frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 10 February 2025
DOI 10.3389/frai.2025.1506074

OPEN ACCESS

EDITED BY
Jose Santamaria Lopez,
University of Jaén, Spain

REVIEWED BY
Sumeet Sehra,
Conestoga College, Canada
Archana Talhar Belge,
Thakur College of Engineering and Technology, India
Pinaki Mitra,
Indian Institute of Technology Guwahati, India

*CORRESPONDENCE
Vladimir Muliukha
✉ vladimir.muliukha@spbstu.ru

RECEIVED 04 October 2024
ACCEPTED 23 January 2025
PUBLISHED 10 February 2025

CITATION
Konstantinov A, Kozlov B, Kirpichenko S, Utkin L and Muliukha V (2025) Dual feature-based and example-based explanation methods.
Front. Artif. Intell. 8:1506074.
doi: 10.3389/frai.2025.1506074

Dual feature-based and example-based explanation methods

Andrei Konstantinov, Boris Kozlov, Stanislav Kirpichenko, Lev Utkin and Vladimir Muliukha*

Department of Artificial Intelligence Technologies, Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

A new approach to the local and global explanation based on selecting a convex hull constructed for the finite number of points around an explained instance is proposed. The convex hull allows us to consider a dual representation of instances in the form of convex combinations of extreme points of a produced polytope. Instead of perturbing new instances in the Euclidean feature space, vectors of convex combination coefficients are uniformly generated from the unit simplex, and they form a new dual dataset. A dual linear surrogate model is trained on the dual dataset. The explanation feature importance values are computed by means of simple matrix calculations. The approach can be regarded as a modification of the well-known model LIME. The dual representation inherently allows us to get the example-based explanation. The neural additive model is also considered as a tool for implementing the example-based explanation approach. Many numerical experiments with real datasets are performed for studying the approach. A code of proposed algorithms is available. The proposed results are fundamental and can be used in various application areas. They do not involve specific human subjects and human data.

KEYWORDS

machine learning, explainable AI, neural additive network, dual representation, convex hull, example-based explanation, feature-based explanation

1 Introduction

Many machine learning models, including neural networks, have the black-box nature due to their complexity and the obscurity of their internal workings. Therefore, to explain how predictions are obtained for their corresponding inputs, specific explanation methods are required. This requirement affects many applications, especially those in medicine, finance, and safety maintenance. As a result, many successful methods and algorithms have been developed to satisfy this requirement (Arya et al., 2019; Belle and Papantonis, 2021; Guidotti et al., 2019; Liang et al., 2021; Molnar, 2019; Murdoch et al., 2019; Ras et al., 2022; Zablocki et al., 2021; Zhang Y. et al., 2021).

There are many definitions and interpretations of the explanation. We understand explanation as an answer to the question which features of an instance or a set of instances significantly impact the black-box model prediction or which features are most relevant to the prediction. Methods answering this question can be referred to as *feature importance* methods or the *feature-based explanation*. Another group of explanation methods is called the *example-based* explanation methods (Molnar, 2019). The corresponding methods are based on selecting influential instances from a training set having the largest impact on predictions to compare the training instance with the explainable one.

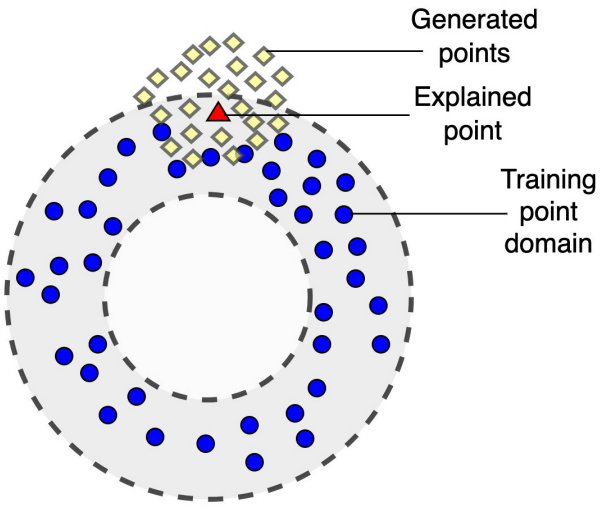

FIGURE 1
Illustration of a case of out-of-domain data when generated points may be out of the training point domain.

Feature importance explanation methods, in turn, can be divided into two groups: local and global. Methods from the first group explain the black-box model predictions locally around a test instance. Global methods explain a set of instances or the entire dataset. The well-known local explanation method is the Local Interpretable Model-Agnostic Explanation (LIME) (Ribeiro et al., 2016). In accordance with this method, a surrogate model is constructed and trained, which approximates the black-box model at a point. The surrogate model in LIME is the linear regression whose coefficients can be interpreted as the feature importance measures. In fact, LIME can be regarded as a method of the linear approximation of a complex non-linear function implemented by the black-box model at a point. LIME is based on using a simple regression model. Agarwal et al. (2021) proposed to generalize LIME using the generalized additive model (GAM) (Hastie and Tibshirani, 1990) instead of the simple linear regression and its implementation by means of neural networks of a special form. The GAM is a more general and flexible model in comparison with the original linear model. The corresponding surrogate model using the GAM is called the neural additive model (NAM).

Another important method, which is used for the local as well as global explanations, is SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017; Strumbelj and Kononenko, 2010). The method is based on applying game-theoretic Shapley values (Shapley, 1953) which can be interpreted as average marginal contributions of features to the black-box model prediction. SHAP can be also viewed as a method of the linear approximation of the black-box model predictions.

One of the important shortcomings of LIME is that it uses the perturbation technique which may be difficult to implement or may be even incorrect for some datasets, for example, for images. Moreover, it may provide incorrect results for high-dimensional data of a complex structure. The perturbation technique may generate a disturbed dataset especially when dealing with image data. A slight change in the data can lead to significant changes in images, often losing their meaning. Examples and an analysis of this pitfall as well as other pitfalls of LIME are considered in Molnar et al. (2020). The dual representation proposed in the study does not deal with images and allows us to overcome this difficulty. Another problem is that points generated in accordance with the perturbation technique may be located out of the training point domain, i.e., these points can be viewed as out-of-domain (OOD) data. This case is shown in Figure 1 where training points and generated points are depicted by small circles and by diamonds, respectively. The explained point is depicted by the triangle. A machine learning black-box model learned on points from the training domain may provide quite incorrect predictions for generated points which are outside of the domain. As a result, the approximating linear function constructed by using the generated points may be also incorrect.

One of the shortcomings of SHAP is that it is also computationally expensive when there is a large number of features due to considering all possible coalitions whose number is 2^m, where m is the number of features. Therefore, the computational time grows exponentially. Several simplifications and approximations have been proposed to overcome this difficulty (Strumbelj and Kononenko, 2010, 2011, 2014; Utkin and Konstantinov, 2022). However, they do not cardinally solve the problem of high-dimensional data. Moreover, there is another difficulty of SHAP, which is rarely mentioned. According to SHAP, the black-box model prediction is computed for instances composed from subsets of features and some values of removing features introduced by using some rules. If to use the example depicted at Figure 1, then new instances in SHAP may be located inside or outside the ring bounding the training data domain where the black-box model provides incorrect predictions.

To partially solve the above problems, we propose a new explanation method which is based on applying two approaches: the *convex hull* of training data and the *duality* concept. The convex hull machine learning methods (Yousefzadeh, 2020) analyze relationship between a convex hull of a training set and the decision boundaries for test instances. The duality is a fundamental concept in various field. We use the dual representation of data assuming the linear space in the local area around the explainable instance.

The idea behind the proposed method is very simple. We propose to find the convex hull of a subset of training data consisting on K instances which are close to the explainable instance. By using extreme points of the corresponding convex polytope, each point inside the convex hull can be expressed through the linear combination of the extreme points. Coefficients λ of the linear combination are proposed to be regarded as a new feature vector which determines the corresponding point. They can be viewed as probabilities defined in the unit simplex of probabilities. Since the coefficients belong to the unit simplex, then they can be uniformly generated from the simplex such that each dual feature vector λ corresponds to the feature vector in the Euclidean space (the feature space of training data). A generated feature vector in the Euclidean space is computed through extreme points of the convex hull. As a result, we get a new dual dataset which generates instances

in a local area around the explainable instance. The surrogate linear model is constructed by using this new dual dataset whose elements may have a smaller dimension defined by K or by the number of extreme points of the convex hull. Hence, we get important elements of the generated vectors of coefficients. Due to the linear representation of the surrogate (explanation) model, the important features in the Euclidean space can be simply computed from the important dual coefficients of the linear combinations by means of solving a simple optimization problem.

Another important idea behind the proposed dual representation is to consider the example-based explanation. It turns out that the dual explanation inherently leads to the example-based explanation when we study how each dual feature λ_i contributes into predictions. The contribution can be determined by applying well-known surrogate methods, for example, LIME or the neural additive model (NAM) (Agarwal et al., 2021), but the corresponding surrogate models are constructed for features λ but not for initial features.

For the local explanation, we construct the convex hull by using only a part of training data. Though the same algorithm can be successfully applied to the global explanation. In this case, the convex hull covers the entire dataset.

Our contributions can be summarized as follows:

1. A new feature-based explanation method is proposed. It is based on the dual representation of datasets such that generation of new instances in carried out by means of generating points from the uniform distribution in the unit simplex. In other words, the method replaces the perturbation process of feature vectors in the Euclidean space by the uniform generation of points in the unit simplex, which is simpler and is carried out by many well-known algorithms (Rubinstein and Kroese, 2008; Smith and Tromble, 2004). The generation resolves the problem of out-of-domain data and reduces the number of hyperparameters which have to be tuned for perturbing new instances.
2. A new example-based explanation method is proposed. It is again based on the dual representation of datasets and uses well-known explanation models NAM, accumulated local effect (Apley and Zhu, 2020), the linear regression model. The explanation method provides shape function which describe contributions of the dual features into the predictions. In sum, the model chooses the most influential instances among a certain number of nearest neighbors for the explained instance.
3. The proposed methods are illustrated by means of numerical experiments with synthetic and real data. The code of the proposed algorithm can be found in https://github.com/Kozlov992/Dual-Explanation.

The study is organized as follows. Related work can be found in Section 2. A brief introduction to the convex hull, the explanation methods LIME, SHAP, NAM, and example-based methods is given in Section 3. A detailed description of the proposed approach applied to the feature-based explanation and the example-based explanation is available in Section 4. Numerical experiments with synthetic data and real data studying the feature-based explanation are given in Section 5. Section 6 provides numerical examples illustrating example-based explanation. Advantages and limitations of the proposed methods are discussed in Section 7. Concluding remarks can be found in Section 8.

2 Related work

2.1 Local and global explanation methods

The requirement of the black-box model explanation led to development of many explanation methods. A large part of methods follows from the original LIME method (Ribeiro et al., 2016). These methods include ALIME (Shankaranarayana and Runje, 2019), Anchor LIME (Ribeiro et al., 2018), LIME-Aleph (Rabold et al., 2020), SurvLIME (Kovalev et al., 2020), LIME for tabular data (Garreau and von Luxburg, 2020a,b), GraphLIME (Huang et al., 2022), etc.

To generalize the simple linear explanation surrogate model, several neural network models, including NAM (Agarwal et al., 2021), GAMI-Net (Yang et al., 2021), and AxNNs (Chen et al., 2020), were proposed. These models are based on applying the GAM (Hastie and Tibshirani, 1990). Similar explanation models, including Explainable Boosting Machine (Nori et al., 2019) and EGBM (Konstantinov and Utkin, 2021), were developed using the gradient boosting machine.

