Information Sciences 622 (2023) 211–234

Contents lists available at ScienceDirect

Information Sciences

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

EFFECT: Explainable framework for meta-learning in automatic classification algorithm selection[☆]

Xinyue Shao[a], Hongzhi Wang[a,b,*], Xiao Zhu[a], Feng Xiong[a], Tianyu Mu[a], Yan Zhang[a]

[a] Harbin Institute of Technology, No. 92, West Dazhi Street, Harbin, China
[b] Peng Cheng Laboratory (PCL), No. 2, Xingke 1st Street, Nanshan, Shenzhen, China

A R T I C L E I N F O

Article history:
Received 18 July 2022
Received in revised form 19 November 2022
Accepted 27 November 2022
Available online 5 December 2022

Keywords:
Explainability
Meta-learning
Causal reasoning
Counterfactual interpretation

A B S T R A C T

With the growing convergence of artificial intelligence and daily life scenarios, the application scenarios for intelligent decision methods are becoming increasingly complex. The development of various machine learning algorithms has benefited all disciplines of study, but choosing which algorithm is most suitable for a certain problem among a large number of algorithms is a challenge that every field must overcome. Another challenge at the practical application level is that machine learning algorithms currently trained with large amounts of data are primarily black-box and uninterpretable. This indicates that these methods pose potential risks and are difficult to rely on, thus hindering their application in sensitive fields such as finance and healthcare. The first challenge can be overcome by using meta-learning to combine data and prior knowledge to efficiently and automatically select the machine learning models. The second challenge remains to be addressed due to the lack of interpretability of traditional meta-learning techniques and deficiencies in transparency and fairness. Achieving the interpretability of meta-learning in autonomous algorithm selection for classification is crucial to balance the need for high accuracy and transparency of machine learning models in practical application scenarios. This paper proposes **EFFECT**, an interpretable meta-learning framework that can explain the recommendation results of meta-learning algorithm selection and provide a more complete and accurate explanation of the recommendation algorithm's performance on specific datasets combined with business scenarios. Extensive experiments have demonstrated the validity and correctness of this framework.

1. Introduction

The proliferation of data-collecting sources and the simplicity of data acquisition have resulted in an exponential increase in the quantity of data available for analysis and decision-making. Numerous data mining methods have been proposed to enable significant association mining and predictive analysis from massive data. Since different algorithms have varying induction biases and the performance varies widely for different datasets, selecting the most suitable method for a given problem has proven to be a significant challenge.

☆ Fully documented templates are available in the elsarticle package on http://www.ctan.org/tex-archive/macros/latex/contrib/elsarticle CTAN.
* Corresponding author at: Harbin Institute of Technology, No. 92, West Dazhi Street, Harbin, China.
E-mail addresses: shaoxinyue@hit.edu.cn (X. Shao), wangzh@hit.edu.cn (H. Wang), zhuxiao_momo@163.com (X. Zhu), xiong257246@outlook.com (F. Xiong), mutianyu@hit.edu.cn (T. Mu), zhangy@hit.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.ins.2022.11.144
0020-0255/© 2022 Elsevier Inc. All rights reserved.

The work of algorithm selection was previously usually performed by domain experts based on their expertise and a series of tests, which is labor-intensive and time-consuming. Brazdil [1]advocated applying meta-learning to the selection of machine learning algorithms by utilizing machine learning techniques as meta-algorithms to identify the mapping between problem meta-features and algorithm performance. The meta-algorithm employs a classification algorithm when determining the optimal strategy for the user. By training on metadata extracted from prior experience, meta-learning can recommend algorithms for new tasks. However, numerous studies have demonstrated that meta-learning classification algorithms may not always produce optimal results. Due to the black-box nature of the meta-algorithm, the user is not provided with any further necessary information when the recommendation results are not ideal [2]. To address such a situation, interpretable research on meta-learning algorithms is essential.

After choosing the suitable algorithm for the task, another big challenge is integrating the application scenario to fully carry out the interpretable work. Understanding the correct decision mechanism of black-box algorithms is certain to increase human confidence in machine learning models. Perhaps on paper, we can sacrifice interpretability for precision. High precision is a priority now, and people do not care why these models make such forecasts. For practical applications, however, such unexplainable predictions achieved by fitting massive data may have a series of pitfalls and cause serious social problems as follows.

1. **Hard to gain human trust.** Regarding the internal decision mechanism, the analysis of the decision algorithm is still at an uncertain level, the so-called black box problem. Such approaches that only allow users to blindly accept predictions without explanations and comprehension of the decision-making process cannot generally gain sufficient trust from users to be widely used in large numbers in sensitive areas.
2. **Bias and discrimination.** Data sampling limitations and bias may cause data-driven machine learning algorithms to create bias. For example, PredPol and COMPAS are two predictive policing tools [3]. The police use PredPol to predict where crimes will occur and accordingly enhance patrols. Courts employ COMPAS to evaluate the parole eligibility of suspects based on their criminal risk. However, both PredPol and COMPAS are racially discriminatory. PredPol can expand surveillance of non-white populations, and COMPAS discriminates against black criminal suspects. Such biases and discriminations can undermine social justice and generate social strife.
3. **Potential security risks.** Unexplainable algorithms increase hazards. Black-box models with superior performance on the training set frequently make mistakes that are impossible for humans to make. A well-trained neural network incorrectly recognizes a school bus as an ostrich after adding invisible noise to some images [4]. A unique backdoor approach is proposed by generating backdoor instances to deceive algorithm instances to gain target labels [5]. Experiments demonstrate that specially-designed glasses can mislead face recognition algorithms. Given the widespread use of image recognition in autonomous driving and facial recognition, this error poses a significant security threat.

As can be seen from the risks described above, existing machine learning models can still not generally be applied at scale in high-risk areas where they can have a significant impact. [6] discusses the explainability based on five points: why, who, what, when, and how, and clarifies that explainability is crucial in scenarios such as autonomous driving, medical diagnosis, and government decision-making. Black-box algorithms need a methodological technique that can reasonably explain their decision mechanisms and processes to help them move from written to practical applications and gain users' trust.

Based on the preceding discussion, two significant algorithm selection challenges require immediate attention. The first one is how to efficiently select the most suitable algorithm for a specific problem among many algorithms while providing a rationale and explanation for the selection process. The second one is how to undertake sufficient explainability work for the chosen algorithm within the context of the problem to boost user confidence in the decision algorithm. Meta-learning is a way for achieving efficient automatic algorithm selection [7,8]. Nevertheless, the existing meta-learning method remains a mystery that cannot deliver more accurate information than the predictions. Therefore, the above two challenges can be reformulated as: **a).** How to supply a decision basis and explanation for meta-learning algorithm automated selection; **b).** How to perform adequate explainable work for the selected algorithm within the problem context.

To address these two challenges, this paper proposes the concept of *meta-learning explainability*, which consists of *meta-explainability* and *recommendation-explainability*, involving two aspects of interpretation. On the one hand, *meta-explainability* is the explainability of the meta-algorithm of algorithm selection, which summarizes the underlying meta-knowledge of algorithm selection. On the other hand, *recommendation-explainability* refers to the explainability of the learning algorithm recommended by meta-learning for a specific problem. More specifically, *meta-explainability* is the interpretation of meta-learning algorithms that hopefully helps users understand how meta-learning algorithms perform algorithm selection based on metadata; *recommendation-explainability* is the interpretability study of the algorithms selected by meta-learning and helps users understand how the selected algorithm makes predictions in the context of a specific problem. The former discusses why meta-learning selected a particular learning algorithm for a specific task, and the latter describes the logic of the decisions made by the selected learning algorithm in the task. They encompass the entirety of the explanation pipeline, including algorithm selection and the interpretation of the chosen algorithms in their subsequent downstream activities. Both of these aspects pose challenges.

For *meta-explainability*, implementing interpretable meta-learning in algorithm selection is difficult since accuracy and explainability cannot coexist. The penalty of improving the accuracy of optimal algorithm recommendations is the increased complexity of the meta-learning model, both in terms of increased computational cost and, more importantly, higher uninterpretability. When optimizing performance, interpretability declines when algorithm performance and model complexity coexist, and until recently, this seemed unavoidable according to [9,10]. Fig. 1 depicts the relationship between model interpretability and performance. Therefore, the primary concern of this paper is how to achieve the *meta-explainability* of automatic algorithm selection while ensuring recommendation accuracy.

For *recommendation-explainability*, the most critical problem is that the criteria for determining what interpretation users need for various scenarios remain unclear [11]. For this reason, Judea Pearl[12] divides the explanations into three cognitive levels from low to high, the first level of the theory being **association**. The second level is **intervention**, and the third level is **counterfactual**. According to this theoretical structure, the current data-driven machine learning algorithms are situated in this first level based on **association**, which inevitably learns spurious relationships. To further filter out the explanations that truly reflect the decision reasoning process from the association relationships with spurious relationships as much as possible, it is necessary to expand the observed phenomena through **intervention** experiments, apply **counterfactual** reasoning that eliminates the falsity, and discover consistent cause-and-effect explanations. Most available interpretable tools [13–15] implicitly assume that data features are independently and identically distributed, ignoring causal relationships between features, leading to less-than-true explanations. Therefore, the second concern of this paper is how to introduce interventions and counterfactuals to further reveal the decision mechanism of the black-box algorithm in the scenario and enhance the model's credibility.

According to the three-level theory of interpretation, we believe that when applying the algorithm to specific problem scenarios, the explanation can take place in two ways. On the one hand, it is based on **intervention**, in which each feature's importance for prediction is quantified by perturbing the features and evaluating the degree of change in the prediction. On the other hand, generating **counterfactuals** for instance is an approach for providing the user with suggestions for revising the prediction. The intervention-based and counterfactual-based interpretations achieve explainability in two ways: the intervention-based interpretation evaluates the feature importance by perturbing them and observing the changes in predictions, whereas the counterfactual-based interpretation assists users in achieving the expected results by providing reasonable suggestions for feature modification.

Numerous studies have been proposed in both directions of explainability research [16–20], but all of them ignore causality and make the unreasonable assumption that data features are independently and identically distributed, i.e., when one feature is modified, it has no effect on other feature values. In fact, there is a causal relation between features [21], i.e., when the value of one variable is changed, the associated feature variable is also changed.

Motivated by this, we attempt to adopt causal relations to achieve the *recommendation-explainability*. In the intervention-based interpretation, we slightly perturbed each feature and updated its latent factors in conjunction with the causal model, and finally the degree of change in the predicted outcome was used as the feature influence score. In the counterfactual interpretation, we prioritized the features with high feature influence scores for modification and updated the latent factors according to the causal model until the desired goal was achieved.

To address the two major problems in algorithm selection proposed in this paper, we proposed a novel interpretation framework for meta-learning in automatic classification algorithm selection called **EFFECT**, an **E**xplainable **F**ramework **F**or m**E**ta-learning in automatic **C**lassifica**T**ion algorithm selection, which assists users in selecting the suitable machine learning model for a new task more efficiently and satisfies the *meta-explainability*. In addition, the framework can also help users better understand the prediction mechanism of the selected model in the task without going deeper into the model, understand the impact of each feature on the results, and satisfy the *recommendation-explainability*. The contributions of this paper are as follows:

1. We propose a comprehensive framework for the explicable meta-learning, **EFFECT**, based on causal relations. To our knowledge, this is the first study on explicable meta-learning in the automatic classification algorithm selection.
2. To achieve *meta-explainability*, we develop an interpretable learning algorithm recommendation approach based on the dataset's meta-features, which allows for further exploration of the relationship between the dataset and the algorithm while maintaining accuracy.
3. To improve *recommendation-explainability*, we combine causality from two levels, **intervention-based** and **counterfactual-based**, to provide in-context explanations for the recommended models. For the intervention-based explanations, we present a novel feature importance metric that combines causality as an interpretable indicator based on observation data expanded by intervention experiments. For counterfactual-based explanations, we provide a counterfactual generation method based on greedy strategies, which prioritizes features with high scores on the interpretable indicator for modification.
4. We evaluate the accuracy and validity of each module of the **EFFECT** framework in real datasets, and the superiority of our method is fully demonstrated in comparison with existing methods in various fields.

