Research Article

Lin Zheng* and Yixuan Lin

A multiorder feature tracking and explanation strategy for explainable deep learning

https://doi.org/10.1515/jisys-2022-0212
received June 15, 2022; accepted March 08, 2023

Abstract: A good AI algorithm can make accurate predictions and provide reasonable explanations for the field in which it is applied. However, the application of deep models makes the black box problem, i.e., the lack of interpretability of a model, more prominent. In particular, when there are multiple features in an application domain and complex interactions between these features, it is difficult for a deep model to intuitively explain its prediction results. Moreover, in practical applications, multiorder feature interactions are ubiquitous. To break the interpretation limitations of deep models, we argue that a multiorder linearly separable deep model can be divided into different orders to explain its prediction results. Inspired by the interpretability advantage of tree models, we design a feature representation mechanism that can consistently represent the features of both trees and deep models. Based on the consistent representation, we propose a multiorder feature-tracking strategy to provide a prediction-oriented multiorder explanation for a linearly separable deep model. In experiments, we have empirically verified the effectiveness of our approach in two binary classification application scenarios: education and marketing. Experimental results show that our model can intuitively represent complex relationships between features through diversified multiorder explanations.

Keywords: model interpretability, multiorder feature interaction, deep model explanation, feature-tracking strategy, multiorder explanation

MSC 2020: 68T07

1 Introduction

Currently, in the era where more data are easily accessible, the exploration and prediction of many fields can be completed by computers, thereby reducing the burden on human beings. In this process, machine learning (ML) algorithms try to predict results or provide decisions by learning from large amounts of information [1,2]. However, unlike humans, most ML algorithms cannot explain the reason for predictions or decisions, which is often mentioned as a black box problem [3] in the ML field. The black box problem refers to a model's lack of interpretability [4], meaning that we cannot understand the model's internal mechanisms by only observing its parameters [5]. To address the lack of interpretability, [3,6] propose delving into a black box model in three ways: model explanation, outcome explanation, and model inspection. Different approaches can start from features, also known as feature attribution or a feature-based

* **Corresponding author: Lin Zheng,** Department of Computer Science, College of Engineering, Shantou University, Shantou 515063, China, e-mail: lzheng@stu.edu.cn
Yixuan Lin: Department of Computer Science, College of Engineering, Shantou University, Shantou 515063, China, e-mail: 20yxlin2@stu.edu.cn

explanation [7]. For example, the Shapley additive explanation (SHAP) approach [8] is a well-known additive feature attribution method that can intuitively quantify the importance of features. Moreover, applying the SHAP approach to tree-based models can further explain the contributions of feature interactions [9,10], which are attributed to the naturally interpretable characteristics of tree-based models, especially methods such as decision trees [3,11].

In particular, gradient-boosting decision trees (GBDTs) [12] are not only effective in automatic feature selection and model prediction [13] but also prove to have reliable interpretability in many application fields [14,15]. In a GBDT approach, important features can be found through a prediction optimization process based on gradient boosting [16,17]. Based on this, researchers found that there exists a smallest subset of features, which is sufficient to explain the prediction results [18,19]. In traditional ML models, GBDTs are utilized to simulate the behavior of the original approach to achieve model explanations or directly employ the path in the tree to intuitively present feature relations [18].

With the development of deep learning, however, it becomes more difficult for GBDTs to simulate model behavior. For instance, some deep models add a multilayer perceptron [20] or an attention mechanism [21] to the original factorization model [22] except for feature interactions, making the calculation of feature importance (FI) more complicated [23,24]. Therefore, some researchers try to make use of the tree structure to extract rules and interpret the deep neural network (DNN) [25] or to replace the last layer of the neural network with a decision tree to achieve interpretability [26]. Compared with changing the structure of DNNs, other scholars adopt a loosely coupled mode when using GBDTs. This method is named tree-enhanced embedding model (TEM), and it uses trees to select cross-features as the input of a neural network [27]. Initializing the DNN with a tree can ensure feature tracking from input to prediction and explanation [28].

However, trees and DNNs have different feature representations. Trees are generally accustomed to using feature values to represent features, while deep models are usually trained with feature embedding vectors [29,30]. Specifically, a GBDT selects those features that it considers to be beneficial for prediction based on the original feature values; but embedding vectors provide richer feature semantics from a higher dimension [31,32]. Nevertheless, the selected features are no longer represented by their original values after being sent to a deep model, resulting in: (1) *the tree model and the deep model are inconsistent in terms of feature representation*. On the other hand, (2) *the inconsistency of feature representations makes feature interactions more difficult to understand*. The existing tree-based approaches [9,10] that explain feature interactions by their values cannot be directly applied to explain the interactions between two vectors, meaning that the interpretability of trees cannot be taken advantage of. Furthermore, when more than two feature vectors interact in a deep model and (3) *the deep model has higher-order interactions, traditional tree-based explainable approaches are no longer applicable*.

The *main challenge* in addressing the above limitations is finding a way to track features from a tree to a deep model without harming the interpretability of the tree or the performance of the deep model. This requires us to make the tree-based explanation consistent with the prediction of the deep model without modifying their model structures. As a result, we need to design an effective feature-tracking strategy that ensures that the model prediction and the interpretation mechanisms are relatively independent. Inspired by the research of natural language processing, we believe that the memory mechanism [33,34] is an effective way to provide independent storage for features. Such an independent feature storage method can provide consistent representations for features, but the memory mechanism is powerless for high-order feature interactions.

In this work, we adopt the explanation in the literature [35] to term the *high-order feature interaction* as interaction modeling that combines more than or equal to two features. This is common in deep models such as neural factorization machines (NFM) [20] and neural tensor networks [31,32]. For example, in NFM, the bi-interaction layer acts as a high-order feature interaction of order 2, the second-order feature interaction. Furthermore, NFM is a *multiorder* model because it has a second-order component of bi-interaction, and includes a first-order linear part.

There are *two difficulties* in utilizing memory-based feature representations to track multiorder interactions: the multiple orders of some models are not linearly separable and the modeling of each order is mostly different for a certain model. Fortunately, existing studies have shown that Taylor expansions are applicable to most DNNs [36,37], that is, most DNNs can be approximately expanded into several linearly addable parts with different orders. Furthermore, Taylor expansions can be applied to explain different orders of network layers. Motivated by an extension of Taylor's theorem in neural networks, we speculate that the different order parts of a deep model can be leveraged to explain its prediction outcomes separately if that deep model is linearly separable or approximately linearly separable. Under this assumption, as long as the methods of feature combination are different, explanations of different orders can be achieved by consistent features, thus solving the second problem. Therefore, we propose a GBDT-based interpretable strategy for deep models named multiorder feature-tracking explanation (MFTE), which employs consistent memory representations to track features from GBDTs to a multiorder deep model and produce prediction-oriented explanations. The detailed contributions of our work are summarized as follows:

- **Novelty:** Different from existing methods that explain features directly, we design a novel memory representation to make feature consistent for the explanation.
- **Diversity:** We contribute a diverse and clear way of explaining prediction results according to different feature orders by a multiorder constraint strategy.
- **Usability:** We have experimentally verified the usability of MFTE and the intuitiveness of its explanations in two application domains: education and marketing.
- **Efficiency:** Experimental results show that the deep model equipped with MFTE can show multiorder explanations while maintaining performance advantages.

The remaining contents of the article are organized as follows. In Section 2, we discuss some research work related to the proposed approach. Then, the framework and details of our strategy are introduced in Section 3 and instantiated in Section 4, respectively. In Section 5, we did a wealth of experiments in the fields of education and marketing to verify the interpretability of MFTE; subsequently, the experimental results are discussed and summarized. Finally, we make conclusions of our work and further provide a future perspective in Section 6.

2 Related work

In this section, we first introduce feature attribution approaches for solving black box problems. Then, we study tree-based explanation methods in ML, and finally, we discuss existing explainable approaches based on both trees and deep models.

2.1 Feature attribution for black box problems

In ML, the black box problem does not have a standardized definition. However, scholars generally believe that the black box problem is caused by a model's lack of interpretability [3,4]. Molnar [5] argues that interpretability is the degree to which a human can understand the cause of a decision or a prediction. Lipton and Guidotti et al. [3,6] proposed achieving interpretability by using different explanation methods, such as model explanation, outcome explanation, and model inspection. The feature-based explanation [7] can be considered a model-oriented explanation or an outcome-/prediction-oriented explanation, which is formally terminized as a feature attribution because it directly captures the importance of the features [38,39]. In addition to eliminating redundant features in the data preprocessing stage, feature selection methods analyze the impact of features on the results and mine behavior information. For example, Kim et al. [40] proposed a model-based feature selection method to explain the importance of features in malware classification. The typical feature attribution method is the SHAP approach [8], which implements a feature-based explanation through the additive nature of FI. SHAP can calculate the local contribution of features. Moreover, its combination with trees can reduce computational complexity [41]. A tree-based SHAP can further calculate the importance of feature interactions [11], because the interpretable nature of the tree structure allows decision trees to automatically mine the relationships between features [9,10]. Therefore, maximizing the interpretable advantages of decision trees and allocating values for features is a valuable research problem in the field of interpretable ML.

2.2 Tree-based explanation

Among various decision trees, GBDTs [12] have been widely used because of their automatic feature select ability and excellent prediction effect [13,42,43]. The gradient boosting process of GBDTs ensures consistency in the feature selection and prediction result explanations [16,17]. For instance, Stojić et al. [14] applied extreme gradient boosting (XGBoost) [13] to predict the distribution and migration of chemical substances in the environment and generate SHAP values to explain important features; Fernández [15] adopted random forests to monitor bank stability in the United States and make multiway interpretations of important variables. Consequently, GBDTs have inherent advantages in selecting necessary features for prediction. Nevertheless, Shih et al. [18] proposed the concept of prime implicant (PI), i.e., explanations based on the tree structure. They believe that there are the smallest feature subsets related to the prediction results, and these subsets are sufficient for prediction and interpretation. Furthermore, [19] standardize the definition of necessary feature subsets. Izz et al. [44] propose a way to calculate PI explanations in decision tree learning. In addition to explaining with features, GBDTs can also explain the original model by simulating the behavior of that model [3]. In particular, the path in the tree is an important explanatory tool to visually show the logical relationship between features [18]. In this work, we will make full use of the interpretable advantages of GBDTs to explain deep models. The following subsection introduces the research progress on trees and deep models.