Another large part of explanation methods is based on the original SHAP method (Strumbelj and Kononenko, 2010) which uses Shapley values (Lundberg and Lee, 2017) as measures of the feature contribution into the black-box model prediction. This part includes FastSHAP (Jethani et al., 2022), Kernel SHAP (Lundberg and Lee, 2017), Neighborhood SHAP (Ghalebikesabi et al., 2021), SHAFF (Benard et al., 2022), TimeSHAP (Bento et al., 2021), X-SHAP (Bouneder et al., 2020), ShapNets (Wang et al., 2021), etc.

Many explanation methods, including LIME and its modifications, are based on perturbation techniques (Fong and Vedaldi, 2019, 2017; Petsiuk et al., 2018; Vu et al., 2019), which stem from the well-known property that contribution of a feature can be determined by measuring how a prediction changes when the feature is altered (Du et al., 2019). The main difficulty of using the perturbation technique is its computational complexity when samples are of the high dimensionality.

Another interesting group of explanation methods, called the example-based explanation methods (Molnar, 2019), is based on selecting influential instances from a training set having the largest impact on the predictions and its comparison with the explainable instance. Several approaches to the example-based method implementation were considered in Adhikari et al. (2019), Cai et al. (2019), Chong et al. (2022), Crabbe et al. (2021), and Teso et al. (2021).

In addition to the aforementioned methods, there are a huge number of other approaches to solving the explanation problem, for example, Integrated Gradients (Sundararajan et al., 2017), and Contrastive Examples (Dhurandhar et al., 2018). Detailed surveys of many methods can be found in Adadi and Berrada (2018), Arrieta et al. (2020), Bodria et al. (2023), Burkart and Huber (2021), Carvalho et al. (2019), Islam et al. (2022), Guidotti et al. (2019), Li et al. (2022), Rudin (2019), and Rudin et al. (2021).

2.2 Convex hull methods and the convex duality concept

Most papers considering the convex hull methods study the relationship between location of decision boundaries and convex hulls of a training set. The corresponding methods are presented in Chau et al. (2013), El Mrabti et al. (2024), Gu et al. (2020), Nemirko and Dula (2021a), Nemirko and Dula (2021b), Renwang et al. (2022), Rossignol et al. (2024), Singh and Kumar (2021), Wang et al. (2013), Yousefzadeh (2020), and Zhang X. et al. (2021). Boundary of the dataset's convex hull is studied in Balestriero et al. (2021) to discriminate interpolation and extrapolation occurring for a sample. Efficient algorithms for efficient computation of the convex hull for training data are presented in Khosravani et al. (2016).

The concept of duality was also widely used in machine learning models starting from duality in the support vector machine and its various modifications (Bennett and Bredensteiner, 2000; Zhang, 2002). This concept was successfully applied to some types of neural networks (Ergen and Pilanci, 2020, 2021), including GANs (Farnia and Tse, 2018), to models dealing with the high-dimensional data (Yao et al., 2018).

At the same time, the aforementioned approaches did not apply to explanation models. Concepts of the convex hull and the convex duality may be a way to simplify and to improve the explanation models.

3 Preliminaries

3.1 Convex hull

According to Rockafellar (1970), a domain produced by a set of instances as vectors in Euclidean space is convex if a straight line segment that joins every pair of instances belonging to the set contains a vector belonging to the domain. A set $\mathcal{S}$ is convex if, for every pair, $\mathbf{u}, \mathbf{v} \in \mathcal{S}$, and all $\lambda \in [0,1]$, the vector $(1-\lambda)\mathbf{u} + \lambda\mathbf{v}$ belongs to $\mathcal{S}$.

Moreover, if $\mathcal{S}$ is a convex set, then for any $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_t$ belonging to $\mathcal{S}$ and for any non-negative numbers $\lambda_1, ..., \lambda_t$ such that $\lambda_1 + ... + \lambda_t = 1$, the sum $\lambda_1\mathbf{x}_1 + ... + \lambda_t\mathbf{x}_t$ is called a convex combination of $\mathbf{x}_1, ..., \mathbf{x}_t$. The *convex hull* or *convex envelope* of set $\mathcal{X}$ of instances in the Euclidean space can be defined in terms of convex sets or convex combinations as the minimal convex set containing $\mathcal{X}$, or the intersection of all convex sets containing $\mathcal{X}$, or the set of all convex combinations of instances in $\mathcal{X}$.

3.2 LIME, SHAP, NAM, and example-based methods

Let us briefly introduce the most popular explanation methods.

LIME (Ribeiro et al., 2016) proposes to approximate a black-box explainable model, denoted as f, with a simple function g in the vicinity of the point of interest $\mathbf{x}$, whose prediction by means of f has to be explained, under condition that the approximation function g belongs to a set of explanation models G, for example, linear models. To construct the function g, a new dataset consisting of generated points around $\mathbf{x}$ is constructed with predictions computed be means of the black-box model. Weights $w_{\mathbf{x}}$ are assigned to new instances in accordance with their proximity to point $\mathbf{x}$ by using a distance metric, for example, the Euclidean distance. The explanation function g is obtained by solving the following optimization problem:

$$\arg\min_{g \in G} L(f, g, w_{\mathbf{x}}) + \Phi(g). \tag{1}$$

Here, L is a loss function, for example, mean squared error, which measures how the function g is close to function f at point $\mathbf{x}$; $\Phi(g)$ is the model complexity. A local linear model is the result of the original LIME such that its coefficients explain the prediction.

Another approach to explaining the black-box model predictions is *SHAP* (Lundberg and Lee, 2017; Strumbelj and Kononenko, 2010), which is based on a concept of the Shapley values (Shapley, 1953) estimating contributions of features to the prediction. If we explain prediction $f(\mathbf{x}_0)$ from the model at a local point $\mathbf{x}_0$, then the i-th feature contribution is defined by the Shapley value as

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right], \tag{2}$$

where $f(S)$ is the black-box model prediction under condition that a subset S of the instance $\mathbf{x}_0$ features is used as the corresponding input; N is the set of all features.

It can be seen from Equation 2 that the Shapley value ϕ_i can be regarded as the average contribution of the i-th feature across all possible permutations of the feature set. The prediction $f(\mathbf{x}_0)$ can be represented by using Shapley values as follows (Lundberg and Lee, 2017; Strumbelj and Kononenko, 2010):

$$f(\mathbf{x}_0) = \mathbb{E}[f(\mathbf{x})] + \sum_{j=1}^{m} \phi_j. \tag{3}$$

To generalize LIME, *NAM* was proposed in Agarwal et al. (2021). It is based on the generalized additive model of the form $y(\mathbf{x}) = g_1(x_1) + ... + g_m(x_m)$ (Hastie and Tibshirani, 1990) and consists of m neural networks such that a single feature is fed to each subnetwork and each network implements function $g_i(x_i)$, where g_i is a univariate shape function with $\mathbb{E}(g_i) = 0$. All networks are trained jointly using backpropagation and can learn arbitrarily complex shape functions (Agarwal et al., 2021). The loss function for training the whole neural network is of the form:

$$L = \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{m} g_k(x_k^{(i)}) \right)^2, \tag{4}$$

where $x_k^{(i)}$ is the k-th feature of the i-th instance; n is the number of training instances.

The representation of results in NAM in the form of shape functions can be considered in two ways. On the one hand, the functions are more informative, and they show how features contribute into a prediction. On the other hand, we often need to have a single value of the feature contribution which can be obtained by computing an importance measure from the obtained shape function.

NAM significantly extends the flexibility of explanation models due to possibility to implement arbitrary functions of features by means of neural networks.

According to Molnar (2019), an instance or a set of instances are selected in *example-based explanation methods* to explain the model prediction. In contrast to the feature importance explanation (LIME, SHAP), the example-based methods explain a model by selecting instances from the dataset and do not consider features or their importance for explaining. In the context of obtained results, the example-based methods are represented by influential instances (points from the training set that have the largest impact on the predictions) and by prototypes (representative instances from the training data). It should be noted that instances used for explanation may not belong to a dataset and are combinations of instances from the dataset or some points in the dataset domain. The well-known method of K nearest neighbors can be regarded as an example-based explanation method.

4 Materials and methods

4.1 Dual explanation

Let us consider the method for dual explanation. Suppose that there is a dataset $\mathcal{T} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_t, y_t)\}$ of t points $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i = (x_1^{(i)}, ..., x_m^{(i)}) \in \mathcal{X} \subset \mathbb{R}^m$ is a feature vector consisting of m features, y_i is the observed output for the feature vector $\mathbf{x}_i$ such that $y_i \in \mathbb{R}$ in the regression problem and $y_i \in \{1, 2, ..., C\}$ in the classification problem with C classes. It is assumed that output y of an explained black-box model is a function $f(\mathbf{x})$ of an associated input vector $\mathbf{x}$ from $\mathcal{X}$.

To explain an instance $\mathbf{x}_0 \in \mathcal{X}$, an interpretable surrogate model g for the black-box model f is trained in a local region around $\mathbf{x}_0$. It is carried out by generating a new dataset $\mathcal{S}$ of n perturbed samples in the vicinity of the point of interest $\mathbf{x}_0$ similarly to LIME. Samples are assigned by weights $w_{\mathbf{x}}$ in accordance with their proximity to the point $\mathbf{x}$. By using the black-box model, output values y are obtained as function f of generated instances. As a result, dataset $\mathcal{S}$ consists of n pairs $(\mathbf{x}_i, f(\mathbf{x}_i))$, $i = 1, ..., n$. Interpretable surrogate model g is now trained on $\mathcal{S}$. Many explanation methods such as LIME and SHAP are based on applying the linear regression function

$$g(\mathbf{x}) = a_1 x_1 + ... + a_m x_m = \mathbf{a}\mathbf{x}^{\mathrm{T}}, \tag{5}$$

as an interpretable model because each coefficient a_i in g quantifies how the i-th feature impacts on the prediction. Here $\mathbf{a} = (a_1, ..., a_m)$. It should be noted that the domain of set $\mathcal{S}$ coincides with the domain of set $\mathcal{T}$ in the case of the global explanation.

Let us consider the convex hull $\mathcal{P}$ of a set of K nearest neighbors of instance $\mathbf{x}_0$ in the Euclidean space. The convex hull $\mathcal{P}$ forms a convex polytope with d vertices or extreme points $\mathbf{x}_i^*$, $i = 1, ..., d$. Then, each point $\mathbf{x} \in \mathcal{P}$ is a convex combination of d extreme points:

$$\mathbf{x} = \sum_{i=1}^{d} \lambda_i \mathbf{x}_i^*, \text{ where } \lambda_i \geq 0, \ \sum_{i=1}^{d} \lambda_i = 1. \tag{6}$$

This implies that we can uniformly generate a vector in the unit simplex of possible vectors λ consisting of d coefficients $\lambda_1, ..., \lambda_d$, denoted Δ^{d-1}. In other words, we can consider points in the unit simplex Δ^{d-1} and construct a new dual dataset $\mathcal{D} = \{(\lambda^{(1)}, z_1), ..., (\lambda^{(n)}, z_n)\}$, which consists of vectors $\lambda^{(j)} = (\lambda_1^{(j)}, ..., \lambda_d^{(j)})$, and the corresponding values $z_j, j = 1, ..., n$, computed by using the black-box model f as follows:

$$z_j = f\left(\sum_{i=1}^{d} \lambda_i^{(j)} \mathbf{x}_i^*\right), \tag{7}$$

i.e., z_j is a prediction of the black-box model when its input is vector $\sum_{i=1}^{d} \lambda_i^{(j)} \mathbf{x}_i^*$.

In sum, we can train the "dual" linear regression model (the surrogate model) for explanation on dataset $\mathcal{D}$, which is of the form:

$$h(\lambda) = b_1 \lambda_1 + ... + b_d \lambda_d = \mathbf{b}\lambda^{\mathrm{T}}, \tag{8}$$

where $\mathbf{b} = (b_1, ..., b_d)$ is the vector of coefficients of the "dual" linear regression model.