2. Related work

The scope and coverage of machine learning are expanding because of its rapid development. Suitable algorithm selection techniques are needed to apply machine learning to new tasks. [1] suggested using meta-learning to learn how to choose machine learning models. Decision trees are a meta-algorithm used by the STATLOG project to present appropriate machine learning models. Gama [22] suggested using three different regression techniques to quantify the error and suggested uti-

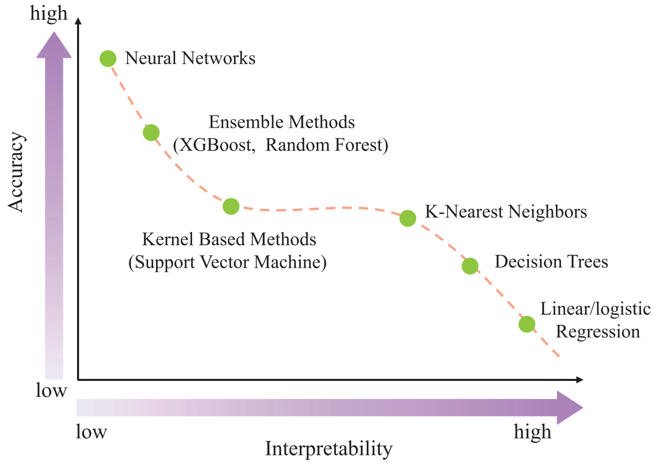


Fig. 1. The trade-off between model accuracy and interpretability.

lizing a regression algorithm as a meta-algorithm to anticipate how well learning algorithms will work. Bradzil [23] was the first to construct and evaluate the ranking of machine learning algorithms using the k-NN algorithm. To achieve automatic algorithm selection, the random forest was used as a meta-algorithm in [7,8].

The advantage of employing classification algorithms as meta-algorithms is that they have a vast choice of algorithms, but the output of classification algorithms is a single optimal algorithm. When the selected algorithm is not optimal, the disadvantage of the classification algorithm as a meta-algorithm is that it does not give the user alternative algorithm information. For these reasons, this paper argues that it is necessary to conduct interpretable research on the meta-learning process. In this section, we present further details of the latest research works in explainability.

Several surveys [24–27] have already helped us summarize a list of the numerous interpretable approaches that have been proposed in recent years. The explainability of artificial intelligence(XAI) consists of the ante-hoc explanation and the post-hoc explanation. In general, the ante-hoc explanation refers to the process of making a model intrinsically interpretable by training a simple interpretable model or incorporating explainability into a particular model structure. The post-hoc explanation refers to the ability to interpret a trained black-box model using interpretable approaches. Post-hoc signifies that explainability is achieved by viewing the world through the lens of the black-box model. Typical post-hoc methods include feature attribution, proxy models, etc. The explainable methods in this paper belong to the category.

Models with the ante-hoc explainability have the built-in transparency, and several types of machine learning methods can be regarded as having strong explainability. For instance, linear regression models express the output as a linearly weighted sum of each input feature. The importance of each feature to the output can be understood from the weights obtained during learning. In contrast, the logistic regression model converts the output to the classification probability using a logit function based on linear regression. The decision tree model fits the sample to a continuous cut of the high-dimensional space of input features, and the learned values of the cut can be interpreted as decision-making rules. Nevertheless, models with the ante-hoc explainability represent only a minor portion of machine learning. To accomplish post hoc explainability, we must create interpretation tools for the majority of the black-box algorithms.

Studies of the post hoc explainability of decision methods can be broadly classified into model-specific and model-agnostic approaches depending on whether they are restricted to a specific model class. We will concentrate on model-agnostic methods in this paper since they have the advantage of isolating the interpretation from the method [28], and exhibit more flexibility, thus allowing users to utilize them in conjunction with whatever method they are interested in as needed.

Visualization, feature attribution, proxy models, and counterfactual explanations are representative model-agnostic interpretation methods. In the remaining section, the relevant work of these methodologies will be discussed.

2.1. Visualization explanation

In image recognition tasks where neural networks are widely used, researchers frequently employ visualization of the network's internal properties to comprehend the information gained within the network. Each convolutional kernel in a convolutional network can function as an analysis detector. Using deconvolution to transfer hidden layer features back to the pixel space, and visualization to analyze the features of each layer and how they change with model training, [29] proposed visualization of the maximum activation regions based on differences in the selection of local image content by different convolutional kernels. [30] proposed CAM, a backpropagation-based input reconstruction visualization method. This method highlights the most relevant regions of the input image to the prediction results by linearly superimposing the weights associated with the prediction categories with the activation maps of the convolutional layers. To improve the convenience of

CAM, [31] proposed Grad-CAM++, which can be used for more complex models based on image tagline generation and image quizzing. In addition, [32] found that the visualization results generated by the CAM were not visually clean enough, so they proposed a visually interpretable confidence score-based method called Score-CAM. Based on Score-CAM and existing image enhancement techniques, [33] proposed Augmented Score-CAM.

2.2. Feature attribution explanation

Feature attribution is a class of interpretable methods that calculate the importance of individual units in the input with higher importance indicating more significant impact. Ranking or metering the input features and quantifying the effects of each element on the decision output can help us understand which inputs drive the black-box algorithm to the specific result. Scholars have proposed numerous methods for calculating the feature's importance. Some are based on statistical metrics; for example, [13] proposed the PDP, which uses a partial dependence function to calculate the average marginal effect of a given feature on the expected output. [34] explicitly addressed the problem of feature associations by presenting ALE, which remains valid even in the presence of feature associations. [35] proposed a game-theoretic SHAP method based on Shapley values to calculate the marginal benefit contribution of the features. Some feature importance calculation methods are based on the gradient obtained by backpropagation; for example, [36] attributed the deep network prediction performance to the input features. Two fundamental attribution axioms are identified; i.e., sensitivity and implementation invariance, and then a new attribution method called "integral gradient" was designed based on these two axioms. [37] proposed a method to calculate feature contributions for GBDT using the residuals of each node. [38] introduced a model-agnostic method based on high-dimensional model representation (HDMR) to analyze local feature contributions and explain supervised learning models.

2.3. Proxy model explanation

Training proxy models is an additional method for attaining model-agnostic explainability. Proxy models often require two conditions: (1) the proxy model must be able to fit the output of the original black-box model, and (2) the proxy model must be more interpretable than the original model to intuitively comprehend its decision logic. Additive models and decision tree models are frequent proxy models. The LIME model proposed by [15] builds a locally interpretable linear model to approximate the predictions of the local black-box approach, and the linear model's coefficients indicate the features' contribution. [16] pointed out that the linear model in LIME cannot determine the coverage, based on which anchors are proposed to describe the local behavior of the model in terms of rule sets, enabling the user to infer the model behavior based on the rules. [39] proposed a method to interpret the output of a black-box model at the semantic level, which trains an additive model based on the concept of semantic submodules for a complex neural network so that the output of the target network can be split into the sum of the contributions of different semantic submodules. [40] proposed a new method for transforming any type of decision forest into interpretable decision trees. The method generates a tree model that approximates the predictive performance of the XGBoost model while providing better interpretability.

2.4. Counterfactual explanation

Counterfactual interpretation is a generic interpretable paradigm that assists users in determining how to change the prediction output with minimal changes. A counterfactual is a conditional assertion that describes a hypothetical problem with a prior that represents the hypothetical condition and its successor. It explains how the world would be if the initial condition had occurred. The characteristic of this interpretation is that it does not explicitly answer the question "why" the prediction was made but instead tries to answer the question in another way, i.e., by helping the user understand what features the user needs to change to achieve a specific output. For example, a bank uses a loan risk prediction model to determine whether a person can get a loan. If someone's loan request is denied, a counterfactual explanation can provide feedback to help them change their features so that their conditions can be classified as creditworthy, thereby achieving the goal of getting a loan. There have been many recent attempts at generating counterfactuals; the most prominent class of approaches describes the counterfactual as an optimization problem whose goal is to minimize the loss function to discover the data points that meet the constraints and are closest to the original instances. This class of approaches mainly proposed different innovations and enhancements in the description of the loss function, the corresponding optimization algorithm, and the definition of the distance function [41,18,42]. In addition, [43] designed a model-based counterfactual interpretation framework and applied an umbrella sampling technique in training, which enhanced the synthesizer to capture the impact of rare events in the data thus strengthening the counterfactual interpretation. [44] has developed a new framework, GRACE, which discovers key features of samples, generates comparison samples based on these features, and provides explanations of the reasons for a given model prediction. [45] described a new approach to qualitative and quantitative counterfactual explanation generation. [33] proposed a new method for generating interpretable counterfactuals and contrastive explanations for models by exploring the inner workings of DCNNs.

This paper is the first interpretable research in the field of meta-learning in automatic classification algorithm selection and clarifies that the explainability of meta-learning includes *meta-explainability* and *recommendation-explainability*. In *meta-*

explainability, a series of experiences that can help analysts to select models are summarized; in *recommendation-explainability*, causality is fully combined to achieve feature influence computation and counterfactual generation.

3. Framework overview

This section introduces the proposed **EFFECT** framework. We divide the task of implementing the interpretable algorithmic automatic selection into the following three parts: explainable meta-learner, causality intervention observer, and counterfactual generator. The model recommendation part constructs a meta-learning model that enables automatic algorithm selection. Back-propagation based on this recommendation model reveals the degree of influence of each meta-feature on the output, thus achieving *meta-explainability*. Calculating the feature influence and generating counterfactuals improves the *recommendation-explainability*. By integrating causation with the quantification of feature relevance, the feature influence computation enables intervention-based interpretation. Counterfactual generation accomplishes counterfactual-based interpretation by suggesting feature modifications to alter the prediction outcomes. The overall workflow is shown in Fig. 2.

The first component is the **explainable meta-learner**. This component takes common machine learning datasets and the best-performing models that we tested on each dataset as the training set. This paper constructs a meta-feature extractor to retrieve the meta-features describing the dataset, which combined with the optimal algorithm constitute the metadata set. For this metadata set, a meta-learner is constructed, which is trained to recommend suitable algorithms for the dataset, and by calculating the integral gradients, a series of experiences that can help analysts to select models can be inducted. This component will be described in greater detail in Section 4.

The second component is the **causality intervention observer**. In response to the limitation that existing feature attribution methods are based on the independent homogeneous distribution of the data features, we propose a method that combines causality to determine the importance of each feature for this algorithm. The input has two parts. The first is a new task requiring a suitable algorithm, for which we extract meta-features and select the appropriate algorithm for the corresponding dataset. The second is the causal relationship among features in the dataset, which we formally express as a directed acyclic graph using a structural causal model. To integrate causality for a more precise interpretation, the latent factors for each feature in the causal structure model need to be discovered using the latent factors search. The causality-based influence computation then allows us to quantify the influence of the features on the prediction. This component will be described in greater detail in Section 5.

The third component is the **counterfactual generator**, which proposes a greedy counterfactual generation algorithm that satisfies the causal relationships among the features. Since this is an instance-based method, a specific instance is needed as a query, and immediately after that, the target class of counterfactual generation needs to be specified. For example, the query in the figure is the credit information of a man whose loan application is rejected. He wants his credit classification to be judged as good so that the loan application will be approved. The final output of the method is a suggestion that helps the query to cross the decision boundary and be predicted as a target class. This part will be described in detail in Section 6.

4. Explainable meta-learner

The explainable meta-learner aims to automatically select the most appropriate algorithm from many options applicable to a given new task and reveal the intrinsic logic of the meta-learner for algorithm selection, thus achieving *meta-explainability*. In this section, we divide the task of the explainable meta-learner into two parts, which are (1) implementing automatic algorithm selection based on meta-learning and (2) implementing explainability for the algorithm selection process.

Before describing how to implement an automatic algorithm selection meta-learner, we define the algorithm selection problem.

4.1. Problem definition

Definition 1. (Algorithm Selection) Given a new task P_{new}, the goal of algorithm selection is to find an algorithm that has the best performance on that task. The selection process can be formally represented as follows. Extract the measurable feature sets $F_{new} = \{f_1, f_2, \cdots, f_n\}$ in the problem space of P_{new}. The algorithm space $\mathscr{A} = \{A_1, A_2, \cdots, A_m\}$ represents the set of candidate algorithms, and the performance space $Y = \{y_1, y_2, \cdots, y_m\}$ is the corresponding performance with evaluation metric T of each method.

