Contents lists available at ScienceDirect

Expert Systems With Applications

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

A model-agnostic, network theory-based framework for supporting XAI on classifiers

Gianluca Bonifazi [a], Francesco Cauteruccio [a], Enrico Corradini [a], Michele Marchetti [a], Giorgio Terracina [b], Domenico Ursino [a,*], Luca Virgili [a]

[a] DII, Polytechnic University of Marche, Italy
[b] DEMACS, University of Calabria, Italy

ARTICLE INFO

Keywords:
Explainable Artificial Intelligence
Model-agnostic XAI systems
Graph theory
Feature relevance
Feature dyscrasia
Sensitivity analysis

ABSTRACT

In recent years, the enormous development of Machine Learning, especially Deep Learning, has led to the widespread adoption of Artificial Intelligence (AI) systems in a large variety of contexts. Many of these systems provide excellent results but act as black-boxes. This can be accepted in various contexts, but there are others (e.g., medical ones) where a result returned by a system cannot be accepted without an explanation on how it was obtained. Explainable AI (XAI) is an area of AI well suited to explain the behavior of AI systems that act as black-boxes. In this paper, we propose a model-agnostic XAI framework to explain the behavior of classifiers. Our framework is based on network theory; thus, it is able to make use of the enormous amount of results that researchers in this area have discovered over time. Being network-based, our framework is completely different from the other model-agnostic XAI approaches. Furthermore, it is parameter-free and is able to handle heterogeneous features that may not even be independent of each other. Finally, it introduces the notion of dyscrasia that allows us to detect not only which features are important in a particular task but also how they interact with each other.

1. Introduction

The past decade has witnessed a widespread diffusion of Artificial Intelligence (AI, for short) and related research activities in various fields (Di Vaio, Palladino, Hassan, & Escobar, 2020; Jan et al., 2023; Kumar & Martin, 2023; Tunyasuvunakool et al., 2021; Ullah, Al-Turjman, Mostarda, & Gagliardi, 2020; Yu, Beam, & Kohane, 2018). Among the most driving areas of AI we have Machine Learning (ML, for short) and Deep Learning (DL, for short). In these areas, many of the proposed approaches are "black-box" ones (Dong, Wang, & Abbas, 2021; Pouyanfar et al., 2018), that is, they are able to solve, even egregiously, the problems for which they were designed but using internal mechanisms that are not transparent or easily interpretable (Henelius, Puolamäki, Boström, Asker, & Papapetrou, 2014; Moradi & Samwald, 2021). Classification is one of the most common problems involving highly accurate and precise black-box solutions. While such classifiers may be acceptable in many areas, there are just as many where the result of a classification cannot be accepted without understanding how it was obtained. Consider, just as an example, the classifiers used in the medical field to support physicians' diagnoses. This issue led to the emergence of the research field known as Explainable AI (XAI, for short) (Adadi & Berrada, 2018; Barredo Arrieta et al., 2020; Gunning & Aha, 2019; Yoo & Kang, 2021; Zini & Awad, 2022). Researchers in this area aim to study and design AI systems that can provide transparent and interpretable explanations for the decisions and actions of black-box subsystems or separate systems (Kaur, Uslu, Rittichier, & Durresi, 2022; Li et al., 2022).

With the explosion of DL, the number of black-box models has become impressive, and this has led to a corresponding increase in the extent of research on XAI. One of the most interesting directions of it deals with the study and development of "model-agnostic" XAI approaches. This term is used to denote all those XAI approaches that can be employed to interpret and explain the decisions of any black-box system, without an a priori knowledge of the type of model on which it is based. Model-agnostic systems are extremely general, and investing in them yields considerable returns because they can be applied to understand the behavior of very varied black-box systems.

* Corresponding author.
E-mail addresses: g.bonifazi@univpm.it (G. Bonifazi), f.cauteruccio@univpm.it (F. Cauteruccio), e.corradini@univpm.it (E. Corradini), m.marchetti@pm.univpm.it (M. Marchetti), terracina@mat.unical.it (G. Terracina), d.ursino@univpm.it (D. Ursino), l.virgili@pm.univpm.it (L. Virgili).

https://doi.org/10.1016/j.eswa.2023.122588
Received 23 January 2023; Received in revised form 13 November 2023; Accepted 13 November 2023
Available online 23 November 2023

On the other hand, model-agnostic approaches are also very difficult to design because they must have a high abstraction level with respect to the black-box models they want to interpret.

In this paper, we aim to provide a contribution in this setting by proposing a model-agnostic framework for classifier explainability. Our framework is based on network theory. It assumes to work on a black-box classifier model whose behavior is unknown. A set of instances, all characterized by the same set of features, is given as input to this classifier, which assigns a class to each instance. Our framework builds and maintains a fully connected network. The nodes in the network represent instances; the direction of the arc between two nodes is an indicator of the confidence level with which the classifier classified the corresponding instances. Starting from this network, our framework first computes the "dyscrasia" of each feature for all the instances. This measure is used to determine how "effective" a certain feature proves to be in discriminating instances. From the values of dyscrasia, and taking into account both the constructed network and the confidence information it stores, our framework computes the relevance of each feature during the classification task (Dabkowski & Gal, 2017; Fong & Vedaldi, 2017; Lundberg & Lee, 2017; Razmjoo, Xanthopoulos, & Zheng, 2017; Strumbelj & Kononenko, 2010). For this purpose, it uses a version of PageRank (Brin & Page, 1998) that we have customized to solve this problem. The knowledge of the features that contributed most to the classification of a set of instances provides us with valuable information about the black-box classifier. In fact, the extraction of such knowledge is recognized as one of the most interesting problems to be addressed in the field of XAI (Barredo Arrieta et al., 2020; Burkart & Huber, 2021; Lundberg & Lee, 2017; Ribeiro, Singh, & Guestrin, 2016; Štrumbelj, Kononenko, & Šikonja, 2009).

To complete the proposed framework, we introduce some parameters that allow a sensitivity analysis to be performed each time our framework is used in a given application context. This analysis is important because it helps to verify that, on the one hand, our framework is not sensitive to noise or outliers, and, on the other hand, it is able to intercept important and real changes in the characteristics of features and to adjust its evaluations accordingly.

As mentioned above, our approach is based on network theory (Newman, 2018). This choice is motivated by the fact that the network-based representation is extremely general and flexible. Moreover, network theory has been extensively studied in the past, both at a general level and in a variety of application contexts (Camacho, Panizo-LLedot, Bello-Orgaz, Gonzalez-Pardo, & Cambria, 2020; Gosak et al., 2018; Sporns, 2022). Consequently, it is possible to take advantage of all the results obtained in this research field and to adapt them to the objective and the scenario of this paper.

Our framework has several strengths. First, being network-based, it adopts a completely different way of proceeding from all other existing model-agnostic approaches in the literature (Nagahisarchoghaei et al., 2023). Therefore, it can contribute to the emergence of a new category of approaches in this area. Moreover, it is parameter-free in that it does not require the user to enter any input parameters. In addition, it is capable of operating not only on homogeneous features but also on heterogeneous ones. Still, it introduces the notion of dyscrasia, which allows the user to identify not only which features are important but also how they interact with each other. Again, unlike many other approaches, ours does not require features to be independent of each other. There are also several other strengths, as well as some weaknesses, that characterize our framework. Both of them will be discussed in detail in Section 3.7.

Summarizing, the main contributions of this paper are as follows:

- We propose a framework to support XAI on classifiers. Our framework makes use of network theory. It is also model-agnostic and thus can operate on multiple types of classifiers.
- We present a new measure, called dyscrasia, which indicates how consistent and capable of supporting classification a feature is.
- We exploit the new notion of dyscrasia and network theory to compute the relevance of each feature in the classifier that our framework is intended to explain. This relevance is computed first for each of the instances provided as input to the classifier and, next, for the set of instances taken as a whole.
- We present quantitative parameters to measure the sensitivity of the proposed approach, and thus its ability to withstand noise, while being flexible to capture real changes in features that might affect the behavior of the classifiers to be explained.

The outline of this paper is as follows: in Section 2, we provide an overview of related work. In Section 3, we describe our framework in detail and we highlight its computational complexity, strengths and weaknesses. In Section 4, we present three case examples, with the aim of helping to better understand how our framework works. In Section 5, we illustrate the experiments we performed to evaluate the goodness of our approach. Finally, in Section 6, we draw some conclusions and delineate some possible future developments.

2. Related work

In recent years, the topic of XAI attracted great interest among computer science researchers (Barredo Arrieta et al., 2020). As evidence of this, according to Semantic Scholar, the number of papers on this topic amounts to more than 4000 in the past five years (Semantic Scholar, 2022). Furthermore, if we just look at three recent surveys on XAI published in 2022 and 2023 (Banerjee & Barnwal, 2023; Chinu & Bansal, 2022; Nagahisarchoghaei et al., 2023), we can see that they collectively refer to nearly 500 papers. These surveys provide taxonomies on XAI approaches and present opportunities and challenges in this area. Among the challenges, they explicitly mention the improvement on XAI models to address the problem of black-box models. Our paper falls just in this track.

In its most general sense, XAI aims to define approaches capable of making machine learning models more explainable while maintaining high performances. Its ultimate goal is to enable users to understand, deepen and trust the AI systems that permeate all current life scenarios (Barredo Arrieta et al., 2020; Gunning & Aha, 2019). Several taxonomies to classify XAI approaches have been proposed in the literature. One of the most general of them was presented in Barredo Arrieta et al. (2020). It first divides AI approaches into two macro-categories, namely transparent models and models needing post-hoc explainability. A model is considered transparent if explanations of its behavior and results can be obtained through its direct observation. Examples of transparent models are decision trees and linear logistic regression. If the behavior and results of a learning model cannot be explained transparently, then it falls into the category of those needing post-hoc explainability. This term collectively denotes a set of very heterogeneous methods, each aiming to provide an explanation of how an existing machine learning approach (viewed as a black-box) behaves providing outputs from given inputs. Post-hoc explainability methods are divided into model-specific and model-agnostic. The first category comprises all those methods operating on specific machine learning models, for example models based on neural networks or Support Vector Machines. Instead, the second category includes those methods that can be applied on any machine learning model, regardless of the internal process or internal representation of data. This category can be further divided into subcategories; for example, we can consider explanation by simplification approaches (Ahern et al., 2019; Ribeiro et al., 2016), feature relevance explanation approaches (Henelius et al., 2014; Lundberg & Lee, 2017), and so on.

In the following, we focus on model-agnostic methods because our approach belongs to that category. In particular, we examine approaches that analyze the features of the underlying learning model through an alternative representation of them. This is because our approach adopts this way of proceeding. Furthermore, we use the terminology described in Burkart and Huber (2021) to distinguish between local and global explainability techniques for supervised models. Specifically, global techniques take into account the model, the feature and the set of all instances provided in input. In contrast, local techniques consider the model, the features and a single instance and provide information valid only for the behavior of the model on that specific instance.

Among model-agnostic methods that analyze features, the approach described in Razmjoo et al. (2017) exploits sensitivity analysis to perform a ranking of feature importance. In particular, it introduces a new definition of sensitivity specifically designed for this purpose. The sensitivity of a feature is based on the concept of redundancy. A feature is considered redundant if perturbing its value does not lead to a change in the ranking result. The approach is lightweight; therefore, it is used in the setting of online feature importance. Both the approach described in Razmjoo et al. (2017) and ours use the concept of sensitivity, although with very different meanings. Both of them provide feedback that allows the sensitivity to be updated whenever a new instance must be classified. However, the sensitivity update in Razmjoo et al. (2017) only takes into account the new instance and the values of features. Instead, our approach considers the changes made by the new instance on the whole network of the previous instances. As a consequence, it employs not only the values of features but also the relationships between instances.

In Lundberg and Lee (2017), the authors propose SHAP (SHapley Additive exPlanations), an approach that associates an importance value with a feature for a given prediction. SHAP is based on a cooperative game theory technique, in which features are represented as players cooperating to achieve the same goal. The approach of Lundberg and Lee (2017) and ours share a common goal, i.e., evaluating the importance of features. However, the ways in which they achieve that goal are completely different. In Henelius, Puolamäki, and Ukkonen (2017), the authors propose a model for interpreting black-box classifiers based on interactions among features. Specifically, the approach of Henelius et al. (2017) exploits the various types of interactions to define groups of attributes that are important for a given class. This approach can be considered orthogonal to our own. In fact, the latter is based on associations between instances and does not consider associations between features. Instead, the former considers associations between features and does not consider associations between instances.