2.3 Explanation based on trees and deep models

The main difference between deep models and traditional ML models is the representation of features. Traditional models generally adopt feature values to represent features, while deep models often use feature embedding vectors to train the model [29,32]. However, the structure of a deep model [20] is usually more complicated than a traditional feature interaction model [22]. For example, the nonlinear feature relations in a multilayer perceptron or in an attention mechanism [21] are difficult to capture by GBDTs. Therefore, to avoid direct tracking and explanation of feature interactions, Zilke et al. [25] utilized the advantages of the tree structure to extract rules from DNNs. Other researchers argue that the tree structure has limitations for understanding a deep model, so they try to achieve interpretability by replacing the last layer of the neural network with a decision tree [26]. However, these methods have changed the original model structure to varying degrees and cannot allow the deep model to take advantage of the tree-selected features. Existing studies have demonstrated that decision trees can initialize neural networks and improve performance [28], which encourages researchers to use tree-selected features as input for a deep model. For example, TEM [27] is a loosely coupled model that uses GBDTs to choose important cross-features as the input of a neural network and explain feature interaction via attention weights [45]. TEM ensures the interpretability and completeness of the feature attribution process, meaning that it achieves feature tracking from input to prediction. However, it still cannot solve the problem of inconsistent feature representation, and the cross-features are fixed; thus, it cannot continue to learn feature interactions in the following deep model. In recent years, some progress has been made in capturing feature interactions of deep models. Researchers have shown that Taylor expansions are applicable to most deep models [36,37], which allow the multiple orders of feature interactions to be represented separately. In this work, we mainly study how to improve external feature representations such as memory [33,34] to capture the semantics of feature interactions and then cooperate with GBDTs to explain the prediction results of a deep model in a different way.

3 The MFTE strategy

This section mainly introduces the framework of the MFTE strategy in Section 3.1 and the detailed design of the model in Section 3.2.

3.1 The framework of MFTE

In Figure 1, we illustrate the entire training process of MFTE, including feature selection and consistent representation, collaborative training between a deep model (on the right), relevant explainable constraints (on the left), and the generation of explanations.

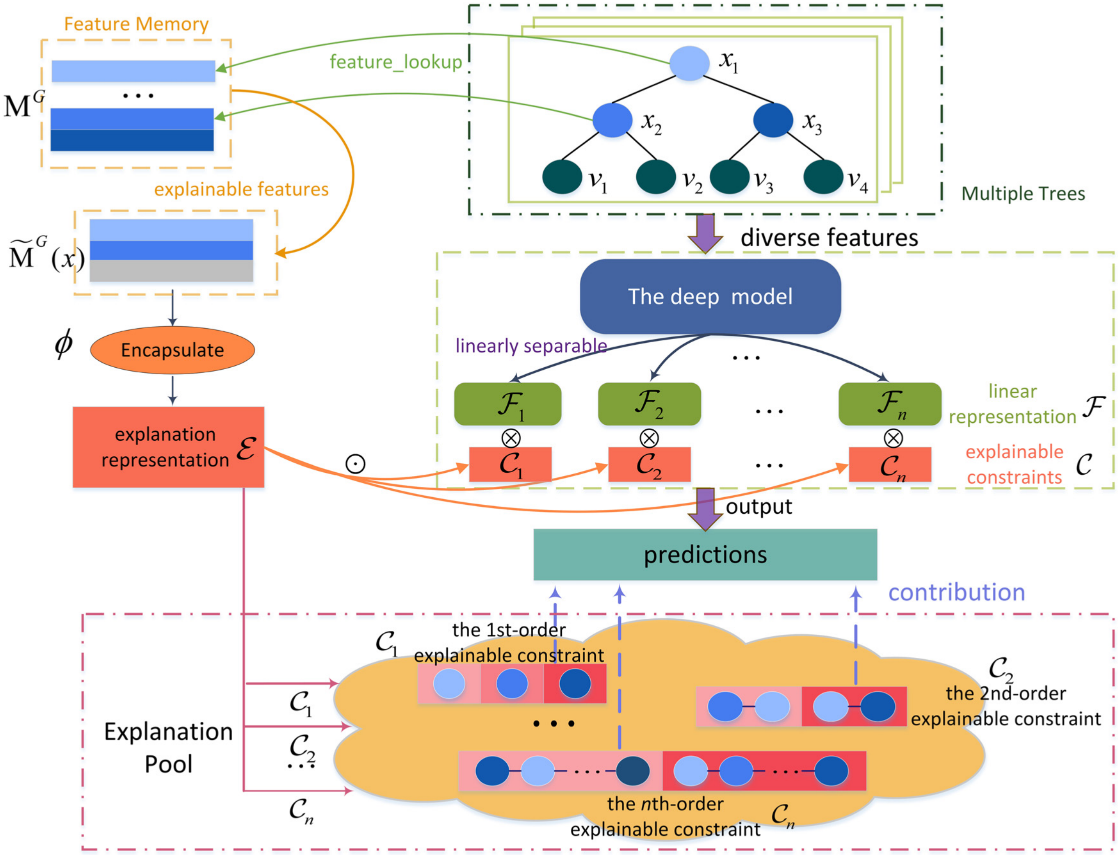


Figure 1: The framework of MFTE.

First, one of the important functions of GBDTs is feature selection, which is the basis for GBDTs' interpretability. On the other hand, deep models generally take features as input. Therefore, MFTE takes advantage of GBDTs by adopting the features that trees select to benefit prediction as the input of a deep model. To ensure a distinct input, MFTE employs multiple trees to select important features from all features of the original data. These diverse features are responsible for both prediction and interpretation; they run through the entire process of model training, prediction, and result explanation. Consequently, we design an independent feature representation method based on the memory mechanism [33,34] to achieve the full tracking of various features. In particular, we allocate an independent feature memory M^G for diverse features to support explainable storage, encapsulation, and representation.

The left side of Figure 1 shows the explainable feature and consistent representation mechanism. We borrow a feature-lookup operation to link the features selected from the trees with the memory and achieve the independent representation of feature memory $\tilde{\mathrm{M}}^G(x)$. However, purely independent representation cannot keep track of feature changes, especially when the deep model has multiorder feature interactions. To explore the explainable advantages of $\tilde{\mathrm{M}}^G(x)$, we design an encapsulating process ϕ to reorganize $\tilde{\mathrm{M}}^G(x)$ into a new explanation representation $\mathcal{E}$. The purpose of encapsulation is to track and interpret the deep model's multiorder feature interactions, because the encapsulating process can functionalize the memory features and uniformly represent the contribution of each order of the deep model. This kind of order-separated interpretation design requires the deep model to be linearly separable, such as in NFM [20] and attentional factorization machine (AFM) [21]. Specifically, assuming that the highest dimension of the input features is n (for example, $\prod_{i=1}^{n} x_i$), a certain model can be expressed as a function $\mathcal{F}$ of n partial combinations $\{\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_n\}$ from the first-order features to the nth-order feature interactions [35], where $\mathcal{F}_1$ only contains first-order features, $\mathcal{F}_2$ only contains second-order feature interactions, and so on. In this work, we consider a type of model $\mathcal{F}$ that can be expressed or approximately expressed as a linear combination from $\mathcal{F}_1$ to $\mathcal{F}_n$, namely $\mathcal{F} = \mathcal{F}_1 + \mathcal{F}_2 + \cdots + \mathcal{F}_n$.

Intuitively, n-order feature interactions have n different contributions to the prediction results. Therefore, we propose an explainable constraint method to quantify and explain the contribution of each order separately according to a prediction result. In particular, the constraint method combines n explanation representations $\mathcal{E}$ to generate an explainable constraint set $\{C_1, C_2, \ldots, C_n\}$. Each element in the constraint set corresponds to and constrains a suborder function of model $\mathcal{F}$. Motivated by the extension of Taylor's theorem in neural networks, we make a loosely coupled relationship between $\mathcal{F}$ and C. Thus, the deep model can be expressed as an n-order linear combination under n explainable constraints. Compared to the coefficients of Taylor expansions, our constraint coefficients are not derived from the original model but are actually extended consistent feature representations to improve the interpretability of that model. Therefore, through collaborative training between the explainable constraints and the deep model, n explainable constraints can separately express the contributions of n suborder functions without excessively interfering with $\mathcal{F}$'s prediction performance. The generation of the explainable constraint set and the details of loose coupling are introduced in Section 3.2.

In the explanation stage, MFTE allows different constraints C_i to explain the contributions of feature modeling belonging to different $\mathcal{F}_i$ in the predicted results. For example, C_1 represents the contribution of a single feature to the result; C_2 represents the contribution of pairwise feature interactions to the result; and C_n can explain the contribution of high-order nonlinear feature interactions to the result. We store the n-order explainable constraints in an explanation pool to allow MFTE to present different orders of explainable representation according to actual needs. For example, we can select only the first-order features, only the second-order feature interactions, or both for explanation. In particular, according to different selected features, we can retrieve the original GBDTs and visualize tree paths containing these features, thereby implementing multiorder and diversified explanations. Thus, this kind of feature explanation based on tree paths naturally achieves feature consistency between predictions and explanations.