The algorithm selection process is to find $A_{new} \in \mathscr{A}$ with the best performance, which can be expressed as Eq. 1. The selection process is shown in Fig. 3.

$$A_{new} = \arg\max_{A_i \in \mathscr{A}} T(A_i(F_{new})) \tag{1}$$

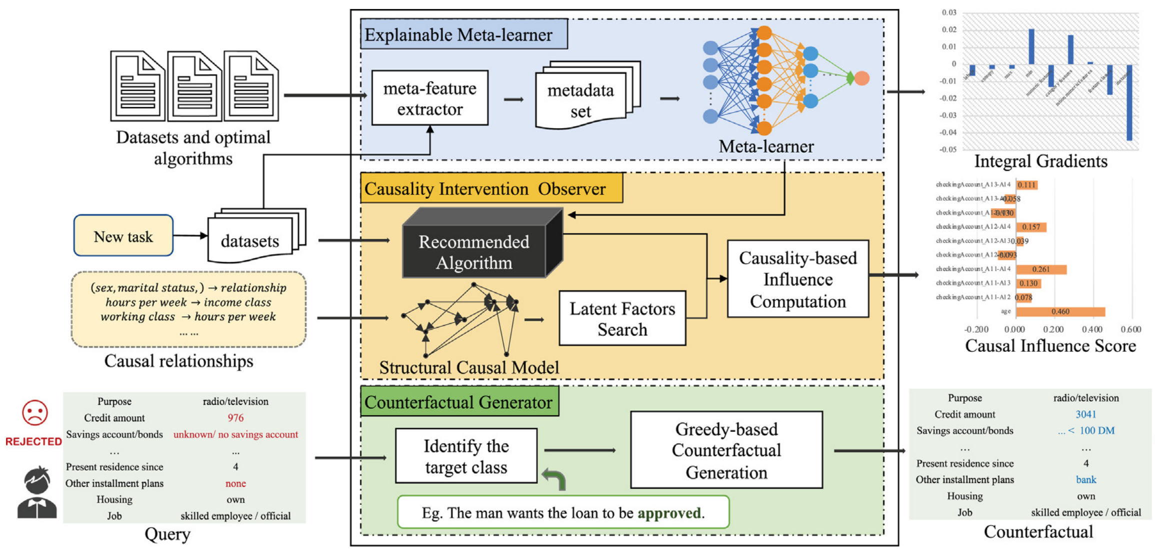

Datasets and optimal algorithms
Explainable Meta-learner
meta-feature extractor
metadata set
Meta-learner
Integral Gradients
New task
datasets
(sex, marital status,) → relationship
hours per week → income class
working class → hours per week
Causal relationships
Causality Intervention Observer
Recommended Algorithm
Structural Causal Model
Latent Factors Search
Causality-based Influence Computation
Causal Influence Score
Counterfactual Generator
Identify the target class
Greedy-based Counterfactual Generation
Eg. The man wants the loan to be approved.
REJECTED
APPROVED
Query
Counterfactual

Fig. 2. The workflow of **EFFECT**. The framework is comprised of the explainable meta-learner, causality intervention observer, and counterfactual generator. The explainable meta-learner trains a recommendation network as a meta-algorithm for automatic algorithm selection and achieves meta-explainability by computing the integral gradient. The causality intervention observer combines the causality to quantify the impact of each feature. The counterfactual generator generates counterfactuals across decision boundaries for each instance query.

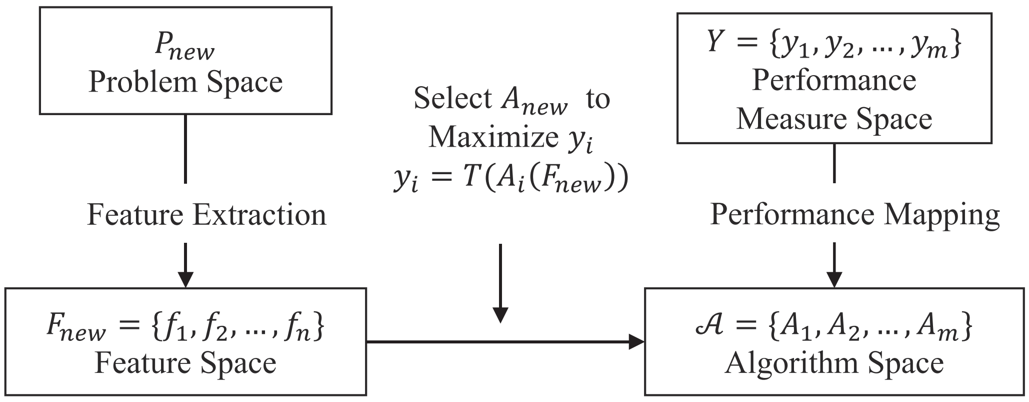

P_{new} Problem Space
Feature Extraction
$F_{new} = \{f_1, f_2, \dots, f_n\}$ Feature Space
Select A_{new} to Maximize y_i
$y_i = T(A_i(F_{new}))$
$Y = \{y_1, y_2, \dots, y_m\}$ Performance Measure Space
Performance Mapping
$\mathcal{A} = \{A_1, A_2, \dots, A_m\}$ Algorithm Space

Fig. 3. The algorithm selection process.

The most straightforward approach to algorithm selection is to test all available algorithms on this new task and then select the best-performing algorithm, which requires significant computational resources and time costs. Another approach is guided by expert empirical knowledge, but it is not always reliable and does not promptly take into account the latest proposed algorithms. To improve the efficiency and accuracy of the algorithm selection while reducing the computational cost and labor cost, we propose the **meta-learner**.

4.2. Meta-learner

The meta-learner is a meta-learning-based tool for automatic algorithm selection to reduce the computation of test-by-test with the knowledge experience of previous algorithm selection. There are various definitions of meta-learning, but in a nutshell, it is learning how to learn. The approach based on meta-learning overcomes algorithm selection challenges by optimizing the learning behavior of learning algorithms, essentially extracting valuable information from the potential features of a high number of instances to generate predictions about the behavior of new cases.

The algorithm selection challenge can be treated as a metadata-based classification problem. In preparation, each candidate algorithm is applied to each dataset, its performance is evaluated, and the algorithm with the best performance is chosen as the label for that dataset. The meta-features of each dataset are then extracted as input to a meta-classifier, and the best-performing algorithm labels are the output that conforms to the mapping between the dataset meta-features and the algorithm performance. Several essential techniques in the process need to be studied; i.e., extraction of the meta-features of tasks and construction of the meta-learning algorithms. Before discussing which and how the meta-features are extracted, a formal definition of meta-features is given in this paper.

Definition 2. (Meta-feature). Let $\mathscr{P}$ denote the set of all tasks, and $\mathscr{S}$ denote the set of statistical algorithms chosen. For each task $P_i \in \mathscr{P}$, the meta-feature is the feature extracted by the algorithm $S_j \in \mathscr{S}$, which is denoted as $mf_{ij} = S_j(P_i)$. $mf_i = \{m_{i1}, \cdots, m_{i|\mathscr{S}|}\}$is the meta-feature description for the task $P_i \in \mathscr{P}$.

To enable a fully comprehensive description of a task, this paper constructs a meta-feature extractor that is capable of extracting three types of meta-features for each task, including simple features, statistical features, and informative features. Simple features represent the basic structure of the task data, such as its size, number of attributes, and categories. Statistical meta-features, such as the geometric mean and variance, mainly indicate the data concentration trend and the features' dispersion. Information meta-features reflect the degree of correlation between the different attributes and the consistency of the data, such as the information entropy of the attribute variables. For each task, the meta-features vector and its optimal algorithm are then collected as metadata.

Metadata can serve as the foundation for meta-learner training. Next, we discuss how to construct a meta-learner that can accurately and efficiently select algorithms for new tasks. Given the superior performance of neural networks for classification tasks, we construct a neural network that mainly consists of one input layer, two hidden layers, and one output layer as a meta-learner. Before prediction, a simple mapping is applied to select the algorithm with the highest probability as the output label to make the classification result more than just the probability vector of each category. The meta-learner is illustrated in Fig. 4. In reality, each hidden layer contains a completely connected operation, an activation function, and a dropout; however, for the sake of simplicity, the activation function and dropout layer following the hidden layer is not displayed in the figure. If a total of n meta-features are extracted for each task $P_i \in \mathscr{P}$, the meta-feature description of P_i is $Mf_i = \{mf_1, \cdots, mf_n\}$. The weights W and bias B between the layers are initialized according to the parameters, and the meta-feature description is normalized and concatenated to a vector as the input.

If there are k neurons in the first hidden layer and d neurons in the second hidden layer, the final output of the first hidden layer is $H_1 = \{h_{11}, \cdots, h_{1n}\}$, and the output of the second layer is $H_2 = \{h_{21}, \cdots, h_{2d}\}$. H_1 can be calculated according to the input Mf_i, connection weight parameters $W_i \in \mathbb{R}^{n\times k}$, and bias $B_1 \in \mathbb{R}^{1\times k}$ between the input and the first hidden layers.

$$H_1 = \psi(W_1 Mf_i + B_1) \quad i = 1, 2, \cdots, t \tag{2}$$

where $\psi(\cdot)$ denotes the activation function, and t denotes that there are t tasks in the training set.

To further enhance the nonlinear representation of the network, a nonlinear function is incorporated as the activation function. The *ReLU* activation function is adopted in the recommendation network. Its computation cost is substantially lower than that of the *sigmoid* and *tanh* functions, which need derivation and division, and it solves the problem of gradient disappearance in the *sigmoid* function backpropagation. It significantly reduces the occurrence of overfitting by making a portion of the neuron's output zero. For a given element x, the *ReLU* activation function is defined as:

$$\phi(x) = max(x, 0) \tag{3}$$

The connections between the two hidden layers, as well as the connection between the second layer and the output layer, which contains a list of candidate algorithms, are similar to those previously described. The output layer's result is not an obvious description of the likelihood that X belongs to each class, so a *softmax* function is required to convert the output values to a probability distribution.

$$\hat{y}_i = softmax(A_i) = \frac{e^{A_i}}{\sum_{j=1}^{m} e^{A_i}} \quad i = 1, 2, \cdots, m \tag{4}$$

where $\hat{y}_i$ denotes the probability that Mf_i is predicted as class i. Creating a label probability vector $Y_i = \{y_1, \cdots, y_m\}$ for the true value of the sample Mf_i, with $y_i = 1, i$ is the valid class of Mf_i, and the remaining values equal 0. Finally, the cross-entropy loss function is used to compute the difference between the prediction probability and label probability.

$$\mathscr{L} = \frac{1}{t}\sum_{i=1}^{t} L_i = -\frac{1}{t}\sum_{i=1}^{t}\sum_{c=1}^{m} y_{ic}\log(\hat{y_{ic}}) \tag{5}$$

where t denotes the number of tasks in the training set, m denotes the number of candidate algorithms, and y_{ic} denotes the prediction probability that the algorithm A_c has the best performance in the task with the meta-feature vector Mf_i.

After iterative training of the forward propagation, computation of the loss values, backward propagation, and gradient update, the meta-learner can recommend the appropriate algorithm automatically based on the meta description of a task.

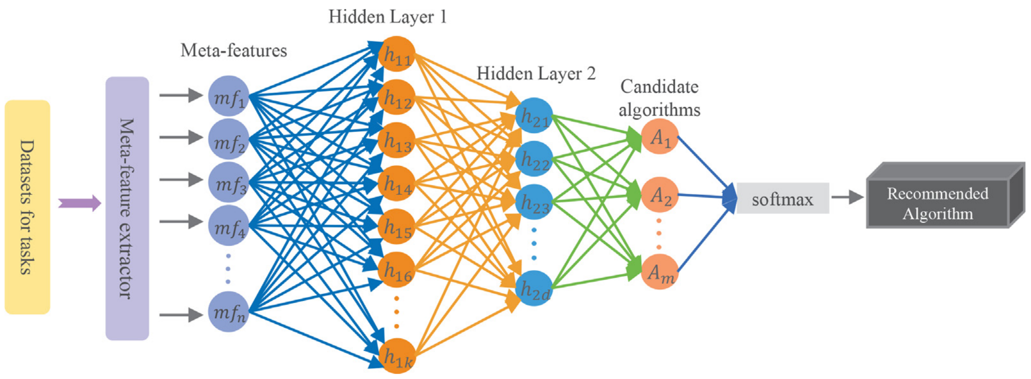


Fig. 4. Meta-learner.

