heterogeneity of models in the ensemble from reaching a consensus on what the actual prediction should be [9]. A consensus is even less likely to be reached on which features should be the main drivers of the input–output relationship. However, such information is essential, e.g. when developing strategies to reduce the uncertainty of the ensemble and arrive at more confident and accurate models.

To elucidate the source of predictive uncertainty, we require tools to pinpoint the features contributing to it. Understanding a model's prediction in terms of input features has been tackled extensively within the field of Explainable AI [10,11] with many successes, e.g. in image classification [12,13]. In contrast, the explanation of predictive uncertainty has received little attention. So-called 'model-agnostic' explanation methods (e.g. [14–16]) are designed to explain potentially *any* machine learning function. Yet, the question arises whether these approaches are suitable to extract faithful explanations of predictive uncertainty or whether the explanation of uncertainty needs to be specifically addressed.

In this paper, we contribute new insights to the problem of explaining uncertainty for the common case where it is estimated as the *variance* over an ensemble of predictions. We find that uncertainty estimates are dominated by second-order effects, with these effects further categorizable as (1) single-feature quadratic contributions and (2) joint-feature bilinear contributions. Our investigation shows that classical explanation techniques do not make such a distinction, leading them to entangle these effects and perform unfaithfully.

We propose a novel second-order Explainable AI method for uncertainty prediction that accounts for these second-order effects. Our method derives from the identification of elementary product structures in the uncertainty function, based on which the second-order effects can be efficiently attributed to input features, in particular by disentangling single-feature from joint feature contributions. Our method

* Corresponding author at: Department of Mathematics and Computer Science, Freie Universität Berlin, Arnimallee 14, Berlin 14195, Germany.
E-mail address: gregoire.montavon@fu-berlin.de (G. Montavon).

https://doi.org/10.1016/j.patcog.2024.111171
Received 6 February 2024; Received in revised form 14 September 2024; Accepted 5 November 2024
Available online 14 November 2024

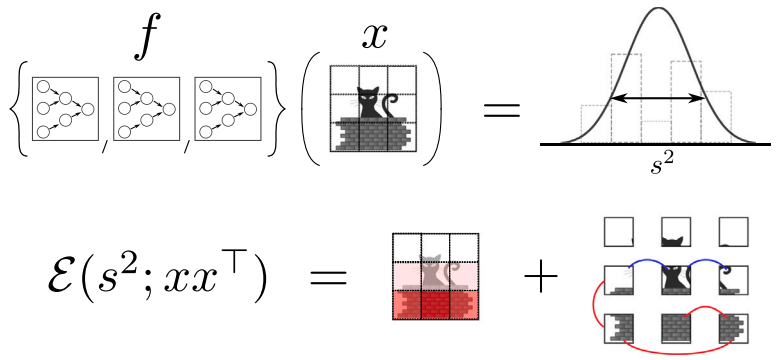

Fig. 1. Illustration of an ensemble model (top left), its prediction and predictive uncertainty (top right), and an illustrative cartoon example of our proposed predictive uncertainty explanation in terms of features and feature interactions (bottom). Red patches and connecting lines highlight features and feature interactions, adding to predictive uncertainty; blue connecting lines highlight feature interactions that decrease predictive uncertainty. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is applicable to general neural network structures, including highly nonlinear ones, and integrates with existing explanation frameworks such as LRP [12], Integrated Gradients [16] and Shapley Values [14]. Our method is shown schematically in Fig. 1 (the way our method operates is illustrated in more detail in Fig. 2).

Through systematic benchmark experiments, we show that our method yields substantially more faithful explanations of predictive uncertainty than a simple and naive application of classical Explainable AI methods to the uncertainty model. In addition, we demonstrate through showcase examples (1) how the identification of features contributing to uncertainty by our method can help to consolidate a dataset and produce a more accurate ML model as a result and (2) how our method helps to gain novel insights into a real-world dataset. Demo code and code to reproduce our main results are available at https://github.com/florianbley/XAI-2ndOrderUncertainty.

2. Related work

This section reviews the literature related to our work, focusing on three main categories: the development of methods for identifying uncertain ML model decisions, approaches to explain the uncertainty behind model predictions, and Explainable AI techniques that identify second- or higher-order effects in the decision behavior of general ML models.

2.1. Estimating uncertainty

Predictive uncertainty can be obtained directly by modeling the output distribution (e.g. [17,18]). However, this modeling approach applied in the context of deep neural networks often leads to overconfident uncertainty estimates [19]. Consequently, many uncertainty estimation techniques have been proposed to remedy overconfidence in deep models and produce more accurate assessments.

In [20], the authors established Monte Carlo dropout (MC dropout) at test time to estimate uncertainty as the variance of model predictions. Variants of the idea based on dropping blocks of features have also been proposed [21]. In [22], the authors pursued a similar approach, this time using stochastic Batch Normalization at test time. The authors of [23] treated model parameters as Gaussian distributed and used stochastic weight averaging (SWAG) to estimate the approximate posterior parameters. The authors of [24] employed ensembles of randomly initialized deep networks to estimate predictive variation. In [25], the authors consider a regression setting and assume, in addition to the conditional target distribution, a higher latent distribution yielding the distribution parameters of the former. They then show theoretically how training the latent distribution can be regarded as an evidence-gathering process, which yields for each prediction an amount of 'virtual' evidence from which the authors deduct uncertainty measures. For classification problems, Deep Prior Networks, as discussed in [26], estimate the parameters of a latent Dirichlet distribution for the conditional class probabilities in a similar way to allow for accurate predictive uncertainty measures.

2.2. Explaining uncertainty

While much research has focused on estimating the predictive uncertainty of deep models, there has been significantly less exploration into explaining predictive uncertainty, e.g. in terms of input features.

In [27], the authors performed a gradient-based Sensitivity Analysis of predictive uncertainty to explain which features contribute to uncertainty. The authors of [28] theoretically explored the approach of applying Shapley Values [14] to various uncertainty measures such as entropy and information gain and tested their approach in covariate shift and feature selection applications. In [29], the authors extended perturbation-based explainers to the entropy measures of predictive distributions. Specifically, they resampled feature values from a marginal feature value distribution to measure their effect on the predictive uncertainty measure. The authors of [30] modified the Integrated Gradients method by using gradient descent to move towards a low-uncertainty region near the input point, incorporating contextual information into the reference point. While all of these works propose a way to score input features according to their relevance, none of them address second-order effects. In particular, they do not disentangle the contributions of individual features from those of feature pairs, which our analysis and experiments show to be essential for high accuracy.

Finally, and in contrast to the above methods based on relevance scoring, the authors of [31] used *counterfactual explanations* to explain predictive uncertainty. Specifically, they trained a generative model and solved the optimization problem of finding a minimally altered sample in latent space while maximally reducing model uncertainty. While the authors could demonstrate good human interpretability of counterfactuals, their approach necessitates training a generative model. This adds additional complexity for the practitioner and renders explanations exposed to potential biases in the generator.