3.2 The design of MFTE

In the part of generating diverse features, we define multiple GBDTs as $G^{\mathrm{multi}} = \{G_1, G_2, \ldots, G_T\}$, which represents T trees. Consider a single tree $G = (X, V)$ with feature nodes $X = \{x_1, x_2, \ldots, x_n\}$ and leaf nodes

$V = \{v_1, v_2, \ldots, v_m\}$. We train multiple trees by ensemble learning and use the following equation to select features:

$$G(x_i) = \begin{cases} x_i, & \text{if } x_i \in X_r \\ -1, & \text{otherwise}, \end{cases} \tag{1}$$

where $X_r \subset X$ is a set recording the feature nodes that ultimately fall on the rth leaf node v_r. The diverse features are generated by the features selected by T trees, thus we have T feature sets. The features in each feature set are used as input to the deep model on one side and used to find explainable feature vectors $\tilde{\mathrm{M}}^{G}(x)$ from the feature memory M^{G} on the other side. In particular, supposing that the feature memory M^{G} stores original vectors of all features, we select the explainable feature vectors according to the diverse features via a feature-lookup process defined as follows:

$$\tilde{\mathrm{M}}^{G}(x) = \mathrm{M}^{G} \sqcap G(x), \tag{2}$$

where $\sqcap$ indicates the selection of the corresponding vector according to the feature identification number. $\tilde{\mathrm{M}}^{G}(x)$ is the selected explainable feature set, where $\tilde{\mathrm{M}}^{G}(x_i)$ denotes the ith feature vector with k dimensions. To obtain the feature-related explanation representation, we apply ϕ to package the explainable features. The encapsulating process is defined as follows:

$$\mathcal{E} = \phi(\tilde{\mathrm{M}}^{G}(x)). \tag{3}$$

Here, the encapsulation function ϕ can be different according to different order subfunctions. For the first-order subfunction, ϕ can be a self-defined function. For subfunctions of order two or above, we need to consider the combination operation when performing encapsulation to obtain n explainable constraints. Since the explainable constraint of each order is related to T trees, we finally accumulate T-encapsulated features from T trees to obtain a single constraint. In particular, let $G_t(x)$ represent the features selected by the tth tree. Thus, $\phi(\tilde{\mathrm{M}}^{G_t}(x))$ indicates the encapsulated features from the tth tree according to equation (3). For n arbitrary combinations of explainable features $\{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$, we define the explainable constraints C as follows:

$$\begin{aligned} C_1 &= \sum_{t=1}^{T} C_1^t = \sum_{t=1}^{T} \mathcal{E}_{i_1}^t = \sum_{t=1}^{T} \left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right) \right) \right), \\ C_2 &= \sum_{t=1}^{T} C_2^t = \sum_{t=1}^{T} \left(\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t \right) = \sum_{t=1}^{T} \left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right) \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_2}\right) \right) \right), \\ &\cdots \\ C_n &= \sum_{t=1}^{T} C_n^t = \sum_{t=1}^{T} \left(\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t \odot \cdots \odot \mathcal{E}_{i_n}^t \right) \\ &= \sum_{t=1}^{T} \left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right) \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_2}\right) \odot \cdots \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_n}\right) \right) \right), \end{aligned} \tag{4}$$

- $\odot$: the combination operation that can be instantiated into specific operations
- $C_1 = \sum_{t=1}^{T} \mathcal{E}_{i_1}^t$: the first-order explainable constraint
- $C_2 = \sum_{t=1}^{T} (\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t)$: the second-order explainable constraint
- $C_n = \sum_{t=1}^{T} (\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t \odot \cdots \odot \mathcal{E}_{i_n}^t)$: the nth-order explainable constraint

where $i_l \in \{1, 2, \ldots, n\}$ and the combination operation $\odot$ is instantiated as element-wise multiplication in our experiments. Then, the n constraints are loosely coupled with the corresponding n-order subfunctions via a constraint process. Without loss of generality, we define an n-order linearly separable deep model containing a feature-independent variable $\mathcal{F}_0$ as follows:

$$\mathcal{F}^{\text{deep}} = \mathcal{F}_0 + \mathcal{F}_1 + \mathcal{F}_2 + \cdots + \mathcal{F}_n. \tag{5}$$

Let $C = \{C_1, C_2, \dots C_n\}$ represent the explainable constraints with respect to subfunctions $\{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n\}$. The constraint process is defined as follows:

$$\begin{aligned}\tilde{\mathcal{F}} &= \mathcal{F}_0 + C_1 \otimes \mathcal{F}_1 + C_2 \otimes \mathcal{F}_2 + \cdots + C_n \otimes \mathcal{F}_n \\ &= \mathcal{F}_0 + \sum_{t=1}^{T} C_1^t \otimes \mathcal{F}_1 + \sum_{t=1}^{T} C_2^t \otimes \mathcal{F}_2 + \cdots + \sum_{t=1}^{T} C_n^t \otimes \mathcal{F}_n \\ &= \mathcal{F}_0 + \sum_{t=1}^{T}\left(\mathcal{E}_{i_1}^t\right) \otimes \mathcal{F}_1 + \sum_{t=1}^{T}\left(\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t\right) \otimes \mathcal{F}_2 + \cdots + \sum_{t=1}^{T}\left(\mathcal{E}_{i_1}^t \odot \mathcal{E}_{i_2}^t \odot \cdots \odot \mathcal{E}_{i_n}^t\right) \otimes \mathcal{F}_n \\ &= \mathcal{F}_0 + \sum_{t=1}^{T}\left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right)\right)\right) \otimes \mathcal{F}_1 + \sum_{t=1}^{T}\left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right) \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_2}\right)\right)\right) \otimes \mathcal{F}_2 \\ &\quad + \cdots + \sum_{t=1}^{T}\left(\phi\left(\tilde{\mathrm{M}}^{G_t}\left(x_{i_1}\right) \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_2}\right) \odot \cdots \odot \tilde{\mathrm{M}}^{G_t}\left(x_{i_n}\right)\right)\right) \otimes \mathcal{F}_n\end{aligned} \tag{6}$$

- $\tilde{\mathcal{F}}$: the final prediction result
- $\mathcal{F}_0$: indicating variables that are feature-independent
- $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_n$: subfunctions representing suborder feature interactions
- $\otimes$: the constraint operation that can be instantiated into specific functions.

It can be observed that the final prediction result $\tilde{\mathcal{F}}$ is a linear combination of n subfunctions after being constrained. In a simple case, the constraint operation $\otimes$ can be multiplication to achieve collaborative training. Specifically, the constraint term and the subfunction become the gradient of each other during the gradient optimization process. In this case, the nth-order constraint C_n is naturally optimized as an explainable representation of the contribution of the nth-order function $\mathcal{F}_n$. The constraint operation is superior to the cross-feature mechanism of TEM [27], because the cross-feature mechanism fuses multiorder feature information before the deep model training, so that the information of each order feature cannot be tracked separately. In addition to being consistent with each order of the deep model, MFTE does not need to modify the original loss function during training. For example, in the two domain applications of this work, we employ the binary cross-entropy of the original deep model as the MFTE loss function L to solve the binary classification problem as follows:

$$L(\mathcal{F}, \tilde{\mathcal{F}}) \triangleq -\frac{1}{N}\sum_{h=1}^{N}[F_h \cdot \log(\tilde{F}_h) + (1 - F_h)\cdot \log(1 - \tilde{F}_h)]. \tag{7}$$

Here, N denotes the number of samples in a training batch, F_h indicates a binary ground-truth value that can be 1 or 0, and $\tilde{F}_h$ is the predicted value corresponding to F_h. By minimizing $L(\mathcal{F}, \tilde{\mathcal{F}})$, we can complete the training of the entire model.

4 The instantiation of MFTE

Considering that MFTE can be applied to the n-order linearly separable deep model, we adopt NFM [20] to instantiate our strategy. The difference between instantiated MFTE and NFM is that each order feature of NFM is not constrained by the feature-tracking strategy available in MFTE. Because NFM is a general deep version of factorization machines (FMs) [22,35] and is directly linearly separable, it is a multiorder feature interaction model that contains first- and second-order subfunctions. Consequently, we employ the first two orders of the constraints C_1 and C_2 to constrain the first- and second-order parts of NFM, respectively. Let $G(x)$ represent the union feature set selected by T trees. By using the tree-selected features $G(x_i)$ in equation (1) as input, the combined NFM model with explainable constraints is defined as follows:

$$\tilde{\mathcal{F}}_{\mathrm{NFM}} = \mathcal{F}_0 + C_1 \otimes \mathcal{F}_1 + C_2 \otimes \mathcal{F}_2 = w_0 + \sum_{t=1}^{T} C_1^t \otimes \sum_{i=1}^{n} w_i G(x_i) + \sum_{t=1}^{T} C_2^t \otimes f_{\mathrm{deep}}\left(\sum_{i=1}^{n}\sum_{j=i+1}^{n} (G(x_i)\mathbf{v}_i \odot G(x_j)\mathbf{v}_j)\right) \tag{8}$$

- w_0: model bias that is feature-independent
- w_i: model parameters of the first-order feature modeling
- f_{deep}: neural network with L deep layers for the second-order feature modeling
- $\mathbf{v}$: model embedding vectors corresponding to features.

We specifically define the constraint operation $\otimes$ as feature corresponding multiplication. For example, we multiply $w_i G(x_i)$ by C_1^i and leverage C_2^{ij} to multiply and constrain $(G(x_i)\mathbf{v}_i \odot G(x_j)\mathbf{v}_j)$, where the formal definitions of C_1^i and C_2^{ij} are provided in the following discussion.

First, based on the explainable feature memory $\tilde{\mathrm{M}}^G(x)$, we specify the encapsulation function ϕ as follows:

$$\phi : \begin{cases} e = s^T \mathrm{ReLU}(\mathrm{W} \cdot \tilde{\mathrm{M}}^G(x) + b) \\ \mathcal{E} = \mathrm{softmax}(e) \end{cases} \tag{9}$$

where the explanation representation $\mathcal{E}$ is employed for encapsulating the explainable feature memory $\tilde{\mathrm{M}}^G(x)$. $\mathrm{W} \in \mathbb{R}^{d \times k}$ and $b \in \mathbb{R}^d$ are the parameters of a `Relu`-activated dense layer; and $s \in \mathbb{R}^d$ is used to adjust the shape of the explanation representation to fit a specific feature. d indicates the dimension of hidden layers and k denotes the dimension of features. For the ith feature memory $\tilde{\mathrm{M}}^G(x_i)$, the corresponding first-order explainable constraint is specified as follows:

$$\begin{aligned} e_i^t &= s^T \mathrm{ReLU}\big(\mathrm{W} \cdot \tilde{\mathrm{M}}^{G_t}(x_i) + b\big) \\ C_1^i &= \sum_{t=1}^{T} p^T \mathcal{E}_i^t = \sum_{t=1}^{T} p^T \mathrm{softmax}(e_i^t), \end{aligned} \tag{10}$$

where p is a weight matrix that can change the shape of C_1^i to adapt to the combination of subfunctions. Furthermore, given the jth feature memory $\tilde{\mathrm{M}}^G(x_j)$, we specify the second-order constraint C_2 based on the feature memory interaction as follows:

$$\begin{aligned} e_i^t \odot e_j^t &= s^T \mathrm{ReLU}\big(\mathrm{W}\big(\tilde{\mathrm{M}}^{G_t}(x_i) \odot \tilde{\mathrm{M}}^{G_t}(x_j)\big) + b\big) \\ C_2^{ij} &= \sum_{t=1}^{T} q^T (\mathcal{E}_i^t \odot \mathcal{E}_j^t) = \sum_{t=1}^{T} q^T \mathrm{softmax}(e_i^t \odot e_j^t). \end{aligned} \tag{11}$$

The combination operation $\odot$ denotes the elementwise multiplication to represent feature memory interaction. q indicates a weight matrix that can transfer the shape of C_2^{ij} to adapt to the combination of subfunctions. After C_1^i and C_2^{ij} are trained, they can represent the contributions of the first-order features and the second-order interactions to the prediction results, thus achieving diversification of explanations. Furthermore, we will demonstrate the detailed advantages of this diversified and multiorder explanation in the experiments section.