In Štrumbelj et al. (2009), the authors propose an approach to analyze subsets of features. For each instance, this approach aims to explain the decisions made by the underlying machine learning model. For this purpose, it suitably aggregates the contributions of features. The approach of Štrumbelj et al. (2009) and ours have the same general goal, which is the explainability of the underlying learning model, albeit their way of proceeding is completely different. In addition, the approach of Štrumbelj et al. (2009) also aims to analyze the interactions between subsets of features. Consequently, it has an exponential computation time against the number of features. Our approach analyzes only individual features and does not consider subsets of features and their interactions. In the face of this limitation, it has a polynomial computation time against the number of features. In Wei, Zhao, Feng, He, and Yu (2020), the authors propose DFIFS (Dynamic Feature Importance-based Feature Selection) whose main goal is feature selection. DFIFS uses a dynamic index, called DFI (Dynamic Feature Importance), to evaluate the importance of a feature. This index has two facets. The first concerns feature importance, which is evaluated using the Gini index. The second focuses on feature redundancy, which is evaluated through the Maximum Information Coefficient. The approach of Wei et al. (2020) and ours share the idea of constructing a global index based on multiple facets. However, the overall goal and the characteristics of facets and indexes are completely different in the two approaches.

In Ribeiro et al. (2016), the authors propose LIME (Local Interpretable Model-agnostic Explanations), a local explainability technique that aims to provide interpretations for a classifier's predictions. Its strategy is to linearly approximate the classifier for a certain prediction. To this end, it modifies the input of the model locally and evaluates the effects this modification has on the output. The model is mainly applied on input data whose representation is human-interpretable, such as images or bag-of-word models. LIME aims to provide an explanation by returning the so-called evidences, that is, relationships between features (e.g., relationships between words in a text or between patches in an image) and the model's prediction. LIME's output is a list of explanations, i.e., an enumeration of features and the importance each of them had in the prediction. LIME and our approach share the same goal, which is to determine the most important features. However, the methods they use to achieve their goal are completely different. Furthermore, our approach is data-independent, meaning that it can be applied on any input dataset, regardless its internal data representation. Instead, to produce very good results, LIME needs a human-interpretable representation of data. The authors of Ahern et al. (2019) extend LIME by proposing NormLIME, which aggregates local interpretations returned by LIME to obtain a global importance parameter relative to all features used in the model. There is an important similarity between our approach and NormLIME in that both operate in two stages, first calculating local importance values and then using the latter to obtain a global importance value. However, the two approaches use very different methods to achieve the same goal. These methods are orthogonal to each other and could be integrated into a single approach in the future.

In Ucer, Ozyer, and Alhajj (2022), the authors propose a network-based classifier called GSNAc (Generalized Social Network Analysis-based Classifier). GSNAc uses network analysis techniques to perform its task; in this respect, it is similar to our approach. It represents the input dataset by means of a network whose nodes represent instances and whose arcs denote the similarities among nodes. Classification is done by analyzing network arcs. GSNAc does not deal precisely with explainability. However, the visualization through different network layouts allowed by it provides a first possibility to interpret the results in a user-friendly way. GSNAc and our approach, while sharing the use of network analysis as a mean to achieve their goals, have different purposes; in fact, our approach is focused on explainability, while GSNAc is essentially a classifier. It could be used as the learning method on which to make our approach operate.

In Ienco, Meo, and Botta (2008), the authors propose a document categorization approach based on PageRank. Specifically, in this approach, PageRank is used, along with a random walk, to determine the ranking of features that best represent documents. PageRank is employed to sort the features in the dataset. In particular, the PageRank of a feature indicates the probability to find that feature along with the other ones in the dataset. The approach of Ienco et al. (2008) and ours share the use of network analysis and PageRank. However, in Ienco et al. (2008) the latter is used for feature selection, while in our case it is employed as an intermediate measure in the context of feature relevance computation. Also, our approach is model-agnostic while the approach of Ienco et al. (2008) is specific for documents. In Akhiat, Asnaoui, Chahhou, and Zinedine (2021), the authors propose a feature selection method using a network-based data representation. In the network exploited by this approach, nodes represent features while arcs denote relationships between features. The network is fully connected. Each arc has a weight obtained by computing the AUC score after applying a decision tree on the dataset. The ultimate goal of the approach of Akhiat et al. (2021) is the identification of communities formed by important nodes. For this reason, it can be considered more of a data pre-processing approach than an explainability one. The approach of Akhiat et al. (2021) and ours share the use of network computations to achieve their goals. However, these computations are very different in the two cases.

In Roffo, Melzi, Castellani, and Vinciarelli (2017), the authors propose a network-based algorithm for feature selection. This algorithm uses a network whose nodes represent features; the presence of an arc indicates the probability that both features associated with the corresponding nodes are relevant. The weights of the features are computed through a Probabilistic Latent Semantic Analysis approach, which models the probabilities of feature co-occurrences as a multinomial distribution. Finally, the approach uses a methodology called Infinite Feature Selection, which considers all paths between nodes to identify and quantify redundancies between features. Both the approach of Roffo et al. (2017) and ours are network-based. However, the way they operate is very different. In fact, the former uses the network to analyze subsets of features based on the cost of the paths connecting them; then, it selects features based on the importance of each of them relative to all the others. Instead, our approach does not compare features with each other, but computes the relevance of each feature with the goal of performing the explainability of the underlying learning model.

Differently from most of the model-agnostic and network-based approaches illustrated above that aim at feature selection, our approach is conceived for the computation of feature relevance. Although these two activities show similarities, they have important differences. In fact, feature selection is generally performed before classification for determining the features to be exploited by the latter. It is rarely carried out after this process, as support for classifier explainability. The computation of feature relevance can be performed before classification to support feature selection (albeit there are some feature selection methods that do not calculate the relevance of the features involved). Furthermore, it becomes extremely valuable after classification to support classifier explainability. It is precisely with the latter perspective that it is employed within the framework proposed in this paper.

We conclude this literature review by taking a look at how our framework relates to approaches belonging to technology areas related to, but different from, AI. In particular, we want to consider the landscape of industrial applications. This is a rapidly evolving field that is deeply intertwined with real world physical systems. In this area, multiple dynamic models are emerging alongside traditional dynamic models under the switched system paradigm. Here, our approach can make a valuable contribution that is orthogonal to those provided by past approaches. Indeed, while previous research approaches (e.g., Song, Song, Stojanovic and Song, 2023; Song, Sun, Song and Stojanovic, 2023; Sun, Song, Song, & Stojanovic, 2022) where concerned with addressing the complexity of control systems, our framework can be used to address a different goal, namely the interpretability and transparency of these advanced systems. As these become increasingly complex, the ability to understand, interpret and trust them becomes paramount. Our framework, being specialized in just such capabilities (although they are applied to AI systems in this paper), can become an excellent support for managing the interpretability and transparency of advanced industrial systems.

3. A network-based framework for classifier explainability

In this section, we illustrate our proposed model and framework for the explainability of classifiers. Fig. 1 shows a visual representation of the workflow of our framework.

Let $\mathcal{I} = \{I_1, I_2, \dots, I_l\}$ be a set of instances to be classified and let $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$ be the set of possible classes. Let $\mathcal{F} = \{F_1, F_2, \dots, F_n\}$ be the set of features characterizing the instances of $\mathcal{I}$. Accordingly, given an instance $I_i \in \mathcal{I}$, it can be represented by the set $\mathcal{F}_i = \{F_{1_i}, F_{2_i}, \dots, F_{n_i}\}$ of the values of its features. In particular, $F_{k_i} \in \mathcal{F}_i$ indicates the value of the feature F_k in the instance I_i. Our framework assumes that each feature F_k can be numeric, categorical or textual.

3.1. A network-based model for representing the classification of a set of instances

Suppose we have a classification model $\mathcal{M}$ and that $\mathcal{M}$ has been already trained. Let $\mathcal{I}$ be the set of instances to classify and let $\mathcal{C}$ be the set of possible classes. For each instance $I_i \in \mathcal{I}$, $\mathcal{M}$ assigns a class of $\mathcal{C}$ to it with a confidence level c_i.[1] The latter is a value in the real interval $[0, 1]$; the higher it is, the more confident $\mathcal{M}$ is in classifying I_i.

The behavior of $\mathcal{M}$ in classifying the instances of $\mathcal{I}$ can be represented by a network $\mathcal{N}$, whose nodes denote the instances of $\mathcal{I}$ and whose arcs are indicators of the confidence level with which $\mathcal{M}$ classified the instances associated with the corresponding nodes. More formally, we represent $\mathcal{N}$ as:

$$\mathcal{N} = \langle N, A \rangle \tag{3.1}$$

Here, N is the set of nodes of $\mathcal{N}$. There is a node $n_i \in N$ for each instance $I_i \in \mathcal{I}$. Since there is a biunivocal correspondence between the nodes of $\mathcal{N}$ and the instances of $\mathcal{I}$, in the following we will use the terms "node" and "instance", as well as the symbols n_i and I_i, interchangeably. It is possible to define a function $\gamma(\cdot)$, which receives a node n_i and returns the confidence c_i with which $\mathcal{M}$ classified I_i.

A is the set of arcs of $\mathcal{N}$. There is an arc of A for each pair of nodes (n_i, n_h) of $\mathcal{N}$. The arc is directed from n_i to n_h if $c_i < c_h$; conversely, if $c_h < c_i$, it is directed from n_h to n_i. Finally, if $c_i = c_h$, its direction is set randomly.

3.2. Assessing the dyscrasia of the occurrences of a feature during classification

In this section, we define an approach to quantitatively assess the dyscrasia (i.e., the "dysfunction", lack of coordination) $\delta(F_{k_i}, F_{k_h})$ between the values F_{k_i} and F_{k_h} of the feature F_k for the instances I_i and I_h. This assessment, which takes into account a single feature, is a first step in the overall reasoning that includes all features and that we will consider below. However, it is essential for understanding the next steps and, therefore, we decided to describe it in detail in this section.

Preliminarily, it is necessary to clarify the reference context and the phenomenon we want to capture through the definition of $\delta(F_{k_i}, F_{k_h})$. As for the first aspect, suppose that the confidence c_i with which $\mathcal{M}$ classified I_i is smaller than the confidence c_h with which $\mathcal{M}$ classified I_h. This implies the existence of an arc from n_i to n_h (see Section 3.1). Regarding the second aspect, we point out that the concept of dyscrasia aims to capture any "disharmony" in the role that the two occurrences F_{k_i} and F_{k_h} of the same feature F_k played during the classification of I_i and I_h carried out by $\mathcal{M}$.

The reasoning we make to capture such a disharmony is as follows:

- If $\mathcal{M}$ classified I_i and I_h in the same class, $\delta(F_{k_i}, F_{k_h})$ is the greater the more F_{k_i} and F_{k_h} have dissimilar values and, at the same time, the confidences c_i and c_h with which $\mathcal{M}$ classified I_i and I_h are low (which indicates that the possibility that $\mathcal{M}$ made a classification error is significant).
- If $\mathcal{M}$ classified I_i and I_h into different classes, $\delta(F_{k_i}, F_{k_h})$ is the greater the more F_{k_i} and F_{k_h} have similar values, the confidence c_h is high and the confidence c_i is low (which implies that the possibility that $\mathcal{M}$ classified I_h correctly while classifying I_i incorrectly, is relevant).

[1] Our classifier model assumes that each instance can be assigned to exactly one class.

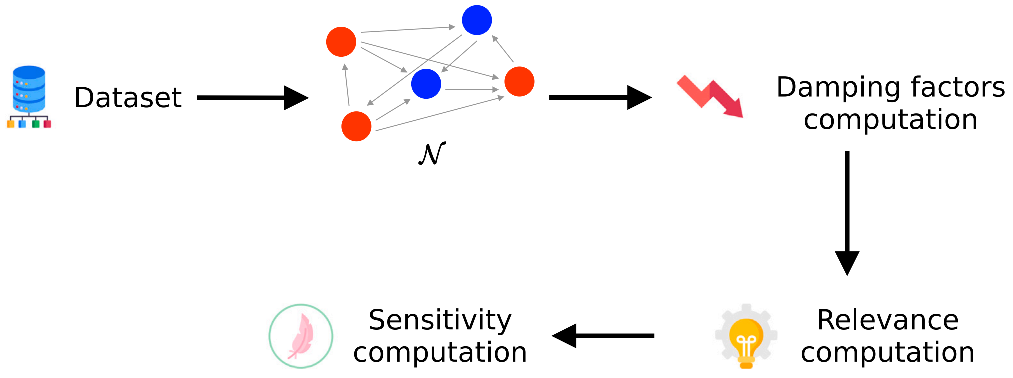

Fig. 1. Workflow of our framework.