The surrogate model can be trained by means of LIME or SHAP with the dual dataset $\mathcal{D}$.

Suppose that we have trained the function $h(\lambda)$ and computed coefficients $b_1, ..., b_d$. The next question is how to transform these coefficients to coefficients $a_1, ..., a_m$ which characterize the feature contribution into the prediction. In the case of the linear regression, coefficients of function $g(\mathbf{x}) = a_1 x_1 + ... + a_m x_m$ can be found from the condition:

$$g\left(\sum_{i=1}^{d} \lambda_i^{(j)} \mathbf{x}_i^*\right) = h(\lambda_j), \tag{9}$$

which has to be satisfied for all generated λ_j. This obvious condition means that predictions of the "primal" surrogate model with coefficients $a_1, ..., a_m$ has to coincide with predictions of the "dual" model.

Introduce a matrix consisting of extreme points

$$\mathbf{X} = \left(\mathbf{x}_i^{*\mathrm{T}}\right)_{i=1}^{d}. \tag{10}$$

Note that, $\lambda_i = 1$ and $\lambda_j = 0, j \neq i$, for the i-th extreme point. This implies that the condition (Equation 9) can be rewritten as

$$g\left(\mathbf{x}_i^*\right) = h(0, ..., 1_i, ..., 0) = b_i. \tag{11}$$

By using Equation 5, we get

$$g\left(\mathbf{x}_i^*\right) = \mathbf{a}\mathbf{x}_i^{*\mathbf{T}} = b_i. \tag{12}$$

Hence, there holds

$$\mathbf{a}\mathbf{X} = \mathbf{b}. \tag{13}$$

It follows from the above that

$$\mathbf{a} = \mathbf{X}^{-1}\mathbf{b}, \tag{14}$$

where $\mathbf{X}^{-1}$ is the pseudoinverse matrix.

Generally, the vector $\mathbf{a}$ can be computed by solving the following unconstrained optimization problem:

$$\mathbf{a}_{opt} = \arg\min_{\mathbf{a} \in \mathbb{R}^m} \|\mathbf{a}\mathbf{X} - \mathbf{b}\|^2. \tag{15}$$

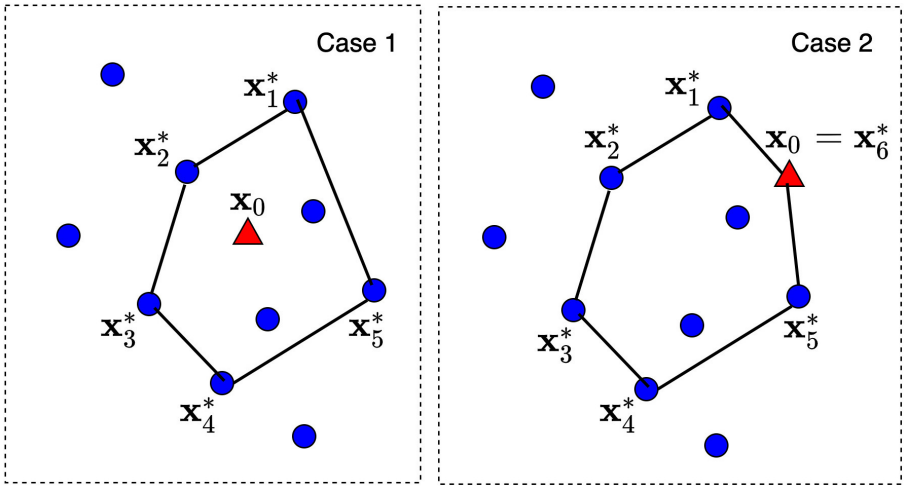

FIGURE 2
Two cases of the explained point location and the convex polytops constructed from K nearest neighbors.

Require: Training set $\mathcal{T}$; the black-box model f; explainable point $\mathbf{x}_0$; the number of nearest neighbors K

Ensure: Important features of $\mathbf{x}_0$ (vector $\mathbf{a} = (a_1, \ldots, a_m)^T$ of the linear surrogate model coefficients)

1: Determine a set $\mathcal{T}_K$ of K nearest neighbors for $\mathbf{x}_0$ adding $\mathbf{x}_0$ itself
2: Construct the largest convex hull $\mathcal{P}$ of $\mathcal{T}_K$
3: Find extreme points of $\mathcal{P}$ and their number $d \leq K+1$
4: Generate uniformly n points $\lambda^{(j)}$, $j = 1, \ldots, n$, from the unit simplex Δ^{d-1}
5: Find predictions z_i of the black-box model in accordance with associated input $\sum_{i=1}^{d} \lambda_i^{(j)} \mathbf{x}_i^*$ for all $i = 1, \ldots, n$
6: Construct a new dual dataset $\mathcal{D} = \{(\lambda^{(1)}, z_1), \ldots, (\lambda^{(n)}, z_n)\}$
7: Train the linear regression (Equation 8) on dataset $\mathcal{D}$ and find the vector of coefficients $\mathbf{b} = (b_1, \ldots, b_d)^T$
8: Find vector $\mathbf{a}$ by solving optimization problem (Equation 15)

Algorithm 1. The dual explanation algorithm.

In the original LIME, perturbed instances are generated around $\mathbf{x}_0$. One of the important advantages of the proposed dual approach is the opportunity to avoid generating instances in accordance with a probability distribution with parameters and to generate only uniformly distributed points $\lambda^{(j)}$ in the unit simplex Δ^{d-1}. Indeed, if we have image data, then it is difficult to perturb pixels or superpixels of images. Moreover, it is difficult to determine parameters of the generation to cover instances from different classes. According to the dual representation, after generating vectors $\lambda^{(j)}$, new vectors $\mathbf{x}_j$ are computed by using Equation 6. This is similar to the mixup method (Zhang et al., 2018) to some extent that generates new samples by linear interpolation of multiple samples and their labels. However, in contrast to the mixup method, the prediction is obtained as the output of the black-box model (see Equation 7), but not as the convex combination of one-hot label encodings. Another important advantage is that instances corresponding to the generated set $\mathcal{D}$ are totally included in the domain of the dataset $\mathcal{T}$. This implies that we do not get anomalous predictions $f(\mathbf{x}_i)$ when generated $\mathbf{x}_i$ is far from the domain of the dataset $\mathcal{T}$.

Another question is how to choose the convex hull of the predefined size and, hence, how to determine extreme points $\mathbf{x}_i^*$ of the corresponding convex polytope. The problem is that the convex hull has to include some number of points from dataset $\mathcal{T}$ and the explained point $\mathbf{x}_0$. Let us consider K nearest neighbors around $\mathbf{x}_0$ from $\mathcal{T}$, where K is a tuning parameter satisfying condition $K \geq d$. The convex hull is constructed on these $K+1$ points (K points from $\mathcal{T}$ and one point $\mathbf{x}_0$). Then, there are d points among K nearest neighbors which define a convex polytope and can be regarded as its extreme points. It should be noted that d depends on the dataset analyzed. Figure 2 illustrates two cases of the explained point location and the convex polytopes constructed from $K = 7$ nearest neighbors. The dataset consists of 10 points depicted by circles. A new explained point $\mathbf{x}_0$ is depicted by the red triangle. In Case 1, point $\mathbf{x}_0$ lies in the largest convex polytope with $d = 5$ extreme points $\mathbf{x}_1^*, ..., \mathbf{x}_5^*$ constructed from seven nearest neighbors. The largest polytope is taken in order to envelop as large as possible points from the dataset. In Case 2, point $\mathbf{x}_0$ lies outside the convex polytope constructed from nearest neighbors. Therefore, this point is included into the set of extreme points and $d \leq K+1$. As a result, we have $d = 6$ extreme points $\mathbf{x}_1^*, ..., \mathbf{x}_5^*, \mathbf{x}_6^* = \mathbf{x}_0$.

To identify whether the newly added point can be expressed as convex combination of the existing vectors, the Farka's lemma (Dinh and Jeyakumar, 2014) can be applied.

Points $\lambda^{(j)}$ from the unit simplex Δ^{d-1} are randomly selected in accordance with the uniform distribution over the simplex. This procedure can be carried out by means of generating random

numbers in accordance with the Dirichlet distribution (Rubinstein and Kroese, 2008). There are also different approaches to generate points from the unit simplex (Smith and Tromble, 2004).

Finally, we write Algorithm 1 implementing the proposed method.

Figure 3 illustrates steps of the algorithm for explanation of a prediction provided by a black-box model at the point depicted by the small triangle. Points of the dataset are depicted by small circles. The training dataset $\mathcal{T}$ and the explained point are shown in Figure 3A. Figure 3B shows set $\mathcal{T}_K$ of $K = 13$ nearest points such that only two points $(0.05, 0.5)$ and $(1.0, 0.1)$ from training set $\mathcal{T}$ do not belong to set $\mathcal{T}_K$. The convex hull and the corresponding extreme points are shown in Figure 3C. Points uniformly generated in the unit simplex are depicted by means of small crosses in Figure 3D. It is interesting to point out that the generated points are uniformly distributed in the unit simplex, but not in the convex polytope as it is follows from Figure 3D. We uniformly generate vectors λ, but the corresponding vectors $\mathbf{x}$ are not uniformly distributed in the polytope. One can see from Figure 3D that generated points in the initial (primal) feature space tend to be located in the area where the density of extreme points is largest. This is a very interesting property of the dual representation. It means that the method takes into account the concentration of training points and the probability distribution of the instances in the dataset.

The difference between points generated by means of the original LIME and the proposed method is illustrated in Figure 4 where the left picture (Figure 4A) shows a fragment of Figure 1 and the right picture (Figure 4B) illustrates how the proposed method generates instances.

The proposed method requires finding all the extreme points (vertices) of the convex hull of a given point $\mathbf{x}_0 \in \mathbb{R}^d$ and its nearest neighbors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}$. When the dimension d is small, these extreme points can be computed in time $O(2^{O(d \log d)} n^2) = O(n^2)$ (Ottmann et al., 2001). In general, determining whether $\mathbf{x}_i$ is an extreme point can be done by checking the condition

$$conv(\{\mathbf{x}_j\}_{j=0}^{n-1}) \neq conv(\{\mathbf{x}_j\}_{j=0}^{n-1} \setminus \{\mathbf{x}_i\}), \tag{16}$$

where $conv(P)$ denotes the convex hull of the set P.

The above condition is equivalent to solving a feasibility problem that can be formulated as a linear program. This linear program involves n variables and $n + d$ constraints and can be solved using the interior-point method described in Vaidya (1989). For each point, the time complexity of this procedure is $O((n + d)^{3/2} n \log(n))$, resulting in an overall complexity of

$$O((n + d)^{3/2} n^2 \log(n)). \tag{17}$$

In the extreme case, when $d \gg 1$, we can use the AVTA algorithm (Awasthi et al., 2018) to approximate the set of extreme points of $\{\mathbf{x}_j\}_{j=0}^{n-1}$. This algorithm has the time complexity $O(n^2(d + t^{-2}))$, where $t \in (0, 1)$. The approximation becomes more precise as $t \to 0$.

The dual approach can work best when applied to analysis of potential outliers. In that regard, the generation procedure proposed in the study is more robust than the one used in LIME. By choosing the generation region as the convex hull of the explained point nearest neighbors, we reduce the likelihood of creating additional samples that fail to align with the original data distribution. As for hyperparameters, the number of nearest neighbors used to construct the convex hull for the explained point largely depends on the user's preferences and the nature of analyzed data. We can stop incorporating additional neighbors when a certain threshold is reached, such as when the next nearest neighbor is considerably more distant compared to the previous ones. Furthermore, we can choose to exclude a new neighbor if its data features clearly indicate that it would not contribute much to the analysis of the explained point. The number of points to generate can be taken as $k \cdot n$, where k is a real number and n is the number of selected neighbors. By default, $k = 3$. This implies that we can increment the number of generated points until we observe the convergence of dual coefficients. We can also modify the distribution type employed for creating the dual dataset. For instance, if we take new points to be generated mostly in close proximity to the explained point $\mathbf{x} = (x_1, ..., x_d)$, we can sample the points from the Dirichlet distribution with concentration parameters $\alpha_i = 1 + t \cdot x_i$, where $t > 0$.