5 Experiments

To verify the interpretability of our strategy, we selected two datasets in different application fields (education and marketing) for experiments. We conduct various experimental analyses on these two datasets to try to answer the following research questions:

- Research Question 1 (**RQ1**): Can MFTE achieve intuitive explanations without reducing the deep model's performance?

- Research Question 2 (**RQ2**): Can the explanations of MFTE maximize the explainable advantages of trees?
- Research Question 3 (**RQ3**): What are the highlights of MFTE's multiorder explanations, and can it achieve diversified explanations?

To answer the above research questions, we first describe two datasets in Section 5.1. Then, Section 5.2 attempts to solve the performance maintenance problem in RQ1. Next, Section 5.3 answers both RQ2 and RQ3 through the experimental results and detailed analysis. Finally, we discuss the entire experiment in Section 5.4.

5.1 Data description

The **education dataset**[1] [46,47] collects students' online learning behaviors in five online courses launched by Harvard University on the EDX platform, covering the period from the autumn of 2012 to the summer of 2013. The features in this dataset can be roughly divided into two categories. One is related to the students themselves, including *birth_year*, *gender*, *education*, etc. The other represents the interactive behaviors between students and courses, including *total_events* (total events in the server log file, including the number of clicks), *active_days* (the days of a student participating in course activities), *num_chapters* (the number of chapters a student learned), *days* (the number of days between a student's registration and the completion of a course), etc. We employ the dataset to predict and explain whether a student can obtain course certification (ground truth = 1 means the student can obtain course certificates, ground truth = 0 means the student cannot obtain course certificates).

The **marketing dataset**[2] [48] implemented by the marketing team of the Bank of the Portugal (2008–2015), stores information about the telemarketing business to attract clients to subscribe to term deposits. This dataset is used to predict whether a client will subscribe (yes/no) a term deposit, where the ground truths contain two variables: "Yes (ground truth = 1)" and "No (ground truth = 0)." There are four categories of features. The first category is client information, including age and mortgage. The second category is social and economic factors, including *emp.var.rate* (employment variation rate – quarterly indicator), *cons.price.idx* (consumer price index – monthly indicator), *euribor3m* (euribor 3 month rate – daily indicator), *nr.employed* (number of employees – quarterly indicator), etc. The third category is related to the last contact of the current *campaign*, such as *month* (last contact month of year) and *pdays* (number of days that passed by after the client was last contacted from a previous campaign). The final category contains all other features, such as *campaign*.

5.2 Performance evaluations (RQ1)

We have instantiated MFTE (called MFTE_N) with NFM in Section 4. In addition to NFM, other members of the FM family such as the embedding version of FM [30] and AFM are also linearly separable. Therefore, in the experiment, we also instantiated FM (named MFTE_F) and AFM (called MFTE_A), together with MFTE_N as our models. We compare the three instantiated MFTE approaches with FM, NFM, and AFM to evaluate whether the MFTE strategy can maintain the same performance as the original models. In addition, XGBoost, an advanced method representing GBDTs, is also employed for performance comparison, because we can combine XGBoost and SHAP to implement feature-attribution explanations in subsequent experiments. Finally, we introduce TEM as another comparison method, because it is an interpretable method, where a deep model accepts tree-selected features as input and shows good performance and explanation effects. All comparison methods are as follows:

[1] https://doi.org/10.7910/DVN/26147.

[2] http://archive.ics.uci.edu/ml/datasets/Bank+Marketing.

- **FM** [22] – the embedding version [30] of Factorization Machine;
- **NFM** [20] – the Neural Factorization Machine;
- **AFM** [21] – the Attentional Factorization Machine;
- **XGBoost** [13] – A Scalable Tree Boosting System;
- **TEM** [27] – the Tree-enhanced Embedding Model;
- **MFTE$_F$** – our multiorder feature-tracking explanation instantiated model for FM;
- **MFTE$_N$** – our multiorder feature-tracking explanation instantiated model for NFM;
- **MFTE$_A$** – our multiorder feature-tracking explanation instantiated model for AFM.

On the education dataset, we uniformly set the regularized terms of the XGBoost and TEM models to 0.01, and the regularization values of the FM family models and the corresponding MFTE instantiated models are set to 0.001. For optimal learning rates, FM/MFTE$_F$ are set to 0.00001; NFM/MFTE$_N$ and AFM/MFTE$_A$ are adjusted to 0.000001 and 0.0001, respectively; XGBoost is set to 0.0001 and TEM is optimized to 0.00001. In our models, to ensure the diversity of features, we set the number of GBDTs to 30 and the height of all trees to 4. Moreover, XGBoost and the tree model part of the TEM maintain the same settings as our models. On the marketing dataset, the setting of the learning rate and regularized value is consistent with that of the education dataset, but the number of GBDTs and the height of the tree model are set to 38 and 5, respectively. For performance evaluation, we adopt the widely accepted area under the receiver operating characteristic curve (AUC) and F1-measure as metrics. The performance comparison results of all models are shown in Tables 1 and 2.

Table 1: The performance comparison results on the education dataset

Methods	FM	MFTE$_F$	NFM	MFTE$_N$	AFM	MFTE$_A$	XGBoost	TEM
AUC	0.9768	0.9599	0.9416	0.9515	0.9744	**0.9770**	0.9364	0.9356
F1-measure	0.3957	**0.4133**	0.3729	0.3978	0.3950	0.4076	0.2423	0.3583

The bold values indicate the best performance on AUC or F1-measure.

Table 2: The performance comparison results on the marketing dataset

Methods	FM	MFTE$_F$	NFM	MFTE$_N$	AFM	MFTE$_A$	XGBoost	TEM
AUC	0.7213	0.7416	0.7583	**0.7725**	0.7625	0.7673	0.7657	0.7657
F1-measure	0.3411	0.3958	0.4061	0.4192	0.3906	**0.4215**	0.4131	0.3694

The bold values indicate the best performance on AUC or F1-measure.

It can be observed that the two explainable models MFTE$_F$ and MFTE$_A$ achieve the best performance on F1 and AUC, respectively, with the education dataset. Besides the AUC value of the MFTE$_F$ being slightly weaker than that of the FM, both MFTE$_N$ and MFTE$_A$ have improved performance over their original models. On the marketing dataset, MFTE$_A$ and MFTE$_N$ outperform other models on F1 and AUC. The comparison results show that although MFTE is designed for explanation, it can maintain the performance of the original deep models, which provides a positive answer to RQ1. In subsequent experiments, we focus on the explanation comparison of MFTE with other methods.

5.3 Explanation comparison (RQ2 and RQ3)

The explainable effect is mainly reflected in whether the model properly shows the contribution of features. General tree models (such as XGBoost) directly employ FI [13,38] to explain the contribution of each feature. SHAP provides both first- and second-order explanation tools for trees to calculate diverse contribution values. TEM is a typical method combining trees and a deep network. It uses the cross-features selected by the trees as the input of the deep model and leverages the attention of the cross-features to explain the results. In contrast, our model achieves diverse explanation effects through multiorder feature tracking. Specifically, the comparable explanation methods in this section are summarized as follows:

- **FI** – a traditional explainable approach based on XGBoost [13,38].
- **SHAP** – a representative FI explanation method, including first- and second-order explanations. We employ the tree-version SHAP [10].
- **TEM** – an explainable method based on tree and only provides cross-feature explanation extracted from tree components [27].
- **MFTE** – our approach that supports multiorder explanations. We employ the MFTE_N version to provide first- and second-order explanations.

To better compare the explanation effects, we separately analyze the explanation results of the two fields in Sections 5.3.1 and 5.3.2, respectively. In each section, we separate the first- and second-order explanations and compare them to better present the experimental results.

5.3.1 Evaluation in the education field

5.3.1.1 Statistical analysis

First, we provide a simple statistical feature correlation analysis. The result shows that if a student certified in the course has the greatest correlation with the *explored* feature, then the correlation value is 0.5. In addition, when ground truth = 0, more than half of the *active_days* values are concentrated in the range of [0, 15]. For over 60% of the students, the corresponding *active_days* values are located in [15,50], when ground truth = 1.

5.3.1.2 The first-order explanations

Statistical analysis mainly describes the distribution of feature values, but it is impossible to know the exact feature contributions to the prediction. Tree-based models can calculate the FI of the whole model and regard it as the model explanation. We apply FI-based XGBoost to the education dataset and obtain FI rankings from largest to smallest: *total_events*, *num_chapters*, *days*, *active_days*. The higher the importance of the feature, the greater its contribution to the prediction result. However, like statistical analysis, FI reflects the global contribution of all the features, so there is only one sort of FI and it is fixed. Compared with FI, SHAP can flexibly calculate the contribution value of each feature to the prediction result of a single sample. To evaluate SHAP's explanation toward individuals, we randomly selected two students, representing those who obtained a certificate and those who did not and listed the relevant feature values and ground truths in Table 3. Correspondingly, Figures 2 and 3 show the SHAP values of the two samples.

Table 3: The relevant feature values of the two samples on education dataset

stu_id	explored	days	total_events	active_days	num_chapters	ground truth
1571	0	216	8	1	1	0 (uncertified)
2264	1	239	2707	50	10	1 (certified)

The red values represent that the features play positive roles in predicting that the student can obtain the certificate, whereas the blue ones play negative roles. In particular, the longer the length of the color bar, the greater the absolute value of the feature's contribution. Therefore, the most contributing feature is *num_chapters* in Figure 2. By searching the relevant feature values in Table 3, we observe that *num_chapters*

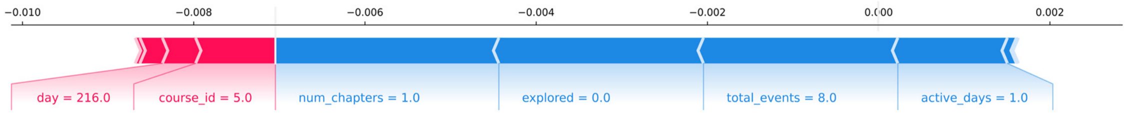


Figure 2: The first-order explanation of SHAP for an uncertified student with id = 1571.