2.3. Higher-order explanations

Various methods have been proposed to extract explanations in terms of multiple interacting features. The Shapley Taylor Index [32] extends the Shapley Value approach to highlight the contribution of feature interactions to a prediction. Likewise, Integrated Hessian [33] is an extension of Integrated Gradients [16] that enables an explanation

in terms of feature interactions. The method operates by computing a double path integral of the predictive model's Hessian matrix towards the data point. BiLRP [34] is a second-order method that specifically addresses the explanation of product-type similarity models of pairs of data points. Predicted similarities can then be robustly attributed to pairs of input features associated to each data point. GNN-LRP [35] is a higher-order explanation method developed for Graph Neural Networks, which decomposes the model's prediction in terms of sequences of connected edges in the graph.—Our proposed approach differs from the methods above by specifically targeting the question of explaining uncertainty. In particular, our method explicitly identifies the second-order structure of uncertainty predictions, leading to explanations that are reliable and fast to compute.

3. Proposed method for explaining uncertainty

In this section, we derive our proposed second-order approach to explaining predictive uncertainty, specifically, attributing predictive uncertainty to the input features. As a starting point, we assume that the uncertainty estimate we want to explain is given by the variance over the predictions of an ensemble of M neural networks:

$$s^2 = \frac{1}{M}\sum_{m=1}^{M}\big(y_m - \bar{y}\big)^2 \tag{1}$$

with y_m the output of model m, and $\bar{y}$ the average prediction of the different models. This formulation is general enough to include any uncertainty quantification method relying on the variance of model predictions such as deep ensembles [24], MC dropout [20], MC batch normalization [22] or SWAG [23]. We observe that the predictive variance stated in Eq. (1) can be rewritten as a linear combination of prediction products, i.e.,

$$s^2 = \sum_{m,m'} b_{m,m'} \cdot y_m y_{m'}, \tag{2}$$

where $b_{m,m'} = \frac{1}{M}\cdot 1_{\{m=m'\}} - \frac{1}{M^2}$ are the coefficients of the linear combination with $1_{\{\cdot\}}$ the indicator function, and where $\sum_{m,m'}$ is a nesting of two sums, each sum running over all models in the ensemble.

We now focus on the problem of attributing the predictive variance s^2 to the input features. Denote by $\mathcal{E}(\cdot)$ the process of attributing what is given as an argument to the input features, as defined later. The linearity observed in Eq. (2) lets us reduce the problem of attributing the variance s^2 to:

$$\mathcal{E}\Big(\sum_{m,m'} b_{m,m'} \cdot y_m y_{m'}\Big) = \sum_{m,m'} b_{m,m'} \cdot \mathcal{E}\big(y_m y_{m'}\big). \tag{3}$$

In other words, the problem of attributing the predictive variance can be treated as solving simpler subproblems (attributing products of model outputs) and linearly combining the results.

Previous works, such as [34], have demonstrated that product structures are mathematically more naturally attributed to pairs of input features. For example, the product of two linear models is a quadratic function, and its monomials (products of pairs of features) form a natural basis for explanation. We denote the process of attributing to such a basis as $\mathcal{E}(\cdot\,; xx^\top)$ (as opposed to $\mathcal{E}(\cdot\,; x)$ for an attribution to individual features). Inspired by [34], we propose to attribute a product of two ML outputs as the outer product of their respective first-order explanations:

$$\mathcal{E}\big(y_m y_{m'}\,;\, xx^\top\big) = \mathcal{E}(y_m; x) \otimes \mathcal{E}(y_{m'}; x) \tag{4}$$

We provide some justification for Eq. (4) in Section 4, in particular, we find that it allows for maintaining useful properties of an explanation, such as conservation and zero-scores for irrelevant features. As a final step, combining Eqs. (3)–(4), we can finally observe that the overall explanation becomes the covariance over the first-order explanations of each ML model in the ensemble:

$$\mathcal{E}\big(s^2\,;\, xx^\top\big) = \mathrm{Cov}_m\big(\mathcal{E}(y_m; x)\big), \tag{5}$$

a matrix of size $d \times d$ encoding the contribution of each pair of features to the uncertainty. The proof is given in Supplementary Note A. In practice, our proposed explanations can be computed in two steps:

1. Compute one classical explanation for the output of each model in the ensemble, e.g., using LRP [12], resulting in a matrix of size $d \times M$, call it $\mathtt{R}$.
2. Compute the covariance over these explanations (e.g. `numpy.cov(R,R)`), resulting in the desired matrix of size $d \times d$.

Our method is illustrated and contrasted with a classical explanation workflow in Fig. 2. If we neglect the cost of computing the covariance matrix from the individual explanations (step 2), our method has M times the computational cost of a classical explanation of a single model in the ensemble. The cost of our method is thus also comparable to that of applying a classical first-order method on the whole ensemble. Hence, our second-order analysis does not cause any significant computational overhead compared to using a first-order approach.

In the presence of *multidimensional* targets (e.g. output time series), predictive uncertainty can be modeled as the sum of the variances of the individual output dimensions. The explanation then becomes a sum of covariances, specifically $\mathcal{E}\big(\sum_k s_k^2\,;\, xx^\top\big) = \sum_k \mathcal{E}(s_k^2\,;\, xx^\top)$.

Note that our framework allows the user to choose which first-order explanation technique to use within Eq. (5). We refer to the use of our method alongside a specific first-order explanation technique by adding to the latter the prefix 'Cov'. For example, if one computes $\mathcal{E}(y_m; x)$ using LRP, we refer to the resulting second-order explanation of uncertainty, i.e., the output of Eq. (5), as 'CovLRP'. Likewise, if the underlying attribution technique is Gradient × Input (GI), one gets 'CovGI'. There is no specific restriction on the choice of underlying attribution technique, except for the satisfaction of basic conservation properties (cf. Propositions 1 and 2).

4. Theoretical properties

We show in this section that our second-order method for attributing predictive uncertainty inherits certain properties of the first-order explanation method it builds upon.

Proposition 1 (*Conservation*). *If for each member m of the ensemble, the corresponding output y_m is attributed to the input features in a way that is conservative, i.e., if $\sum_i \mathcal{E}(y_m; x)_i = y_m$, then the attribution of predictive uncertainty s^2 according to Eq. (5) is also conservative, i.e., $\sum_{ij} \mathcal{E}(s^2; xx^\top)_{ij} = s^2$.*

One way of proving this is to combine Eqs. (3) and (4) and observe that summing all elements of the resulting matrix expression yields s^2. The detailed proof is given in Supplementary Note B.

Proposition 2 (*Preservation of Irrelevance*). *If all models in the ensemble are invariant to a given feature i, and if the explanations of each prediction reflect that invariance by assigning a score of zero, i.e., if $\forall_m : \mathcal{E}(y_m; x)_i = 0$, then it results that $\mathcal{E}(s^2; xx^\top)_{jk} = 0$ for all pairs (j,k) where $j = i$ or $k = i$. In other words, the feature i neither contributes to uncertainty on its own nor in interaction with other features.*

This property is straightforward to demonstrate from an inspection of Eq. (4) where features that have been attributed zero by the first-order explanations preserve their score of zero after the product operation.

4.1. Reductions for special cases

We now show that our second-order uncertainty explanations (which we can compute using Eq. (5)) reduce to simple and intuitive forms for special cases of models and underlying attribution techniques.