4.3. Meta-explainability

Even though the meta-learner has enabled automatic algorithm selection, it is still a "black box", and users cannot intuitively comprehend the correlations between meta-features and recommendation algorithms. To achieve *meta-explainability*, we try to understand the impact of each meta-feature on the selection algorithm by computing its gradient in the network.

There is a vast family of explainable methods that utilize backpropagation by calculating the gradient of each input as feature importance. The direct gradient is based on the Taylor expansion, which uses the $|\,[\nabla_x F(x)]_i\,|$ as the i^{th} feature's importance indicator.

$$F(x+\Delta x) - F(x) \approx \langle \nabla_x F(x), \Delta x\rangle = \sum_i [\nabla_x F(x)]_i \Delta x_i \tag{6}$$

By computing the gradient for fully connected layers, convolutional layers, etc., it can explain specific prediction outcomes in many instances; in the network, it can more accurately assign the feature's importance value and reflect its importance. However, it has an obvious drawback that the gradient tends to 0 when the input feature values are in the gradient saturation stage and cannot reveal valid information. The meta-learner proposed in this paper contains the activation function *ReLU*, which has a gradient saturation phase in its negative half-axis. Therefore, the direct gradient's sensitivity is insufficient as a measure of the importance of the meta-learner features.

To overcome the shortcomings of the direct gradient, we adopt an interpretable technique called the integral gradient [36]. This is a strategy that combines the direct gradient and back-propagation-based attribution and adheres to the sensitivity and realization invariance assumptions. By calculating the integral gradient of the meta-earner, we can understand the influence of individual features on the selected algorithm.

This study then briefly discusses the idea of the integral gradient and analyzes why it can circumvent the gradient disappearing problem. First, an approximate expansion of Eq. 6 is performed to obtain Eq. 7.

$$F(\overline{x}) - F(x) \approx \sum_i [\nabla_x F(x)]_i [\overline{x} - x]_i \tag{7}$$

Eq. 7 can be regarded as the overall cost of moving from $\overline{x}$ to x as $F(\overline{x}) - F(x)$, which is the sum of the costs of each feature. Since the cost of each feature is approximately $[\nabla_x F(x)]_i [\overline{x} - x]_i$, we can use $|\,[\nabla_x F(x)]_i [\overline{x} - x]_i\,|$ as an important indicator of the i^{th} feature. Based on this perspective, the defect of the direct gradient can be interpreted as the imprecision of Eq. 7 leading to the insensitivity of $|\,[\nabla_x F(x)]_i [\overline{x} - x]_i\,|$. The integral gradient solves this problem by finding a more precisely equivalent analogous expression.

In the integral gradient, a path function $\gamma(\alpha) = x + \alpha \times (\overline{x} - x), \alpha \in [0, 1]$ is defined, representing a parameter connecting x to $\overline{x}$, where$\gamma(0) = x, \gamma(1) = \overline{x}$. The original gradient formula can be modified as:

$$\begin{aligned} F(\overline{x}) - F(x) &= F(\gamma(1)) - F(\gamma(0)) \\ &= \int_0^1 \frac{dF(\gamma(\alpha))}{d\alpha}\, d\alpha \\ &= \int_0^1 \langle \nabla_\gamma F(\gamma(\alpha)), \gamma'(\alpha)\rangle\, d\alpha \\ &= \sum_i \int_0^1 [\nabla_\gamma F(\gamma(\alpha))]_i [\gamma'(\alpha)]_i\, d\alpha \end{aligned} \tag{8}$$

Intuitively, since the integral gradient takes into account the gradient of all points on the entire path, it is no longer limited by the fact that the gradient at a point is 0. Therefore, $|\,[\int_0^1 \nabla_\gamma F(\gamma(\alpha))|_{\gamma(\alpha)=x+\alpha\times(\overline{x}-x)}\, d\alpha]_i [\overline{x} - x]_i\,|$ can be used as the important measure of the i^{th} feature. From the preceding equation, it is evident that the integral gradient algorithm only considers the model's input and output, ignoring the model's internal. Since the function is differentiable everywhere, the integrated gradient satisfies the implementation invariance. The integral gradient takes into account the gradients of all points along the path by selecting an infinite number of integral points between the baseline and the input value for integration and summation, so it is no longer limited by the gradient of 0 at a certain point, thus solving the gradient saturation problem and satisfying the sensitivity.

The detailed workflow of this component is shown in Fig. 2. The meta-learner automatically recommends an appropriate algorithm based on the meta description of the task and is interpretable in conjunction with the integral gradient, which explicitly provides the user with a direct relationship between the meta description and the selected algorithm. Thus, the **explainable meta-learner** enables the *meta-explainability* of meta-learning in the algorithm selection process.

5. Causality intervention observer

Meta-learning explainability also includes the *recommendation-explainability*, which explores the explainability of recommended algorithms in the problem context. Based on the three-layer theory [12], this paper will implement *recommendation-explainability* from two perspectives: intervention-based and counterfactual-based. In this section, we will introduce the **causality intervention observer**, which enables the intervention-based explanation in conjunction with causality.

In the intervention-based approaches, the typical investigated question is "What will happen to the outcome if a change is made to the input?" Existing methods of this type primarily calculate the direct effect of feature changes on the output without considering the causal relationships among the features. A causal relationship between the features shows that changes in certain aspects induce changes in other features and that changes in the outcomes occur from the combined effect of both features. Therefore, an intervention-based approach without causality is insufficient. We propose the observer, which depicts causality as a structural causal model(SCM) and determines the causal descendants of each feature by performing a latent factor search on the SCM. It can fully incorporate causality into the explanation.

Before presenting specific intervention-based explanatory methods, this section will give a brief introduction to causality as well as structural causal models(SCM).

5.1. Structural causal model

We have repeatedly emphasized that explainability cannot be separated from causality because causality exists in many fields, including economics, law, medicine, and physics, although it is difficult to describe. In layman's words, causality exists when one event causes the occurrence of another, and the latter is the result of the former. It is important to note that causation differs from correlation. The correlation is symmetric, and the correlation between the random variables A and B can be defined as:

$$corr(A,B) = \frac{Cov(A,B)}{\sqrt{Var(A)Var(B)}} \tag{9}$$

Obviously $corr(A,B) = corr(B,A)$, so the symmetry of the correlation is proved. Causality is asymmetric, which means that event A causes event B but does not indicate that event B triggers event A.

Experts from numerous domains have created various representations of causality models, such as causality diagrams, structural equations, logical statements, etc., to vividly convey causality. Judea [12] pointed out that the structural causal model (SCM) is the clearest and most understandable of them all, and is defined as follows.

Definition 3. (Structural Causal Model). The structural causal model(SCM) is defined as an ordered triad $\mathscr{M} =< U, V, F >$, where U denotes a set of exogenous variables, and the model itself does not determine the values of the exogenous variables. $V = \{V_1, \cdots, V_n\}$ represents a set of endogenous variables, and the endogenous variables are determined by $U \cup V$. $F = \{f_1, \cdots, f_n\}$ means the group of functions, each f_i describes how the value of the corresponding endogenous factor V_i is determined by the values of the exogenous and antecedent endogenous factors. Formally, $V_i = f_i(v_{\pi_i}, u_i)$, where π_i is the number of V_i's father nodes, (v_{π_i} is the corresponding value, and u_i is the value of the variable in the exogenous that acts directly on V_i.

Since SCM is a generalization of the Bayesian networks[12], it is represented by a directed acyclic graph(DAG) $\mathscr{G}$. The edges of $\mathscr{G}$ are the relations described by functions in F, and the nodes are the variables in U and V. If $X = f(Y)$, a directed edge in $\mathscr{G}$ will point from X to Y, indicating that X is the parent node of Y. Fig. 5 is an example of an SCM of basketball performance, where endogenous variables $V = \{gender, height, performance\}$, exogenous variables $U = \{u_1, u_2, u_3\}$, and $F = \{f_1, f_2, f_3\}$.

5.2. Latent factors search

Representing the cause relationships as SCM is more concise and very understandable and facilitates determining which other features are potentially affected by each feature's intervention.

As indicated in the basketball performance example, when we try to intervene on gender and observe its effect on performance, it is evident that gender not only directly affects basketball performance, but it also affects the final basketball performance by indirectly affecting height. If one wants to accurately measure the calculated combined effect of gender on performance, it also needs to be assessed in conjunction with the latent effect of gender on height. In this paper, we refer to such features causally influenced by the intervened feature as latent factors, which are represented as descendants nodes in the SCM.

Thus, identifying latent factors can be formulated as a challenge involving the identification of descendants in a DAG $\mathcal{G}$. This section discusses the latent factors search algorithm that can discover the causal descendants of each feature with precision and efficiency.

The flow of the latent factors search algorithm is shown in Algorithm 1. Throughout the search process, the algorithm incorporates the characteristics of the SCM itself to develop a pruning strategy. Since not all features are causally related in the actual problem scenario, the latent factors of those features that do not appear in the SCM are denoted as $\varnothing$. If every node in V has been traversed in the latent factors, the search is terminated, and the latent factors are output. These output latent factors are the causal descendants of this feature node, and they should be fully accounted for in the causality-based influence computation.

Algorithm 1. Latent Factors Search

```
Input: SCM $\mathcal{G}$, Initial feature $X$, Endogenous variables $V$, Exogenous
       variables $U$
Output: Latent factors $Latent$
1  if $X \notin (U \cup V)$ then
2      return $\varnothing$;
3  end
4  Initialize the search queue $Queue \leftarrow Queue \cup X$;
5  while $Queue \neq \varnothing$ and $V \neq \varnothing$ do
6      $t \leftarrow Queue.top()$;
7      childlist = getChildNode($t, \mathcal{G}, V$);
8      for $d \in childlist$ do
9          $Latent \leftarrow Latent \cup d$;
10         $Queue \leftarrow Queue \cup d$;
11         $V \leftarrow V \setminus d$;
12     end
13 end
14 return $Latent$
```

5.3. Causality-based influence computation

Given the latent factors of a feature, the causality-based influence computation consists of two steps. The first step is to measure the impact of the intervention on its latent factors, whose values would change as a result of the intervention. The second step is to re-predict the output corresponding to the updated feature values after the intervention and quantify the degree of predicted change due to the intervention, thus achieving the intervention-based explanation for feature importance.

For the first step, there is no sequential relationship within the set of latent factors obtained from the search because the causal relationships between features are complex. Because not every latent factor is only or directly influenced by the target feature, it is not possible to directly update the latent factors in arbitrary order after intervening with the target feature;

therefore, it is necessary to construct a linear sequence for the vertices in the SCM to guide the updated order of the latent factors. As previously indicated, the SCM is a DAG, hence, it is possible to build a topological sequence to determine the updated order of the latent factors.

Definition 4. (Topological Sorting). Topological sorting for a DAG $\mathcal{G}$ is to arrange all vertices in $\mathcal{G}$ into a linear sequence such that any pair of vertices u and v in the graph, if edge $(u, v) \in E(\mathcal{G}), u$ will appear before v in the linear sequence. In general, such a linear sequence is called topological order.

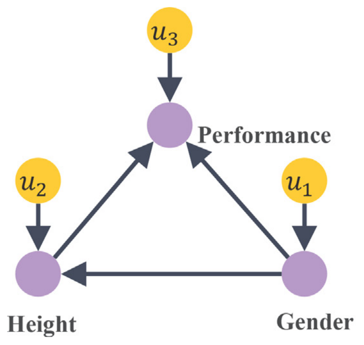
It is ensured that any parent nodes that might affect the factor are updated first by updating the latent factor values in topological order. After determining the update order, we need to identify how to quantify the influence of features. In this paper, we use the change in the prediction after applying intervention to the feature as a measure of the feature influence, motivated by the intuition that the greater the contribution of a feature to the prediction, the greater the impact it will have on the prediction when the feature is intervened. A sufficiently small intervention Δx_i is added to the given target feature x_i , whose feature influence value S_i is, which is defined as follows.

$$S_i = \frac{\hat{y} - y}{\Delta x_i} = \frac{M(X') - M(X)}{\Delta x_i} \tag{10}$$

where M is the recommended algorithm, X is the original input, y is the output of the recommended algorithm without intervention, X' is the input of the features after updating the latent factors affected by x_i based on causality, and $\hat{y}$ is the prediction after intervening x_i and updating the latent factors. The entire calculation process is shown in Algorithm 2. In this way the influence of each feature on the model prediction can be quantified in conjunction with causality, thus achieving the goal of *recommendation-explainability*.