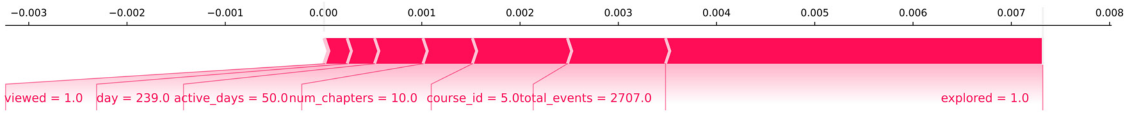


Figure 3: The first-order explanation of SHAP for a certified student with id = 2264.

= 1, indicating that the student with id = 1571 has only read one chapter. The salient feature of Figure 3 is *explored* (*explored* = 1), indicating that the student (id = 2264) has explored the whole course. Although the predictions of the two students are consistent with the ground truths, the tree-based SHAP value depends on the original feature value. When the original feature value is relatively small, it will affect its feature contribution value.

For MFTE, the first-order explanation values corresponding to the two samples are shown in Table 4. It can be observed that the first-order explanation of each feature has multiple values corresponding to multiple trees, meaning that MFTE achieves the first-order diversified explanations. The order of average feature contributions of the student with id = 1571 is: *explored* > *active_days* > *days* and the order of average feature contributions of the student with id = 2264 is: *active_days* > *num_chapters* > *total_events*.

According to the feature values in Table 3, the values of *active_days* and *num_chapters* of the student with id = 2264 is 50 and 10, respectively (the total number of days is 71 and the total number of chapters is 11). Thus, the first-order explanation of MFTE makes sense and MFTE further provides more diversified explanations.

5.3.1.3 The high-order explanations

Figure 4 shows the second-order SHAP explanation heatmap corresponding to the two samples, where we only select the features with large SHAP values and visualize their second-order contributions. The second-order feature explanations mainly indicate the contributions of the feature interactions to the predictions. Consequently, we do not consider the interaction value between the feature and itself, that is, the contribution value on the diagonal. In this case, for the student with id = 1571, the *course_id-total_events* feature interaction has the largest negative contribution (−0.00064) to the prediction. In contrast, for the student with id = 2264, the *course_id-viewed* and *num_chapters-course_id* feature interactions both have the largest positive contributions (0.00019) to the prediction.

Although the SHAP values are calculated by training the XGBoost approach, the second-order explanations of SHAP do not reflect the relationship of features in the trees. In contrast, the TEM model introduces “cross features” in the trees and takes the cross-feature embeddings as the input to a deep attention model. Figure 5 shows the attention values of the features included on the cross-feature paths (e.g., v66) and uses them as an explanation.

Table 5 lists the relevant cross-feature paths corresponding to Figure 5. For example, path *v*23 contains cross-features *explored*, *num_chapters*, *active_days* and *days*, meaning that they are on the same tree. However, these cross feature paths in the trees are fixed, and these features cannot be tracked continuously in the subsequent deep model training, causing the trees to be out of touch with the deep model.

In Figure 6, we illustrate both the first-order explanations (represented by rectangular boxes) and the second-order explanations (represented by ellipses) of MFTE. The greater the absolute value of the explanation, the greater its contribution to the result. Thus, the five features in the figure are the important features that contribute massively to the results. Moreover, the feature selection paths containing these

Table 4: The first-order explanation values of the two samples in MFTE

stu_id	explored	days	total_events	active_days	num_chapters	prediction
1571	(0, −1.47)*	(0.071, 0.407, −1.44)	(0.0002, −0.0257)	(−1.067, −0.0013)	(−0.374, −0.0214)	Uncertified
2264	(−0.0013, −0.0002)	(−0.003, −0.0043)	(0.0041, 0.235)	(2.6, 1.42, 3)	(0.298, 0.0012, −0.008, 1.234)	Certified

* The values in brackets are the explainable values of features on different trees, and the number of explainable values represents the times of the features selected by the trees.

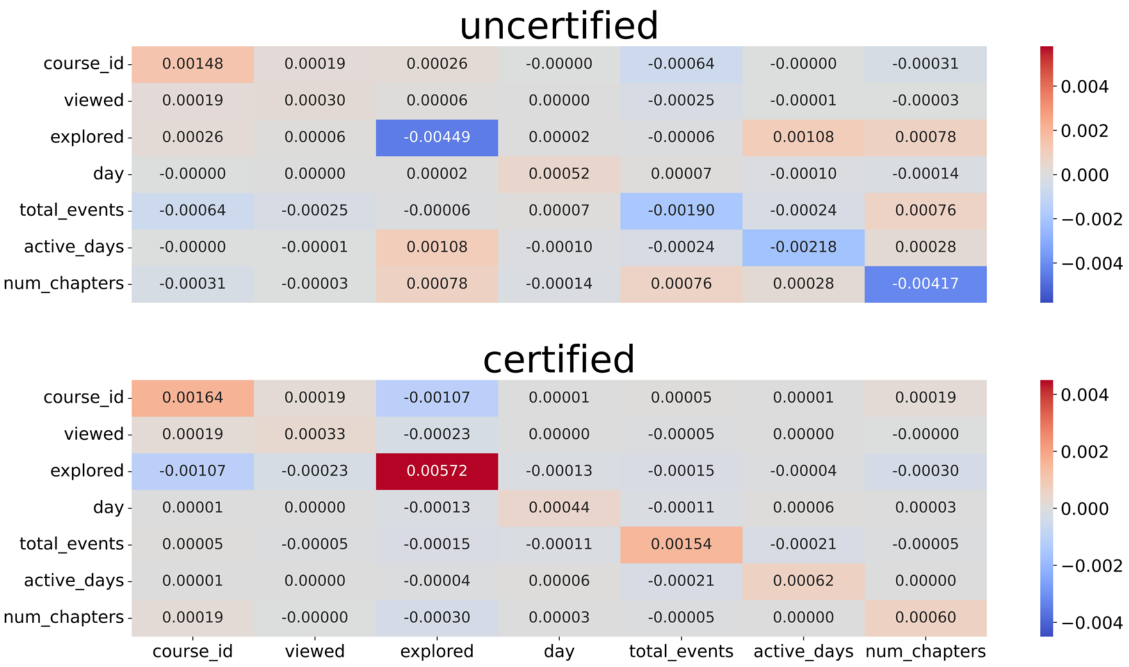


Figure 4: The second-order explanations of the uncertified & certified predictions in SHAP.

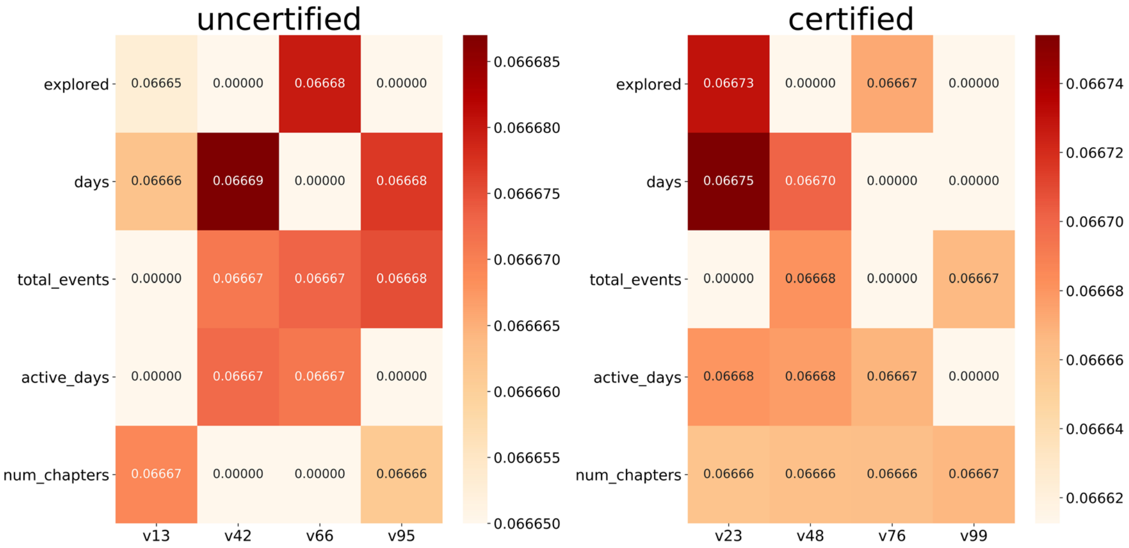
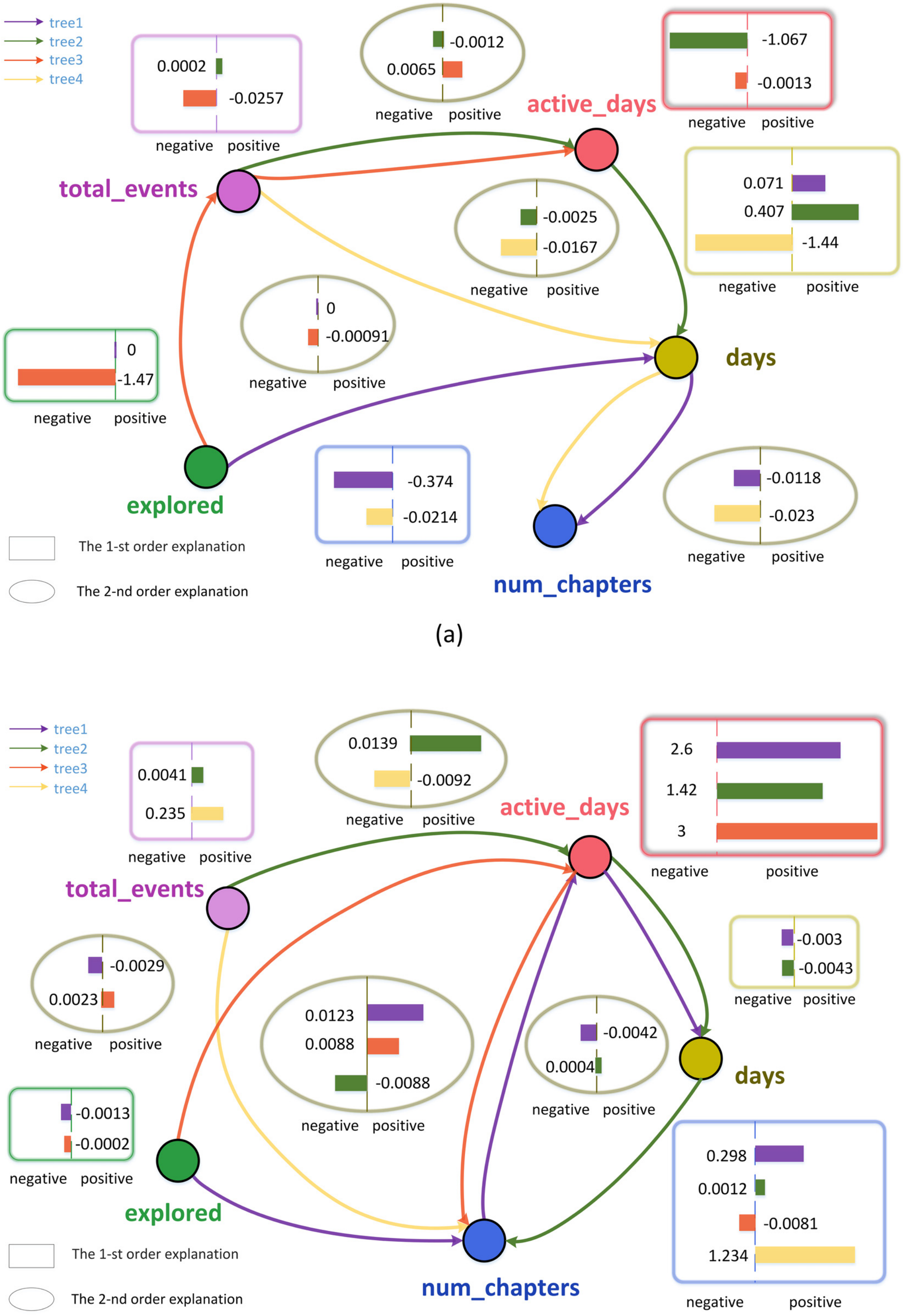


Figure 5: The attention explanations of the uncertified and certified predictions in TEM.