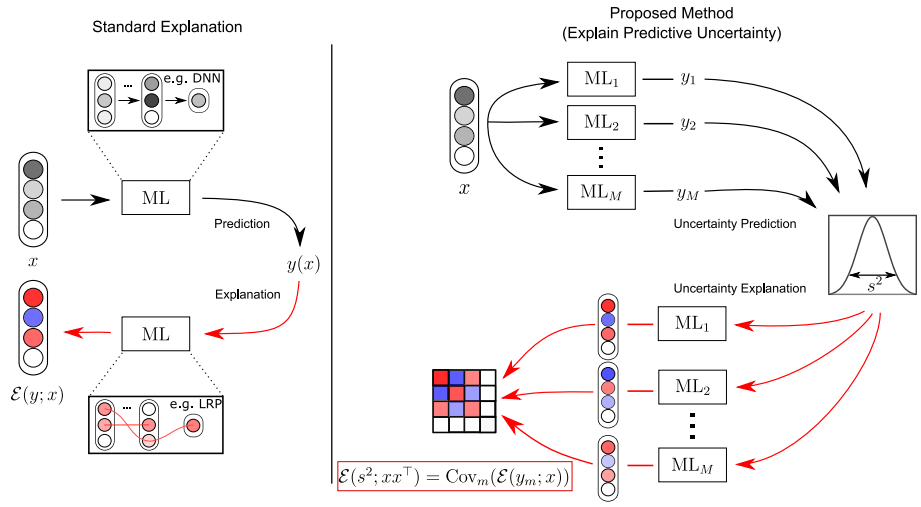

Fig. 2. Left: Classical explanation workflow, commonly used for attributing the output of a neural network model to individual input features (elements of x). Right: Proposed explanation method for explaining predictive uncertainty. Predictive uncertainty (estimated by the variance over an ensemble's predictions) is attributed to elements of $xx^\top$ (a second-order explanation) by computing a covariance over the explanations associated with each member of the ensemble.

Proposition 3. *If each model in the ensemble is a linear homogeneous function of the input, i.e., if $y_m = w_m^\top x$ with w_m the vector of parameters of model m and if the underlying first-order attribution method consists of the element-wise product of the weights and the input,*[1] *i.e., $\mathcal{E}(y_m; x) = w_m \odot x$, then the outcome of our analysis reduces to $\mathcal{E}(s^2; xx^\top) = \text{Cov}_m(w_m) \odot xx^\top$.*

In other words, Proposition 3 states that for a feature to contribute to uncertainty on its own, it must be present in the data, and the models in the ensemble must disagree about the effect of that feature for the prediction. Likewise, for two distinct features to *jointly* contribute to uncertainty, both must be expressed in the data, and the models should disagree, however, in some correlated manner. Specifically, some models should respond *positively* to the two features, and other models should disagree with the first models by responding *negatively* to the two features. Proposition 3 can be demonstrated by the chain equations: $\mathcal{E}(s^2; xx^\top) = \text{Cov}_m(\mathcal{E}(y_m)) = \text{Cov}_m(w_m \odot x) = \text{Cov}_m(w_m) \odot xx^\top$.

We now give a closed-form expression for a classical gradient-based method, relating the proposed second-order explanation technique to the Hessian of the predicted uncertainty.

Proposition 4. *If each model in the ensemble is a piecewise linear function of the input and if the underlying explanation method is Gradient × Input, then the outcome of our analysis can be interpreted in terms of the Hessian of s^2 w.r.t. the input features. Specifically, we get $\mathcal{E}(s^2; xx^\top) = \frac{1}{2}\nabla^2 s^2(x) \odot xx^\top$ where ∇^2 denotes the Hessian operation, and x is the feature vector.*

A proof is given in Supplementary Note C. This result can be seen as a generalization of the form of Proposition 3, where the Hessian operation is a local estimation of the discrepancy between the members of the ensemble.

5. Summarizing uncertainty explanations

In many applications of Explainable AI, the human interpreter requires as an explanation a d-dimensional heatmap instead of a high-dimensional matrix of feature interactions, which may be difficult to visualize and interpret. To retrieve an explanation over individual features, we consider two approaches. Let R denote the original explanation computed via Eq. (5) and its elements given by: $R_{ij} = \mathcal{E}(s^2; xx^\top)_{ij}$ A first approach to summarizing the original explanation consists of retaining only its diagonal elements, i.e., keeping terms of the explanation that can be unambiguously attributed to individual features, and disregarding interactions between features, i.e., $r_i^{\text{diag}} = R_{ii}$. We refer to it with the suffix 'diag' in our experiments. Because diagonal terms are variance computations, the resulting explanation is strictly non-negative and thus only sees features as sources of uncertainty and never as inhibitors of uncertainty. An alternative summarization technique consists of also including joint contributions and redistributing them equally to the two associated features, i.e., $r_i^{\text{marg}} = R_{ii} + \sum_{j:j\neq i}\left(\frac{1}{2}R_{ij} + \frac{1}{2}R_{ji}\right)$. The resulting explanation is also expressible as the column-wise sum over the original explanation of size $d \times d$. It can be further interpreted as the covariance between an input feature's relevance and the total explained output (i.e. a marginalization over the input features). In the experiments, we refer to this way of summarizing the explanation with the suffix 'marg'.

6. Quantitative evaluation

In the following, we proceed with a quantitative benchmark evaluation of our proposed covariance-based explanation technique against a number of existing techniques and baselines. We primarily consider the CovLRP instantiation of our method, which corresponds to injecting LRP explanations into Eq. (5). We use the generalized LRP-γ rule (cf. Supplementary Note D) and heuristically set its parameter γ to 0.2 at each layer in MLP models. In CNN models, due to the increased depth of the architecture, we chose a smaller γ value of 0.1 in the fully-connected layers and a value of 0.2 in the convolutional layers. We experiment with the 'diag' and 'marg' summarization techniques described in Section 5, and we refer to the resulting methods as CovLRP-diag and CovLRP-marg, respectively.

6.1. Baselines

We compare our proposed approach with a number of 'classical' explanation methods applied directly to the uncertainty function of interest $s^2(x)$. These include Gradient × Input (GI) [36]; Integrated Gradients (IG) [16], which explains any function output by computing a path integral of the input gradients along a segment connecting the data point and some reference point (in our experiments, the training data mean); Shapley Value Sampling (SVS) [14], a perturbation-based explanation technique readily implemented in Captum AI (https://captum.ai/); LIME [15,37], a surrogate-based explanation technique,

[1] We note that LRP, Integrated Gradients, Gradient × Input or Shapley values with a zero-valued reference point reduce to this simple attribution (cf. [11]).

Table 1
AUFC scores of different explanation techniques for *deep ensembles* built on a selection of datasets. Lower AUFC values indicate better, more faithful explanations. For each dataset, the best-performing explanation technique is shown in bold, and the second best is shown with an underlining.