Algorithm 2. Causality-based Influence Computation

```
   Input: Feature representation $X = \{x_0, \cdots, x_i, \cdots, x_n\}$, Initial feature
          $x_i$, Decision method $M$, SCM $\mathcal{G}$
   Output: Feature Influence $S_i$
 1 $y = M(X)$ // the original output;
 2 $Latent_i = LatentFactorsSearch(\mathcal{G}, x_i, V, U)$;
 3 while $node\ set \neq \varnothing$ do // Get Topological Sorting of $\mathcal{G}$
 4     Add the nodes with degree 0 to $Order$;
 5     Delete these nodes and all edges from them;
 6 end
 7 $x'_i = x_i + \Delta x_i$;
   // add perturbations to target feature;
 8 for $node : Order$ do // execute in topological order
 9     if $node \in Latent$ then
10         $node = update(node)$// for each node, construct a
           predictor based on the $\mathcal{G}$;
11     end
12 end
13 $S_i = \frac{\hat{y}-y}{\Delta x_i} = \frac{M(X')-M(X)}{\Delta x_i}$;
14 return $S_i$
```



$$
\begin{aligned}
&V = \{\text{Gender}, \text{Height}, \text{Performance}\}\\
&U = \{u_1, u_2, u_3\}\\
&F = \{f_1, f_2, f_3\}\\
&\text{Gender} = f_1(u_1)\\
&\text{Height} = f_2(\text{Gender}, u_2)\\
&\text{Performance} = f_3(\text{Gender}, \text{Height}, u_3)
\end{aligned}
$$

Fig. 5. Structural causal model(SCM) of basketball performance.

6. Counterfactual generator

Counterfactual reasoning is a fundamental way of thinking in human consciousness [46], where people often consider whether events will change by assuming that some aspects of the events have changed. Counterfactual explanations provide users with feedback on which they can act to achieve the desired outcomes in the future. This section introduces the **counterfactual generator**, which enables *recommendation-explainability* based on counterfactuals. Before introducing specific counterfactual-based interpretation methods, this section briefly describes what a counterfactual explanation is and gives a related definition.

6.1. Counterfactual definition

The counterfactual is a desired goal-oriented explanation that does not explicitly answer the question of "why" the prediction was made but instead tries to answer the question in another way by helping the user understand which features the user needs to change to achieve the desired prediction. Frequently, what users want is not an explanation of the importance of each feature but rather a piece of information on how to alter the expected prediction through effort. For different users, depending on their own different situations, the counterfactual generator proposed in this paper can give them different suggestions to help them achieve their goals, which can also be considered as a method that can provide personalized explanations for different users.

For instance, if one requests a loan at a bank, the bank will utilize the black-box decision method to assess the applicant's creditworthiness and decide whether to grant the loan. If the applicant's application is denied as a result of the decision method, the applicant will be left wondering why the application was rejected. The conventional explanation technique can assist the applicant in determining which factors are critical in the decision, whereas the counterfactual explanation can teach the applicant how to modify their behavior to obtain a loan authorization.

We then define the counterfactual explanation based on comparable similarity introduced by Lewis [47] as follows.

Definition 5. (Counterfactual Explanation) Given an instance $x \in X$ and a black-box classifier $f_\theta : \mathbb{R}^d$. If $f_\theta(x) = t$, then the counterfactual of x is defined as:

$$
x^{cf} = \arg\min_{x^* \in C} l(x^*, x) \qquad \text{s.t.} f_\theta(x) = t, f_\theta(x^*) = t' \tag{11}
$$

where x^{cf} is the counterfactual explanation for instance x, C is the counterfactual universe of the observed data space, and l is the distance metric between x^* and x. Although $t, t' \in \mathbb{R}^d$ are the predicted outputs of f_θ for x and x^*, respectively, $t' \neq t$ denotes counterfactual explanations that cause the model predictions to be reversed.

6.2. Greedy-based counterfactual generation

As indicated by the definition, the counterfactual is a sample of data from some distribution that can be used to invert model predictions as needed while remaining similar to the query input. It is difficult for causality to be directly involved in the counterfactual generation, which results in self-contradiction in the generated counterfactual explanations. Despite the existence of numerous counterfactual generation methods, it remains challenging to effectively generate counterfactual explanations with causality. For this reason, this section proposes a **greedy-based counterfactual generation** method that combines causal constraints with counterfactual explanations.

The fundamental issue with counterfactual generation at the moment is that the generated counterfactuals are unrealistic since the causal relationships and the restrictions of the value domain are ignored. According to the conventional method, counterfactual generation is an optimization problem with the goal of minimizing the loss function to identify the data point that best satisfies the constraint and is closest to the initial input. However, the current approach does not take into account that such generated counterfactual explanations are unfeasible due to the lack of consideration of causality between the features. This is because causality is difficult to express in the functional form to participate in the objective function.

The greedy-based counterfactual generation approach provided in this section ultimately includes causality through instructions based on causal influence scores and a latent factors search. Based on the discussion in Section 5.3, it can be argued that features with stronger causal influence scores significantly impact the prediction. Therefore, if the goal is to modify the prediction with minimal cost, it is reasonable to intervene by first selecting the features that have the greatest influence.

Algorithm 3 shows the whole greedy-based counterfactual generation process. For a given instance x, the algorithm applies a greedy strategy to discover the counterfactual explanation x^{cf} as close to x as possible; it can be roughly divided into the following four steps.

1. Compute the causal influence score, denoted S, for each feature in x according to Causality-based Influence Computation. The feature x_k with the highest score $S_k \in S$ is then selected for modification.
2. Considering the causal relationships among the features, the latent factors influenced by x_k need to be updated, which can be searched through the Latent Factors Search. The list of latent factors affected by x_k is denoted as $Latent_k$.
3. The latent factor values in $Latent_k$ are updated in topological order, ensuring that any parent node that may affect the factor is updated first. The update value relies on our prior construction of a predictor for each node based on the causality graph $\mathcal{G}$.
4. After all factors in $Latent_k$ have been updated, map x^{cf} back to the domain space to ensure that the generated counterfactual conforms to the value domain constraint. P is a projection that maps the x^{cf} to the value domain space, which ensures that the final x^{cf} looks more realistic and, that the domain space dom(X) contains maxima, minima, datatypes, etc.
5. Feed x^{cf} into the original decision model and observe whether its prediction changes. If so, then consider it as the counterfactual explanation that we are looking for; otherwise, repeat the above steps.

Algorithm 3. Counterfactual Interpretation Generation

```
Input: Decision method $M$, Query instance $X = \{x_0, \cdots, x_n\}$,
       Learning rate $\alpha$
Output: Counterfactual Interpretation $x^{cf}$
1  Initialize $x^{cf} \leftarrow x$;
2  $t \leftarrow M(x)$;
3  while $M(x^{cf}) = t$ do
4      compute all feature influence score of $X$, and denote as $S$;
5      choose the $S_k = max\{S_0, \cdots, S_n\}$;
6      $x_k = x_k - \alpha \times S_k$ ;
7      $Latent_k = LatentFactorsSearch(\mathcal{G}, x_k, V, U)$;
8      for $node : Order$ do // Order is the topological order of $\mathcal{G}$
9          if $node \in Latent_k$ then
10             $node = update(node)$// for each node, construct a
                predictor based on the $\mathcal{G}$;
11         end
12     end
13     $x^{cf} \leftarrow P(x^{cf}, dom(X))$;
14 end
15 return $x^{cf}$
```

7. Experiments

In this section, we conducted extensive experiments to verify the efficiency of each module in the **EFFECT** framework.

7.1. Evaluation on explainable meta-learner

In this section, we will evaluate the module's accuracy in recommending algorithms for various problems and the method's performance on specific problems.

7.1.1. Experimental settings

We created a metadata set[1] for the dataset by collecting data on target attribute categories, the number of category attributes, the number of numeral attributes, the information entropy of the attribute values within each category, and other factors. Each dataset has 23 meta-features as shown in Table 1, whose label corresponds to the classification method that performed the best on that dataset throughout the test. During the training process, we set the dropout rate to 0.5 and the learning rate to 0.001 for each layer.

7.1.2. Accuracy and validity

We chose KNN and decision tree, two interpretable machine learning models, to test the algorithm selection on metadata sets with SVM, random forest, XGBoost, and the neural network model meta-learner proposed in this paper in order to experimentally demonstrate the idea that increasing the accuracy of the optimal algorithm recommendations decreases the interpretability. The experimental results are depicted in Fig. 6. Even though the decision tree and KNN models have strong interpretability, it is evident that the algorithm selection accuracy for both of these models is often inferior to that of the other uninterpretable models. Although with better interpretability, the linear regression model is generally used for regression tasks rather than classification tasks, thus we did not take it as the baseline. As shown in Fig. 6, the experimental results are generally consistent with Fig. 1 given in the literature, and the complex model may achieve higher accuracy in our algorithm selection task.

To evaluate the accuracy of the meta-learner, we selected the following models as the baselines based on a variety of criteria, including performance and explainability. The decision tree and KNN are representatives of interpretable machine learning models; SVM represents the kernel-based method; XGBoost and random forest are examples of ensemble methods, with random forest also being the method used for algorithm selection in [7,8]; CNN, ResNet [48], and FT-Transformer [49] are representatives of state-of-the-art techniques, where FT-Transformer [49] is an improved version of Transformer [50] on tabular data. The experimental results are shown in Table 2. Although the decision tree and KNN are more interpretable due to their simpler structures, their performance in terms of accuracy, precision, and other metrics is slightly worse. CNN, ResNet [48], and Transformer [50] are some of the latest baseline methods, and their accuracy rates are above 0.7. However, their structural designs may be suitable for processing images, text, and other classification tasks for unstructured data, whereas the features of tabular data, are relatively shallow when compared to these image and text data. Therefore, these methods do not provide a significant advantage in our algorithm selection task. The accuracy of the meta-learner is higher than 0.8, and the precision, recall, and F-measure are all higher than 0.9, indicating that it performs the algorithm selection task substantially more effectively than the baselines.

To further validate whether the recommended algorithms are still effective in specific problems when the selection algorithms are inconsistent with the labels in the metadata set, a subset of the dataset with inconsistent predictions and labels is selected. We measure the *precision*, *recall*, and *F-measure* score of the prediction methods and labeled methods on each dataset of the subset.

To facilitate comparisons between the recommended and label methods, we introduce three metrics: $\mathscr{F}^1 = \frac{Precision_1}{Precision_2}, \mathscr{F}^2 = \frac{Recall_1}{Recall_2}, \mathscr{F}^3 = \frac{F-measure_1}{F-measure_2}$.

The larger the value of these indicators, the more similar the effect of the two approaches is. If the two methods are consistent, all of these indicators equal 1. As shown in Table 3, although the recommended technique is inconsistent with the label method, its performance still is relatively comparable to the label method. Except for the $monks - problems - 1$ dataset, all of the other datasets had $\mathscr{F}^1, \mathscr{F}^2, \mathscr{F}^3$ values greater than 0.9. The method recommended in this module, especially in $heart - statlog$, has the same excellent performance as the label method. To summarize, this module is capable of recommending efficient and effective decision methods for different problems.

7.1.3. Explainability

For explainability, we test the meta-learner in terms of integral gradients and make an attempt to establish a link between some tough datasets and successful decision methods based on them. As illustrated in Fig. 7, the AdaboostM1 is an improved method of Adaboost that is more adaptable to multi-class single-label tasks. The integral gradient of this

[1] Metadata: https://github.com/XinyueShao/EFFECT/tree/main/datasets (All datasets used in this paper can be found in this link.)

Table 1
Meta-features and their meanings.