Table 5: The tree paths of the uncertified and certified predictions in TEM

tree_path	Details of the cross-features on the paths
v_{23}	[explored = 1] & [num_chapters = 10] & [active_days = 50] &[days = 239]
v_{42}	[total_events = 8] & [active_days = 1] & [days = 216]
v_{48}	[total_events = 2707] & [active_days = 50] & [days = 239] & [num_chapters = 10]
v_{66}	[explored = 0] & [total_events = 8] & [active_days = 1]
v_{76}	[explored = 1] & [active_days = 50] & [num_chapters = 10]



(a)

(b)

Figure 6: The first- and second-order explanations of the uncertified & certified predictions in MFTE. (a) A multiorder explanation for predicting uncertified with stu_id = 1571, and (b) a multiorder explanation for predicting certified with stu_id = 2264.

features in the original four trees are randomly retrieved and displayed in Figure 6. We employ straight lines with arrows to represent the optimal feature selection path from the root of the tree to its leaf nodes and adopt different colors to represent the different trees. The first-order explanation in the figure corresponds to the tree nodes, and the second-order explanation corresponds to directed edges. The positive or negative explainable values denote the positive or negative contribution of the tree features to the predicted results. In this way, the total contribution of the tree can be obtained by combining the contributions of the tree nodes and edges. If the overall contribution is greater than 0, it indicates that a tree predicts that the student will obtain a course certificate.

The explanation characteristic of MFTE is that it can illustrate the feature selection path in trees, making the relationship between first- and second-order explanations clear. Moving forward, the multi-order explanations are less disturbed by the original feature values. For example, take the student with id = 1571. Figure 6(a) shows that the feature *days* in different trees have both a large positive and negative impact on the predicted results. Table 3 shows that the feature value of *days* is 216, which may make people think that the student has been studying for a long time. In this case, we need to further investigate the second-order explanations to analyze the impact of *days*. It can be found that the interactions of the three pairs of features: *num_chapters-days*, *active_days-days*, and *explored-days*, all contribute negatively to the results, because the original values of these three features interacting with days are very small. In addition, MFTE believes that the largest negative second-order contribution comes from the feature interaction of *explored-total_events*. Combining these two conditions (*explored* = 0 and *total_events* = 8) greatly increases the probability that the student will not be able to obtain course certification. For the student with id = 2264 in Figure 6(b), MFTE automatically learns two second-order feature interaction pairs that contribute massively: *active_days-num_chapters* and *total_events-active_days*. Table 3 shows that *total_events* (i.e., the number of clicks) of the student with id = 2264 is as high as 2707. Therefore, the multiorder explanations of MFTE can complement each other to better understand the students' certification results.

5.3.2 Evaluation in the marketing field

5.3.2.1 Statistical analysis

According to the correlation analysis, the correlations between marketing result (ground truth) and features *emp.var.rate*, *cons.price.idx*, *euribor3m*, *nr.employed* and *campaign* are −0.3, −0.14, −0.31, −0.35, and −0.066, respectively. The smaller the negative correlation values of these four features, the more likely the marketing result is to be successful (ground truth = 1), otherwise it may be fail (ground truth = 0). For clients with successful marketing result, the corresponding *euribor3m* values are mainly distributed in [0.68, 1.4], while for most clients with failed marketing results, the *euribor3m* values are around 5.

5.3.2.2 The first-order explanations

We apply FI-based XGBoost to the marketing dataset and obtain the important "weight" of all the features, whereas the top-3 important features are: *euribor3m*, *month* and *pdays*. To compare with tree-based SHAP, we randomly select a sample of failed marketing (client_id = 2556) and a sample of successful marketing (client_id = 1666) as examples. The first-order SHAP contribution values of the two samples are shown in Figures 7 and 8, respectively. Table 6 lists the relevant and important feature values according to the SHAP results.

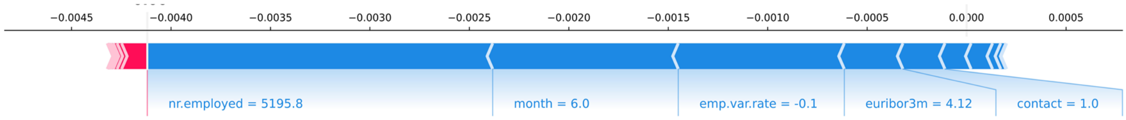


Figure 7: The first-order SHAP explanation of a failed marketing sample with id = 2556.

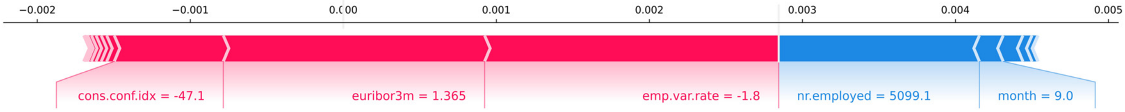


Figure 8: The first-order SHAP explanation of a successful marketing sample with id = 1666.

Table 6: The relevant feature values of the two samples on the marketing dataset

Client_id	emp.var.rate	cons.price.idx	euribor3m	nr.employed	campaign	ground truth
2556	−0.1	93.2	4.12	5195.8	1	0 (failure)
1666	−1.8	93.075	1.365	5099.1	1	1 (success)

For the sample with id = 2556, the feature *nr.employed* has the largest contribution value and the relevant feature value is 5195.8. In our correlation analysis results, the correlation value between *nr.employed* and the ground truth is −0.31, meaning that the smaller the feature value of *nr.employed*, the more likely the marketing is to succeed. In this case, the feature value 5195.8 is higher than the average value 5167, which indicates that the first-order SHAP explanation is consistent with the correlation analysis results. For the successful sample with id = 1666, the feature with the largest contribution value is *emp.var.rate* and its SHAP value is 0.002.

Table 7 lists the first-order explanations of MFTE. For the sample with id = 2556, the top-3 important features are: *emp.var.rate*, *euribor3m*, and *nr.employed*. For the sample with id = 1666, its top-3 important features are: *nr.employed*, *euribor3m*, and *emp.var.rate*. The contribution values of two samples can be compared with feature values to explain the results. In addition, the feature *emp.var.rate* is important in both samples because it appears three times on four trees. Therefore, the first-order explanations of MFTE can intuitively reflect the importance of features through the number of contributions and their specific values.

Table 7: The first-order explanation values of the two samples in MFTE

Client_id	emp.var.rate	cons.price.idx	euribor3m	nr.employed	campaign	prediction
2556	(−0.62, −0.75, −0.043)	(0.001)	(−0.36)	(−0.0214, −0.64)	(0.051, −0.019)	Failure
1666	(0.99, −0.12, 0.091)	(−0.13)	(1.6, −0.0011)	(−0.051, 1.9)	(0.38, 0.00013)	Success

5.3.2.3 The high-order explanations

Figure 9 shows the second-order explanations of SHAP for the two samples. Their most important contributions of feature interaction are both *euribor3m-emp.var.rate*. In particular, the second-order SHAP values of *euribor3m-emp.var.rate* of the sample with id = 2556 and id = 1666 are −0.00019 and 0.00064, respectively. Through the color of the contribution values, we can intuitively see that the second-order explanations of SHAP are reasonable.

In contrast, the explanations of TEM in Figure 10 can provide more details for the predictions with the help of the tree paths in Table 8. For the sample with id = 2556, the highest attention value comes from the feature *emp.var.rate* on the path *v*135. The other two features that have massive contributions on the path *v*135 are *month* and *nr.employed*. It indicates that the three features interact on the same tree and have an important impact on the prediction. Similarly, for the sample with id = 1666, the key path is *v*83 because it contains four important cross-features, whereas one of them has the highest attention value.

The symbol in Figure 11 is consistent with Figure 6. It can be observed that the first-order explanations of feature *campaign* includes both positive and negative contributions in both samples. In this case, by further observing the interaction between feature *campaign* and feature *euribo3m*, it can be seen that

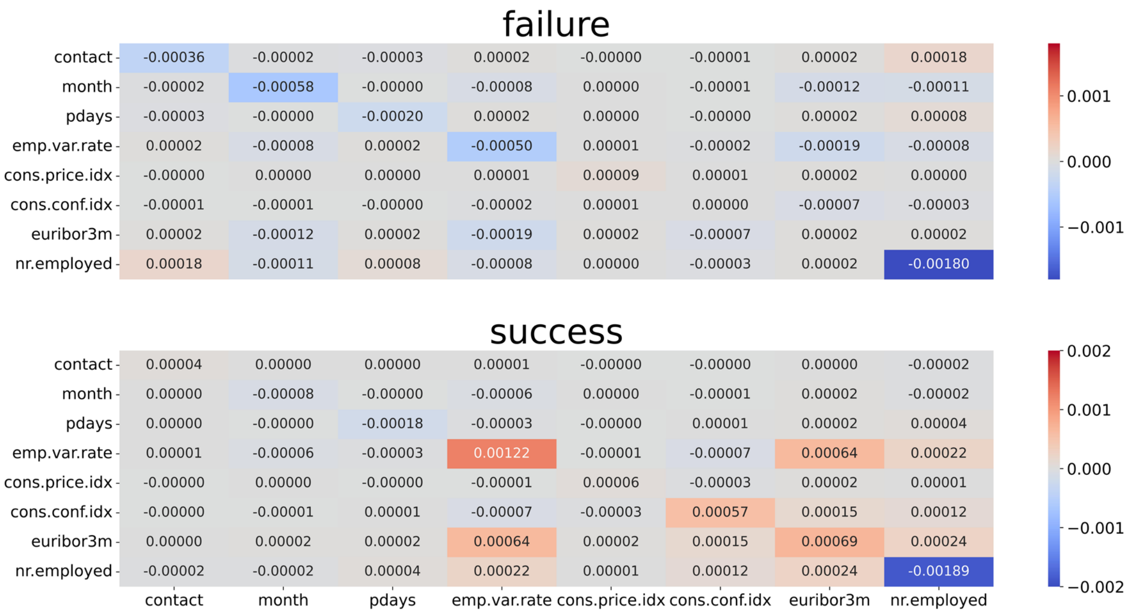


Figure 9: The second-order explanations of the failure and success predictions in SHAP.