4.2 Example-based explanation and NAM

It turns out that the proposed method for the dual explanation inherently leads to the example-based explanation. An example-based explainer justifies the prediction on the explainable instance by returning instances related to it. Let us consider the dual representation (Equation 8). If we normalize coefficients $\mathbf{b} = (b_1, ..., b_d)$ as

$$v_i = \frac{b_i}{\sum_{j=1}^{d} b_j}, \tag{18}$$

then new coefficients $(v_1, ..., v_d)$ quantify how extreme points $(\mathbf{x}_1^*, ..., \mathbf{x}_d^*)$ associated with $(\lambda_1, ..., \lambda_d)$ impact on the prediction. The greater the value of v_i, the greater contribution of $\mathbf{x}_i^*$ into a prediction. Hence, the linear combination of extreme points

$$\mathbf{x} = \sum_{i=1}^{d} v_i \mathbf{x}_i^* \tag{19}$$

allows us to get an instance $\mathbf{x}$ explaining $\mathbf{x}_0$.

An outstanding approach considering convex combinations of instances from a dataset as the example-based explanation was proposed in Crabbe et al. (2021). In fact, we came to the similar example-based explanation by using the dual representation and constructing linear regression surrogate model for new variables $(\lambda_1, ..., \lambda_d)$.

The example-based explanation may be very useful when we apply NAM (Agarwal et al., 2021) for explaining the black-box prediction. By using dual dataset $\mathcal{D} = \{(\lambda^{(1)}, z_1), ..., (\lambda^{(n)}, z_n)\}$, we train NAM consisting of d subnetworks such that each subnetwork implements the shape function $h_i(\lambda_i)$. Figure 5 illustrates a scheme of training NAM. Each generated vector λ is fed to NAM such that each its variable λ_i is fed to a separate neural subnetwork. For the same vector λ, the corresponding instance $\mathbf{x}$ is computed by using Equation 6, and it is fed to the black-box model. The loss function

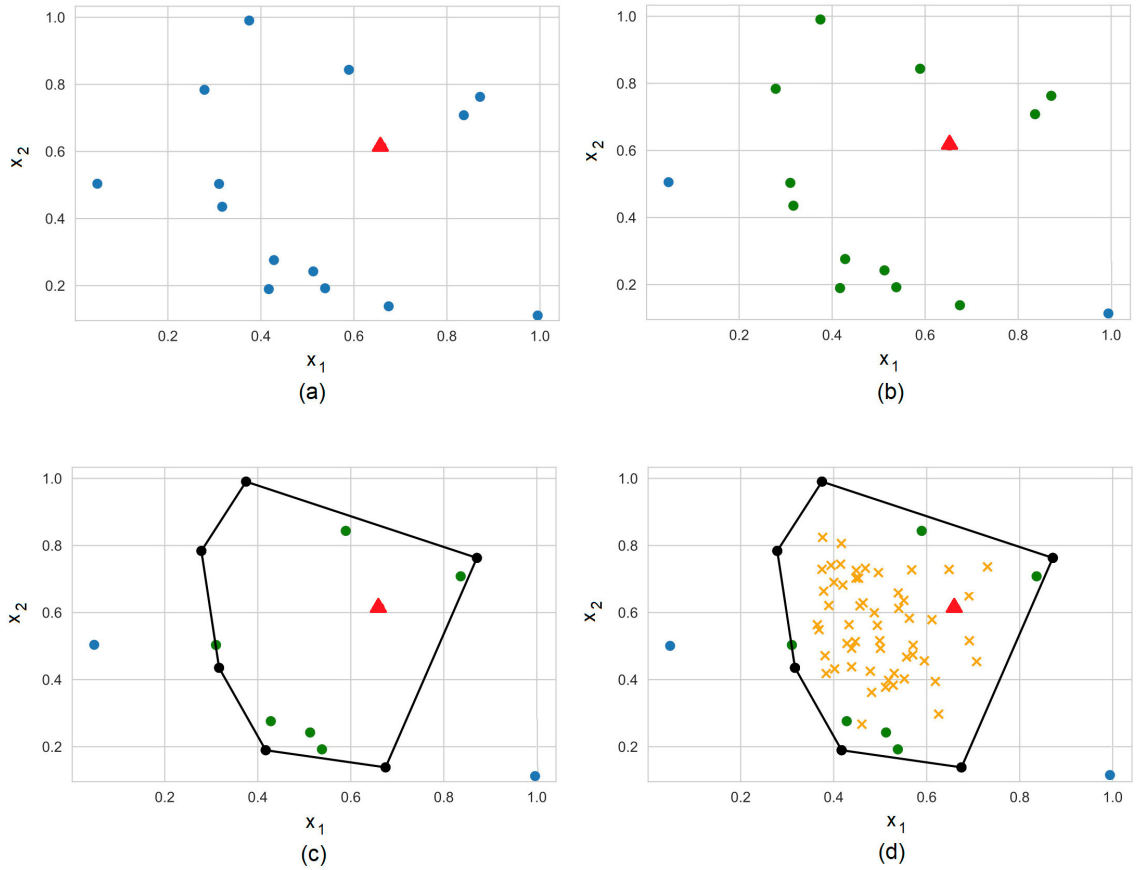

FIGURE 3
Steps of the algorithm for explanation of a prediction provided by a black-box model at the point depicted by the small triangle. **(A)** Training dataset (circles) and the explainable example (triangle). **(B)** 13 nearest neighbors. **(C)** The convex hull around the explainable example. **(D)** Generated examples (crosses) in the convex hull.

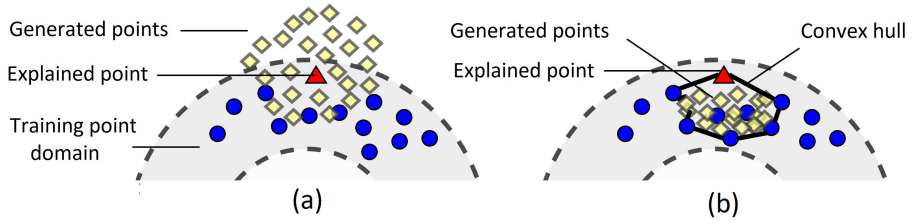

FIGURE 4
Generated points in the original LIME **(A)** and in the proposed dual method **(B)**.

for training the whole neural network is defined as the difference between the output z of the black-box model and the sum of shape functions $h_1, ..., h_d$ implemented by neural subnetworks for the corresponding vector λ, i.e., the loss function L is of the form:

$$L = \sum_{i=1}^{n} \left(z_i - \sum_{k=1}^{d} h_k \left(\lambda_k^{(i)} \right) \right)^2 + \alpha R(\mathbf{w}), \tag{20}$$

where $\lambda_k^{(i)}$ is the k-th element of vector $\boldsymbol{\lambda}^{(i)}$; R is a regularization term with the hyperparameter α which controls the strength of the regularization; $\mathbf{w}$ is the vector of the neural network training parameters.

The main difficulty of using the NAM results, i.e., shape functions $h_k(\lambda_k)$, is how to interpret the shape functions for explanation. However, in the context of the example-based explanation, this difficulty can be simply resolved. First, we study how a shape function can be represented by a single value characterizing the importance of each variable λ_k, $k = 1, ..., d$. The shape function is similar to the partial dependence plot (Friedman, 2001; Molnar, 2019) to some extent. The

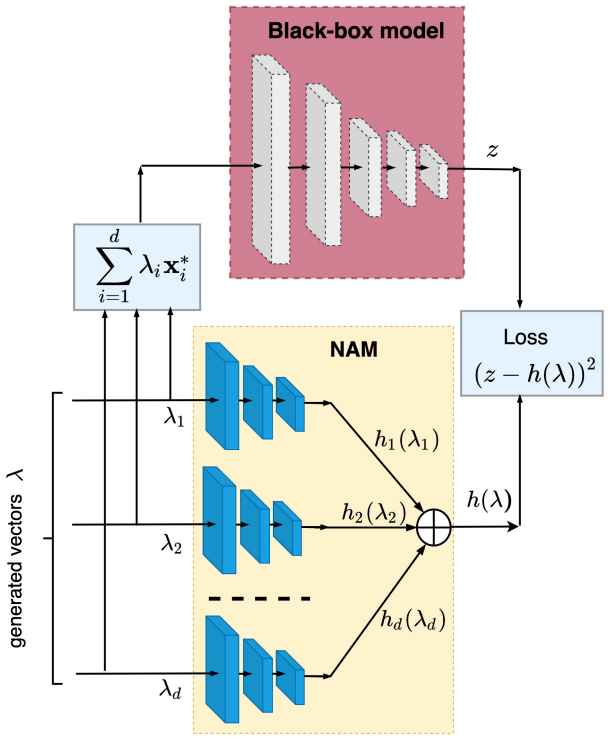

FIGURE 5
Scheme of training NAM on the generated set of random vectors λ.

importance of a variable (λ_k) can be evaluated by studying how rapidly the shape function, corresponding to the variable, is changed. The rapid change of the shape function says that small changes of the variable significantly change the target values (z). The above implies that we can use the importance measure proposed in Greenwell et al. (2018), which is defined as the deviation of each unique variable value from the average curve. In terms of the dual variables, it can be written as:

$$I(\lambda_k) = \sqrt{\frac{1}{r-1}\sum_{i=1}^{r}\left(h_k\left(\lambda_k^{(i)}\right) - \frac{1}{r}\sum_{i=1}^{r} h_k\left(\lambda_k^{(i)}\right)\right)^2}, \qquad (21)$$

where r is a number of values of each variable λ_k, which are analyzed to study the corresponding shape function.

Normalized values of the importance measures can be regarded as coefficients v_i, $i = 1, ..., d$, in Equation 19, i.e., they show how important each extreme point or how each extreme point can be regarded as an instance which explains instance $\mathbf{x}_0$.

An additional important advantage of the dual representation is that shape functions for all variables λ_k, $k = 1, ..., d$, have the same scale because all variables are in the interval from 0 to 1. This allows us to compare the importance measures $I(\lambda_k)$ without the preliminary scaling which can make results incorrect.

5 Numerical experiments with the feature-based explanation

5.1 Example 1

First, we consider the following simplest example when the black-box model is of the form:

$$f(\mathbf{x}) = 10x_1 - 20x_2 - 2x_3 + 3x_4 + 0x_5 + 0x_6 + 0x_7 + \xi$$
$$= \mathbf{a}\mathbf{x} + \xi, \ \ \xi \sim \mathcal{N}(0, 0.1).$$

Let us estimate the feature importance by using the proposed dual model. We generate $n = 1000$ points $\mathbf{x}_i$, $i = 1, ..., N$, with components uniformly distributed in interval $[0, 1]$, which are explained. For every point $\mathbf{x}_i$, the dual model with $K = 10$ nearest neighbors is constructed by generating 30 vectors $\lambda^{(i)} \in \mathbb{R}^7$ in the unit simplex. By applying Algorithm 1, we compute optimal vector $\mathbf{a}^{(i)} = (a_1, ..., a_7)^{\mathrm{T}}$ for every point $\mathbf{x}_i$. We expect that the mean value $\bar{\mathbf{a}}$ of $\mathbf{a}^{(i)}$ over all $i = 1, ..., N$ should be as close as possible to the true vector of coefficients $\mathbf{a}$ forming function $f(\mathbf{x})$. The corresponding results are shown in Table 1. It can be seen from Table 1 that the obtained vector $\bar{\mathbf{a}}$ is actually close to vector $\mathbf{a}$.