Dataset (d)	Model	CovLRP		LRP	GI	IG	SVS	SA [27]	LIME	std[a]
		diag	marg							
Bias Correction (21)	DeepEns	**0.352**	0.444	0.411	0.559	0.546	0.513	0.600	0.478	±0.034
California Housing (8)	DeepEns	**0.344**	0.370	0.415	0.430	0.394	0.391	0.480	0.408	±0.029
EPEX-FR (96)	DeepEns	**0.044**	0.052	0.106	0.113	0.099	0.062	0.245	0.093	±0.035
kin8nm (8)	DeepEns	0.391	0.388	0.462	0.427	0.405	0.386	0.481	**0.382**	±0.013
Seoul Bike Sharing (98)	DeepEns	**0.268**	0.294	0.293	0.350	0.338	0.329	0.394	0.382	±0.025
Wine Quality (11)	DeepEns	0.482	**0.471**	0.526	0.517	0.500	0.495	0.594	0.528	±0.026
YearPredictionMSD (90)	DeepEns	**0.155**	0.173	0.184	0.264	0.273	0.195	0.422	0.310	±0.020
Bias Correction	MCDropout	**0.514**	0.517	0.568	0.651	0.530	0.672	0.721	0.940	±0.039
California Housing	MCDropout	**0.674**	0.691	0.728	0.812	0.703	0.787	0.963	0.950	±0.059
EPEX-FR	MCDropout	**0.085**	0.091	0.137	0.146	0.119	0.125	0.292	0.311	±0.018
kin8nm	MCDropout	**0.483**	0.486	0.568	0.586	0.498	0.593	0.634	0.619	±0.058
Seoul Bike Sharing	MCDropout	**0.520**	0.590	0.555	0.640	0.568	0.676	0.727	0.974	±0.057
Wine Quality	MCDropout	0.661	**0.657**	0.713	0.729	0.662	0.767	0.807	0.813	±0.052
YearPredictionMSD	MCDropout	**0.215**	0.258	0.253	0.391	0.273	0.403	0.622	0.560	±0.047
YearPredictionMSD	DeepEns-5	**0.128**	0.148	0.155	0.197	0.212	0.153	0.377	0.274	±0.022
YearPredictionMSD	DeepEns-10	**0.155**	0.173	0.184	0.264	0.273	0.195	0.422	0.310	±0.020
YearPredictionMSD	DeepEns-20	**0.162**	0.183	0.247	0.250	0.267	0.218	0.519	0.336	±0.033
YearPredictionMSD	DeepEns-40	0.180	**0.179**	0.235	0.267	0.277	0.213	0.503	0.325	±0.025
EPEX-FR	ConvNet	**0.085**	0.101	0.210	0.159	0.108	0.087	0.279	0.339	±0.012
Seoul Bike Sharing	ConvNet	**0.231**	0.308	0.422	0.331	0.306	0.321	0.336	0.327	±0.057

[a] For conciseness, we report only the maximum standard deviations over the different explanation methods.

and finally, plain LRP [12,38] (i.e., without our proposed covariance-based formulation). Because LRP relies on a computational graph to perform attribution, we represent Eq. (1) as an additional layer on top of the M ensemble neural networks. This additional top layer consists of the activation function, i.e., $a(y_m) = (y_m - \bar{y})^2$ with the mean $\bar{y}$ treated as constant, followed by linear aggregation. We further include Sensitivity Analysis (SA), which scores input features according to the model output's partial derivatives, and which was specifically proposed in [27] to explain predictive uncertainty.

6.2. Datasets

We perform our evaluation on multiple regression datasets, which include the kin8nm dataset,[2] as well as five datasets from the UCI Machine Learning Repository, namely the Bias Correction, California Housing, Wine Quality, YearPredictionMSD and Seoul Bike Sharing datasets. For the latter dataset, where the input data to the prediction is not strictly defined, we treated the prediction as a time series problem and used a concatenation of past and present data of a given day as the input representation.

The datasets were processed in a consistent manner: they were shuffled and split into training and testing sets, with 75% of the data allocated to training and the remaining 25% reserved for testing.[3] All datasets were standardized by subtracting the training data mean and dividing by the training data standard deviation per feature.

In addition, we considered the EPEX-FR dataset proposed in [39], which serves as a benchmark for predicting day-ahead electricity prices. This data, publicly available at the authors' GitHub repository (https://github.com/jeslago/epftoolbox), spans from January 2011 to December 2016. The task is to forecast the next 24 day-ahead electricity prices in France based on the next 24 forecasted values for French electricity demand and renewable electricity production and the previous 48 h of day-ahead prices. The data is divided temporally, with data from 2016 reserved for testing and data from 2011 to 2015 used for training, as recommended in [39]. As with the UCI datasets, EPEX-FR was centered and standardized.

[2] https://www.cs.toronto.edu/delve/data/kin/desc.html.

[3] An exception is YearPredictionMSD, which comes with a predefined training-test split.

6.3. AUFC evaluation metric

Good explanations should be able to identify the subset of features that are most relevant to the model output. This quality of an explanation is often evaluated using pixel-flipping [12,40] (which we refer to as feature-flipping in the context of our tabular data). The feature-flipping procedure consists of ranking the input features in order of relevance according to the explanation. One then iteratively flips (i.e., removes) features from most to least relevant (i.e., starting with the most positive scores and terminating with the most negative ones). As features are being removed one after another, we keep track of the output of the network (in our case, the uncertainty score s^2), thereby creating a 'flipping curve'. The faster the curve decreases, the better the explanation. We summarize this decreasing behavior using the area under the flipping curve (AUFC). Details of the computation of AUFC scores are provided in Supplementary Note E. In our experiment, we report the AUFC averaged over the 100 test examples with the highest predictive uncertainty.

6.4. Results

We perform our evaluation on a diverse set of datasets and models, including classical deep ensembles and ensembles derived from applying Monte-Carlo Dropout [20], a different uncertainty estimator based on the generation of multiple predicting instances through the *dropout* mechanism, and set the dropout rate to 0.1. The ensemble size is set to 10, but we also experimented with ensemble sizes from 5 to 40. The deep ensemble consists of Multi-Layer Perceptrons (MLP) with three layers composed of 900, 600, and 300 neurons and ReLU activations, respectively. We also test deep ensembles of CNN instances with three convolutional layers of 16, 8, and 4 channels, respectively, and two fully-connected 100-neuron layers. The CNN ensembles were trained on the EPEX-FR and Seoul Bike datasets transformed into a channelized format. The channelized EPEX-FR dataset consists of 3 channels of 48 values, while the channelized Seoul Bike dataset consists of 9 channels of length 10. All models undergo 100 training epochs. During each epoch, we evaluate the loss on a 10% validation set held out from the training data and save the best-performing model for the final application. Uncertainty is calculated as the variance of model predictions according to Eq. (1). The AUFC score of each explanation technique on each dataset-model pair is shown in Table 1.

We observe that CovLRP systematically achieves the highest explanation accuracy, specifically, CovLRP-diag, which drops feature interaction terms. Our approach not only improves over a naive application of LRP to predictive uncertainty but also over a wide range of diverse baselines. Furthermore, applying Cov(·)-diag on top of other explanation methods also leads to similar explanation improvements, as we show in Supplementary Note F. We can further demonstrate the advantage of Cov(·)-diag over the first-order approach in synthetic experiments where the produced explanation can be compared to some ground truth (results in Supplementary Note F).

Overall, the superior performance of our proposed explanations compared to first-order methods (including computationally more expensive ones) underscores the benefit of integrating second-order effects into the explanation procedure. The superiority of the diagonal ('diag') over the marginal ('marg') summarization may appear surprising, given that the diagonal does not include all the evidence for uncertainty. We explain this result by the inherent difficulty of attributing feature interactions to individual features. These interactions are inherently more sensitive to changes in feature values and thus only locally informative. In contrast, single-feature contributions used in the diagonal summarization constitute a more global (and thereby more robust) form of explanation. Overall, our results also demonstrate the benefit of disentangling these two types of second-order contributions, which are otherwise entangled in the simpler baseline explanations.