Meta-feature number	The meaning of meta-feature
mf_0	Class number in target attribute.
mf_1	Class information entropy of target attribute.
mf_2	Maximum proportion of single class in target attribute.
mf_3	Minimum proportion of single class in target attribute.
mf_4	Number of numeric attributes.
mf_5	Number of category attributes.
mf_6	Proportion of numeric attributes.
mf_7	Total number of attributes.
mf_8	Records number in the dataset.
mf_9	Class number in category attribute with the least classes.
mf_{10}	Class information entropy in category attribute with the least classes.
mf_{11}	Maximum proportion of single class in category attribute with the least classes.
mf_{12}	Maximum proportion of single class in category attribute with the least classes.
mf_{13}	Class number in category attribute with the most classes.
mf_{14}	Class information entropy in category attribute with the most classes.
mf_{15}	Maximum proportion of single class in category attribute with the most classes.
mf_{16}	Minimum proportion of single class in category attribute with the most classes.
mf_{17}	Minimum average value in numeric attributes.
mf_{18}	Maximum average value in numeric attributes.
mf_{19}	Minimum variance in numeric attributes.
mf_{20}	Maximum variance in numeric attributes.
mf_{21}	Variance of average value in numeric attributes.
mf_{22}	Variance of variance in numeric attributes.

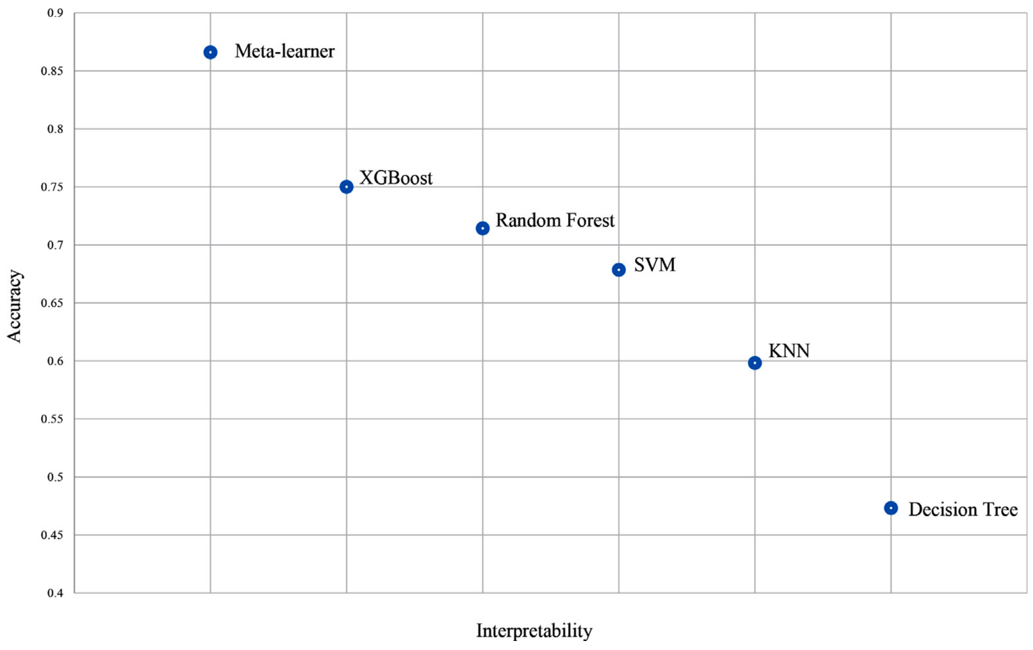


Fig. 6. The algorithm selection accuracy of different machine learning models.

method is significantly larger than that of other methods on the feature of the number of target attribute classes, making it superior to other methods in multi-class tasks. The random forest is more capable than other approaches for classification problems involving a high number of category features, and our studies also demonstrate that the integrated gradient of this method is greater than that of other methods when the number of category attributes is considered.

Additionally, our experiments demonstrate that the method performs better as the minimum proportion of individual categories in the target attribute or the minimum proportion of individual categories in the category attribute with the most categories in the category attribute increases, but its performance may degrade as the category attribute with the most categories in the category attribute's information entropy or the maximum average value in the numeral increases. Besides, MLP differs from the random forest in that it is better suited to datasets with a large number of numerical features, as evidenced by the fact that its integrated gradient is positive regardless of the maximum mean, minimum variance, or maximum

Table 2
Performance of recommended methods.

Methods	Accuracy	Precision	Recall	F-measure
Decision Tree	0.473	0.234	0.310	0.222
KNN	0.598	0.547	0.467	0.479
SVM	0.679	0.610	0.761	0.654
Random Forest	0.714	0.703	0.677	0.662
XGboost	0.75	0.648	0.642	0.625
Meta-learner	**0.866**	**0.902**	**0.933**	**0.903**
CNN	0.714	0.471	0.558	0.475
ResNet [48]	0.759	0.728	0.711	0.682
FT-Transformer [49]	0.777	0.751	0.745	0.718

Table 3
Performance comparison of recommended and labeled methods.

Dataset	Label Algorithm			Recommended Algorithm			Other Metrics		
	Precision	Recall	*F-measure*	Precision	Recall	*F-measure*	$\mathscr{F}^1$	$\mathscr{F}^2$	$\mathscr{F}^3$
anneal	0.993	0.993	0.993	0.982	0.981	0.981	0.989	0.988	0.988
D41	0.623	0.629	0.623	0.620	0.623	0.621	0.995	0.990	0.997
D74	0.854	0.855	0.855	0.851	0.848	0.849	0.996	0.992	0.993
heart-statlog	0.822	0.822	0.822	0.822	0.822	0.822	1.00	1.00	1.00
liver-disorders	0.624	0.629	0.625	0.620	0.623	0.621	0.994	0.990	0.994
monks-problems-1	0.459	0.461	0.456	0.397	0.398	0.397	0.865	0.863	0.871
sick	0.987	0.987	0.987	0.979	0.979	0.979	0.992	0.992	0.992

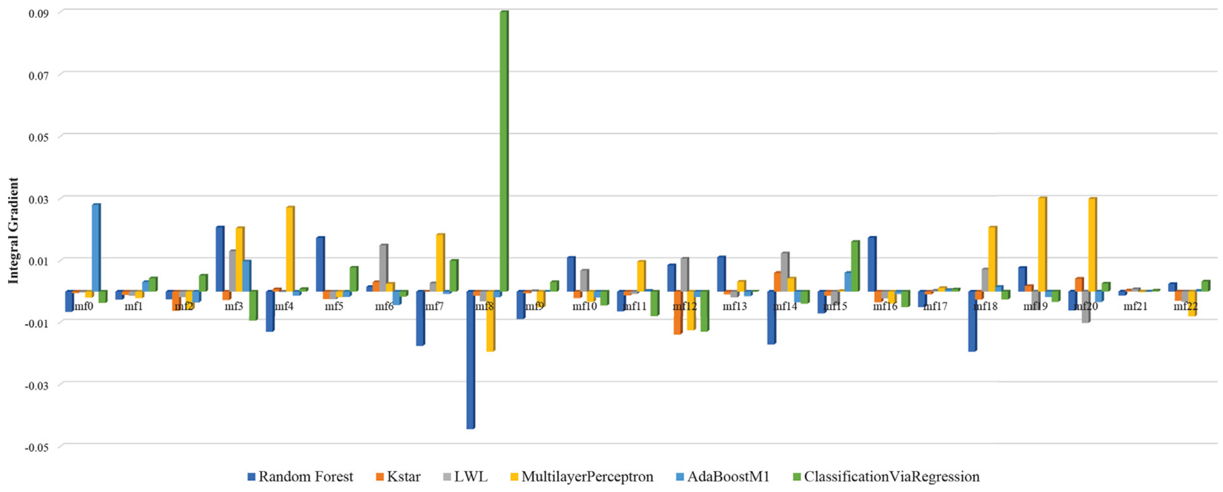


Fig. 7. The integral gradient of the explainable meta-learner.

variance in numerical attributes, and that its gradient value is significantly higher than that of other methods when the number of numerical attributes is taken into consideration. The ClassficationViaRegression method for classification by regression requires more instances to achieve a fit for more accurate classification.

7.2. Evaluation on causality-based influence computation

The purpose of this part is to evaluate whether the causal influence scores proposed in this module more accurately capture the causal relationships in the domain.

7.2.1. Experimental settings

In this part of the evaluation, we consider two real-world datasets (Adult[2] and German-Credit[3]). The Adult dataset contains information on 14 categories, including age, gender, occupation, education level, etc. The objective is to forecast if an individ-

[2] Adult: https://archive.ics.uci.edu/ml/datasets/adult.
[3] German Credit: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

ual's annual income reaches 50 k. The German-Credit covers 24 dimensions of personal, financial, and demographic information about the bank account holder. The objective is to categorize loan borrowers as having a good or bad credit risk. Additionally, we apply the causal graphs provided in the [46] for both the Adult and German-Credit datasets.

7.2.2. Baselines

LIME [15]:This method allows visualization of feature importance scores and feature heatmaps to provide interpretation. SHAP [35]:an interpretation method based on cooperative game theory that provides an interpretation for black box models by calculating the Shapley value.

7.2.3. Causal influence score

LIME's performance on the two datasets is shown in Fig. 8 and Fig. 9, respectively. Its drawback is that it requires the input of specific instances to perform the calculations, which means that the model is interpreted differently for different instances and does not provide a global interpretation for the user. The performance of the SHAP on the two datasets is shown in Fig. 10 and 11, respectively. It provides the shapely values of the *top-20* features that have the most significant impact on the model prediction from a global perspective, but it is computationally inefficient, especially on the adult data-set, so the method cannot be directly extended for use with large datasets or model interpretation. The performance of our proposed causal influence scores on the two datasets is shown in Fig. 12 and Fig. 13, respectively. In comparison to the LIME and SHAP, our method calculates the impact scores of category features independently for each change in categorical features allowing us to more correctly capture the effect of specific feature values on the model. Our findings highlight causa-

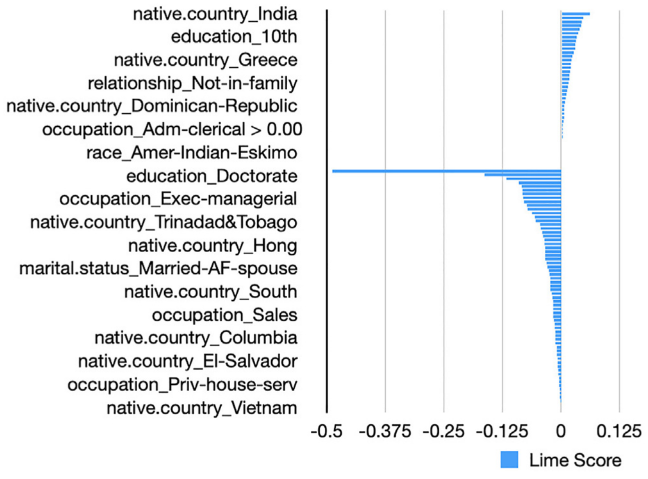


Fig. 8. LIME on Adult.

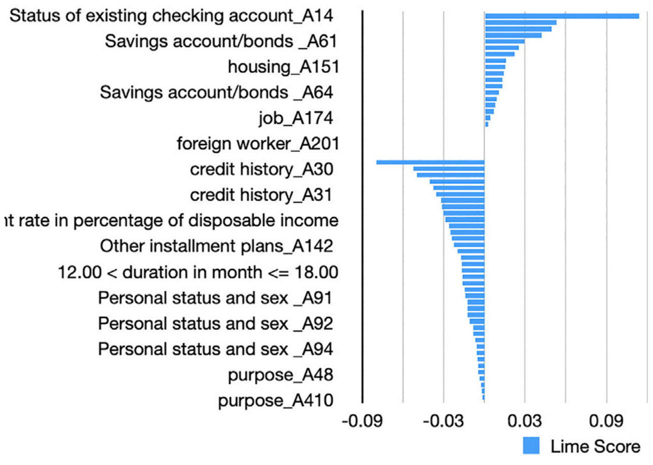


Fig. 9. LIME on German-Credit.