5.2 Example 2

Let us consider another numerical example where the non-linear black-box model is investigated. It is of the form:

$$f(\mathbf{x}) = -x_1^2 + 2x_2 + \xi, \ \ \xi \sim \mathcal{N}(0, 0.05).$$

We take $N = 400$ and generate two sets of points $\mathbf{x}$. The first set contains $\mathbf{x}$ whose features are uniformly generated in the interval $[0, 1]$. The second set consists of $\mathbf{x}$ whose features are uniformly generated in the interval $[15, 16]$. It is interesting to note that the feature x_1 is more important for the case of the second set because x_1^2 rapidly increases whereas x_1^2 decreases when we consider the first set and x_2 is more important in this case.

We take $K = 6$ and generate 30 vectors $\lambda^{(i)}$ uniformly distributed in the unit simplex for every $\mathbf{x}$ to construct the linear model $h(\lambda^{(i)})$. Mean values of the normalized importance of features x_1 and x_2 obtained for the first set are -0.3 and 0.86 and for the second set are -0.95 and 0.37. These results completely coincide with the importance of features considered above for two subsets.

5.3 Example 3

A goal of the following numerical example is to consider a case when we try to get predictions for points lying outside bounds of data on which the black-box model was trained as it is depicted in Figure 1. In this case, the predictions of generated instances may be inaccurate and can seriously affect quality of many explanation models, for example, LIME, which uses the perturbation technique.

The initial dataset consists of $n = 400$ feature vectors $\mathbf{x}_1, ..., \mathbf{x}_n$ such that there holds

$$\mathbf{x}_i = \begin{pmatrix} x_i^{(1)} \\ x_i^{(2)} \end{pmatrix} = \rho \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix}, \qquad (22)$$

TABLE 1 Values of the importance measures in Example 1 in accordance with the explanation approach LR.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
a	10	−20	−2	3	0	0	0
$\bar{\mathbf{a}}$	9.98	−20.01	−2.02	2.97	0.11	−0.02	0.03

where parameter ρ^2 is uniformly distributed in interval $[0, 2^2]$; parameter φ is uniformly distributed in interval $[0, 2\pi]$.

The observed outputs $y_i = f(\mathbf{x}_i)$ are defined as

$$f(\mathbf{x}_i) = \left(x_i^{(1)}\right)^2 + \left(x_i^{(2)}\right)^2 + \xi, \;\; \xi \sim \mathcal{N}(0, 0.05). \tag{23}$$

We use two black-box models: the KNN regressor with $k = 6$ and the random forest consisting of 100 decision trees, implemented by means of the Python Sckit-learn. The above black-box models have default parameters taken from Sckit-learn.

We construct the explanation models at $l = 100$ testing points $\mathbf{x}_{1,test}, ..., \mathbf{x}_{l,test}$ of the form Equation 22, but with parameters ρ^2 uniformly distributed in $[1.9^2, 2^2]$ and φ uniformly distributed in $[0, 2\pi]$. It can be seen from the interval of parameter ρ that a part of generated points can be outside bounds of training data $\mathbf{x}_1, ..., \mathbf{x}_n$. Figure 6 shows the set of instances for training the black-box model and the set of testing instances for evaluation of the explanation models.

The dual model is constructed in accordance with Algorithm 1 using $K = 6$ nearest neighbors. We generate 30 dual vectors $\lambda^{(j)}$ to train the dual model. We also use LIME and generate 30 points having normal distribution $\mathcal{N}(\mathbf{x}_{j,test}, \Sigma)$, where $\Sigma = \text{diag}(0.05, 0.05)$. Every point has a weight generated from the normal distribution with parameter $\nu = 0.01$.

To compare the dual model and LIME, we use the mean squared error (MSE) which measures how predictions of the explanation model $g(\mathbf{x})$ are close to predictions of the black-box model $f(\mathbf{x})$ (KNN or the random forest). It is defined as

$$MSE = \frac{1}{l}\sum_{j=1}^{l}\left(f(\mathbf{x}_{j,test}) - g(\mathbf{x}_{j,test})\right)^2.$$

Values of the MSE measures for the dual explanation model and for the original LIME, when KNN is used as a black-box model, are 0.01 and 0.02, respectively. It can be seen from the results that the dual model provides better results in comparison with LIME because some generated points in LIME are located outside the training domain. Values of the MSE measures for the dual explanation model and for the original LIME, when the random forest is used as a black-box model, are 0.005 and 0.014, respectively.

5.4 Example 4

Let us perform a similar experiment with real data by taking the dataset "Combined Cycle Power Plant Data Set" (https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant) consisting of 9568 instances having 4 features. We use Z-score normalization (the mean is 0 and the standard deviation is 1) for feature vectors from the dataset. Two black-box models implemented by using the KNN regressor with $K = 10$ and the random forest regressor consisting of 100 decision trees. The testing set consisting of $l = 200$ new instances is produced as follows. The convex hull of the training set in the 4-dimensional feature space is determined. Then, vertices of the obtained polytope are computed. Two adjacent vertices $\mathbf{x}_{j_1}$ and $\mathbf{x}_{j_2}$ are randomly selected. Value λ is generated from the uniform distribution on the unit interval. A new testing instance $\mathbf{x}_{j,test}$ is obtained as $\mathbf{x}_{j,test} = \lambda\mathbf{x}_{j_1} + (1-\lambda)\mathbf{x}_{j_2}$. Then, we again select adjacent vertices and repeat the procedure for computing testing instances l times. As a result, we get the testing set $\mathbf{x}_{j,test}$, $j = 1, ..., l$.

The dual model is constructed in accordance with Algorithm 1 using $K = 10$ nearest neighbors. We again generate 30 dual vectors $\lambda^{(j)}$ to train the dual model. We also use LIME and generate 30 points having normal distribution $\mathcal{N}(\mathbf{x}_{j,test}, \Sigma)$, where $\Sigma = \text{diag}(0.05, 0.05, 0.05, 0.05)$. Every point has a weight generated from the normal distribution with parameter $\nu = 0.5$.

Values of the MSE measures for the dual explanation model and for the original LIME, when KNN is used as a black-box model trained on dataset "Combined Cycle Power Plant Data Set", are 84 and 0173, respectively. It can be seen from the results that the dual model provides better results in comparison with LIME because some generated points in LIME are located outside the training domain. Values of the MSE measures for the dual explanation model and for the original LIME, when the random forest is used as a black-box model, are 110 and 282, respectively. One can again see from the above results that the dual models outperform LIME.

6 Numerical experiments with the example-based explanation

6.1 Example 1

We start from the synthetic instances illustrating the dual example-based explanation when NAM is used. Suppose that the explained instance $\mathbf{x}_0$ belongs to a polytope with six vertices $\mathbf{x}_1, ..., \mathbf{x}_6$ ($d = 6$). The black-box model is a function $f(\mathbf{x})$ such that

$$f(\mathbf{x}) = f\left(\sum_{k=1}^{6}\lambda_k\mathbf{x}_k\right) = h(\lambda) = h(\lambda_1, ..., \lambda_6)$$
$$= 15\lambda_1 + 22\lambda_2 + 0\lambda_3 + 40(1-\lambda_4)\sin(3.14\cdot\lambda_4) + 0\lambda_5 + 0\lambda_6. \tag{24}$$

$n = 2,000$ vectors $\lambda^{(i)} \in \mathbb{R}^6$, $i = 1, ..., n$, are uniformly generated in the unit simplex Δ^{6-1}. For each point $\lambda^{(i)}$, the corresponding prediction z_i is computed by using the black-box function $h(\lambda)$. NAM is trained with the learning rate 0.0005, with hyperparameter $\alpha = 10^{-4}$, the number of epochs is 300, and the batch size is 128.

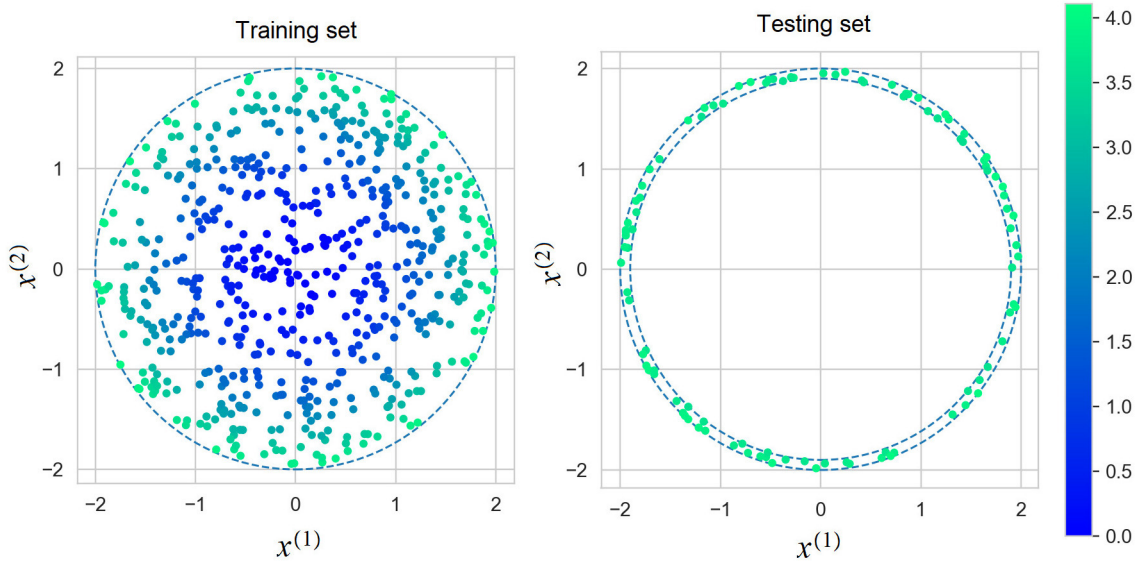

FIGURE 6
Instances for training the black-box models **(the left picture)** and testing points for evaluation of the explanation models **(the right picture)**.

TABLE 2 Values of the importance measures in Example 1 in accordance with explanation approaches: ALE, LR, and NAM.

	Importance measures					
	$I(\lambda_1)$	$I(\lambda_2)$	$I(\lambda_3)$	$I(\lambda_4)$	$I(\lambda_5)$	$I(\lambda_6)$
ALE	0.172	0.259	0.000	0.569	0.000	0.000
LR	0.182	0.245	0.054	0.405	0.062	0.052
NAM	0.157	0.238	0.012	0.569	0.012	0.012

To determine the normalized values of the importance measures $I(\lambda_i)$, $i = 1, ..., 6$, we use three approaches. The first one is to apply the method called accumulated local effect (ALE) (Apley and Zhu, 2020), which describes how features influence the prediction of the black-box model on average. The second approach is to construct the linear regression model (LR) by using the generated points and their predictions obtained by means of the black-box model. The third approach is to use NAM.

The corresponding normalized values of the importance measures for $\lambda_1, ..., \lambda_6$ obtained by means of ALE, LR, and NAM are shown in Table 2. It should be noted that the importance measure $I(\lambda_i)$ can be obtained only for NAM and ALE. However, normalized coefficients of LR can be interpreted in the same way. Therefore, we consider results of these models jointly in all tables. One can see from Table 2 that all methods provide similar relationships between the importance measures $I(\lambda_1)$, $i = 1, ..., 6$. However, LR provides rather large values of $I(\lambda_3)$, $I(\lambda_5)$, $I(\lambda_6)$, which do not correspond to the zero-valued coefficients in Equation 24.

Shape functions illustrating how functions of the generalized additive model depend on λ_i are shown in Figure 7. It can be clearly seen from Figure 7 that the largest importance λ_2 and λ_4 have the highest importance. This implies that the explained instance is interpreted by the fourth and the second nearest instances.