7. Use case 1: Identifying underrepresented features in CelebA

High predictive uncertainty commonly arises when a model makes predictions for data points dissimilar from the observed training data [9]. Such a covariate shift may be caused by measurement biases or insufficient and unrepresentative training data collection from the whole data population.

As a consequence of insufficient training data collection, some input features may remain underrepresented at training time. When these features appear at test time, the model is ill-prepared to interpret their effect on the prediction task. Thus, the model prediction is unreliable, and predictive uncertainty is high. In this case, explaining predictive uncertainty in terms of underrepresented features can enable the user to precisely diagnose what is missing in the current data and, subsequently, gather additional training data to improve the model.

This section will demonstrate that our uncertainty explanation is able to reveal underrepresented high-level features at test time and how retraining on a consolidated dataset reduces uncertainty attributed to the originally underrepresented feature. To show this, we consider a setting of a model suffering from missing features in the training data. We perform fine-tuning by introducing new data points with the missing feature, improving model accuracy. We accompany this model improvement with our Explainable AI method to explain the relevant features inducing uncertainty of the original model and explain the reduction of uncertainty after fine-tuning.

In this use case, we consider the CelebA dataset,[4] which contains over 200,000 celebrity face images and multiple annotated visual features per image. In addition, we consider the CelebA-HQ extension,[5] which adds to 30,000 CelebA images detailed segmentation masks for the visual features. We will use the 30,000 CelebA-HQ images as the test set. The CelebA dataset allows us to simulate removing a visual feature at training but not at test time, and the segmentation masks of CelebA-HQ allow us to aggregate relevance attributed to these visual features.

We trained two ensembles of five VGG-16 [41] networks on a male/female classification task. We trained the first ensemble on a subset of the training data, from which we removed all images exhibiting a particular feature (hats and eyeglasses). To create the second ensemble, we fine-tuned a copy of the first ensemble using the previously omitted data. Thus, while the first ensemble has no concept of the omitted features, the fine-tuned one has. Fine-tuning enhanced test accuracy from 97.9% to 98.2% when the 'eyeglasses' feature was originally omitted, and from 98% to 98.3% when the hat feature was originally omitted.

Following training, we consider the subset of 100 CelebA-HQ test data points of the omitted feature with the greatest uncertainty reduction after fine-tuning. On this experimental dataset, we may expect that the ground-truth cause of the original ensemble's uncertainty lies in the underrepresented feature. A truthful uncertainty explanation is then expected to highlight this feature when explaining the uncertainty of the original ensemble. When comparing uncertainty explanations before and after fine-tuning, it is expected that fine-tuning would reduce the relevance of previously underrepresented features. To verify this, we apply CovLRP-diag,[6] identifying for each instance the pixel-wise contributions to predictive uncertainty.

In Fig. 3, we show on the left a T-SNE embedding of the CelebA dataset, where our experimental dataset (images for which fine-tuning leads to a maximum reduction in uncertainty) is highlighted in red. On the right, we display a selection of those input images and their pixel-wise uncertainty explanations before and after fine-tuning. Pixel-wise explanations confirm that omitted features ('eyeglasses' and 'hat') are a primary source of predictive uncertainty in the original ensemble and that fine-tuning on the full data significantly reduces these sources of uncertainty. Our observations are confirmed quantitatively by the histograms in the middle column, which measure the uncertainty attributed to the different CelebA visual features averaged over the whole experimental data and highlight that the reduction in uncertainty is primarily attributable to the 'eyeglasses' and 'hat' features.

8. Use case 2: Insights into German day-ahead electricity prices

Practitioners of Explainable AI are often motivated by the prospect of performing data science and extracting new insights from large datasets that would otherwise be too complex for human investigation. To uncover interesting features within a dataset, an ML model can be trained on the data, and Explainable AI techniques can then be applied to highlight input features that are relevant for predicting an output [42,43]. In the following, we demonstrate through a practically relevant use case how Explainable AI's ability to characterize input–output relationships can be extended to the case where the output has the structure of an uncertainty estimator.

We considered the task of predicting German day-ahead electricity prices on the EPEX-DE dataset as in [39], with a particular interest in price volatility. We organized the dataset into one target series of 24 future hourly day-ahead prices and three input channels (x_1, x_2, x_3). These three input channels, representing past prices, renewable energy production, and electricity demand (i.e., grid load), respectively, were organized as series of 48 entries. The past price series consists of the previous 48 hourly prices. The electricity demand and renewable energy production series each consist of 24 historical values and 24 forecasts for the next day.

We applied deep ensembles to predict price volatility.[7] We trained a deep ensemble of 10 convolutional neural networks with three convolutional layers, three dense layers, and 24 output neurons. In all layers, we used the ReLU activation function. We used data from the

[4] https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

[5] https://mmlab.ie.cuhk.edu.hk/projects/CelebA/CelebAMask_HQ.html.

[6] The underlying LRP explanations are computed using the generalized LRP-γ with $\gamma = 0.1$ in the convolutional layers and $\gamma = 0.01$ in the dense layers.

[7] While price volatility can, in principle, be learned from price time series and predicted directly (using, e.g., a classical neural network), such a direct approach will tend to underestimate volatility on unseen data [19].

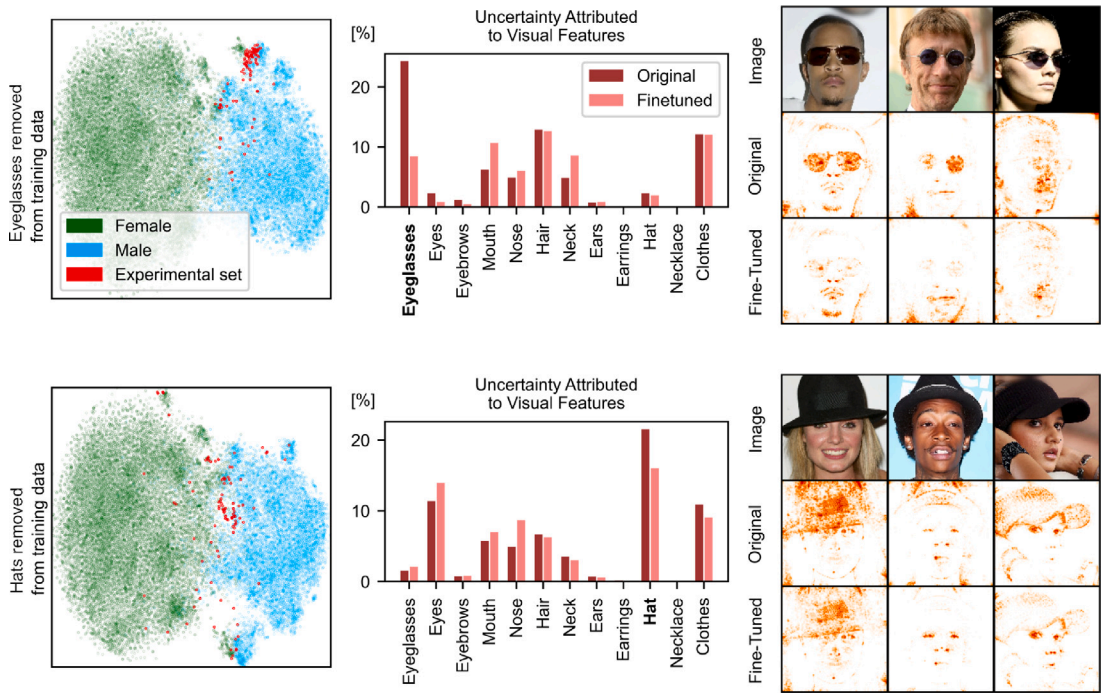

Fig. 3. Comparison of the uncertainty attribution of the original and the fine-tuned ensemble for two different underrepresented features. In the upper row, the original ensemble was trained without 'eyeglass' images; in the lower row, the original ensemble was trained without 'hat' images. *Left:* T-SNE visualization of the original ensemble's hidden activations for the test data points, with the actual class labels colored in green-blue and the experimental set in red. *Middle:* Share of uncertainty attributed to different visual features for the original and the fine-tuned ensemble, highlighting the primary role of 'eyeglass' and 'hat' features in reducing uncertainty. *Right:* Heatmap explanations of the original and the fine-tuned ensemble for three images from the experimental data, illustrating on a pixel-wise basis the reduction of 'hat' and 'eyeglass' features as contributors to uncertainty. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