To better understand the concept of dyscrasia, let us consider a real-life example from a dataset we used in the experiments (see Section 5.4.3). Specifically, we consider the ratings of a restaurant in Yelp posted by customers who have frequented it. The feature “age of customer” is an example of a feature that may exhibit dyscrasia. Indeed, there may be customers with very different ages who give the same rate to the restaurant and, conversely, customers with the same age who give very different rates to it. In contrast, the feature “gluten intolerance by customer” is an example of a feature that does not exhibit dyscrasia. In fact, if the restaurant does not provide food for this type of intolerance, it will presumably receive a negative rating from all gluten-intolerant customers.

From a formal point of view, the dyscrasia $\delta(F_{k_i}, F_{k_h})$ can be defined as:

$$\delta(F_{k_i}, F_{k_h}) = \begin{cases} \varepsilon(n_i) \cdot \varepsilon(n_h) \cdot \lambda(F_{k_i}, F_{k_h}) & \text{if } \mathcal{M} \text{ assigned } I_i \text{ and } I_h \\ & \text{to the same class} \\ \varepsilon(n_i) \cdot \gamma(n_h) \cdot [1 - \lambda(F_{k_i}, F_{k_h})] & \text{otherwise} \end{cases} \tag{3.2}$$

Here:

- $\lambda(\cdot, \cdot)$ is a function that receives the values F_{k_i} and F_{k_h} assumed by the feature F_k for I_i and I_h and returns a value in the real interval $[0, 1]$ indicating the dissimilarity degree between F_{k_i} and F_{k_h}. Clearly, $\lambda(\cdot, \cdot)$ depends on the type of F_k. For example, in the case F_k is numerical, it could return the (suitably normalized) difference between the two values. Instead, in the case F_k is textual, it could return the (suitably normalized) string dissimilarity value.
- $\gamma(\cdot)$ returns the confidence of $\mathcal{M}$ in classifying the instance received in input. It has been defined in Section 3.1.
- $\varepsilon(\cdot)$ returns the possible error of $\mathcal{M}$ in classifying the instance received in input. It is defined as: $\varepsilon(n_i) = 1 - \gamma(n_i)$.

Note that, in the definition of $\delta(F_{k_i}, F_{k_h})$, the factor $\varepsilon(n_i)$ is present both in the case where $\mathcal{M}$ has assigned I_i and I_h to the same class and in the opposite case. This is because we are considering directed arcs, in this case from n_i to n_h. As mentioned above, the presence of an arc from n_i to n_h implies that the confidence $\gamma(n_i)$ with which $\mathcal{M}$ classified I_i is less than the confidence $\gamma(n_h)$ with which $\mathcal{M}$ classified I_h. Consequently, we assume that, in the presence of high dyscrasia, the greater responsibility is due to the classification of I_h, on which $\mathcal{M}$ had shown greater confidence than I_i. This assumption brings us to the formula of dyscrasia seen above, in which $\varepsilon(n_i)$ is always present. In fact: *(i)* if the dyscrasia $\delta(F_{k_i}, F_{k_h})$ is high and the dissimilarity $\lambda(F_{k_i}, F_{k_h})$ is high, then $\mathcal{M}$ has misclassified I_h in the same class as I_i; consequently, $\varepsilon(n_i) = 1 - \gamma(n_i)$ and $\varepsilon(n_h) = 1 - \gamma(n_h)$ come into play in the formula; *(ii)* if the dyscrasia $\delta(F_{k_i}, F_{k_h})$ is high and the dissimilarity $\lambda(F_{k_i}, F_{k_h})$ is low, then $\mathcal{M}$ has misclassified I_h into a different class from the one of I_i, and this depends on the high confidence $\gamma(n_h)$ it had in making that classification; consequently, $\varepsilon(n_i)$ and $\gamma(n_h)$ come into play in the formula. The presence of $\varepsilon(n_i)$ is meant to account for any error that $\mathcal{M}$ made in the classification of I_i, which, albeit less decisive than the classification of I_h, still takes on some weight.

3.3. Assessing the relevance of a feature during classification

As we mentioned in the previous section, the dyscrasia between the occurrences of a feature is a preparatory step for the core task of our approach. This regards the computation of the relevance of a feature during a classification process carried out by a (possibly) black-box classifier. In this section, we describe this task in detail.

Preliminarily, recall that, given two nodes n_i and n_h, an arc from n_i to n_h indicates that the confidence with which $\mathcal{M}$ classified I_i is less than or equal to the confidence with which $\mathcal{M}$ classified I_h. As a consequence, given a node $n_i \in N$, its incoming arcs start from nodes whose associated instances were classified with lower or equal confidence. Conversely, its outgoing arcs end in nodes whose corresponding instances were classified with higher or equal confidence. The two sets of nodes introduced above can be defined as follows:

$$\begin{aligned} N_i^{out} &= \left\{ n_h | n_h \in N, n_h \neq n_i, (n_i, n_h) \in A \right\} \\ N_i^{in} &= \left\{ n_h | n_h \in N, n_h \neq n_i, (n_h, n_i) \in A \right\} \end{aligned} \tag{3.3}$$

N_i^{out} (resp., N_i^{in}) is thus the set of nodes connected to n_i via an outgoing (resp., incoming) arc. All these nodes have a confidence higher (resp., lower) than or equal to the confidence c_i with which $\mathcal{M}$ classified I_i.

Let us now consider the feature F_k whose relevance during the classification process we want to assess. Clearly, F_k takes on a specific value F_{k_i} for each instance $I_i \in \mathcal{I}$. Therefore, in order to assess the relevance of F_k during the classification process, we must first assess the relevance of F_{k_i}.

From the above modeling, we can observe that the node n_i corresponding to I_i is connected through its outgoing arcs to the nodes of N_i^{out}, each of which has a confidence higher than or equal to c_i. On the other hand, it is connected through its incoming arcs to the nodes of N_i^{in}, each of which has a confidence lower than or equal to c_i. Therefore, in determining the role of F_k in the classification task, n_i can act as a “guide” for the nodes of N_i^{in}, while it should be “guided” by the nodes of N_i^{out}. One way to formalize this reasoning consists of adapting the PageRank centrality (Brin & Page, 1998) to this context.

Recall that, given a network $\overline{\mathcal{N}} = \langle \overline{N}, \overline{A} \rangle$, and given a node $\overline{n_i} \in \overline{N}$, the PageRank centrality $\varrho(\overline{n_i})$ of $\overline{n_i}$ is defined as:

$$\varrho(\overline{n_i}) = \frac{1-d}{|\overline{N}|} + d \cdot \left(\sum_{\overline{n_h} \in \overline{N_i^{in}}} \frac{\varrho(\overline{n_h})}{|\overline{N_h^{out}}|} \right) \tag{3.4}$$

Observe that this formula is recursive. In it:

- $|\overline{N}|$ is the cardinality of $\overline{N}$.
- $\overline{N_i^{in}}$ is the set of nodes connected to the arcs incoming in $\overline{n_i}$.
- $\overline{N_h^{out}}$ is the set of nodes connected to the arcs outgoing from $\overline{n_h}$.
- d is called "damping factor" and is used to weigh the contribution that the nodes associated with the arcs incoming to $\overline{n_i}$, and their PageRank centralities, provide in determining the PageRank centrality of $\overline{n_i}$. The other component of the formula of $\varrho(\overline{n_i})$ is fixed and is equal to $\frac{1-d}{|\overline{N}|}$. In the original PageRank formula, d is fixed and is equal to 0.85.

By adapting the general formula of the PageRank centrality seen above to our reference context, we have that the relevance $\rho(F_{k_i})$ of the occurrence F_{k_i} of F_k corresponding to the instance I_i, is given by:

$$\rho(F_{k_i}) = \frac{1-d_{k_i}}{|N|} + d_{k_i} \cdot \left(\sum_{n_h \in N_i^{in}} \frac{\rho(F_{k_h})}{|N_h^{out}|} \right) \tag{3.5}$$

Here, the relevance of F_{k_i} depends on two components. The first is fixed and depends on the number of nodes in the network. The second is variable and depends on the relevance of the feature occurrences related to the starting nodes of the arcs incoming into n_i. The relevance of the feature occurrence F_{k_h} of the node n_h is weighted with respect to the number of arcs outgoing from n_h. In fact, the greater the number of these arcs, the lower the weight of the relevance of F_{k_h}. This is reasonable because the number of arcs outgoing from n_h indicates the number of nodes having a higher confidence than n_h.

The damping factor d_{k_i} in Eq. (3.5) does not have a constant value, as was the case in the original definition of PageRank centrality. Instead, its value varies for each node $n_i \in N$ and depends on the characteristics of n_i. In particular, it depends on the number of nodes outgoing from it, as well as on the dyscrasia between the feature occurrence of each of these nodes and the feature occurrence F_{k_i} of F_k for n_i. More specifically, d_{k_i} can be defined as follows:

$$d_{k_i} = \sigma \left(\frac{\sum_{n_h \in N_i^{out}} \delta(F_{k_i}, F_{k_h})}{|N_i^{out}|} \right) \tag{3.6}$$

The rationale for this formula is as follows: the value of d_{k_i} depends on the magnitude of the dyscrasia between the occurrence of F_k for n_i and the occurrence of F_k for all the ending nodes of the arcs outgoing from n_i, which therefore have a greater confidence than n_i. The definition of damping factor was designed to create a positive correlation between the values of this parameter and those of dyscrasia. Therefore, d_{k_i} assumes high values when the dyscrasia is high. Let us now consider Eq. (3.5): if d_{k_i} is high, the weight of the first term of the formula tends to be very low. The second term depends strongly on the number of arcs incoming to n_i. If that number is low (which happens if $\mathcal{M}$ has not expressed high confidence in the classification of n_i) then the relevance of F_{k_i} will be low. This is right because $\mathcal{M}$ did not express a high confidence on the classification of n_i and, at the same time, F_{k_i} showed a high dyscrasia with feature occurrences of nodes having a higher confidence than n_i.

The function $\sigma(\cdot)$ present in the formula is the sigmoid one. Recall that this function ranges from 0 to 1 when the value of its argument varies from $-\infty$ to $+\infty$. In particular, if the argument can only be non-negative (as in our case), $\sigma(\cdot)$ ranges from 0.5 to 1. The use of the sigmoid function is motivated by the fact that the fraction within it in Eq. (3.6) can tend quickly to 0. In this way, any differences in the number of outgoing arcs and in the dyscrasia would have little impact in determining the value of d_{k_i}. Instead, the sigmoid function tends to amplify the differences in the output when the ones in the input are close to 0, and thus avoids the problem highlighted above. The fact that the value returned by the sigmoid function in our case is between 0.5 and 1 has another positive consequence. Indeed, this prevents the damping factor from being close to 0 for most of the $|N|$ nodes. If this were to happen, the weight of the second term in Eq. (3.5) would tend to 0 and, therefore, all feature occurrences would tend to assume the same value, which would be close to $\frac{1}{|N|}$. This would be a negative aspect since it would nullify the differences between the relevances of the various feature occurrences.