6.2 Example 2

Suppose that the explainable instance $\mathbf{x}_0$ belongs to a polytope with four vertices $\mathbf{x}_1^*, ..., \mathbf{x}_4^*$ ($d = 4$). The black-box model is a function $f(\mathbf{x})$ such that

$$h(\lambda) = \lambda_1^2 + \lambda_1\lambda_2 - \lambda_3\lambda_4 + \lambda_4.$$

$n = 1000$ points $\lambda^{(i)} \in \mathbb{R}^4$, $i = 1, ..., n$, are uniformly generated in the unit simplex Δ^{4-1}. For each point $\lambda^{(i)}$, the corresponding prediction z_i is computed by using the black-box function $h(\lambda)$. NAM is trained with the learning rate 0.0005, with hyperparameter $\alpha = 10^{-6}$, the number of epochs is 300, and the batch size is 128.

Normalized values of $I(\lambda_i)$ obtained by means of ALE, LR, and NAM are shown in Table 3. It can be seen from Table 3 that the obtained importance measures correspond to the intuitive consideration of the expression for $h(\lambda)$. The corresponding shape functions for all features are shown in Figure 8.

6.3 Example 3

Suppose that the explained instance $\mathbf{x}_0$ belongs to a polytope with three vertices $\mathbf{x}_1^*, \mathbf{x}_2^*, \mathbf{x}_3^*$ ($d = 3$):

$$\mathbf{x}_1^* = (-1, -1)^{\mathrm{T}},\ \mathbf{x}_2^* = (0, 2)^{\mathrm{T}},\ \mathbf{x}_3^* = (1, 0)^{\mathrm{T}}.$$

The black-box model has the following function of two features $x^{(1)}$ and $x^{(2)}$:

$$f(\mathbf{x}) = 0.7 \cdot \mathrm{sign}(x^{(1)}) + \mathrm{sign}(x^{(2)})$$

We generate $n = 1,000$ points $\lambda^{(i)} \in \mathbb{R}^3$, $i = 1, ..., n$, which are uniformly generated in the unit simplex Δ^{3-1}. These points correspond to n vectors $\mathbf{x}_i \in \mathbb{R}^2$ defined as

$$\mathbf{x}_i = \lambda_1^{(i)} \cdot (-1, -1)^{\mathrm{T}} + \lambda_2^{(i)} \cdot (0, 2)^{\mathrm{T}} + \lambda_3^{(i)} \cdot (1, 0)^{\mathrm{T}}$$

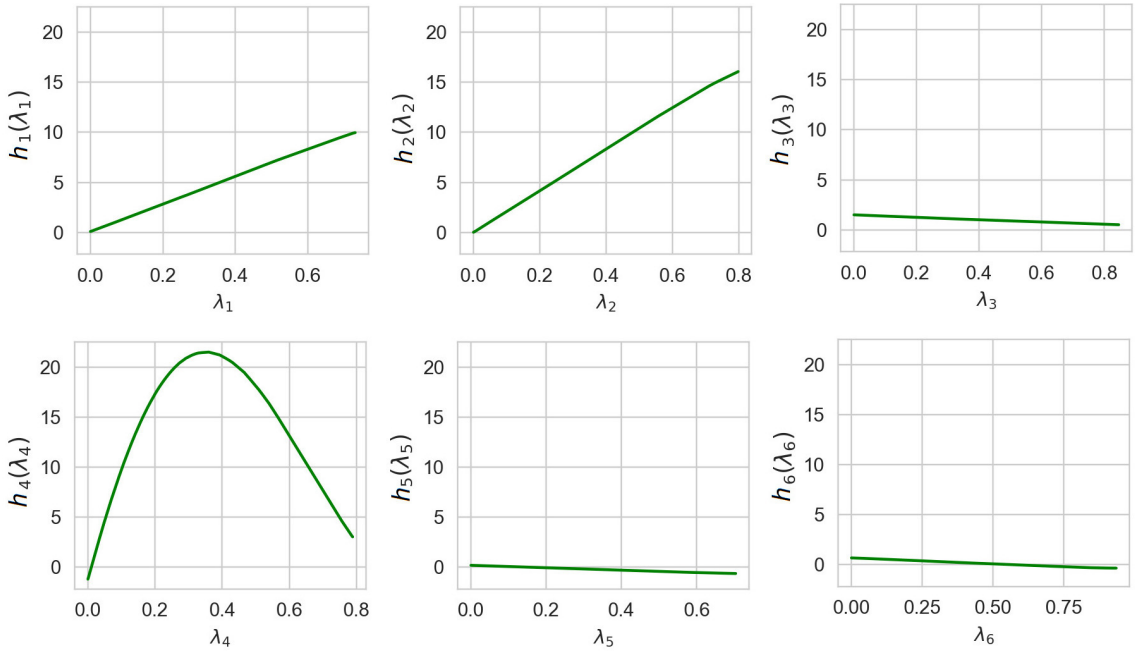

FIGURE 7
Six shape functions obtained in Example 1 for the example-based explanation.

TABLE 3 Values of the importance measures in Example 2 in accordance with three explanation approaches: ALE, LR, and NAM.

	Importance measures			
	$I(\lambda_1)$	$I(\lambda_2)$	$I(\lambda_3)$	$I(\lambda_4)$
ALE	0.392	0.087	0.089	0.432
LR	0.357	0.081	0.112	0.450
NAM	0.306	0.134	0.202	0.358

with the corresponding values of $f(\mathbf{x}_i)$ and shown in Figure 9. It can be seen from Figure 9 that this example can be regarded as a classification task with four classes. Parameters of experiments are the same as in the previous examples, but $\alpha = 0$.

Normalized values of $I(\lambda_i)$ obtained by means of ALE, LR, and NAM are shown in Table 4. It can be seen from Table 4 that the obtained importance measures correspond to the intuitive consideration of the expression for $h(\lambda)$. The corresponding shape functions for all features are shown in Figure 10.

7 Discussion

Let us analyze advantages and limitations of the proposed methods. First, we consider advantages.

1. One of the important advantages is that the proposed methods allow us to replace the perturbation process of feature vectors in the Euclidean space by the uniform generation of points in the unit simplex. Indeed, the perturbation of feature vectors requires to define several parameters, including probability distributions of generation for every feature, and parameters of the distributions. The cases depicted in Figure 1 may lead to incorrect predictions and to an incorrect surrogate model. Moreover, if instances are images, then it is difficult to correctly perturb them. Due to the proposed method, the perturbation of feature vectors is avoided, and it is replaced with uniform generation in the unit simplex, which is simple. The dual approach can be applied to the feature-based explanation as well as to the example-based explanation.
2. The dual representation of data can have a smaller dimension than the initial instances. It depends on K nearest neighbors around the explained instance. As a result, the constructed surrogate dual model can be simpler than the model trained on the initial training set.
3. The dual approach can be also adapted to SHAP to generate the removed features in a specific way.
4. The proposed methods are flexible. We can change the size of the convex hull by changing the number K. It can be applied to different explanation models, for example, to LIME, SHAP, and NAM. The main idea of the adaptation is to use the well-known explanation methods. In particular, LIME can be incorporated into the proposed method by constructing the linear regression for the dual dataset. We can incorporate SHAP for computing the feature contributions of the dual instances $(\lambda^{(i)}, z_i)$. NAM is incorporated to compute the shape functions of features $\lambda_k^{(i)}$, $k = 1, ..., d$. The method can be applied to the local and global explanations. There are different definitions of the global explanation. One of the approaches to define the global explanation, proposed in Ribeiro et al. (2016), is to compute the average feature importance over the feature importances obtained by means of the local explanation for all instances of

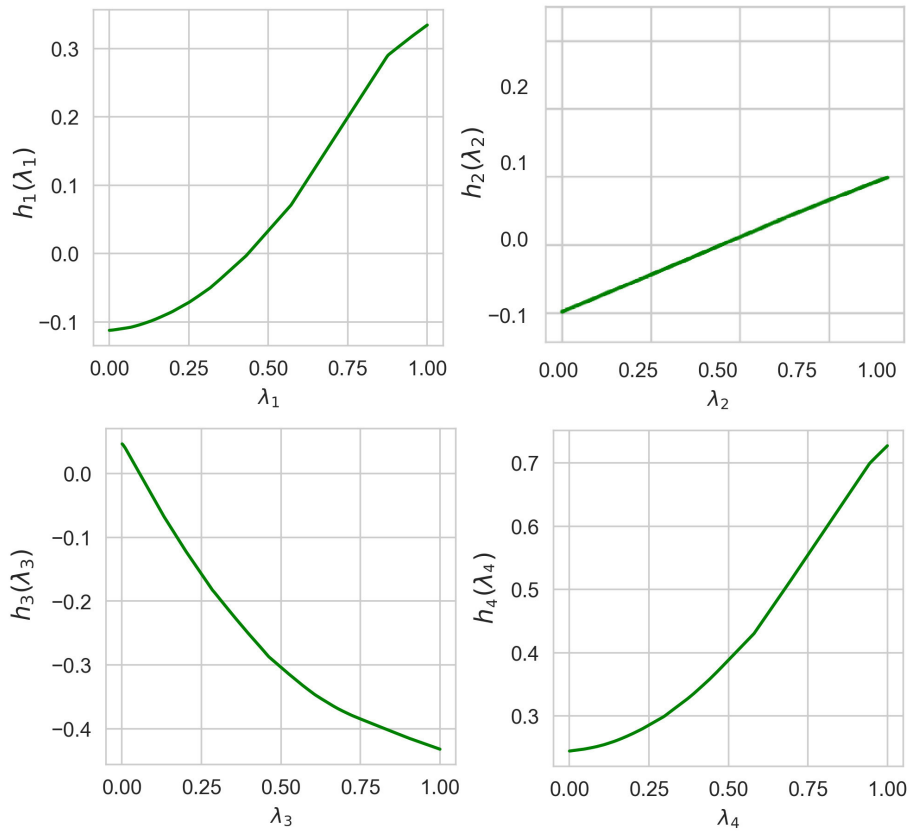

FIGURE 8
Four shape functions obtained in Example 2 for the example-based explanation.

the dataset. This is a computationally difficult problem due to two main factors: (1) constructing a convex hull on the dataset; (2) solving the local explanation problems for all instances in the training set. The first problem can be solved by dividing the whole dataset into subsets with feature vectors that are close in distance construct a convex hull for each subset and solve the "local" problem of global explanation. This can be done, for example, using a decision tree so that leaves of the tree contain close instances. Another way is clustering, for which the assumption is fulfilled that each cluster also contains close instances. The second problem is computationally intensive. Its efficient solution is one of the important directions for further research.

In spite of many advantages of the dual approach, we have to note also its limitations:

1. The advantage of the smaller dimensionality in the dual representation is questionable for the feature-based explanation. If we take a number of extreme points smaller than the data dimensionality, then we restrict the set of generated primal points by some subspace of the initial feature space. This can be a reason of incorrect results. Ways to overcome this difficulty are an interesting direction for further research. However, this limitation does not impact on the example-based explanation because we actually extend the mixup method and try to find influential instances among nearest neighbors.

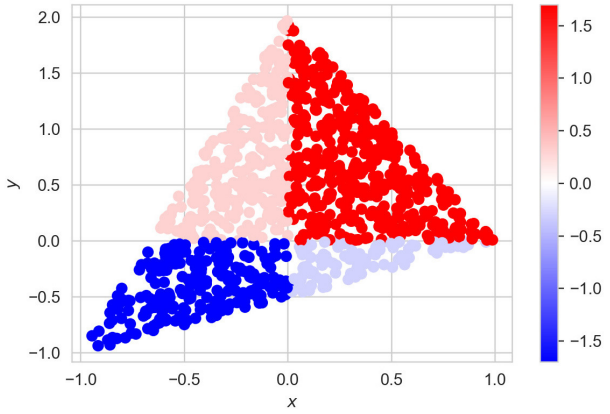

FIGURE 9
Dataset of vectors **x** and the corresponding values of $f(\mathbf{x})$ for Example 3.