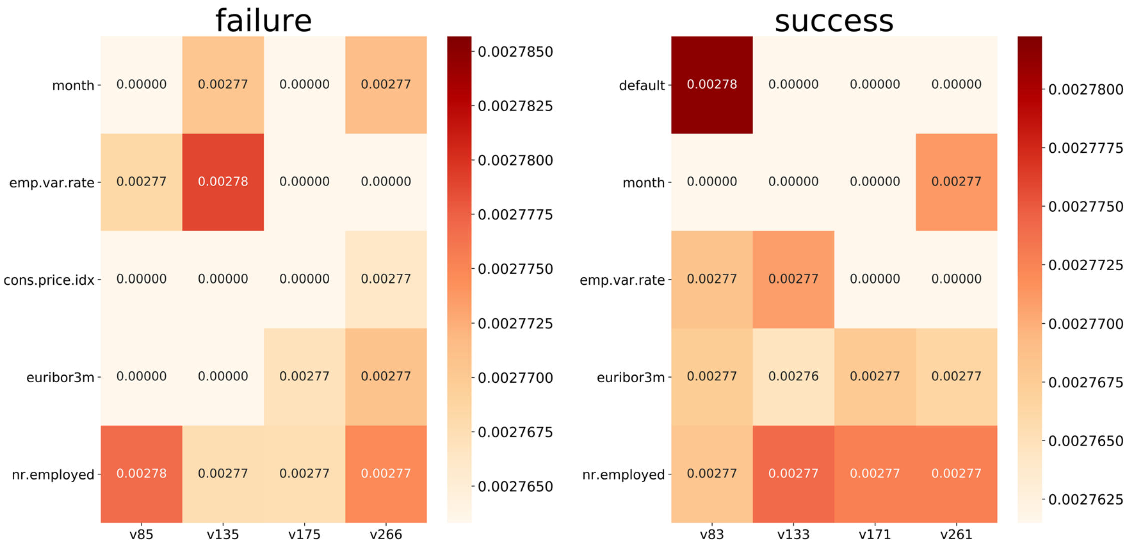
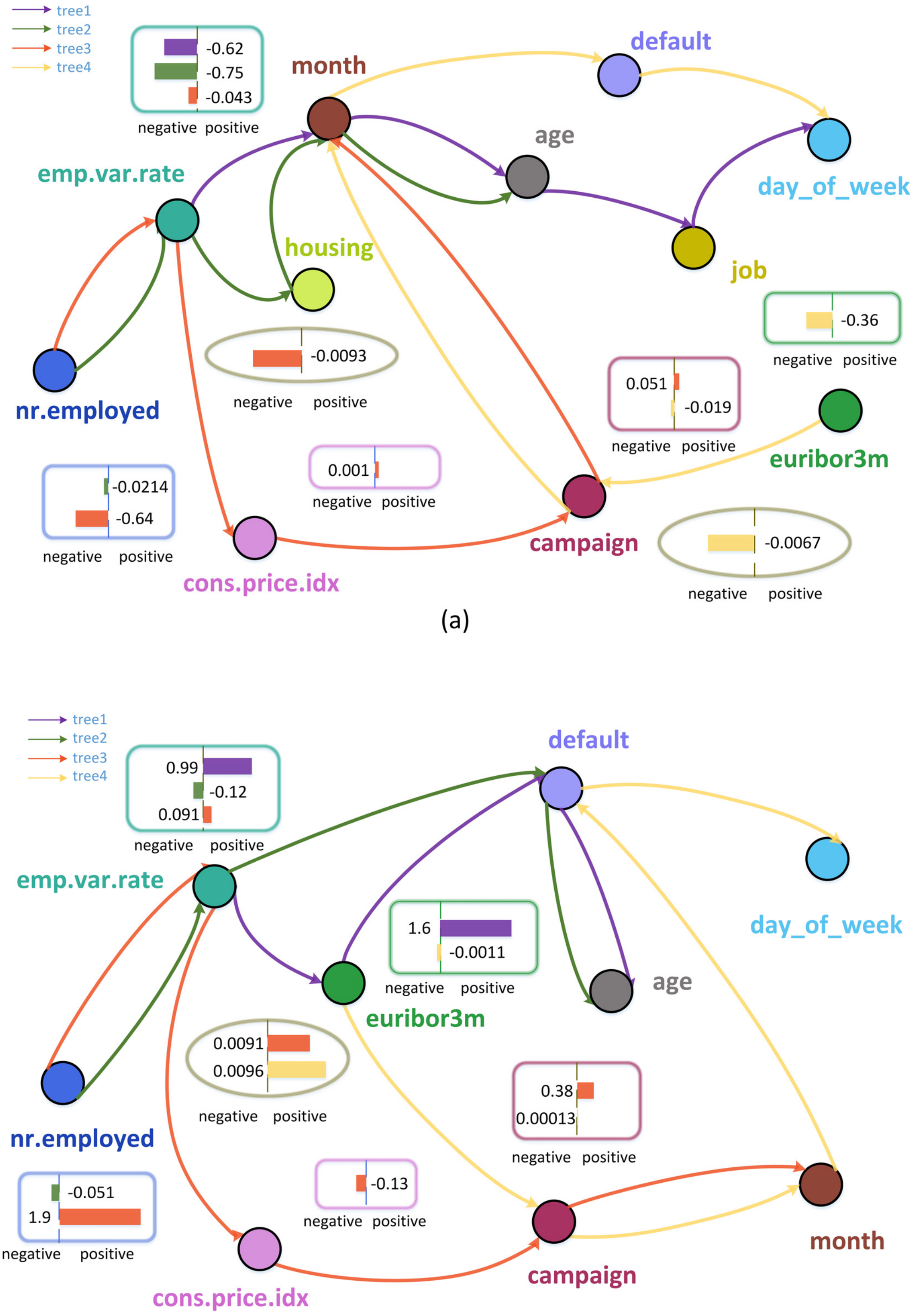


Figure 10: The attention explanations of the failure and success predictions in TEM.

Table 8: The tree paths of the failure and success predictions in TEM

tree_path	**Details of the cross-features on the paths**
v_{83}	[nr.employed = 5099.1] & [emp.var.rate = -1.8] & [euribor3m = 1.365] &[default = no]
v_{85}	[nr.employed = 5195.8] & [emp.var.rate = -0.1]
v_{133}	[nr.employed = 5099.1] & [emp.var.rate = -1.8] & [euribor3m = 1.365]
v_{135}	[nr.employed = 5195.8] & [emp.var.rate = -0.1] & [month = November]
v_{261}	[nr.employed = 5099.1] & [month = April] & [euribor3m = 1.365]



(a)

(b)

Figure 11: The first- and second-order explanations of the failure and success predictions in MFTE. (a) A multiorder explanation for predicting failure with client_id = 2556, and (b) a multiorder explanation for predicting success with client_id = 1666.

feature *campaign* has an indirect contribution to the prediction of the result. Specifically, for the sample with id = 2556 in Figure 11(a), the second-order explanation of *campaign-euribo3m* interaction has a negative value. In contrast, for the sample with id = 1666 in Figure 11(b), the second-order explanation of *campaign-euribo3m* interaction contributes positively to the prediction, which is consistent with the ground truth. According to the marketing dataset, *campaign* means "number of contacts performed during this campaign and for this client," whereas *euribo3m* indicates "euribor 3 month rate – daily indicator." The corresponding feature values in Table 6 show that the *campaign* values of both samples is 1. The difference is that the *euribo3m* value for the fail prediction is 4.12, while the *euribo3m* value for the success prediction is 1.365. Consequently, the second-order contributions from *campaign-euribo3m* of both samples are consistent with the original feature meanings, which means that the second-order explanations of MFTE are appropriate. The experimental results in these two fields empirically answer RQ2 and RQ3. In the next section, we will further answer the three research questions to summarize the experiments.

5.4 Experiment discussion

Our experiments evaluate the MFTE strategy in terms of performance and explanation. In the performance comparisons, most of the deep models equipped with MFTE perform better than the original models. Therefore, while MFTE provides explanations, it can also provide additional help for performance improvement. In terms of explanation evaluation, MFTE can explain for a single sample, which is more advantageous than the global explanation provided by the FI-based XGBoost. Moreover, another highlight of MFTE is that the first- and second-order explanations can be displayed separately, which makes the explanation effect clearer. Therefore, the experiment gave a positive answer to *RQ*1, meaning that MFTE can provide an intuitive explanation without reducing the performance of the model.

In the explanation representations, the visualized tree paths make the relationship of multiorder explanations clear, thereby maximizing the interpretable advantage of the trees. In contrast, although TEM can also record the paths of the trees by cross-features, the paths are fixed during the training of the deep model. This fixed method of cross-features cannot allow a single feature vector to be further trained, nor can it further take advantage of the trees' interpretation advantages. Moreover, MFTE is more personalized because each sample has its own feature selection path, which has advantages over the second-order explanation based on global feature interaction in SHAP and the fixed second-order explanation in TEM. Therefore, the experiments also gave a positive answer to *RQ*2 to confirm that MFTE can maximize the interpretable advantages of the trees.

Furthermore, if a deep model has higher-order feature interactions, MFTE can also make corresponding explanations. In contrast, although SHAP can also perform the first- and second-order explanations for samples, it has higher computational complexity for higher-order interpretations, so the scalability of MFTE is relatively better. The third highlight of MFTE is that we store multiorder explainable constraints in an explanation pool to allow MFTE to present different orders of explainable representation according to actual needs. Therefore, MFTE can show diversified representations for the feature contributions of the same order. Because we utilize the memory mechanism to store the features selected by different trees and achieve the consistency of features from selection to training, to explanation. The above three highlights answered *RQ*3 and confirmed that the explanation of MFTE is diverse.