Having defined the relevance of a single feature occurrence F_{k_i}, we are now able to define the relevance of a feature F_k. In fact, it can be obtained by computing the average of the relevances of all its occurrences. Formally speaking:

$$\rho(F_k) = \frac{\sum_{n_i \in N} \rho(F_{k_i})}{|N|} \tag{3.7}$$

Finally, we can define a function $\alpha(\cdot)$ that receives a classifier $\mathcal{M}$ and returns a value in the real interval $[0, 100]$ indicating the ability of $\mathcal{M}$ to differentiate the feature relevance. $\alpha(\cdot)$ is defined as follows:

$$\alpha(\mathcal{M}) = \frac{max_{\mathcal{M}} - min_{\mathcal{M}}}{MaxCPI_{\mathcal{M}}} \cdot 100 \tag{3.8}$$

Here, $max_{\mathcal{M}}$ (resp., $min_{\mathcal{M}}$) is the maximum (resp., minimum) value taken by the median relevance of a feature when $\mathcal{M}$ is adopted. $MaxCPI_{\mathcal{M}}$ (Maximum Central Percentile Interval) is obtained in the following way: Given the classifier $\mathcal{M}$, first the width of the interval between the values of the relevances located between the 25th and 75th percentiles is computed for each feature. Then, the maximum value of the widths thus constructed is determined. We decided to consider these percentiles because, if we had taken the full interval of relevance values, $\alpha(\cdot)$ would have been sensitive to outliers.

3.4. Measuring the sensitivity of the proposed approach

In the previous section, we presented our approach for determining feature relevance. It is the core of this paper. In this section, we propose a method to measure its sensitivity when new instances are included in the analysis.

The study of sensitivity is crucial in order to be able to monitor whether, how, and to what extent our approach is able to adapt to changes. Ideally, an approach should be stable enough to be unaffected by small changes or outliers that represent only "noise", while it should be flexible enough to adapt to significant changes. In this section, we want to define a quantitative method to measure the sensitivity of our approach. In our case, changes are defined by the number of new instances to be classified and the values of their features. Thus, the presence of a small number of new instances should not substantially affect the behavior of our approach. Significant changes occur when the number of new instances starts to be high and the values of one or more of their features always differ from past values along the same direction. Having made this premise, which allowed us to define the issue to address, let us see how our sensitivity analysis approach works.

Suppose we have a set $\mathcal{I}$ of instances to be classified by the classifier under consideration and a set C of possible classes. Assume also that, at the end of the classification process, we obtained a network $\mathcal{N}$ associated with that process (see Section 3.1). Also, assume that we computed both the relevance $\rho(F_{k_i})$ of each occurrence F_{k_i} that F_k has in correspondence with a given instance I_i and the relevance $\rho(F_k)$ of each feature F_k of the set $\mathcal{F}$ of features associated with $\mathcal{I}$.

Suppose now that we have a new instance I_j to be classified and added to the set $\mathcal{I}$. At the end of this classification process and the application of our approach, we have a new network:

$$\hat{\mathcal{N}} = \langle \hat{N}, \hat{A} \rangle \tag{3.9}$$

In this case, $\hat{N} = N \cup \{n_j\}$ and $\hat{A} = A \cup \hat{A}_j^{in} \cup \hat{A}_j^{out}$. In other words, the set of nodes of $\hat{\mathcal{N}}$ is obtained from the set of nodes of $\mathcal{N}$ by adding the node n_j associated with the instance I_j to it. The set of arcs $\hat{A}$ of $\hat{\mathcal{N}}$ is obtained through the union of the set A of the arcs of $\mathcal{N}$, the set $\hat{A}_j^{in}$ of the arcs incoming into n_j, and the set $\hat{A}_j^{out}$ of the arcs outgoing from n_j.

At this point, given the feature occurrence F_{k_j}, associated with F_k and n_j, we define its estimated relevance $\overline{\rho}(F_{k_j})$ as:

$$\overline{\rho}(F_{k_j}) = \frac{1 - d_{k_j}}{|\hat{N}|} + d_{k_j} \cdot \left(\sum_{n_h \in \hat{N}_j^{in}} \frac{\tilde{\rho}(F_{k_h})}{|\hat{N}_h^{out}|} \right) \tag{3.10}$$

We call $\overline{\rho}(F_{k_j})$ "estimated relevance" because, in the second component of the above formula, we used $\tilde{\rho}(F_{k_h})$, which is the value of the relevance that the feature occurrence F_{k_h}, $n_h \in N_j^{in}$, had before the new instance I_j was added. In other words, in computing the estimated relevance $\overline{\rho}(F_{k_j})$, we do not recompute the relevance of all feature occurrences F_{k_h} (as in principle we should do), but rely on their preexisting values.

Instead, the exact value $\rho(F_{k_j})$ of the relevance of F_{k_j} is obtained by applying the formula represented in Eq. (3.5) to the network $\hat{\mathcal{N}}$. In this case, we have:

$$\rho(F_{k_j}) = \frac{1 - d_{k_j}}{|\hat{N}|} + d_{k_j} \cdot \left(\sum_{n_h \in \hat{N}_j^{in}} \frac{\rho(F_{k_h})}{|\hat{N}_h^{out}|} \right) \tag{3.11}$$

At this point, we can define the function $\Delta(F_{k_j})$ that calculates the difference between the actual and estimated values of the relevance of F_{k_j}. Specifically:

$$\begin{aligned} \Delta(F_{k_j}) &= |\rho(F_{k_j}) - \overline{\rho}(F_{k_j})| \\ &= \left| \frac{1 - d_{k_j}}{|\hat{N}|} + d_{k_j} \cdot \left(\sum_{n_h \in \hat{N}_j^{in}} \frac{\rho(F_{k_h})}{|\hat{N}_h^{out}|} \right) - \frac{1 - d_{k_j}}{|\hat{N}|} \right. \\ &\qquad \left. + d_{k_j} \cdot \left(\sum_{n_h \in \hat{N}_j^{in}} \frac{\tilde{\rho}(F_{k_h})}{|\hat{N}_h^{out}|} \right) \right| \\ &= d_{k_j} \cdot \left| \sum_{n_h \in \hat{N}_j^{in}} \frac{\rho(F_{k_h})}{|\hat{N}_h^{out}|} - \sum_{n_h \in \hat{N}_j^{in}} \frac{\tilde{\rho}(F_{k_h})}{|\hat{N}_h^{out}|} \right| \\ &= d_{k_j} \cdot \sum_{n_h \in \hat{N}_j^{in}} \frac{\left| \rho(F_{k_h}) - \tilde{\rho}(F_{k_h}) \right|}{|\hat{N}_h^{out}|} \end{aligned} \tag{3.12}$$

The function $\Delta(F_{k_j})$ returns a numerical indicator of the variation of the relevance of a feature occurrence F_{k_j} caused by a new instance I_j, when it is added to $\mathcal{I}$ and classified by the classifier underlying our framework.

In a similar way, it is possible to proceed with the other occurrences of F_k; in fact, these will also have changed due to the inclusion of I_j in $\mathcal{I}$ and the consequent transition from $\mathcal{N}$ to $\hat{\mathcal{N}}$. In this way, we obtain a value of $\Delta(F_{k_i})$ for each node $n_i \in \hat{N}$.

Starting from these values, it is possible to define the function $\hat{\Delta}(F_k, I_j)$, which returns the overall variation in the relevance of F_k caused by the new instance I_j:

$$\hat{\Delta}(F_k, I_j) = \frac{\sum_{n_i \in \hat{N}} \Delta(F_{k_i})}{|\hat{N}|} \tag{3.13}$$

In other words, the variation is given by the average of the variations of all feature occurrences that resulted from the insertion of I_j into $\mathcal{I}$. It quantitatively expresses how the relevance of a feature has varied due to the perturbation caused by the insertion of I_j into $\mathcal{I}$ and its next classification performed by the classifier underlying our framework.

Finally, we can define the function $\tilde{\Delta}(F_k, I_j)$ that returns the relative variation in the relevance of F_k caused by the new instance I_j:

$$\tilde{\Delta}(F_k, I_j) = \frac{\sum_{n_i \in \hat{N}} \Delta(F_{k_i})}{\sum_{n_i \in \hat{N}} \rho(F_{k_i})} = \frac{\sum_{n_i \in \hat{N}} |\rho(F_{k_i}) - \overline{\rho}(F_{k_i})|}{\sum_{n_i \in \hat{N}} \rho(F_{k_i})} \tag{3.14}$$

Similarly, we can define the function $\tilde{\Delta}(F_k, IS_j)$, which computes the relative variation in the relevance of F_k caused by a new set of instances IS_j instead of a single instance I_j.

3.5. Analysis of the computational complexity of the proposed framework

Having introduced our framework, in this section we want to evaluate its computational complexity. In performing this task, we consider both time and space complexity. To this end, we define the complexity of applying our framework on a general dataset considering all the tasks it involves. Specifically, we first describe the complexity of creating the network-based model defined in Section 3.1. Then, we evaluate the complexity of assessing the discordance of feature occurrences during classification, as discussed in Section 3.2. After that, we analyze the complexity of assessing the relevance of a feature during classification, as defined in Section 3.3. Finally, we focus on the complexity of measuring the sensitivity of the approach, as illustrated in Section 3.4. We make these assessments first for what concerns time complexity and then for what regards space complexity.

3.5.1. Analysis of the time complexity

We start by analyzing time complexity. Assume we have $|\mathcal{I}| = l$ instances with $|\mathcal{F}| = n$ features characterizing them. In most practical cases, we can assume that $n \ll l$. Furthermore, we assume that the result of the classification task carried out by the model $\mathcal{M}$ is available and preprocessed. Without loss of generality, we assume that we employ two lookup tables (Cormen, Leiserson, Rivest, & Stein, 2001) associating each instance $I_i \in I$ with the confidence level c_i and the set of features $\mathcal{F}_i$, respectively.

Given an instance $I_i \in \mathcal{I}$, accessing its confidence value c_i (resp., the value of its kth feature $F_{k_i} \in \mathcal{F}_i$, $1 \leq k \leq n$) has time complexity $\mathcal{O}(1)$.

Let us consider now the building of the network $\mathcal{N}$. This network has $\mathcal{O}(l)$ nodes and $\mathcal{O}(l \cdot (l-1)) = \mathcal{O}(l^2)$ edges. By representing the network through a classical adjacency matrix, the time complexity of its construction is $\mathcal{O}(l^2)$. Note that, thanks to the lookup tables, deciding the direction of an arc in $\mathcal{N}$ has $\mathcal{O}(1)$ time complexity.

Let us now focus on the analysis of the time complexity of assessing the dyscrasia $\delta(F_{k_i}, F_{k_h})$. Since the time complexity of $\epsilon(\cdot)$ and $\gamma(\cdot)$ is $\mathcal{O}(1)$, the time complexity of $\delta(F_{k_i}, F_{k_h})$ coincides with that of the function $\lambda(F_{k_i}, F_{k_h})$, which returns the dissimilarity degree between F_{k_i} and F_{k_h}. This function may vary depending on the type of F_{k_i} and F_{k_h}. For example, if the type of F_{k_i} and F_{k_h} is numerical and $\delta(F_{k_i}, F_{k_h})$ is the difference between the two values, then its time complexity is $\mathcal{O}(1)$. Instead, if the type is textual, $\delta(F_{k_i}, F_{k_h})$ could be the common Hamming distance, whose time complexity is $\mathcal{O}(m)$, where m is the size of the two strings. Without loss of generality, we assume that the features have size r and consider the maximum time complexity $\mathcal{O}(r^o)$ (for some constant o) among all the dissimilarity degree functions considered for each type of features in our dataset. However, considering that our graph is static, we can precompute all the possible dissimilarity degrees and access them with a $\mathcal{O}(1)$ time complexity.

Let us now analyze the time complexity of assessing the relevance of a feature during classification. Given a node n_i of $\mathcal{N}$, our approach scans its incoming and outgoing arcs to retrieve the sets of nodes N_i^{in} and N_i^{out}. The time complexity of computing these sets is $\mathcal{O}(l)$. Nevertheless, considering that we focus on static graphs, we can store the values of $|N_i^{in}|$ and $|N_i^{out}|$ in such a way as to access them in $\mathcal{O}(1)$. Let us now consider $\rho(F_{k_i})$, i.e., the relevance of the occurrence F_{k_i} of

F_k corresponding to the instance I_i. The formula for its computation (see Eq. (3.5)) is an adaptation of the PageRank centrality; therefore, they share a common complexity. Several approximation algorithms exist in the literature to compute PageRank. The typical PageRank algorithms have computational complexity $\mathcal{O}(log(l))$ (Chung, 2014), given the value of the damping factor (in our case d_{k_i} in Eq. (3.5)). The computation of the value of one instance d_{k_i} of d_k costs $\mathcal{O}(l)$ if we assume that the time complexity of the sigmoid function $\sigma(\cdot)$ is $\mathcal{O}(1)$ and that the result of the dissimilarity degree function is available. We can assume that we compute all values d_{k_i} once and store them for the application of Eq. (3.5). This step, overall, costs $\mathcal{O}(l^2)$ and must be carried out only once. As a consequence, since d_{k_i} is already available, the time complexity of the relevance $\rho(F_{k_i})$ is $\mathcal{O}(log(l))$. It follows from this result that the time complexity of the relevance $\rho(F_k)$ of a feature F_k (see Eq. (3.7)) is $\mathcal{O}(l \cdot log(l))$. By storing the results of this last computation, the time complexity of Eq. (3.8) is $\mathcal{O}(l \cdot n \cdot log(n))$. The last complexity encompasses the time complexity of computing the median, which essentially reduces to a sorting procedure; however, recall that, in most cases, $n \ll l$ and, consequently, in these cases the time complexity of Eq. (3.8) reduces to $\mathcal{O}(l)$.