2. Another problem is that calculation of vertices of the largest convex hull is a computationally hard problem. This problem does not take place for the example-based explanation when the number of nearest neighbors is smaller than the initial data dimensionality.

In spite of the above limitations, the proposed approach has many interesting properties and can be regarded as the first step for developing various algorithms using dual representation. It can have the biggest impact in medicine, where, on the one hand, high-dimensional data take place, and, on the other hand, predictions (diagnoses) need to be explained to believe in them and choose a desirable treatment.

It has been shown in numerical examples with synthetic data that the proposed method outperforms the separate LIME method in terms of accuracy (see, for example, Sections 5.3, 5.4). One of the reasons is that some generated points in LIME may be located outside the training domain. However, LIME can be regarded as a part of the proposed method when it is used for computing coefficients $\mathbf{b} = (b_1, ..., b_d)$ in the dual representation. This implies that the computation time for explanation using the proposed method may exceed the LIME time. At the same time, instances in the obtained dual dataset may have the smaller dimensionality in comparison with the initial data. In this case, the computation time of the proposed method can be comparable with the LIME time.

8 Conclusion

Feature-based and example-based explanation methods in the framework of the dual feature representation have been presented in the study. The methods directly follow from the dual representation. They can be viewed as a basis for their improvement and the development of other methods within the dual representation.

In the example-based explanation, we used NAM as a neural network tool for explaining predictions under condition of considering the dual dataset with new variables $(\lambda_1, ..., \lambda_d)$. However, there are effective explanation methods different from NAM, which are based on the gradient boosting machine (Nori et al., 2019; Konstantinov and Utkin, 2021). The combination of the proposed approach with these methods is an interesting direction for further research.

Another interesting direction for further research is to study how the proposed approach adapts to the example-based image explanation when K nearest neighbors are not determined by the proximity of original images. The search for efficient adaptation algorithms seems to be a relevant and interesting task.

There are interesting results in the linear programming when the significance of dual variables is related to perturbations of coefficients of the primal constraints (Castillo et al., 2006). This peculiarity can be applied to develop new explanation methods.

It should be noted that many applications have features that are not taken into account in the proposed approach, for example, the presence of multimodal data having different dimensions. Adaptation of the approach and the extensions oriented to specific applications are also important issues for further research. An idea behind the problem solution is to reduce different dimensions to one in the dual data representation.

Adversarial settings can produce a complex cluster structure within the feature space. A significant challenge in such scenarios is addressing out-of-distribution points. The proposed method can handle this problem unlike the LIME. To enhance the robustness, we propose two hyperparameters: the configuration of the Dirichlet distribution and the number of the neighbors to construct the convex hull. Proper adjustment of these hyperparameters has the potential to enhance the method's robustness.

The proposed results are fundamental. They are illustrated only with synthetic data or well-known real datasets. Therefore, we do not use personal data which require to implement robust

TABLE 4 Values of the importance measures in Example 3 in accordance with three explanation approaches: ALE, LR, and NAM.

	Importance measures		
	$I(\lambda_1)$	$I(\lambda_2)$	$I(\lambda_3)$
ALE	0.411	0.395	0.194
LR	0.430	0.310	0.260
NAM	0.499	0.338	0.163

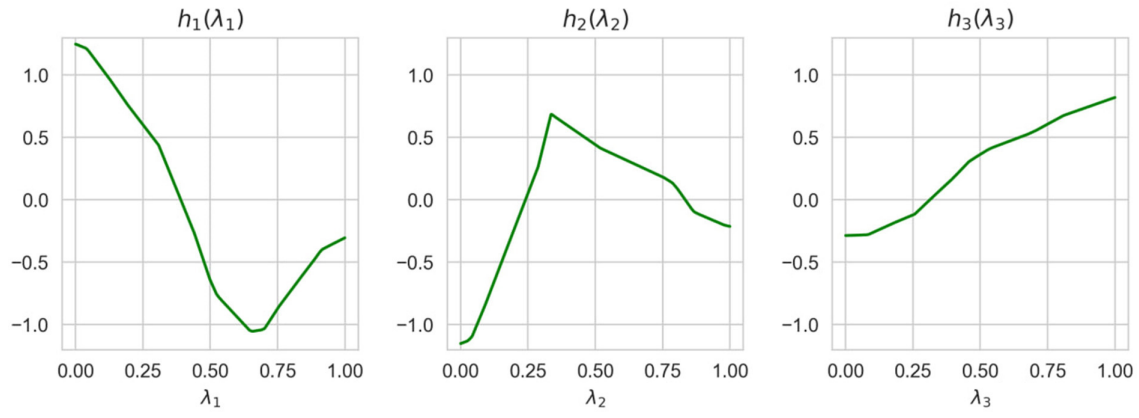

FIGURE 10
Three shape functions obtained in Example 3 for the example-based explanation.

security measures to safeguard sensitive information and prevent unauthorized access. It should be noted that one of the important goals of the proposed results is to provide explanations for the machine learning model decisions and actions making the models transparent. As a result, users have a clear understanding of how the black-box model operates and the factors influencing its outputs. The proposed method belongs to the field of explainable artificial intelligence; thus, we have contributed to the development of transparent and reliable AI systems. Methods of the prediction explanation can improve collaboration between AI developers and domain experts as they can be used to facilitate the feedback exchange between the AI engineer and the expert. Our method can be more useful in domains where the example-based explanations are in demand. The potential risks and biases associated with the proposed method are comparable to those of the LIME method and depend on the data scientist's handling of the data.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://archive.ics.uci.edu/dataset/294/combined+cycle+power+plant.

Author contributions

AK: Conceptualization, Formal analysis, Resources, Methodology, Writing – original draft. BK: Formal analysis, Data curation, Software, Validation, Visualization, Writing – review & editing. SK: Data curation, Software, Validation, Visualization, Writing – review & editing. LU: Writing – review & editing, Conceptualization, Formal analysis, Methodology, Writing – original draft. VM: Conceptualization, Formal analysis, Writing – review & editing, Investigation, Project administration, Resources.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. The research is partially funded by the Ministry of Science and Higher Education of the Russian Federation as part of World-class Research Center Program: Advanced Digital Technologies (Contract No. 075-15-2022-311 dated April, 20 2022).

Acknowledgments

The authors would like to express their appreciation to the referees whose very valuable comments have improved the study.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Generative AI was used in the creation of this manuscript.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Adadi, A., and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* 6, 52138–52160. doi: 10.1109/ACCESS.2018.2870052

Adhikari, A., Tax, D., Satta, R., and Faeth, M. (2019). "LEAFAGE: example-based and feature importance-based explanations for black-box ML models," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (IEEE), 1–7. doi: 10.1109/FUZZ-IEEE.2019.8858846

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., et al. (2021). "Neural additive models: interpretable machine learning with neural nets," in *Advances in Neural Information Processing Systems*, 4699–4711.

Apley, D., and Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc.* 82, 1059–1086. doi: 10.1111/rssb.12377

Arrieta, A., Diaz-Rodriguez, N., Ser, J. D., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion* 58, 82–115. doi: 10.1016/j.inffus.2019.12.012

Arya, V., Bellamy, R., Chen, P.-Y., Dhurandhar, A., Hind, M., Hoffman, S., et al. (2019). One explanation does not fit all: a toolkit and taxonomy of AI explainability techniques. *ArXiv:1909.03012*.

Awasthi, P., Kalantari, B., and Zhang, Y. (2018). "Robust vertex enumeration for convex hulls in high dimensions," in *International Conference on Artificial Intelligence and Statistics (PMLR)*, 1387–1396.

Balestriero, R., Pesenti, J., and LeCun, Y. (2021). Learning in high dimension always amounts to extrapolation. *ArXiv:2110.09485v09482*.

Belle, V., and Papantonis, I. (2021). Principles and practice of explainable machine learning. *Front. Big Data* 39:688969. doi: 10.3389/fdata.2021.688969

Benard, C., Biau, G., Veiga, S. D., and Scornet, E. (2022). "SHAFF: fast and consistent SHApley eFfect estimates via random forests," *in International Conference on Artificial Intelligence and Statistics (PMLR)*, 5563–5582.

Bennett, K., and Bredensteiner, E. (2000). "Duality and geometry in SVM classifiers," in *ICML'00: Proceedings of the Seventeenth International Conference on Machine Learning*, 57–64.

Bento, J., Saleiro, P., Cruz, A., Figueiredo, M., and Bizarro, P. (2021). "TimeSHAP: explaining recurrent models through sequence perturbations," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*, 2565–2573. doi: 10.1145/3447548.3467166

Bodria, F., Giannotti, F., Guidotti, R., Naretto, F., Pedreschi, D., and Rinzivillo, S. (2023). Benchmarking and survey of explanation methods for black box models. *Data Min. Knowl. Disc.* 37, 1719–1778. doi: 10.1007/s10618-023-00933-9

Bouneder, L., Leo, Y., and Lachapelle, A. (2020). X-SHAP: towards multiplicative explainability of machine learning. *ArXiv:2006.04574*.

Burkart, N., and Huber, M. (2021). A survey on the explainability of supervised machine learning. *J. Artif. Intell. Res*. 70, 245–317. doi: 10.1613/jair.1.12228

Cai, C., Jongejan, J., and Holbrook, J. (2019). "The effects of example-based explanations in a machine learning interface," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, 258–262. doi: 10.1145/3301275.3302289

Carvalho, D., Pereira, E., and Cardoso, J. (2019). Machine learning interpretability: a survey on methods and metrics. *Electronics* 8, 1–34. doi: 10.3390/electronics8080832

Castillo, E., Conejo, A., Castillo, C., Mínguez, R., and Ortigosa, D. (2006). Perturbation approach to sensitivity analysis in mathematical programming. *J. Optim. Theory Appl.* 128, 49–74. doi: 10.1007/s10957-005-7557-y

Chau, A., Li, X., and Yu, W. (2013). Large data sets classification using convex-concave hull and support vector machine. *Soft Comput*. 17, 793–804. doi: 10.1007/s00500-012-0954-x

Chen, J., Vaughan, J., Nair, V., and Sudjianto, A. (2020). Adaptive explainable neural networks (AxNNs). *ArXiv:2004.02353v02352*.

Chong, P., Cheung, N., Elovici, Y., and Binder, A. (2022). Toward scalable and unified example-based explanation and outlier detection. *IEEE Trans. Image Process*. 31, 525–540. doi: 10.1109/TIP.2021.3127847

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., et al. (2018). Explanations based on the missing: towards contrastive explanations with pertinent negatives. *ArXiv:1802.07623v07622*.

Dinh, N., and Jeyakumar, V. (2014). Farkas' lemma: three decades of generalizations for mathematical optimization. *Top* 22, 1–22. doi: 10.1007/s11750-014-0319-y

Du, M., Liu, N., and Hu, X. (2019). Techniques for interpretable machine learning. *Commun. ACM* 63, 68–77. doi: 10.1145/3359786

El Mrabti, S., E. L.-,Mekkaoui, J., Hachmoud, A., and Lazaar, M. (2024). An explainable machine learning model for sentiment analysis of online reviews. *Knowl. Based Syst*. 302:112348. doi: 10.1016/j.knosys.2024.112348

Ergen, T., and Pilanci, M. (2020). "Convex duality of deep neural networks," in *Proceedings of the 37 th International Conference on Machine Learning* (PMLR), 108.