In general, the experiments empirically verified the applicability of MFTE in different fields, thereby providing a practical approach for a prediction-oriented explanation.

6 Conclusion and future work

This work investigates a challenging problem in ML application – the black box problem. Our model mainly solves the problem of inconsistent feature representation between the tree model and the deep model and exploits the feature-tracking strategy to track features from the beginning of the tree to the training of the deep model and the explanation of the final result. It intuitively reflects the complex features interaction in the deep model. The experiments verify that our model is not only better than the previous work in performance but also provides more diversified explanations. In addition, we also prove that the feature-tracking strategy is applicable to linear or approximate linear separable deep models and suitable in different application fields.

In future work, we will further investigate our multiorder explanation framework. In particular, we would compare the linearly separable and the approximate linearly separable deep model and try to express their multiorder feature interactions in a unified way. Furthermore, we plan to design a more automated multiorder explanation, so that the prediction and explanations of the results can be more intuitively presented.

Funding information: The work was supported by the Higher Education Special Project of Guangdong Education Science Planning (2021GXJK241), the Guangdong Basic and Applied Basic Research Foundation (2023A1515011240, 2020A1515010531), and the National Natural Science Foundation of China (61902231).

Author contributions: Conceptualization: Lin Zheng; Methodology: Yixuan Lin; Formal analysis and investigation: Yixuan Lin, Lin Zheng; Writing – review and editing: Lin Zheng, Yixuan Lin; Funding acquisition: Lin Zheng.

Conflict of interest: The authors state that there is no conflict of interest.

Data availability statement: The used datasets of this research are available online and have proper citations within the article's contents.

References

[1] Zheng L, Zhu F, Huang S, Xie J. Context neighbor recommender: integrating contexts via neighbors for recommendations. Inform Sci. 2017;414(11):1–18. http://www.sciencedirect.com/science/article/pii/S0020025517307466.

[2] Zheng L, Guo N, Chen W, Yu J, Jiang D. Sentiment-guided sequential recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval; 2020. p. 1957–60.

[3] Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D. A survey of methods for explaining black box models. ACM computing surveys (CSUR). 2018;51(5):1–42.

[4] Lipton ZC. The Mythos of model interpretability: in machine learning, the concept of interpretability is both important and slippery. Queue. 2018;16(3):31–57.

[5] Molnar C. Interpretable machine learning. Lulu. com; 2020.

[6] Guidotti R, Monreale A, Matwin S, Pedreschi D. Black box explanation by learning image exemplars in the latent feature space. 2020. arXiv: http://arXiv.org/abs/arXiv:200203746.

[7] Bhatt U, Weller A, Moura JM. Evaluating and aggregating feature-based model explanations. 2020. arXiv: http://arXiv.org/abs/arXiv:200500631.

[8] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. In: Proceedings of the 31st international conference on neural information processing systems; 2017. p. 4768–77.

[9] Lundberg SM, Erion GG, Lee SI. Consistent individualized feature attribution for tree ensembles. 2018. arXiv: http://arXiv.org/abs/arXiv:180203888.

[10] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, et al. From local explanations to global understanding with explainable AI for trees. Nature Machine Intell. 2020;2(1):56–67.

[11] Sharma P, Mirzan SR, Bhandari A, Pimpley A, Eswaran A, Srinivasan S, et al. Evaluating tree explanation methods for anomaly reasoning: a case study of SHAP TreeExplainer and TreeInterpreter. In: International Conference on Conceptual Modeling. Springer; 2020. p. 35–45.

[12] Friedman JH. Greedy function approximation: a gradient boosting machine. Ann Stat. 2001;29(5):1189–232. http://www.jstor.org/stable/2699986.

[13] Chen T, Guestrin C. XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '16. New York, NY, USA: Association for Computing Machinery; 2016. p. 785–94. doi: 10.1145/2939672.2939785.

[14] Stojić A, Stanić N, Vuković G, Stanišić S, Perišić M, Šoštarić A, et al. Explainable extreme gradient boosting tree-based prediction of toluene, ethylbenzene and xylene wet deposition. Sci Total Environ. 2019;653:140–7.

[15] Fernández JAF. United States banking stability: An explanation through machine learning. Banks Bank Syst. 2020;15(4):137.

[16] He X, Pan J, Jin O, Xu T, Liu B, Xu T, et al. Practical lessons from predicting clicks on ads at Facebook. In: Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. ADKDD'14. New York, NY, USA: Association for Computing Machinery; 2014. p. 1–9. doi: 10.1145/2648584.2648589.

[17] Zheng L, Zhu F, Mohammed A. Attribute and global boosting: a rating prediction method in context-aware recommendation. Comput J. 2017;60(7):957–68. https://academic.oup.com/comjnl/article/60/7/957/2609377.

[18] Shih A, Choi A, Darwiche A. A symbolic approach to explaining Bayesian network classifiers. 2018. arXiv: http://arXiv.org/abs/arXiv:180503364.

[19] Flambeau JKF, Norbert T. Simplifying the explanation of deep neural networks with sufficient and necessary feature-sets: case of text classification. 2020. arXiv: http://arXiv.org/abs/arXiv:201003724.

[20] He X, Chua TS. Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval; 2017. p. 355–64.

[21] Xiao J, Ye H, He X, Zhang H, Wu F, Chua TS. Attentional factorization machines: Learning the weight of feature interactions via attention networks. 2017. arXiv: http://arXiv.org/abs/arXiv:170804617.

[22] Rendle S. Factorization machines with libfm. ACM Trans Intell Syst Technol (TIST). 2012;3(3):1–22.

[23] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences. In: International Conference on Machine Learning. PMLR; 2017. p. 3145–53.

[24] Zhang Y, Tinnno P, Leonardis A, Tang K. A survey on neural network interpretability. 2020. arXiv: http://arXiv.org/abs/arXiv:201214261.

[25] Zilke JR, Menciiia EL, Janssen F. Deepred-rule extraction from deep neural networks. In: International Conference on Discovery Science. Springer; 2016. p. 457–73.

[26] Wan A, Dunlap L, Ho D, Yin J, Lee S, Jin H, et al. NBDT: Neural-backed decision trees. 2020. arXiv: http://arXiv.org/abs/arXiv:200400221.

[27] Wang X, He X, Feng F, Nie L, Chua TS. TEM: Tree-enhanced embedding model for explainable recommendation. In: Proceedings of the 2018 Conference. WWW '18. Republic and Canton of Geneva, CHE: International Conferences Steering Committee; 2018. p. 1543–52. doi: 10.1145/3178876.3186066.

[28] Humbird KD, Peterson JL, McClarren RG. Deep neural network initialization with decision trees. IEEE Trans Neural Networks Learn Syst. 2018;30(5):1286–95.

[29] Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. Red Hook, NY, United States: Curran Associates Inc.; 2013. p. 3111–9.

[30] Bayer I, He X, Kanagal B, Rendle S. A Generic Coordinate Descent Framework for Learning from Implicit Feedback. In: Proceedings of the 26th International Conference on WWW '17. Republic and Canton of Geneva, CHE: International Conferences Steering Committee; 2017. p. 1341–50. doi: 10.1145/3038912.3052694.

[31] Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. Red Hook, NY, United States: Curran Associates Inc.; 2013. p. 926–34.

[32] Chen D, Socher R, Manning CD, Ng AY. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. 2013. arXiv: http://arXiv.org/abs/arXiv:13013618.

[33] Weston J, Chopra S, Bordes A. Memory networks. In: Proceedings of the International Conference on Learning Representations. ICLR '15; 2015.

[34] Sukhbaatar S, Szlam A, Weston J, Fergus R. End-to-end memory networks. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R, editors. Advances in neural information processing systems 28. NIPS '15. Cambridge, MA, United States: MIT Press; 2015. p. 2440–8.

[35] Blondel M, Fujino A, Ueda N, Ishihata M. Higher-order factorization machines. 2016. arXiv: http://arXiv.org/abs/arXiv:160707195.

[36] Bach S, Binder A, Montavon G, Klauschen F, Müller KR, Samek W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One. 2015;10(7):e0130140.

[37] Balduzzi D, McWilliams B, Butler-Yeoman T. Neural Taylor approximations: convergence and exploration in rectifier networks. In: International Conference on Machine Learning. PMLR; 2017. p. 351–60.

[38] Casalicchio G, Molnar C, Bischl B. Visualizing the feature importance for black box models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer; 2018. p. 655–70.

[39] Lafta SA, Ismael MQ. Trip generation modeling for a selected sector in Baghdad city using the artificial neural network. J Intell Sys. 2022;31(1):356–69.

[40] Kim DW, Shin GY, Han MM. Analysis of feature importance and interpretation for malware classification. Comput Materials Continua. 2020;65(3):1891–904.

[41] Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, et al. Explainable AI for trees: from local explanations to global understanding. 2019. arXiv: http://arXiv.org/abs/arXiv:190504610.

[42] Aswad FM, Kareem AN, Khudhur AM, Khalaf BA, Mostafa SA. Tree-based machine learning algorithms in the Internet of Things environment for multivariate flood status prediction. J Intell Syst. 2022;31(1):1–14.

[43] Yekun EA, Haile AT. Student performance prediction with optimum multilabel ensemble model. J Intell Syst. 2021;30(1):511–23.

[44] Izza Y, Ignatiev A, Marques-Silva J. On explaining decision trees. 2020. arXiv: http://arXiv.org/abs/arXiv:201011034.

[45] Arik SÖ, Pfister T. TabNet: attentive interpretable tabular learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35; 2021. p. 6679–87.

[46] Ho A, Reich J, Nesterko S, Seaton D, Mullaney T, Waldo J, et al. HarvardX and MITx: the first year of open online courses, fall 2012-summer 2013. Ho, AD, Reich, J, Nesterko, S, Seaton, DT, Mullaney, T, Waldo, J, & Chuang, I(2014) HarvardX and MITx: The first year of open online courses (HarvardX and MITx Working Paper No 1). 2014.

[47] HarvardX. HarvardX Person-Course Academic Year 2013 De-Identified dataset, version 3.0. 2014. doi: 10.7910/DVN/26147.

[48] Moro S, Cortez P, Rita P. A data-driven approach to predict the success of bank telemarketing. Decision Support Sys. 2014;62:22–31.