Integrating all these partial results, the highest cost of the various steps described above is $\mathcal{O}(l^2)$.

The last task we analyze is measuring the sensitivity of our approach. In this case, we have as input a new instance I_j to be added to the set $\mathcal{I}$. We assume that the result of its classification by $\mathcal{M}$ can be obtained in constant time. Therefore, the first step is the construction of the network $\hat{\mathcal{N}}$ that includes I_j. In our setting, the time complexity of such operation is $\mathcal{O}(l^2)$. At this point, we examine Eq. (3.10), which returns the estimated relevance $\tilde{\rho}(F_{k_j})$. In this case, the time complexity is $\mathcal{O}(l)$. This is because we resort to the values of $\tilde{\rho}(F_{k_h})$ computed before the addition of the new instance I_j. We also assume the availability of the results of the dissimilarity degree function. Instead, the time complexity of computing the exact value $\rho(F_{k_j})$ is the same as seen above, i.e., $\mathcal{O}(l^2)$ to compute the values of all instances d_{k_j} of d_k, and $\mathcal{O}(log(l))$ to compute $\rho(F_{k_j})$. The last step is computing the difference between the actual and estimated values of the relevance of F_{k_j}, i.e., $\Delta(F_{k_j})$. By storing the results of the exact and estimated value of the relevance, the time complexity of $\Delta(F_{k_j})$ is $\mathcal{O}(l)$. Based on this result, we obtain that the time complexity of the overall variation in the relevance of F_k caused by the new instance I_j, denoted by $\hat{\Delta}(F_k, I_j)$ (see Eq. (3.13)), is $\mathcal{O}(l^2)$. Finally, the time complexity of computing the relative variation, defined in Eq. (3.14), relatively to the case in which we store the values of $\Delta(F_{k_i})$ and $\rho(F_{k_i})$, is $\mathcal{O}(l)$. Again, overall, the highest cost of the various steps described above is $\mathcal{O}(l^2)$.

3.5.2. Analysis on the space complexity

Before analyzing the space complexity of our approach, we want to point out that a trade-off between the spatial and temporal aspects is necessary. In fact, as we saw in Section 3.5.1, the approach used by our framework is based on a set of values that, since the underlying network is static, can be computed once and stored for later computations. We used this procedure several times in the previous section.

That said, let us begin by analyzing the space complexity of the lookup tables used to store the confidence levels and feature set. It is $\mathcal{O}(l)$ and $\mathcal{O}(l \cdot n)$, respectively.

Let us now consider the space complexity of building the network $\mathcal{N}$. In our framework, we represent it by an adjacency matrix; therefore, its complexity is $\mathcal{O}(l^2)$. Additionally, for each node n_i of $\mathcal{N}$ we store the values of $|N_i^{in}|$ and $|N_i^{out}|$; the space complexity of this operation is $\mathcal{O}(l)$. We also compute and store the values of the dissimilarity degree function for the features of each pair of instances, and the space complexity of this operation is $\mathcal{O}(l^2 n)$. As mentioned above, in most cases, $n \ll l$; thus, the space complexity of this operation is $\mathcal{O}(l^2)$.

Let us now analyze the space complexity of assessing the relevance of a feature during classification. During this assessment, we compute and store the values of d_{k_i} and the results of the relevance $\rho(F_{k_i})$, and we do it for each instance $I_i \in \mathcal{I}$; the space complexity of this operation is $\mathcal{O}(l \cdot n)$. In addition, in calculating the ability of $\mathcal{M}$ to differentiate feature relevance (see Eq. (3.8)), we compute the median relevance of features; the space complexity of this task is $\mathcal{O}(n)$.

The last step to consider concerns sensitivity measurement. To this end, our framework builds the new network $\hat{\mathcal{N}}$, with a space complexity of $\mathcal{O}(l^2)$. Then, it computes and stores the results of the exact and estimated values of the relevance F_{k_i}, with a space complexity of $\mathcal{O}(l \cdot 2n)$. Finally, it calculates and stores the values of $\Delta(F_{k_i})$, whose space complexity is $\mathcal{O}(l \cdot n)$.

3.5.3. Final considerations

Note that although the complexity outlined above is polynomial in the number of instances (i.e., $\mathcal{O}(l^2)$), it may happen that l becomes really large, even in the order of millions of instances. Clearly, even a quadratic algorithm becomes too expensive in this case. However, given the application scenario of the proposed approach, it is reasonable to apply it on a properly composed subset of representative instances. If one proceeds in this direction, the size of this subset can be determined in such a way as to ensure reasonable performances.

3.6. Positioning of our framework within XAI approaches and its support in practical decision making scenarios

As mentioned in Section 2, many XAI approaches have been proposed in the literature. The main ones fall in the following categories (Banerjee & Barnwal, 2023): *(i) Feature Relevance*, which includes those approaches that identify the features that most explain the output of the model; *(ii) Local explanations*, which includes those approaches that want to explain the operation of a part of the overall system; *(iii) Visualization*, which includes those techniques that want to visualize the behavior of a model, often minimizing the complexity of the problem; they are aimed at users who are not familiar with AI; *(iv) Explanation by example*, which includes those techniques that elicit distinctive data to provide an explanation of the overall behavior of the model; *(v) Text explanation*, which includes those techniques that produce a natural language text to represent the behavior of an AI model and, specifically, the algorithm rationale; *(vi) Model simplification*, which includes those techniques that build a new model that is less complex and more interpretable than the original model.

Along these techniques, several models and methods have been developed (Chinu & Bansal, 2022), such as: *(i) Prototype and criticism*, which provides a representation of all the data for a particular data instance and singles out data that are not represented by the prototypes through criticism; *(ii) Model distillation*, which exploits a student–teacher model that trains a student model (the explanation) behaving similarly to the teacher model (the original model); *(iii) Surrogate model*, which is an interpretable model that mimics the predictions of a black-box model; *(iv) Partial dependence plot*, which displays the marginal effect of one or more features on the expected output; *(v) Decision tree*, which describes the behavior of a machine learning model in the form of trees; *(vi) Rule extraction*, which learns (linear) decision rules representing automatically recognized interaction effects.

In this scenario, our framework falls into the Feature Relevance category and yet it adopts a completely new model and method compared to existing ones. Regarding the usefulness of our framework in practical decision making scenarios, we emphasize that it enables the identification of feature relevance at both the model and instance levels. The identification of feature relevance at the instance level means that when a new instance is processed by the classifier, our framework allows us to identify which features contributed most to its classification. On the other hand, identifying feature relevance at the model level makes it possible to identify on which features to act so that future instances have values for those features that allow them to belong to one class rather than another.

3.7. *Strengths and weaknesses of our framework*

As pointed out in the Introduction, our approach belongs to the category of model-agnostic tools for XAI. However, as can be seen from the detailed taxonomy of XAI approaches presented in a recent survey (Nagahisarchoghaei et al., 2023), our approach, being network-based, adopts a way of proceeding completely different from the one of other existing approaches belonging to the same category, and thus may open new research scenarios in this area. Due to the network-based philosophy it adopts, our framework helps to address several aspects considered as limitations of common model-agnostic approaches (Barredo Arrieta et al., 2020). In the following, we will explain these aspects in detail.

First, our approach is parameter-free, that is, it has no parameters for the user to set. It also does not have to discern between shallow Machine Learning models and Deep Learning models, as others do (Barredo Arrieta et al., 2020). In addition, it is not restricted to classifiers operating on homogeneous features. In particular, since it computes instance-based pairwise feature dissimilarities, and it uses feature-specific functions to do this, it is able to handle feature heterogeneity straightforwardly. Interestingly, this way of computing dissimilarities allows our framework to seamlessly couple with data fusion systems (Barredo Arrieta et al., 2020).

Another advantage of our approach is that the notion of dyscrasia introduced in this paper allows us to detect insights not only into which features are important but also into how these features interact with each other (Barredo Arrieta et al., 2020; Nagahisarchoghaei et al., 2023). Again, while many feature relevance approaches operate by assuming that features are independent of each other (Barredo Arrieta et al., 2020), our framework does not need this assumption to operate.

There are two further strengths that differentiate our framework from existing literature. First, the very same technique allows us to characterize features at both the model and the instance levels; in particular, our framework can determine the relevance of features to the entire model, as well as which features of a single instance contribute most to the classification. In addition, our framework's sensitivity computation procedure can support in determining the possible obsolescence of the underlying classification model. In particular, as will become clear in the experiments, the sensitivity computation that can be performed with our framework allows for the interception of radical changes in the inputs in a certain direction. The presence of such variation, in turn, can be an indicator of model obsolescence and the profitability of updating the model classifier based on the new input characteristics.

Finally, the introduction of the concept of dyscrasia, and thus the possibility of quantifying the disharmony between the values of a given feature during classification, is also a strength of our approach in that it allows us to understand how certain values of a feature can contribute to misclassification or, conversely, to correct classification.

Our framework also has weaknesses that we now examine. First, the model-agnostic nature of our framework, while a strength in terms of applicability, is also a weakness in terms of the depth of explanations obtained. In fact, model-specific XAI approaches could provide deeper insight tailored to the model itself. Furthermore, our framework assumes the assignment of a single label by the classifier; this assumption limits the applicability of our framework in multi-label scenarios.

Another limitation comes from the use of the network-based model. While it provides an intuitive form of explanation and allows a macroscopic view of the context, it could also make the approach computationally expensive when working with huge amounts of data, as seen in Section 3.5. Finally, in its current definition, our framework is not well suited to provide explanations in dynamic contexts, where the underlying data distribution, or even the classifier, continually evolve. Adapting our framework to this more complex scenario is certainly very challenging and is the subject of possible future developments.

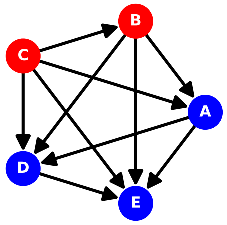

Fig. 2. The network $\mathcal{N}$ after classification.

Table 1
Instances, their features, confidence values and assigned classes.

I_i	F_1	F_2	c_i	C_i
A	0.48	0.52	0.37	1
B	0.66	0.73	0.34	0
C	0.18	0.21	0.21	0
D	0.47	0.53	0.88	1
E	0.45	0.58	0.95	1

4. Case examples

In this section, we want to present some simple case examples to better clarify the formulas underlying our framework. In particular, we consider three cases: In the first one, we examine the behavior of our approach in the presence of a classifier whose features are not coherent in the classification process. In the second one, we assume that all the features contribute to classification in a coherent way. Finally, in the third case, we assume that only one feature contributes to the classification in a coherent way. In all these cases, we suppose that $\mathcal{I}$ consists of five instances: $\mathcal{I} = \{A, B, C, D, E\}$, $\mathcal{C}$ consists of two classes: $\mathcal{C} = \{0, 1\}$ and $\mathcal{F}$ consists of two features: $\mathcal{F} = \{F_1, F_2\}$. We further suppose that F_1 and F_2 are numerical with values between 0 and 1. Accordingly, the function $\lambda(\cdot,\cdot)$ consists of the absolute value of the difference of the values of the parameters received as input. Formally speaking, $\lambda(F_{k_i}, F_{k_h}) = |F_{k_i} - F_{k_h}|$. In all cases, the best classification (and, thus, the one with the highest confidence) is for the instance E, while the worst one (to which the lowest confidence corresponds) is for the instance C.