Ergen, T., and Pilanci, M. (2021). Convex geometry and duality of over-parameterized neural networks. *J. Mach. Learn. Res*. 22, 1–63. Available at: https://typeset.io/pdf/convex-geometryand-duality-of-over-parameterizedneural-3w1w4fo6ik.pdf (accessed January 29, 2025).

Farnia, F., and Tse, D. (2018). "A convex duality framework for gans," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, 1–11.

Fong, R., and Vedaldi, A. (2017). "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision* (IEEE), 3429–3437. doi: 10.1109/ICCV.2017.371

Fong, R., and Vedaldi, A. (2019). "Explanations for attributing deep neural network predictions," in *Explainable AI* (Cham: Springer), 149–167. doi: 10.1007/978-3-030-28954-6_8

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Ann. Stat*. 29, 1189–1232. doi: 10.1214/aos/1013203451

Garreau, D., and von Luxburg, U. (2020a). "Explaining the explainer: a first theoretical analysis of LIME," in *International Conference on Artificial Intelligence and Statistics* (PMLR), 1287–1296.

Garreau, D., and von Luxburg, U. (2020b). Looking deeper into tabular LIME. *ArXiv:2008.11092*.

Ghalebikesabi, S., Ter-Minassian, L., Diaz-Ordaz, K., and Holmes, C. (2021). "On locality of local explanation models," in *Advances in Neural Information Processing Systems*, 18395–18407.

Greenwell, B., Boehmke, B., and McCarthy, A. (2018). A simple and effective model-based variable importance measure. *ArXiv:1805.04755*.

Gu, X., lai Chung, F., and Wang, S. (2020). Extreme vector machine for training on large data. *Int. J. Mach. Learn. Cyber*. 11, 33–53. doi: 10.1007/s13042-019-00936-3

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2019). A survey of methods for explaining black box models. *ACM Comput. Surv*. 51:93. doi: 10.1145/3236009

Hastie, T., and Tibshirani, R. (1990). *Generalized Additive Models*. New York: CRC Press.

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., and Chang, Y. (2022). GraphLIME: local interpretable model explanations for graph neural networks. *IEEE Trans. Knowl. Data Eng*. 35, 6968–6972. doi: 10.1109/TKDE.2022.3187455

Islam, M., Ahmed, M., Barua, S., and Begum, S. (2022). A systematic review of explainable artificial intelligence in terms of different application domains and tasks. *Appl. Sci*. 12, 1–38. doi: 10.3390/app12031353

J., Crabbe, Z. Q., Imrie, F., and van der Schaar, M. (2021). "Explaining latent representations with a corpus of examples," in *Advances in Neural Information Processing Systems*, 12154–12166.

Jethani, N., Sudarshan, M., Covert, I., Lee, S.-I., and Ranganath, R. (2022). "FastSHAP: real-time shapley value estimation," in *The Tenth International Conference on Learning Representations, ICLR 2022*, 1–23.

Khosravani, H., Ruano, A., and Ferreira, P. (2016). A convex hull-based data selection method for data driven models. *Appl. Soft Comput*. 47, 515–533. doi: 10.1016/j.asoc.2016.06.014

Konstantinov, A., and Utkin, L. (2021). Interpretable machine learning with an ensemble of gradient boosting machines. *Knowl. Based Syst*. 222, 1–16. doi: 10.1016/j.knosys.2021.106993

Kovalev, M., Utkin, L., and Kasimov, E. (2020). SurvLIME: a method for explaining machine learning survival models. *Knowl. Based Syst*. 203:106164. doi: 10.1016/j.knosys.2020.106164

Li, X., Xiong, H., Li, X., Wu, X., Zhang, X., Liu, J., et al. (2022). Interpretable deep learning: Interpretations, interpretability, trustworthiness, and beyond. *Knowl. Inf. Syst*. 64, 3197–3234. doi: 10.1007/s10115-022-01756-8

Liang, Y., Li, S., Yan, C., Li, M., and Jiang, C. (2021). Explaining the black-box model: a survey of local interpretation methods for deep neural networks. *Neurocomputing* 419, 168–182. doi: 10.1016/j.neucom.2020.08.011

Lundberg, S., and Lee, S.-I. (2017). "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, 4765–4774.

Molnar, C. (2019). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Available at: https://christophm.github.io/interpretable-ml-book/ (accessed January 29, 2025).

Molnar, C., König, G., Herbinger, J., Freiesleben, T., Dandl, S., Scholbeck, C., et al. (2020). "General pitfalls of model-agnostic interpretation methods for machine learning models," in *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers* (Springer), 39–68. doi: 10.1007/978-3-031-04083-2_4

Murdoch, W., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yua, B. (2019). Interpretable machine learning: definitions, methods, and applications. *ArXiv:1901.04592*.

Nemirko, A., and Dula, J. (2021a). Machine learning algorithm based on convex hull analysis. *Procedia Comput. Sci*. 186, 381–386. doi: 10.1016/j.procs.2021.04.160

Nemirko, A., and Dula, J. (2021b). Nearest convex hull classification based on linear programming. *Patt. Recogn. Image Anal*. 31, 205–211. doi: 10.1134/S1054661821020139

Nori, H., Jenkins, S., Koch, P., and Caruana, R. (2019). InterpretML: a unified framework for machine learning interpretability. *ArXiv:1909.09223*.

Ottmann, T., Schuierer, S., and Soundaralakshmi, S. (2001). Enumerating extreme points in higher dimensions. *Nordic J. Comput*. 8, 179–192. doi: 10.1007/3-540-59042-0_105

Petsiuk, V., Das, A., and Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. *ArXiv:1806.07421*.

Rabold, J., Deininger, H., Siebers, M., and Schmid, U. (2020). "Enriching visual with verbal explanations for relational concepts: combining LIME with Aleph," in *Machine Learning and Knowledge Discovery in Databases: International Workshops of ECML PKDD 2019* (Springer), 180–192. doi: 10.1007/978-3-030-43823-4_16

Ras, G., Xie, N., Van Gerven, M., and Doran, D. (2022). Explainable deep learning: a field guide for the uninitiated. *J. Artif. Intell. Res*. 73, 329–396. doi: 10.1613/jair.1.13200

Renwang, S., Baiqian, Y., Hui, S., Lei, Y., and Zengshou, D. (2022). Support vector machine fault diagnosis based on sparse scaling convex hull. *Measur. Sci. Technol*. 34:035101. doi: 10.1088/1361-6501/aca217

Ribeiro, M., Singh, S., and Guestrin, C. (2016). "Why should I trust You?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. doi: 10.1145/2939672.2939778

Ribeiro, M., Singh, S., and Guestrin, C. (2018). "Anchors: high-precision model-agnostic explanations," in *AAAI Conference on Artificial Intelligence*, 1527–1535. doi: 10.1609/aaai.v32i1.11491

Rockafellar, R. (1970). *Convex Analysis*. Princeton, NJ: Princeton University Press. doi: 10.1515/9781400873173

Rossignol, H., Minotakis, M., Cobelli, M., and Sanvito, S. (2024). Machine-learning-assisted construction of ternary convex hull diagrams. *J. Chem. Inf. Model*. 64, 1828–1840. doi: 10.1021/acs.jcim.3c01391

Rubinstein, R., and Kroese, D. (2008). *Simulation and the Monte Carlo Method, 2nd Edition* New Jersey: Wiley. doi: 10.1002/9780470230381

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell*. 1, 206–215. doi: 10.1038/s42256-019-0048-x

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. (2021). Interpretable machine learning: fundamental principles and 10 grand challenges. *ArXiv:2103.11251*. doi: 10.1214/21-SS133

Shankaranarayana, S. M., and Runje, D. (2019). "Alime: autoencoder based approach for local interpretability," in *Intelligent Data Engineering and Automated Learning-IDEAL 2019: 20th International Conference, Manchester,*

UK, November 14-16, 2019, Proceedings, Part I 20 (Springer), 454–463. doi: 10.1007/978-3-030-33607-3_49

Shapley, L. (1953). "A value for n-person games," in *Contributions to the Theory of Games* (Princeton: Princeton University Press), 307–317. doi: 10.1515/9781400881970-018

Singh, V., and Kumar, N. (2021). Chelm: Convex hull based extreme learning machine for salient object detection. *Multimed. Tools Appl.* 80, 13535–13558. doi: 10.1007/s11042-020-10374-x

Smith, N., and Tromble, R. (2004). *Sampling uniformly from the unit simplex*. Technical Report 29, Johns Hopkins University.

Strumbelj, E., and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.* 11, 1–18. Available at: https://dl.acm.org/doi/pdf/10.5555/1756006.1756007 (accessed January 29, 2025).

Strumbelj, E., and Kononenko, I. (2011). "A general method for visualizing and explaining black-box regression models," in *Adaptive and Natural Computing Algorithms. ICANNGA 2011* (Berlin: Springer), 21–30. doi: 10.1007/978-3-642-20267-4_3

Strumbelj, E., and Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* 41, 647–665. doi: 10.1007/s10115-013-0679-x

Sundararajan, M., Taly, A., and Yan, Q. (2017). "Axiomatic attribution for deep networks," in *34th International Conference on Machine Learning, ICML*, 5109–5118.

Teso, S., Bontempelli, A., Giunchiglia, F., and Passerini, A. (2021). "Interactive label cleaning with example-based explanations," in *Advances in Neural Information Processing Systems*, 12966–12977.

Utkin, L., and Konstantinov, A. (2022). Ensembles of random SHAPs. *Algorithms* 15, 1–27. doi: 10.3390/a15110431

Vaidya, P. (1989). "Speeding-up linear programming using fast matrix multiplication," in *30th Annual Symposium on Foundations of Computer Science* (IEEE Computer Society), 332–337. doi: 10.1109/SFCS.1989.63499

Vu, M., Nguyen, T., Phan, N. R., and Gera, M. T. (2019). Evaluating explainers via perturbation. *ArXiv:1906.02032v02031*.

Wang, D., Qiao, H., Zhang, B., and Wang, M. (2013). Online support vector machine based on convex hull vertices selection. *IEEE Trans. Neural Netw. Learn. Syst.* 24, 593–609. doi: 10.1109/TNNLS.2013.2238556

Wang, R., Wang, X., and Inouye, D. (2021). Shapley explanation networks. *ArXiv:2104.02297*.

Yang, Z., Zhang, A., and Sudjianto, A. (2021). GAMI-Net: An explainable neural networkbased on generalized additive models with structured interactions. *Pattern Recogn.* 120:108192. doi: 10.1016/j.patcog.2021.108192

Yao, D., Zhao, P., Pham, T.-A., and Cong, G. (2018). "High-dimensional similarity learning via dual-sparse random projection," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 3005–3011. doi: 10.24963/ijcai.2018/417

Yousefzadeh, R. (2020). "Deep learning generalization and the convex hull of training sets," in *NeurIPS 2020 Workshop: Deep Learning through Information Geometry*, 1–10.

Zablocki, E., Ben-Younes, H., Perez, P., and Cord, M. (2021). Explainability of vision-based autonomous driving systems: review and challenges. *arXiv:2101.05307*.

Zhang, H., Cisse, M., Dauphin, Y., and Lopez-Paz, D. (2018). Mixup: beyond empirical risk minimization," in *Proceedings of ICLR*, 1–13.

Zhang, T. (2002). On the dual formulation of regularized linear systems with convex risks. *Mach. Learn.* 46, 91–129. doi: 10.1023/A:1012498226479

Zhang, X., Wang, C., and Fan, X. (2021). Convex hull-based distance metric learning for image classification. *Comput. Appl. Mathem.* 40, 1–22. doi: 10.1007/s40314-021-01482-x

Zhang, Y., Tiňo, P., Leonardis, A., and Tang, K. (2021). A survey on neural network interpretability. *IEEE Trans. Emer. Topics Comput. Intell.* 5, 726–742. doi: 10.1109/TETCI.2021.3100641