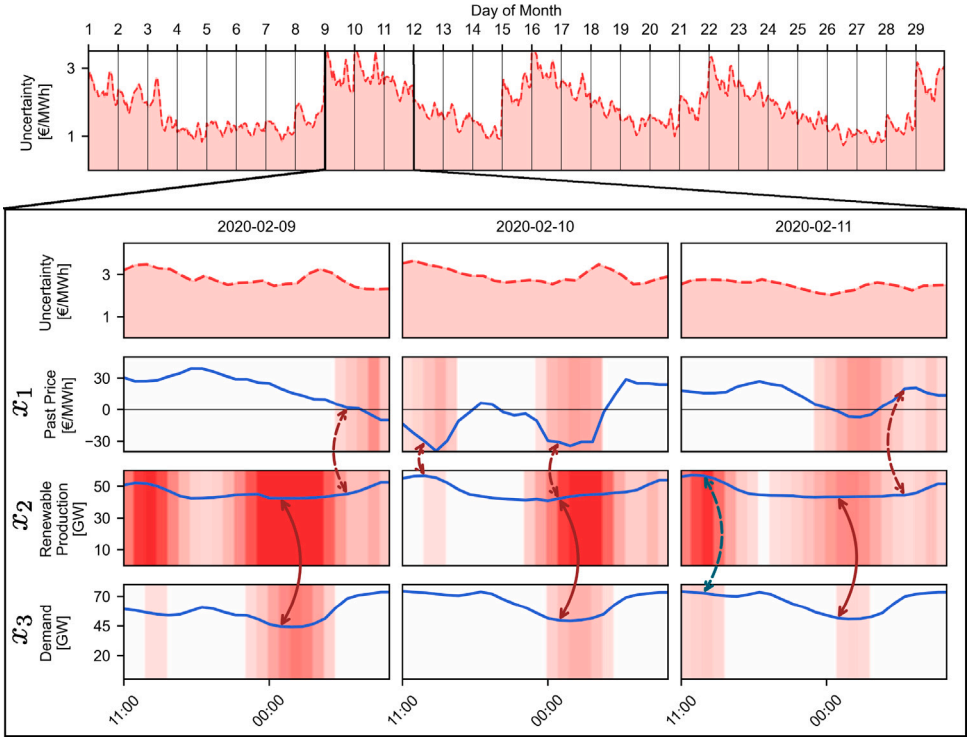

Fig. 4. Predictive uncertainty of day-ahead price prediction and uncertainty relevance analysis of three high-uncertainty days. The upper plot depicts the trained deep ensemble's hourly predictive uncertainty over the course of an entire month. The lower plot depicts the predictive uncertainty for three consecutive days and the 24 last values of the input channels x_1, x_2, x_3. Additionally, the CovLRP attribution of uncertainty onto these three channels is depicted in shades of red for diagonal terms and as two-sided arrows for off-diagonal terms capturing the highest interactions. Solid connecting lines denote strong interactions, and dashed connecting lines denote weaker interactions. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

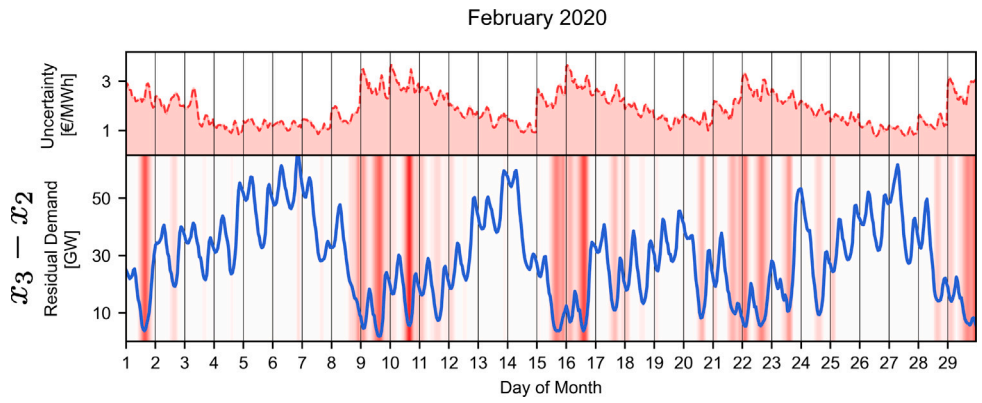

Fig. 5. Contribution of residual demand to uncertainty in February 2020. The upper plot depicts the trained deep ensemble's predictive uncertainty. The lower plot depicts residual demand, calculated as the difference between demand (x_3) and renewable production (x_2), and the shades of red show the sum of contribution to the uncertainty associated with these two features. The figure shows a negative correlation between residual demand and uncertainty and the explanation's focus on low residual demand periods. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

year 2019 for training and data from the year 2020 for validation. The neural networks were randomly initialized, and we stopped training at minimal validation loss. In order to understand price volatility in terms of input features, we then applied CovLRP to the predicted uncertainty. We produced LRP explanations using the generalized LRP-γ rule and set $\gamma = 0.3$ in the convolutional layers and $\gamma = 0.1$ in the dense layers. Because LRP-γ and its generalized variant always assign zero relevance to zero-valued features, potentially biasing the result of the analysis, we performed an affine transformation of the input data before training, where for each channel x_i, we applied the map $x_i \mapsto (1 - x_i, 1 + x_i, 2 - x_i, 2 + x_i)$, thereby forcing features to always have at least one non-zero value after mapping, and thus allowing any feature value to be attributed relevance.

As a starting point for our investigation into volatility-inducing features, we analyze the uncertainty explanation associated with three high-uncertainty days between February 9th and February 11th. During that period, storm 'Sabine' passed over Germany, causing extremely high wind power generation. Fig. 4 displays the predictive uncertainty in February 2020, with a focus on the three days of storm, for which we additionally show the three input channels' time series, their respective contribution to uncertainty (CovLRP's diagonal terms), and the most significant feature interactions (CovLRP's off-diagonal terms normalized by the corresponding diagonal terms[8]). The analysis of relevant feature interactions is simplified by only considering interactions of simultaneous feature values and aggregating interactions over six-hour intervals (e.g., 11 h–17 h, 17 h–23 h, etc.).