4.1. *Worst case example*

In this case example, we assume that the features F_1 and F_2 discriminate very poorly, being both very similar with each other for almost all instances and having no specific correlation with the corresponding classification; the average confidence of the classifier is 0.53. Table 1 shows the instances involved, their features, their classification confidences and the classes assigned to them. Instead, Fig. 2 shows the corresponding network $\mathcal{N}$ obtained at the end of the classification of all the instances.

4.1.1. *Relevance computation*

We begin the computation on the relevance of feature occurrences starting with node C. Since this is the worst instance, i.e., the one with the lowest confidence, there is no arc incoming into C. Consequently, $N_C^{in} = \emptyset$ and $N_C^{out} = \{A, B, D, E\}$. Applying Eq. (3.6), we can compute the values d_{1_C} and d_{2_C} of the damping factors associated with the

Table 2
Value of relevance of the feature occurrences for the worst case example.

Node	$\rho(F_1)$	$\rho(F_2)$	Absolute difference
A	0.1083	0.1084	0.0001
B	0.0309	0.0306	0.0003
C	0.0267	0.0290	0.0023
D	0.1355	0.1358	0.0002
E	0.2033	0.2037	0.0004

feature occurrences F_{1_C} and F_{2_C}, respectively:

$$\begin{aligned} d_{1_C} &= \sigma\left(\frac{\sum_{n_h \in N_C^{out}} \delta(F_{1_C}, F_{1_h})}{|N_C^{out}|}\right) \\ &= \sigma\left(\frac{\delta(F_{1_C}, F_{1_A}) + \delta(F_{1_C}, F_{1_B}) + \delta(F_{1_C}, F_{1_D}) + \delta(F_{1_C}, F_{1_E})}{4}\right) \\ &= \sigma\left(\frac{0.2046 + 0.2503 + 0.4936 + 0.5479}{4}\right) = 0.8665 \\ d_{2_C} &= 0.8548 \end{aligned}$$

We are now able to compute the relevance of F_{1_C} and F_{2_C} by applying Eq. (3.5). In this case, since $N_C^{in} = \emptyset$, the computations are straightforward. Indeed:

$$\rho(F_{1_C}) = \frac{1 - d_{1_C}}{|N|} + d_{1_C} \cdot \sum_{n_h \in N_C^{in}} \frac{\rho(F_{1_h})}{|N_h^{out}|} = \frac{1 - 0.8665}{5} + 0.8665 \cdot 0 = 0.0267$$

$$\rho(F_{2_C}) = \frac{1 - 0.8548}{5} + 0.8548 \cdot 0 = 0.0290$$

Let us now consider node B. It has only one incoming arc, namely the one starting from C. Therefore, $N_B^{in} = \{C\}$ and $N_B^{out} = \{A, D, E\}$. Applying Eq. (3.6), we can compute the values d_{1_B} and d_{2_B} associated with the feature occurrences F_{1_B} and F_{2_B}, respectively:

$$\begin{aligned} d_{1_B} &= \sigma\left(\frac{\sum_{n_h \in N_B^{out}} \delta(F_{1_B}, F_{1_h})}{|N_B^{out}|}\right) \\ &= \sigma\left(\frac{\delta(F_{1_B}, F_{1_A}) + \delta(F_{1_B}, F_{1_D}) + \delta(F_{1_B}, F_{1_E})}{3}\right) \\ &= \sigma\left(\frac{0.2002 + 0.4704 + 0.4953}{3}\right) = 0.8747 \\ d_{2_B} &= 0.8791 \end{aligned}$$

After this, we compute the relevance of the two feature occurrences:

$$\begin{aligned} \rho(F_{1_B}) &= \frac{1 - d_{1_B}}{|N|} + d_{1_B} \cdot \sum_{n_h \in N_B^{in}} \frac{\rho(F_{1_h})}{|N_h^{out}|} = \frac{1 - 0.8747}{5} + 0.8747 \cdot \left(\frac{0.0267}{4}\right) \\ &= 0.0309 \\ \rho(F_{2_B}) &= \frac{1 - 0.8791}{5} + 0.8791 \cdot \left(\frac{0.0290}{4}\right) = 0.0306 \end{aligned}$$

Proceeding in the same way with the remaining nodes, we obtain the values of the relevance of all feature occurrences. They are shown in Table 2. Observe that all relevance values are very low; however, recall that, given the structure of the formulas for the computation of the relevance of the feature occurrences, the corresponding values tend to be flattened downward. Actually, the most important thing to observe is that the absolute difference (i.e., the absolute value of the difference of the relevances of the two feature occurrences) is extremely low; this, coupled with the very low values, reflects the hypothesis of the case example, i.e., that no feature is actually discriminating for the classification of any instance.

Finally, by applying the formula in Eq. (3.7), we can compute the value of the relevance of F_1 and F_2. These values are reported in Table 3.

Table 3
Value of relevance of the features in the worst case example.

Feature	Relevance
F_1	0.1010
F_2	0.1015

4.1.2. Sensitivity computation

Once we have computed the relevance of both the feature occurrences and the features, we proceed to perform a sensitivity analysis by applying the technique described in Section 3.4. To this end, suppose that we need to include in $\mathcal{I}$, and next classify, a new instance G whose features are: $F_{1_G} = 0.51$ and $F_{2_G} = 0.49$. Suppose, also, that the classifier underlying our approach assigned G to the class 0 and that the corresponding confidence c_G is 0.40.

In the new network $\hat{\mathcal{N}}$, obtained after the classification of G, we have that: $\hat{N}_G^{out} = \{D, E\}$ and $\hat{N}_G^{in} = \{A, B, C\}$.

As for the computation of the damping factors, we have that:

$$\begin{aligned} d_{1_G} &= \sigma\left(\frac{\delta(F_{1_G}, F_{1_D}) + \delta(F_{1_G}, F_{1_E})}{2}\right) = \sigma\left(\frac{0.5069 + 0.5358}{2}\right) \\ &= \sigma(0.5213) = 0.9580 \end{aligned}$$

$$\begin{aligned} d_{2_G} &= \sigma\left(\frac{\delta(F_{2_G}, F_{2_D}) + \delta(F_{2_G}, F_{2_E})}{2}\right) = \sigma\left(\frac{0.5069 + 0.5187}{2}\right) \\ &= \sigma(0.5128) = 0.9559 \end{aligned}$$

Applying the formula given in Eq. (3.12), we can compute the values $\Delta(F_{1_G})$ and $\Delta(F_{2_G})$, which represent the difference between the actual and estimated values of the relevance of F_{1_G} and F_{2_G}, respectively.

$$\begin{aligned} \Delta(F_{1_G}) &= d_{1_G} \cdot \sum_{n_h \in \hat{N}_G^{in}} \frac{\left|\rho(F_{1_h}) - \tilde{\rho}(F_{1_h})\right|}{\left|\hat{N}_h^{out}\right|} \\ &= d_{1_G} \cdot \left(\frac{|\rho(F_{1_A}) - \tilde{\rho}(F_{1_A})|}{|\hat{N}_A^{out}|} + \frac{|\rho(F_{1_B}) - \tilde{\rho}(F_{1_B})|}{|\hat{N}_B^{out}|} \right. \\ &\quad \left. + \frac{|\rho(F_{1_C}) - \tilde{\rho}(F_{1_C})|}{|\hat{N}_C^{out}|}\right) \end{aligned}$$

The values $\tilde{\rho}(F_{1_A})$, $\tilde{\rho}(F_{1_B})$ and $\tilde{\rho}(F_{1_C})$ are as shown in Table 2. The values $\rho(F_{1_A})$, $\rho(F_{1_B})$ and $\rho(F_{1_C})$ are obtained by applying the formula of Eq. (3.10) on the new network $\hat{\mathcal{N}}$. Here, we do not report all the steps of this computation because the procedure is the same as the one for the relevance computation carried out in the previous section. The new values obtained are the following:

$$\rho(F_{1_A}) = 0.0698 \quad \rho(F_{1_B}) = 0.0264 \quad \rho(F_{1_C}) = 0.0202$$

At this point, we have that:

$$\begin{aligned} \Delta(F_{1_G}) &= 0.9580 \cdot \left(\frac{|0.0698 - 0.1083|}{3} + \frac{|0.0264 - 0.0309|}{4} \right. \\ &\quad \left. + \frac{|0.0202 - 0.0267|}{5}\right) \\ &= 0.0146 \end{aligned}$$

Proceeding similarly for the feature F_2, we first compute $\rho(F_{2_A})$, $\rho(F_{2_B})$ and $\rho(F_{2_C})$ and obtain:

$$\rho(F_{2_A}) = 0.0698 \quad \rho(F_{2_B}) = 0.0251 \quad \rho(F_{2_C}) = 0.0224$$

Afterwards, we compute $\Delta(F_{2_G})$ as:

$$\begin{aligned} \Delta(F_{2_G}) &= 0.9559 \cdot \left(\frac{|0.0698 - 0.1084|}{3} + \frac{|0.0251 - 0.0306|}{4} \right. \\ &\quad \left. + \frac{|0.0224 - 0.0290|}{5}\right) \\ &= 0.0149 \end{aligned}$$

Table 4
Instances, their features, confidence values and assigned classes.

I_1	F_1	F_2	c_i	C_i
A	0.15	0.75	0.82	1
B	0.80	0.31	0.78	0
C	0.64	0.45	0.65	0
D	0.11	0.82	0.88	1
E	0.02	0.95	0.95	1

Table 5
Value of relevance of the feature occurrences for the best case example.

Node	$\rho(F_1)$	$\rho(F_2)$	Absolute difference
A	0.1251	0.1225	0.0026
B	0.0965	0.0878	0.0087
C	0.0738	0.0657	0.0081
D	0.1565	0.1534	0.0031
E	0.2348	0.2302	0.0047
Avg	0.1374	0.1319	0.0054

Table 6
Instances, their features, confidence values and assigned classes.

I_1	F_1	F_2	c_i	C_i
A	0.15	0.75	0.55	1
B	0.80	0.25	0.53	0
C	0.64	0.50	0.21	0
D	0.11	0.25	0.69	1
E	0.02	0.50	0.87	1

Table 7
Value of relevance of the feature occurrences for the intermediate case example.

Node	$\rho(F_1)$	$\rho(F_2)$	Absolute difference
A	0.1196	0.1021	0.0175
B	0.0853	0.0491	0.0362
C	0.0539	0.0262	0.0277
D	0.1504	0.1354	0.0150
E	0.2261	0.2047	0.0214
Avg	0.1271	0.1035	0.0236

Knowing the real values of $\rho(F_{1_G}) = 0.0395$ and $\rho(F_{2_G}) = 0.0399$ in $\hat{N}$, we can compute the percentage of variation between the estimated and actual values. This is equal to $\frac{0.0146}{0.0395} \cdot 100 = 36.96\%$ for F_{1_G} and to $\frac{0.0149}{0.0399} \cdot 100 = 37.34\%$ for F_{2_G}. These values are quite high. However, this was somewhat expected, considering that: *(i)* the classifier was poorly confident, on average, about its classifications, and *(ii)* the features of G had quite different values from those assumed by the same features for the instances D and E that belong to the same class of G (i.e., class 0).

4.2. Best case example

In this case example, we assume that both features are strongly discriminating. In particular, in this case, a low value of F_1 and a high value of F_2 imply class 1, whereas a high value of F_1 and a low value of F_2 imply class 0; the average confidence of the classifier is 0.82. In Table 4, we report the instances involved, their features, their classification confidences and the classes assigned to them. Due to space constraints we do not report here again all details about relevance computation; instead, we show the overall results directly in Table 5.

Some interesting observations can be drawn from the analysis of this table. In particular, we can see that the more discriminating the values of F_1 and F_2 are, the greater their relevances. As evidence of this, consider the instance E, which is the one with the highest confidence and the most extreme values of F_1 and F_2. The relevance values associated with E for the two features are the highest of all the relevance values associated with the various instances. Instead, consider the instance C, which has the least discriminating values for F_1 and F_2. The relevance values associated with C for the two features are the lowest among all the relevance values associated with the various instances. This confirms that our relevance definition is really able to tell us when a feature contributed to determining the class assigned by the classifier to each instance. Finally, note that the absolute values of the differences between the feature relevances for the instances are all very low. This was expected and desirable since both features in this example are strongly discriminating.