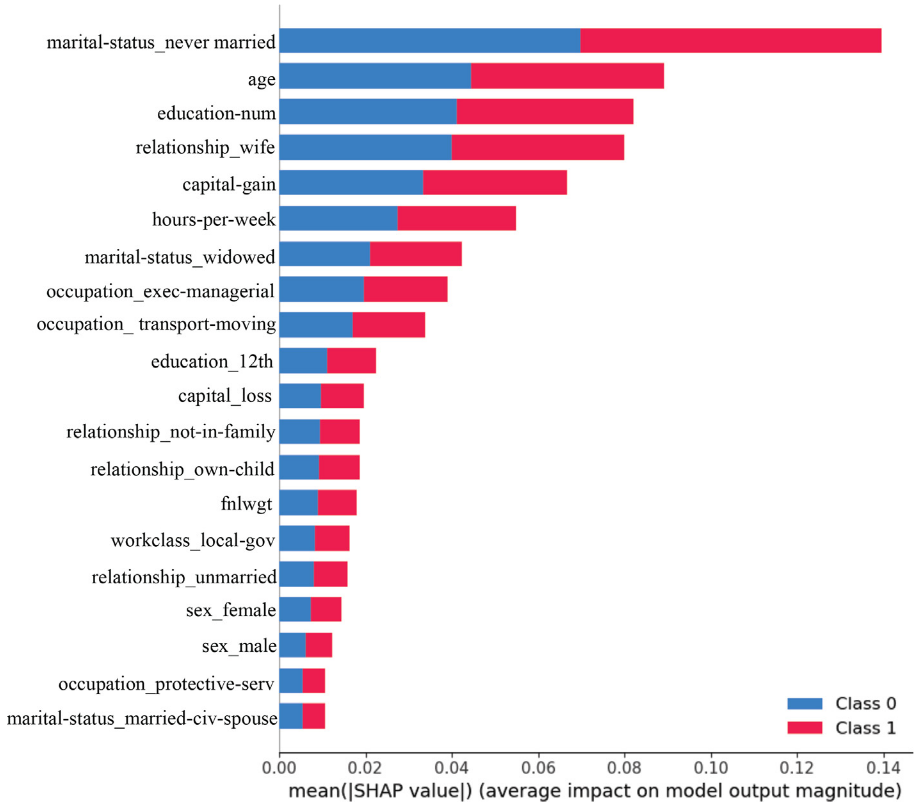


Fig. 10. SHAP on Adult.

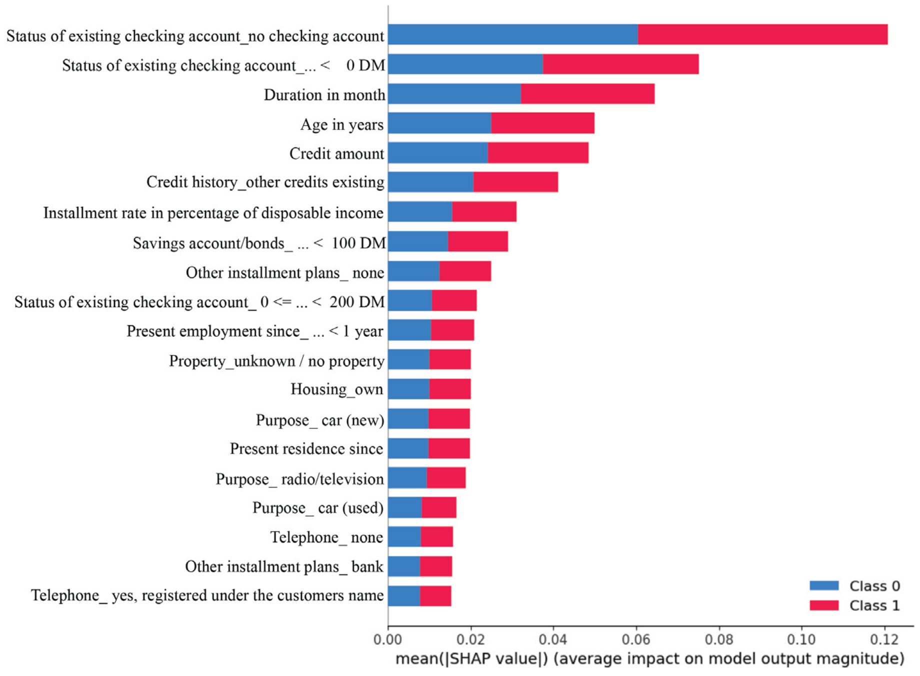


Fig. 11. SHAP on German-Credit.

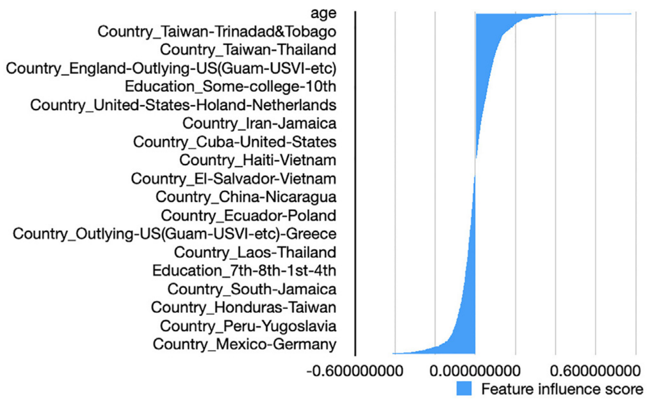


Fig. 12. Causal influence score on Adult.

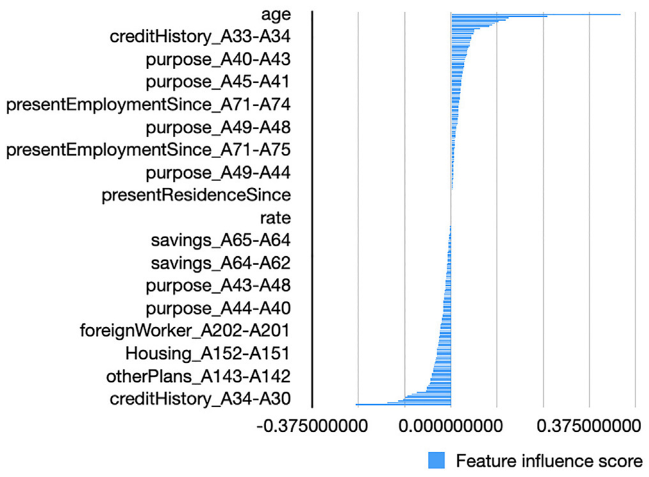


Fig. 13. Causal influence score on German-Credit.

tion, such as how age is typically associated with high income and how having a housing or a growing credit history is likely to result in favorable outcomes for persons for whom the credit algorithm returns negative results.

7.3. Evaluation on counterfactual generator

In this subsection, we generate instance-specific counterfactuals based on the feature influence scores proposed in this paper, and adequately compare and evaluate them with existing popular counterfactual baseline methods in various aspects such as capability, efficiency, and quality of counterfactual generation.

7.3.1. Experimental settings

We continue to use two real-world datasets, Adult and German-Credit, and the chosen metrics are centered on both reliability and efficiency.

7.3.2. Baselines

CEM [17]: For the contrast counterfactual, the proximal gradient descent approach is employed. This is a very representative class of techniques, and as a result, we have chosen it as the comparison model for the baseline comparison.

Proto [18]: To find interpretable counterfactual interpretations of classifier predictions, a method known as prototyping is used. The prototyping method uses class prototypes that have been obtained through class-specific K-d trees to speed up the search for counterfactual instances while also eliminating computational bottlenecks caused by the numerical gradient evaluation of black box models.

7.3.3. Generated counterfactuals

In the specific experiments described in this study, 5, 10, 20, and 50 examples are randomly chosen for the counterfactual generation. As illustrated in Fig. 14, our method generates counterfactuals that are slightly more distant from the original instances than the CEM and Proto methods, since our method considers the causal relationship between features and is capable of dynamically updating the latent factors that are causally affected by the feature while it is being modified. In contrast, the CEM and Proto methods disregard such causality, and in the pursuit of a greater distance from the original instance, these methods sacrifice the truthfulness of the generated counterfactuals, resulting in some implausible counterfactuals, such as the number of people whose census takers believe that the observation has been modified to a negative number, etc.

As far as efficacy is concerned, as shown in Fig. 15, our strategy outperforms Proto by a wide margin. This is especially true on the larger Adult dataset, where the difference is more visible. Despite the fact that the generation efficiency of CEM on the Adult dataset is comparable to that of our method, on the smaller German-Credit dataset, CEM still requires a time investment comparable to that of it on the Adult dataset, whereas our method is able to generate counterfactuals on the German-credit dataset at a significantly lower time expenditure.

To provide a more visual representation of the counterfactuals generated by the counterfactual generator, we provide counterfactual examples for the Adult and German-Credit datasets in Tables 4 and 5, respectively. In these tables, two data

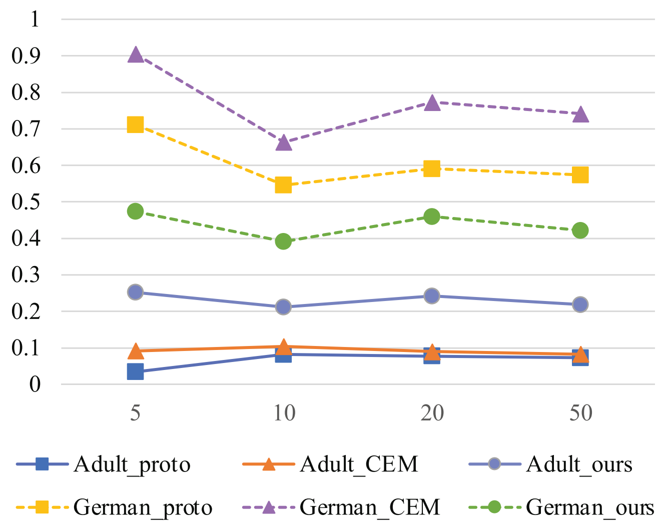


Fig. 14. Average distance.

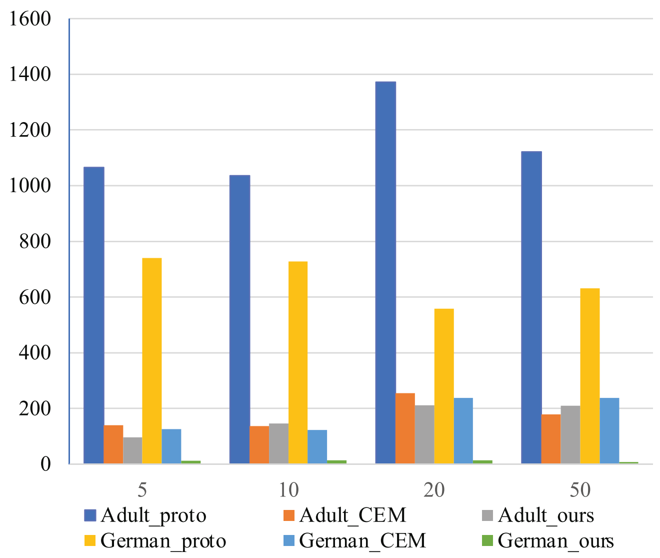


Fig. 15. Average time cost.