We observe that the variable x_1 representing past prices contributes to uncertainty when those prices are negative.[9] Furthermore, renewable production x_2, which is constantly high over these three days, tends to contribute strongly to uncertainty, especially at night. Dips in expected electricity demand (variable x_3) during the night and weekend[10] also contribute to overall uncertainty. Moreover, strong uncertainty-inducing interactions can be observed around midnight between high renewable production (x_2) and low electricity demand (x_3).

The combined effect of high renewable energy production and low electricity demand on uncertainty is intriguing. It leads us to hypothesize that our method has uncovered the residual demand, defined in power markets as the difference between electricity demand (x_3) and renewable energy production (x_2), as a primary driver of uncertainty. When the expected residual demand is low and renewable sources are at peak production, fossil energy producers are compelled to reduce their output. This often results in high down-regulation costs. Depending on fuel prices, producers may choose to sell electricity below production cost to avoid these down-regulation costs. This decoupling between electricity supply and price can result in increased price volatility.

We then test our hypothesis by analyzing the relationship between residual demand ($x_3 - x_2$) and predictive uncertainty over an extended period consisting of all days of February 2020. Results are shown in Fig. 5. We observe that predictive uncertainty negatively correlates to the residual demand. Furthermore, the overall contributions of the associated features x_2 and x_3 to uncertainty are consistently high at residual demand troughs, suggesting a deeper connection between residual demand and uncertainty than a mere data correlation.—As renewable electricity production will undoubtedly increase over the next years, these results suggest that existing ML models based on demand and supply data will become insufficient to forecast day-ahead prices precisely. More generally, this analysis exemplified that the explainability of model uncertainty can help practitioners anticipate trends, such as a gradual decline in the predictive performance of ML models.

9. Conclusion and discussion

Predictive uncertainty, or 'knowing when the model doesn't know', can be critical for real-world predictive systems. So is the ability to *explain* predictive uncertainty to ensure that those uncertainty estimates are 'right for the right reasons'.

We have contributed new insights to the problem of explaining uncertainty in the common and popular case where the latter is computed as the variance over an ensemble of predictions. We highlighted that the structure of predicted uncertainty is dominated by second-order effects, including both single-feature quadratic contributions and joint-feature contributions, a distinction that classical explanation techniques do not make. Putting these insights into algorithms, we have proposed a novel framework for uncertainty explanation that efficiently extracts the second-order feature contributions. Our derivation leads to a general scheme for computing uncertainty explanations, namely a covariance over an ensemble's individual classical explanations. Thus, our method allows to systematically transform classical first-order explanation techniques (LRP, GI, etc.) into more powerful second-order uncertainty explainers (CovLRP, CovGI, etc.).

In a quantitative evaluation, we demonstrated the high performance of our approach, with CovLRP achieving the highest explanation accuracy (as evaluated by a feature-flipping experiment), outperforming classical LRP as well as a number of other competitive baselines. We

[8] Or equivalently, off-diagonal terms of the *correlation* matrix of LRP heatmaps.

[9] Negative prices are often caused by inflexible fossil energy production and low demand.

[10] February 9th, 2020 was a Sunday.

found that the superiority of CovLRP holds consistently across the multiple tabular datasets included in our benchmark, as well as for a variety of uncertainty models (deep ensembles and MC dropout) and neural network architectures (fully-connected and convolutional). We attribute the high performance to the ability of our approach to expose and disentangle the second-order effect, in particular, favoring single-feature contributions over the less robust interaction terms. Furthermore, our CovLRP approach inherits technical advantages of LRP such as applicability to general neural network structures, and high compute efficiency.

We then applied our framework to two practical use cases. Our first use case demonstrated that the proposed method could reveal uncertainty caused by covariate shift at test time. By identifying underrepresented features, our method can guide practitioners in collecting additional training data in a targeted fashion, ultimately improving model performance. In our second use case, we explored an electricity price dataset and focused on predictive uncertainty as a model of price volatility. Our uncertainty explanation revealed the difference between electricity demand and renewable production to be a key factor of uncertainty, enabling a scientist to test the viability of existing ML approaches in the context of a gradual increase in renewable energy production.

Limitations and future work:. So far, our investigation has been limited to *ensemble-based uncertainty estimators*, where uncertainty derives from the disagreement between ensemble members. Although those uncertainty estimators are among the most common and popular, our proposed uncertainty explanation method could be extended in the future to explain other forms of uncertainty, such as in Mixture Density Networks [18], or more diverse sets of models such as fuzzy decision systems [44–46]. An additional limitation of our work is the focus on variance as a measure of uncertainty, which can cause distortions in the presence of strong outliers. Incorporating robust statistics into our explanation framework, disentangling between valid and outlier predictions, would be an important future work. Finally, while our investigation has focused on explanations in terms of the input features (e.g. pixels), further understanding of predictive uncertainty may be more efficiently achieved in a dedicated latent space representing more abstract concepts. Such concept-based uncertainty explanations may be potentially achievable within the framework of virtual inspection layers [47].

CRediT authorship contribution statement

Florian Bley: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Sebastian Lapuschkin:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Wojciech Samek:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Funding acquisition, Conceptualization. **Grégoire Montavon:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded by the German Ministry for Education and Research (refs. 01IS18037A, 01IS18037I, 01IS18025A); the German Research Foundation (DFG) as research unit KI-FOR 5363 (project ID: 459422098); the European Union's Horizon Europe research and innovation programme (EU Horizon Europe) as grant TEMA (101093003); the European Union's Horizon 2020 research and innovation programme (EU Horizon 2020) as grant iToBoS (965221); and the state of Berlin within the innovation support programme ProFIT (IBB) as grant BerDiBa (10174498).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patcog.2024.111171.

Data availability

Data is accessible freely in public repositories.