After computing the relevances of features and the corresponding feature occurrences, we proceed with sensitivity analysis. As in the worst case example, suppose we need to include in $\mathcal{I}$, and next classify, a new instance G whose features are: $F_{1_G} = 0.91$ and $F_{2_G} = 0.15$. Finally, suppose that the classifier assigned G to the class 0 and the corresponding confidence c_G is 0.84.

Operating similarly to what we have done in Section 4.1.1, we can compute the values $\Delta(F_{1_G})$ and $\Delta(F_{2_G})$, which represent the difference between the actual and estimated values of the relevance of F_{1_G} and F_{2_G}, respectively. Specifically, we have that $\Delta(F_{1_G}) = 0.0086$ and $\Delta(F_{2_G}) = 0.0086$.

Knowing the corresponding real values $\rho(F_{1_G})$ and $\rho(F_{2_G})$ in $\hat{\mathcal{N}}$, we can compute the percentage of variation between the estimated and real values. This is equal to 6.30% for F_{1_G} and 6.61% for F_{2_G}. These values are much smaller than those obtained in Section 4.1.1 for the worst case. This result can be easily explained by considering that the underlying classifier was really confident, on average, about its classifications. This ensured that the network $\mathcal{N}$, the value configuration and the relationships between instances are all very stable. Consequently, compared to the worst case example, the entry of a new instance, even with values of features quite different from those of the other instances of the same class, was not able to produce significant alterations in the configuration of the network $\mathcal{N}$ and the corresponding relationships and associated parameters.

4.3. Intermediate case example

In this final case example, we assume that only feature F_1 is strongly discriminating. In particular, in this case, a low (resp., high) value of F_1 implies that the corresponding instance belongs to class 1 (resp., 0). The average confidence of the classifier is 0.57. In Table 6, we report the instances involved, their features, their classification confidences and the classes assigned to them. Again, due to space constraints, we report the values of the relevance of the feature occurrences directly in Table 7.

From the analysis of this table we can observe that the relevance of F_2 is always lower, and in some cases significantly lower, than the one of F_1. We can also observe that F_1 is more capable of discriminating class 1 than class 0. In fact, the values of $\rho(F_1)$ are much higher for the instances A, D and E (that belong to class 1) than for the instances B and C (that belong to class 0). This result could not have been obtained from examining confidence alone, since, for example, the confidence of B is still high (comparable with the one of A, albeit much lower than the one of D and E).

Again, after computing the relevance values of features and the corresponding feature occurrences, we proceed with a sensitivity analysis. As in the other case examples, suppose we need to include in $\mathcal{I}$, and next classify, a new instance G whose features are: $F_{1_G} = 0.91$ and $F_{2_G} = 0.75$. Finally, suppose that the classifier assigned G to the class 0 and the corresponding confidence c_G is 0.57.

In this case, the percentages of variation between the estimated and real values are equal to 7.09% for F_{1_G} and 14.69% for F_{2_G}. These values confirm that the classifier confidence is mostly affected by the discrimination capability of F_1. Instead, the stability of the framework with respect to the feature F_2 is partially compromised by the very different and not characterizing values of F_2 in the instances of the same class.

Table 8
Main characteristics of the datasets used in our experiments.

Dataset	Instances	Features	Classes
Iris	150	4	3
Mammographic Mass	961	4	2

5. Experimental campaign

In this section, we illustrate the tests we conducted on real data to verify the behavior of our approach. In particular, we present our testbed in Section 5.1. In Section 5.2, we describe the experiments on the feature relevance computation we performed and the results we obtained. Finally, in Section 5.3, we illustrate our tests concerning sensitivity computation and the results they returned.

5.1. Testbed

To carry out our tests we used a 2019 MacBook Pro equipped with 16 GB of RAM and 2.6 GHz Intel Core i7 6 core. We also developed a desktop app implementing our approach, called NAFER (Nafer Another FEature Relevance). It is freely available on GitHub at the following address: www.github.com/ecorradini/NAFER.

Furthermore, we selected several classifiers to include as underlying engine of our approach. The classifiers we chose are among those most widely adopted in the past literature (Datta, Sen, & Zick, 2016; Henelius et al., 2017; Strumbelj & Kononenko, 2010). They are:

- Naive Bayes (hereafter, NB) Zhang (2004);
- SVM with polynomial kernel (hereafter, Polynomial SVM) (Chang & Lin, 2011);
- SVM with radial basis function kernel (hereafter, Radial SVM) (Chang & Lin, 2011);
- Multi-Layer Perceptron (hereafter, MLP) He, Zhang, Ren, and Sun (2015);
- Random Forest (hereafter, RF) Breiman (2001).

In particular, we chose these classifiers because our framework is model-agnostic and we want to test and exploit this property of it. In fact, our classifiers are of different types and exhibit different behaviors. Naive Bayes is a probabilistic classifier, unlike SVM, which is non-probabilistic. For the latter, we chose two different kernels, namely: *(i)* the polynomial one, which considers features and their combinations, and *(ii)* the radial one, which separates data using a nonlinear decision-boundary. Multi-Layer Perceptron is a neural network; therefore, it represents a totally black-box model. Finally, Random Forest is an example of an ensemble learning model.

During the test campaign, we used two datasets published on the UCI Machine Learning Repository (Asuncion & Newman, 2007). They are Iris (Fisher, 1936) and Mammographic Mass (Elter, Schulz-Wendtland, & Wittenberg, 2007). The number of instances, features and classes of the two datasets are shown in Table 8. As can be seen from this table, the two datasets are very different in the number of instances, while they are similar in the number of features and classes.

More specifically, the Iris features are the following:

- `sepal_length`: It represents the sepal length of the flower in centimeters; its values range in the real interval [4.3, 7.9];
- `sepal_width`: It denotes the sepal width of the flower in centimeters; its values range in the real interval [2.0, 4.4];
- `petal_length`: It indicates the petal length of the flower in centimeters; its values range in the real interval [1.0, 6.9];
- `petal_width`: It represents the petal width of the flower in centimeters; its values range in the real interval [0.1, 2.5].

Instead, the features of Mammographic Mass are the following:

Table 9
Classifier accuracy with the Iris dataset.

Model	Accuracy
Naive Bayes (Zhang, 2004)	0.93
SVM with polynomial kernel (Chang & Lin, 2011)	0.98
SVM with radial basis function kernel (Chang & Lin, 2011)	0.96
Multi-Layer Perceptron (He et al., 2015)	0.93
Random Forest (Breiman, 2001)	0.96

- `age`: It denotes the patient's age in years; its values range in the integer interval [0, 96];
- `shape`: It indicates the mass shape; its possible values are: round = 1, oval = 2, lobular = 3, and irregular = 4;
- `margin`: It represents the mass margin; its possible values are: circumscribed = 1, microlobulated = 2, obscured = 3, ill-defined = 4, and spiculated = 5;
- `density`: It denotes the mass density; its possible values are: high = 1, iso = 2, low = 3, fat-containing = 4.

All features are numerical; however, the values they can take are very heterogeneous. To homogenize them, we performed a normalization task. For this purpose, we used a min–max scaler (Ahsan, Mahmud, Saha, Gupta, & Siddique, 2021). Given the value F'_{k_i} of a feature, whose maximum (resp., minimum) value is $F'_{k_{max}}$ (resp., $F'_{k_{min}}$), this scaler obtained a normalized value F_{k_i}, belonging to the real interval $[0, 1]$, by means of the following formula:

$$F_{k_i} = \frac{F'_{k_i} - F'_{k_{min}}}{F'_{k_{max}} - F'_{k_{min}}} \tag{5.1}$$

At this point, having all feature occurrences normalized between 0 and 1, we decided to choose as dissimilarity function $\lambda(F_{k_i}, F_{k_h})$ between two feature occurrences F_{k_i} and F_{k_h}, the absolute value of their difference, that is:

$$\lambda(F_{k_i}, F_{k_h}) = |F_{k_i} - F_{k_h}| \tag{5.2}$$

5.2. Relevance computation

In this section, we illustrate the tests on relevance computation that we performed first on the Iris dataset (Section 5.2.1) and then on the Mammographic Mass dataset (Section 5.2.2).

5.2.1. Iris dataset

The first task we performed was the computation of the accuracy of classifiers. In fact, this parameter is the most widely used in the literature for an initial assessment of the performance of classifiers (Han, Kamber, & Pei, 2011). The results we obtained are shown in Table 9. As we can see, these values are very high. Therefore, we can assume that all the classifiers considered are able to guarantee appropriate confidence values and, thus, are eligible for the next steps of our analysis.

Recall that the main objective of our analysis is to check whether there are features having a higher relevance than others and, if so, to detect them. Therefore, feature relevance is the second performance indicator we considered in our tests. It also represents one of the main goals of our paper since it allows us to understand which features are most important during the classification process and, ultimately, sheds light on some aspects of classifier behavior. Starting from this consideration, if all classifiers show no significant differences between the relevance values of the various features, we can reasonably assume that all of them have the same relevance. Conversely, suppose some or all classifiers show different relevance values for the various features, and there is a substantial agreement among them in indicating which features are the most relevant. In that case, we can reasonably conclude that the relevance values of the features are different, and we can determine the most relevant ones. In this case, the best classifiers are those that can best highlight the differences in feature relevances.

Having made this premise, we can continue with the description of our experiment. Its next task involved the computation of the damping factor for the various features and classifiers. We chose to analyze damping factor as a preliminary performance indicator because our formula of relevance is derived from PageRank centrality, where damping factor plays a crucial role (see Eqs. (3.4) and (3.5)). Moreover, in our formula of relevance, the damping factor is not fixed but variable, as shown in Eq. (3.6). The boxplots with the corresponding distributions are reported in Fig. 3.

From the analysis of this figure, we can observe that the various classifiers show completely different behaviors regarding the values of the damping factor. In particular:

- Naive Bayes tends to assign extremely low and similar values to the damping factor for all features.
- Polynomial SVM assigns very different values to the damping factor of the various features. From this point of view, it shows a very good ability to discriminate among features. However, this ability will have to be verified, and hopefully confirmed, by the next computations concerning feature relevances.
- Radial SVM shows some differences in the values of the damping factor. However, these are smaller than the ones associated with Polynomial SVM, since the values of the damping factor tend to settle in a central range between 0.60 and 0.70.
- Multi-Layer Perceptron allows for very varied values of the damping factor between occurrences of the same feature. Instead, median values are all very high. This classifier is less able to differentiate the values of the damping factor among the various features than the two SVM classifiers, albeit it seems better than Naive Bayes.
- Random Forest has a very similar, although less extreme, behavior than Naive Bayes. In any case, it does not show a great ability to differentiate among features.

All these conclusions drawn by analyzing the damping factor are preliminary, although indicative of potential trends. Actually, the final conclusions of interest to us are those that can be drawn by examining the distributions of relevance values associated with the various occurrences of the four features for the five classifiers. They are shown in Fig. 4. By examining this figure, we can draw the following insights:

- There are some classifiers (in particular, Naive Bayes and Random Forest) that fail to capture the differences in relevance existing among features.
- The two SVM classifiers and Multi-Layer Perceptron are able to capture differences in relevance among features, although this ability is held to different degrees by the three classifiers.
- The differences identified by the various classifiers are concordant. In particular, both Polynomial SVM and Radial SVM and, to some extent, Multi-Layer Perceptron show that `petal_length` and `petal_width` are more relevant than `sepal_length` and `sepal_width`.
- The two classifiers that prove most capable of discerning differences in feature relevances are Polynomial SVM and Radial SVM.

The conclusions we drew by examining Fig. 4 are only partially quantitative; in fact, they are in many ways more qualitative than quantitative. So, the next step is to quantify the different classifier abilities to discern differences in feature relevances. In Table 10, we report the median values of the occurrence relevances for each feature and for each classifier. In this case, we used the median as a performance indicator because it is less sensitive to outliers than the mean. The analysis of this table shows that, even at the quantitative level, `petal_length` and `petal_width` are more relevant than `sepal_length` and `sepal_width`.

Table 10
Median relevance of each feature returned by the five classifiers for the Iris dataset.