Table 4
Counterfactual samples on Adult.

age	workclass	education	marital.status	occupation	hours/w	gender	income
49	Private	Some-college	Married-civ-spouse	Sales	44	male	$\leq 50K$
49	Private	masters	Married-civ-spouse	Prof-specialty	42	male	$> 50K$
59	Self-emp-not-Inc.	5th-6th	Married-civ-spouse	Other-service	15	male	$\leq 50K$
59	Federal-gov	HS-grad	Married-civ-spouse	Adm-clerical	39	male	$> 50K$
31	Private	HS-grad	Married-civ-spouse	Craft-repair	20	male	$\leq 50K$
31	Private	Bachelors	Married-civ-spouse	Exec-managerial	50	male	$> 50K$
35	Self-emp-Inc.	Some-college	Married-civ-spouse	Sales	40	Female	$\leq 50K$
35	Self-emp-Inc.	Assoc-acdm	Married-civ-spouse	Prof-specialty	40	Female	$> 50K$
46	Private	7th-8th	Married-spouse-absent	Transport-moving	45	male	$\leq 50K$
46	Private	Bachelors	Married-spouse-absent	Exec-managerial	48	male	$> 50K$
43	Federal-gov	Some-college	Married-civ-spouse	Adm-clerical	40	Female	$\leq 50K$
43	Private	Masters	Married-civ-spouse	Prof-specialty	37	Female	$> 50K$
53	Private	Some-college	Married-civ-spouse	Craft-repair	40	male	$\leq 50K$
53	Federal-gov	Some-college	Married-civ-spouse	Adm-clerical	40	male	$> 50K$
34	Private	Bachelors	Married-civ-spouse	Exec-managerial	45	male	$> 50K$
34	Private	HS-grad	Divorced	Craft-repair	45	male	$\leq 50K$
40	Private	Bachelors	Married-civ-spouse	Exec-managerial	50	male	$> 50K$
40	Private	10th	Married-civ-spouse	Craft-repair	39	male	$\leq 50K$

Table 5
Counterfactual samples on German-Credit

checking account status	credit history	credit amount	savings	property	housing	type
$0 \leqslant \ldots < 200$ DM	existing credits paid back duly till now	12612	$100 \leqslant \ldots < 500$ DM	unknown/ no property	for free	bad
$0 \leqslant \ldots < 200$ DM	existing credits paid back duly till now	12612	$100 \leqslant \ldots < 500$ DM	real estate	own	good
$0 \leqslant \ldots < 200$ DM	existing credits paid back duly till now	888	$\ldots < 100$ DM	car or other	own	bad
$0 \leqslant \ldots < 200$ DM	no credits taken/ all credits paid back duly	888	$\ldots < 100$ DM	car or other	own	good
$\ldots < 0$ DM	delay in paying off in the past	4870	$\ldots < 100$ DM	unknown/ no property	for free	bad
$\ldots \geqslant 200$ DM	delay in paying off in the past	2883	$\ldots < 100$ DM	unknown/ no property	for free	good
$\ldots < 0$ DM	existing credits paid back duly till now	3345	$\ldots < 100$ DM	building society savings agreement	rent	bad
$\ldots < 0$ DM	no credits taken/ all credits paid back duly	3345	$\ldots < 100$ DM	building society savings agreement	rent	good
$\ldots < 0$ DM	existing credits paid back duly till now	1207	$\ldots < 100$DM	building society savings agreement	rent	bad
$\ldots < 0$ DM	existing credits paid back duly till now	1425	$100 \leqslant \ldots < 500$ DM	building society savings agreement	rent	good
$\ldots < 0$ DM	existing credits paid back duly till now	3345	$\ldots < 100$DM	building society savings agreement	rent	bad
$\ldots < 0$ DM	no credits taken/all credits paid back duly	3345	$\ldots < 100$DM	building society savings agreement	rent	good
$\ldots < 0$ DM	no credits taken/all credits paid back duly	2578	$\ldots < 100$DM	unknown	for free	good
$\ldots < 0$ DM	existing credits paid back duly till now	2578	$\ldots < 100$DM	unknown	for free	bad
no checking account	critical account/ other credits existing	976	unknown	car or other	own	good
$\ldots < 0$ DM	critical account/ other credits existing	3041	$\ldots < 100$ DM	car or other	own	bad

lines are shown as a set with the first line representing the original data in the dataset and the second line representing the counterfactual, while the portion marked in blue is the difference between the counterfactual and the original data. Some consistent features are not fully listed due to space limitations.

In Table 4, considering that users are more interested in explaining how they can shift from income predictions less than $50K$ to counterfactuals with income higher than $50K$, we chose more such samples than samples with predictions reduced from higher than 50K to less than 50K. The generated counterfactual explanations that raise incomes above 50K follow the logic of improving education to find a higher-paying job. For example, for the first set of data, the counterfactual suggests that the user should upgrade her education from "some-college" to "masters" so that she can find a job in the "Prof-specialty" field. This fully obeys the causal relationships and verifies the feasibility of the counterfactual generator.

In Table 5, again, considering that users need to improve their credit more than decrease it, we list more counterfactual suggestions that can change the credit prediction from bad to good. For example, for the first data set, the counterfactual suggestions suggest that it would be beneficial for the user to improve her credit and get approved for a bank loan if she could acquire real estate and own her own home. For the second and fourth sets, the user has existing credit that needs to be paid back on time, and it would be beneficial for her to pay back all the credit on time to get approved for a loan. For the third data set, according to the counterfactual interpretation, it would be beneficial for the user to change the status of the existing account from "$\ldots<0$ DM" to "$\ldots\geqslant 200$ DM", then their credit can be changed from bad to good. For the fifth data set, it would be beneficial for the user to increase the amount of savings from "$\ldots<$ 100 DM" to "100 $\leqslant\ldots<$ 500 DM", then the probability of changing the credit to good is significantly increased. In summary, these counterfactual explanations serve as a suggestion to provide users with a guideline on how they can change to get a good rating in the credit forecast and thus obtain a loan.

8. Conclusions

This study argues for the necessity of meta-learning explainability research and divides meta-learning explainability into two aspects, *meta-explainability* and *recommendation-explainability*. This paper proposes and implements the **EFFECT** meta-learning explainability framework, in which a model recommendation is used to achieve the *meta-explainability*. The *recommendation-explainability* is also achieved by feature influence computation and counterfactual generation from the perspective of intervention-based and counterfactual-based, respectively. Future extensions of the work may include explainability in more challenging meta-learning domains, such as meta-learning of time series, and more in-depth counterfactual research that incorporates causality.

CRediT authorship contribution statement

Xinyue Shao: Writing – original draft, Writing – review & editing, Conceptualization, Methodology, Formal analysis, Visualization, Software, Validation, Investigation. **Hongzhi Wang:** Writing – original draft, Writing – review & editing, Conceptualization, Methodology, Supervision, Project administration, Funding-acquisition. **Xiao Zhu:** Visualization, Validation, Conceptualization. **Feng Xiong:** Methodology, Visualization, Writing – original draft. **Tianyu Mu:** Writing – original draft, Resources. **Yan Zhang:** Validation, Supervision, Visualization.

Data availability

Data will be made available on request.

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 paper was supported by NSFC grant (62232005) and the Sichuan Science and Technology Program (2020YFSY0069).

References

[1] P. Brazdil, C.G. Carrier, C. Soares, R. Vilalta, Metalearning: Applications to data mining, Springer Science & Business Media, 2008.
[2] A. Kalousis, Algorithm selection via meta-learning (Ph.D. thesis), University of Geneva, 2002.
[3] W.D. Heaven, Predictive policing algorithms are racist. they need to be dismantled, MIT ZTechnology Review 17 (2020) 2020.
[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199.
[5] X. Chen, C. Liu, B. Li, K. Lu, D. Song, Targeted backdoor attacks on deep learning systems using data poisoning, arXiv preprint arXiv:1712.05526.
[6] A. Rosenfeld, A. Richardson, Explainability in human–agent systems, Auton. Agent. Multi-Agent Syst. 33 (6) (2019) 673–705.
[7] N. Cohen-Shapira, L. Rokach, B. Shapira, G. Katz, R. Vainshtein, Autogrd: Model recommendation through graphical dataset representation, in: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019, pp. 821–830.
[8] T. Mu, H. Wang, C. Wang, Z. Liang, X. Shao, Auto-cash: A meta-learning embedding approach for autonomous classification algorithm selection, Inf. Sci. 591 (2022) 344–364.
[9] D. Gunning, Explainable artificial intelligence (xai), Defense advanced research projects agency (DARPA), nd Web 2 (2) (2017) 1.
[10] A.B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al, Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Inf. Fusion 58 (2020) 82–115.

[11] S. Galhotra, R. Pradhan, B. Salimi, Explaining black-box algorithms using probabilistic contrastive counterfactuals, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 577–590.
[12] J. Pearl, D. Mackenzie, The book of why: the new science of cause and effect, Basic books, 2018.
[13] J.H. Friedman, The elements of statistical learning: Data mining, inference, and prediction, Springer open, 2017.
[14] A. Goldstein, A. Kapelner, J. Bleich, E. Pitkin, Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, J. Comput. Graphical Stat. 24 (1) (2015) 44–65.
[15] M.T. Ribeiro, S. Singh, C. Guestrin, 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, 2016, pp. 1135–1144.
[16] M.T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018.
[17] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, P. Das, Explanations based on the missing: Towards contrastive explanations with pertinent negatives, arXiv preprint arXiv:1802.07623.
[18] A.V. Looveren, J. Klaise, Interpretable counterfactual explanations guided by prototypes, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2021, pp. 650–665.
[19] K. Kaczmarek-Majer, G. Casalino, G. Castellano, M. Dominiak, O. Hryniewicz, O. Kamińska, G. Vessio, N. Díaz-Rodríguez, Plenary: Explaining black-box models in natural language through fuzzy linguistic summaries, Information Sciences.
[20] N. Jain, P.K. Jana, Xrrf: An explainable reasonably randomised forest algorithm for classification and regression problems, Inf. Sci. 613 (2022) 139–160.
[21] B. Schölkopf, F. Locatello, S. Bauer, N.R. Ke, N. Kalchbrenner, A. Goyal, Y. Bengio, Toward causal representation learning, Proc. IEEE 109 (5) (2021) 612–634.
[22] J. Gama, P. Brazdil, Characterization of classification algorithms, in: Portuguese Conference on Artificial Intelligence, Springer, 1995, pp. 189–200.
[23] P.B. Brazdil, C. Soares, J.P. Da Costa, Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results, Mach. Learn. 50 (3) (2003) 251–277.
[24] M. Du, N. Liu, X. Hu, Techniques for interpretable machine learning, Commun. ACM 63 (1) (2019) 68–77.
[25] F.-L. Fan, J. Xiong, M. Li, G. Wang, On interpretability of artificial neural networks: A survey, IEEE Trans. Radiation Plasma Med. Sci.
[26] R.R. Fernández, I.M. De Diego, J.M. Moguerza, F. Herrera, Explanation sets: A general framework for machine learning explainability, Inf. Sci.
[27] W. Ding, M. Abdel-Basset, H. Hawash, A.M. Ali, Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey, Inf. Sci.
[28] M.T. Ribeiro, S. Singh, C. Guestrin, Model-agnostic interpretability of machine learning, arXiv preprint arXiv:1606.05386.
[29] M.D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.
[30] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921–2929.
[31] A. Chattopadhay, A. Sarkar, P. Howlader, V.N. Balasubramanian, Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, 2018 IEEE winter conference on applications of computer vision (WACV), IEEE (2018) 839–847.
[32] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, X. Hu, Score-cam: Score-weighted visual explanations for convolutional neural networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 24–25.
[33] R. Ibrahim, M.O. Shafiq, Augmented score-cam: High resolution visual interpretations for deep neural networks, Knowl.-Based Syst. 252 (2022) 109287.
[34] D.W. Apley, J. Zhu, Visualizing the effects of predictor variables in black box supervised learning models, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 82 (4) (2020) 1059–1086.
[35] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proceedings of the 31st international conference on neural information processing systems, 2017, pp. 4768–4777.
[36] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 3319–3328.
[37] Á. Delgado-Panadero, B. Hernández-Lorca, M.T. García-Ordás, J.A. Benítez-Andrades, Implementing local-explainability in gradient boosting trees: Feature contribution, Inf. Sci. 589 (2022) 199–212.
[38] X. Zhang, L. Wu, Z. Li, Local interpretation of supervised learning models based on high dimensional model representation, Inf. Sci. 561 (2021) 1–19.
[39] R. Chen, H. Chen, J. Ren, G. Huang, Q. Zhang, Explaining neural networks semantically and quantitatively, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9187–9196.
[40] O. Sagi, L. Rokach, Approximating xgboost with an interpretable decision tree, Inf. Sci. 572 (2021) 522–542.
[41] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the gdpr, Harv. JL & Tech. 31 (2017) 841.
[42] R.M. Grath, L. Costabello, C.L. Van, P. Sweeney, F. Kamiab, Z. Shen, F. Lecue, Interpretable credit application predictions with counterfactual explanations, arXiv preprint arXiv:1811.05245.
[43] F. Yang, S.S. Alva, J. Chen, X. Hu, Model-based counterfactual synthesizer for interpretation, arXiv preprint arXiv:2106.08971.
[44] T. Le, S. Wang, D. Lee, Grace: Generating concise and informative contrastive sample to explain neural network model's prediction, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 238–248.
[45] I. Stepin, J.M. Alonso-Moral, A. Catala, M. Pereira-Fariña, An empirical study on how humans appreciate automated counterfactual explanations which embrace imprecise information, Information Sciences.
[46] S. Chiappa, Path-specific counterfactual fairness, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 7801–7808.
[47] D. Lewis, Causation, J. Philos. 70 (17) (1974) 556–567.
[48] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[49] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, Advances in Neural Information Processing Systems 34 (2021) 18932–18943.
[50] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30.