References

[1] W. Xu, J. Pan, J. Wei, J.M. Dolan, Motion planning under uncertainty for on-road autonomous driving, in: ICRA, IEEE, 2014, pp. 2507–2512.
[2] G. Kahn, A. Villaflor, V. Pong, P. Abbeel, S. Levine, Uncertainty-aware reinforcement learning for collision avoidance, 2017, arXiv:1702.01182.
[3] Y.B. Özçelik, A. Altan, Overcoming nonlinear dynamics in diabetic retinopathy classification: A robust AI-based model with chaotic swarm intelligence optimization and recurrent long short-term memory, Fract. Fract. 7 (8) (2023) 598.
[4] İ. Yağ, A. Altan, Artificial intelligence-based robust hybrid algorithm design and implementation for real-time detection of plant diseases in agricultural environments, Biology 11 (12) (2022) 1732.
[5] A. Mehrtash, W.M.W. III, C.M. Tempany, P. Abolmaesumi, T. KapurDEFAULT, Confidence calibration and predictive uncertainty estimation for deep medical image segmentation, IEEE Trans. Med. Imaging 39 (12) (2020) 3868–3878.
[6] M. Abdar, et al., Uncertainty quantification in skin cancer classification using three-way decision-based Bayesian deep learning, Comput. Biol. Med. 135 (2021) 104418.
[7] W. Xi, Z. Li, X. Song, H. Ning, Online portfolio selection with predictive instantaneous risk assessment, Pattern Recognit. 144 (2023) 109872.
[8] G. Montavon, M.L. Braun, T. Krueger, K.-R. Müller, Analyzing local structure in kernel-based learning: Explanation, complexity, and reliability assessment, IEEE Signal Process. Mag. 30 (4) (2013) 62–74.
[9] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J.V. Dillon, J. Ren, Z. Nado, Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift, in: NeurIPS, 2019, pp. 13969–13980.
[10] A. Holzinger, G. Langs, H. Denk, K. Zatloukal, H. Müller, Causability and explainability of artificial intelligence in medicine, WIREs Data Min. Knowl. Discov. 9 (4) (2019).
[11] W. Samek, G. Montavon, S. Lapuschkin, C.J. Anders, K.-R. Müller, Explaining deep neural networks and beyond: A review of methods and applications, Proc. IEEE 109 (3) (2021) 247–278.
[12] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE 10 (7) (2015) e0130140.
[13] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, Int. J. Comput. Vis. 128 (2) (2020) 336–359.
[14] E. Štrumbelj, I. Kononenko, An efficient explanation of individual classifications using game theory, J. Mach. Learn. Res. 11 (1) (2010) 1–18.
[15] M.T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, in: KDD, ACM, 2016, pp. 1135–1144.
[16] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: ICML, Vol. 70, 2017, pp. 3319–3328.
[17] D. Nix, A. Weigend, Estimating the mean and variance of the target probability distribution, in: ICNN, Vol. 1, 1994, pp. 55–60.
[18] C. Bishop, Mixture Density Networks, Tech. Rep., Aston University, 1994.
[19] C. Guo, G. Pleiss, Y. Sun, K.Q. Weinberger, On calibration of modern neural networks, in: ICML, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 1321–1330.
[20] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: ICML, in: JMLR Workshop and Conference Proceedings, vol. 48, JMLR.org, 2016, pp. 1050–1059.
[21] S.H. Yelleni, D. Kumari, S.P. K, K.M. C, Monte Carlo DropBlock for modeling uncertainty in object detection, Pattern Recognit. 146 (2024) 110003.
[22] M. Teye, H. Azizpour, K. Smith, Bayesian uncertainty estimation for batch normalized deep networks, in: ICML, in: Proceedings of Machine Learning Research, vol. 80, PMLR, 2018, pp. 4914–4923.
[23] W.J. Maddox, P. Izmailov, T. Garipov, D.P. Vetrov, A.G. Wilson, A simple baseline for Bayesian uncertainty in deep learning, in: NeurIPS, 2019, pp. 13132–13143.

[24] B. Lakshminarayanan, A. Pritzel, C. Blundell, Simple and scalable predictive uncertainty estimation using deep ensembles, in: NIPS, 2017, pp. 6402–6413.
[25] A. Amini, W. Schwarting, A. Soleimany, D. Rus, Deep evidential regression, in: NeurIPS, 2020.
[26] A. Malinin, M.J.F. Gales, Predictive uncertainty estimation via prior networks, in: NeurIPS, 2018, pp. 7047–7058.
[27] S. Depeweg, J.M. Hernández-Lobato, S. Udluft, T.A. Runkler, Sensitivity analysis for predictive uncertainty, in: ESANN, 2018.
[28] D.S. Watson, J. O'Hara, N. Tax, R. Mudd, I. Guy, Explaining predictive uncertainty with information theoretic shapley values, in: NeurIPS, 2023.
[29] D. Wood, T. Papamarkou, M. Benatan, R. Allmendinger, Model-agnostic variable importance for predictive uncertainty: an entropy-based approach, Data Min. Knowl. Discov. 38 (6) (2024) 4184–4216.
[30] N. Amanova, J. Martin, C. Elster, Finding the input features that reduce the entropy of a neural network's prediction, Appl. Intell. 54 (2) (2024) 1922–1936.
[31] J. Antorán, U. Bhatt, T. Adel, A. Weller, J.M. Hernández-Lobato, Getting a CLUE: a method for explaining uncertainty estimates, in: ICLR, 2021.
[32] M. Sundararajan, K. Dhamdhere, A. Agarwal, The Shapley taylor interaction index, in: ICML, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 9259–9268.
[33] J.D. Janizek, P. Sturmfels, S.-I. Lee, Explaining explanations: Axiomatic feature interactions for deep networks, J. Mach. Learn. Res. 22 (104) (2021) 1–54.
[34] O. Eberle, J. Büttner, F. Kräutli, K.-R. Müller, M. Valleriani, G. Montavon, Building and interpreting deep similarity models, IEEE Trans. Pattern Anal. Mach. Intell. 44 (3) (2022) 1149–1161.
[35] T. Schnake, O. Eberle, J. Lederer, S. Nakajima, K.T. Schütt, K.-R. Müller, G. Montavon, Higher-order explanations of graph neural networks via relevant walks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 7581–7596.
[36] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: ICML, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3145–3153.
[37] Y. Zhang, K. Song, Y. Sun, S. Tan, M. Udell, "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations, arXiv:1904.12991.
[38] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, K.-R. Müller, Layer-wise relevance propagation: An overview, in: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, Springer, 2019, pp. 193–209.
[39] J. Lago, G. Marcjasz, B. De Schutter, R. Weron, Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, Appl. Energy 293 (2021) 116983.
[40] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, Evaluating the visualization of what a deep neural network has learned, IEEE Trans. Neural Netw. Learn. Syst. 28 (11) (2017) 2660–2673.
[41] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, ICLR, 2015.
[42] D. Slijepcevic, et al., Explaining machine learning models for clinical gait analysis, ACM Trans. Comput. Healthc. 3 (2) (2021).
[43] A. Binder, et al., Morphological and molecular breast cancer profiling through explainable machine learning, Nat. Mach. Intell. 3 (4) (2021) 355–366.
[44] W. You, X. Xie, H. Wang, J. Xia, V. Stojanovic, Relaxed model predictive control of T-S fuzzy systems via a new switching-type homogeneous polynomial technique, IEEE Trans. Fuzzy Syst. 32 (8) (2024) 4583–4594.
[45] P. Sun, X. Song, S. Song, V. Stojanovic, Composite adaptive finite-time fuzzy control for switched nonlinear systems with preassigned performance, Internat. J. Adapt. Control Signal Process. 37 (3) (2022) 771–789.
[46] Z. Peng, X. Song, S. Song, V. Stojanovic, Hysteresis quantified control for switched reaction–diffusion systems and its application, Complex Intell. Syst. 9 (6) (2023) 7451–7460.
[47] J. Vielhaben, S. Lapuschkin, G. Montavon, W. Samek, Explainable AI for time series via virtual inspection layers, Pattern Recognit. 150 (2024) 110309.

Florian Bley obtained a Master's degree in Industrial Engineering and Management from the Technische Universität Berlin, Germany, in 2022. He is currently pursuing his Ph.D. at the Machine Learning Group at TU Berlin.

Sebastian Lapuschkin is heading the XAI Research Group at Fraunhofer Heinrich Hertz Institute in Berlin, Germany, since 2021. He received a Ph.D. degree in Machine Learning from the Technische Universität Berlin in 2018.

Wojciech Samek received the Ph.D. degree from the Technische Universität Berlin, Germany, in 2014. He is a Professor with the Department of Electrical Engineering and Computer Science, Technische Universität Berlin, and is jointly heading the Department of Artificial Intelligence, Fraunhofer Heinrich Hertz Institute, Berlin.

Grégoire Montavon received the Ph.D. degree from the Technische Universität Berlin, Germany, in 2013. He is a Guest Professor at the Department of Mathematics and Computer Science at the Freie Universität Berlin, and Research Group Lead in the Berlin Institute for the Foundations of Learning and Data (BIFOLD).