Model	Feature	Relevance
Naive Bayes (Zhang, 2004)	sepal_length	0.014598
	sepal_width	0.014572
	petal_length	0.014696
	petal_width	0.014714
SVM with polynomial kernel (Chang & Lin, 2011)	sepal_length	0.005608
	sepal_width	0.002904
	petal_length	0.007408
	petal_width	0.007660
SVM with radial basis function (Chang & Lin, 2011)	sepal_length	0.009293
	sepal_width	0.009238
	petal_length	0.011012
	petal_width	0.011139
Multi-Layer Perceptron (He et al., 2015)	sepal_length	0.000108
	sepal_width	0.000082
	petal_length	0.001598
	petal_width	0.001454
Random Forest (Breiman, 2001)	sepal_length	0.014313
	sepal_width	0.014280
	petal_length	0.014504
	petal_width	0.014534

Table 11
Values of the function $\alpha(\cdot)$ for the classifiers into consideration and for the Iris dataset.

Model	Value of $\alpha(\cdot)$
Naive Bayes (Zhang, 2004)	1.29%
SVM with polynomial kernel (Chang & Lin, 2011)	37.47%
SVM with radial basis function (Chang & Lin, 2011)	17.62%
Multi-Layer Perceptron (He et al., 2015)	11.43%
Random Forest (Breiman, 2001)	2.50%

Table 12
Classifier accuracy with the Mammographic Mass dataset.

Model	Accuracy
Naive Bayes (Zhang, 2004)	0.77
SVM with polynomial kernel (Chang & Lin, 2011)	0.78
SVM with radial basis function (Chang & Lin, 2011)	0.81
Multi-Layer Perceptron (He et al., 2015)	0.77
Random Forest (Breiman, 2001)	0.76

We computed the values returned by the function $\alpha(\cdot)$ (see Eq. (3.8)) for the five classifiers into examination. We used this function as a performance indicator because it is an excellent indicator of the ability of $\mathcal{M}$ to differentiate the relevance of features. They are shown in Table 11. The analysis of this table gives us an accurate quantitative result of what we had already glimpsed qualitatively in Fig. 4. In particular, it highlights that the best classifier is Polynomial SVM, with a value of $\alpha(\cdot)$ equal to 37.47%, while the second best classifier is Radial SVM, with a value of $\alpha(\cdot)$ equal to 17.62%.

5.2.2. Mammographic mass dataset

Also for this dataset, we initially performed the computation of the classifier accuracy. The results obtained are shown in Table 12. As can be seen from this table, the accuracy values are lower than in the previous case. We are thus in presence of a completely different (albeit equally interesting to investigate) scenario than the one we have seen for the Iris dataset.

Then we computed the distributions of the values of the damping factor for the various classifiers and features involved. The results obtained are shown in Fig. 5. In this case, we can observe that:

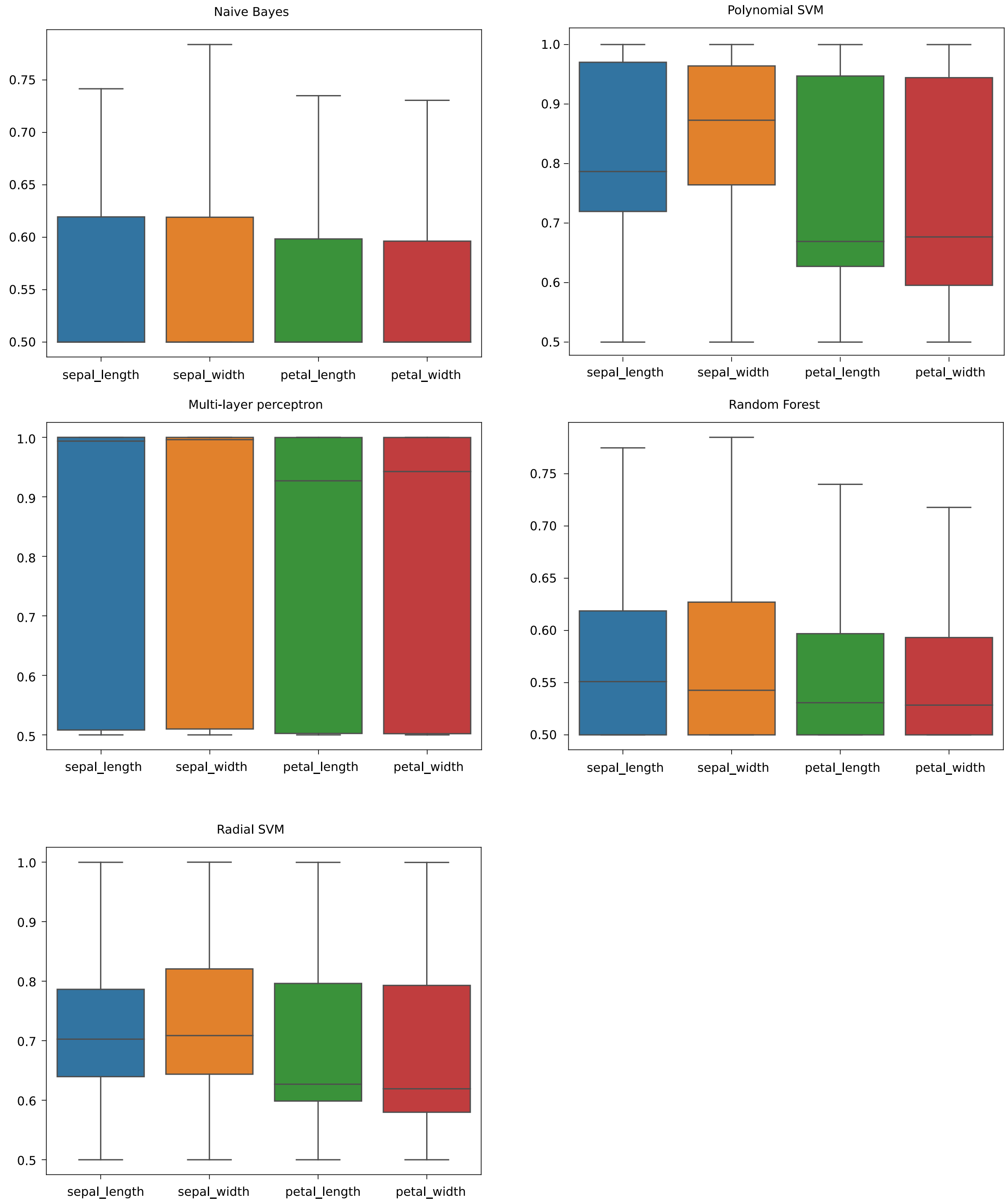

Fig. 3. Distribution of the values of the damping factor for the Iris dataset.

- We have very high values of the damping factor for all the classifiers. In fact, almost always the median is very close to 1.
- Naive Bayes, Random Forest and Radial SVM allow for very varied values of the damping factor among occurrences of the same feature. Such variety is much smaller for Polynomial SVM and Multi-Layer Perceptron.

The examination of the damping factor would seem to suggest that, in this case, there are no major differences either between classifiers or features. However, as in the previous case, the damping factor analysis is only preliminary. In fact, in order to give a definitive answer, it is necessary to consider the relevance of features. In Fig. 6, we report the distributions of the values of feature relevances for the various classifiers and features. From the examination of this figure we can see that:

- There is no substantial difference between the relevance of the various features in any of the classifiers considered.
- Only Naive Bayes is able to identify a slightly greater relevance for the features `shape` and `margin` than for the features `age` and `density`.

In this case, only one classifier (namely, Naive Bayes) indicates a difference between the various features with regard to relevance values,

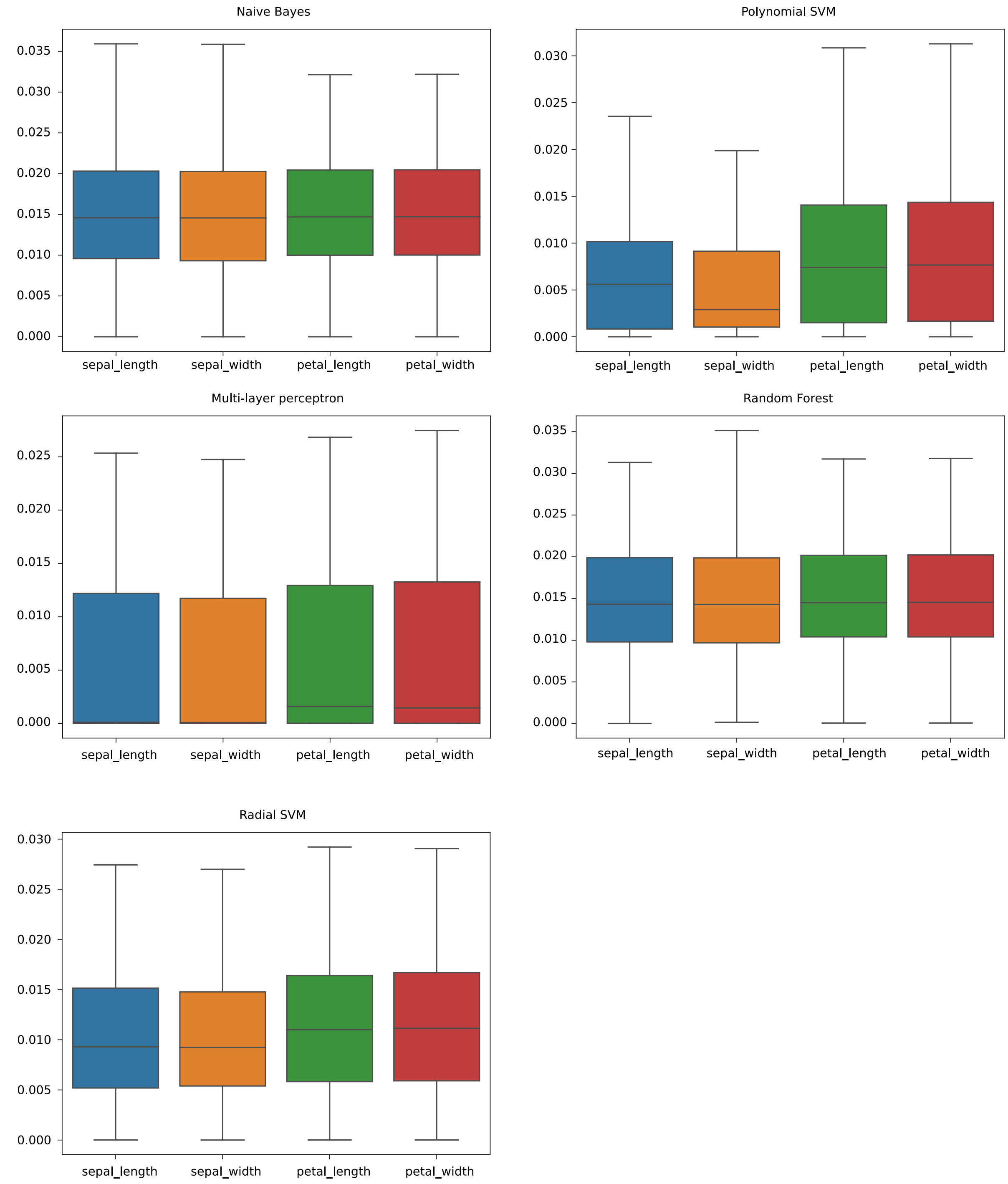

Fig. 4. Distribution of the relevance values for the Iris dataset.

while all the others denote a substantial equality. Also in this case, we do not have contradictions because there is no classifier indicating a higher relevance of one or more features, which are less relevant than features for another classifier. However, a quantitative analysis is still needed to see whether the differences captured qualitatively by Naive Bayes are significant or negligible. In Table 13, we report the mean values of the occurrence relevances for each feature and for each classifier. Instead, in Table 14, we report the values of the function $\alpha(\cdot)$ for the five classifiers into consideration.

From the analysis of these two tables we can deduce that the difference between the relevance values of `shape` and `margin` on the one hand, and `age` and `density` on the other hand, as identified by Naive Bayes, is not negligible. As evidence of this, the value of $\alpha(\cdot)$ relative to Naive Bayes is high, not far from the maximum value of $\alpha(\cdot)$ we had found for the Iris dataset. However, for the Mammographic Mass dataset, all classifiers except Naive Bayes lead $\alpha(\cdot)$ to return very low values. In contrast, for the Iris dataset, there were at least two other classifiers, besides the one associated with the maximum value of $\alpha(\cdot)$, that lead this function to return significant values. As a consequence, we can say that, also at the quantitative level, there is a difference between the relevances of the features in the Mammographic Mass dataset, but this is more attenuated than for Iris.

Actually, we had already realized from the values of accuracy and damping factor that Mammographic Mass represented a very different

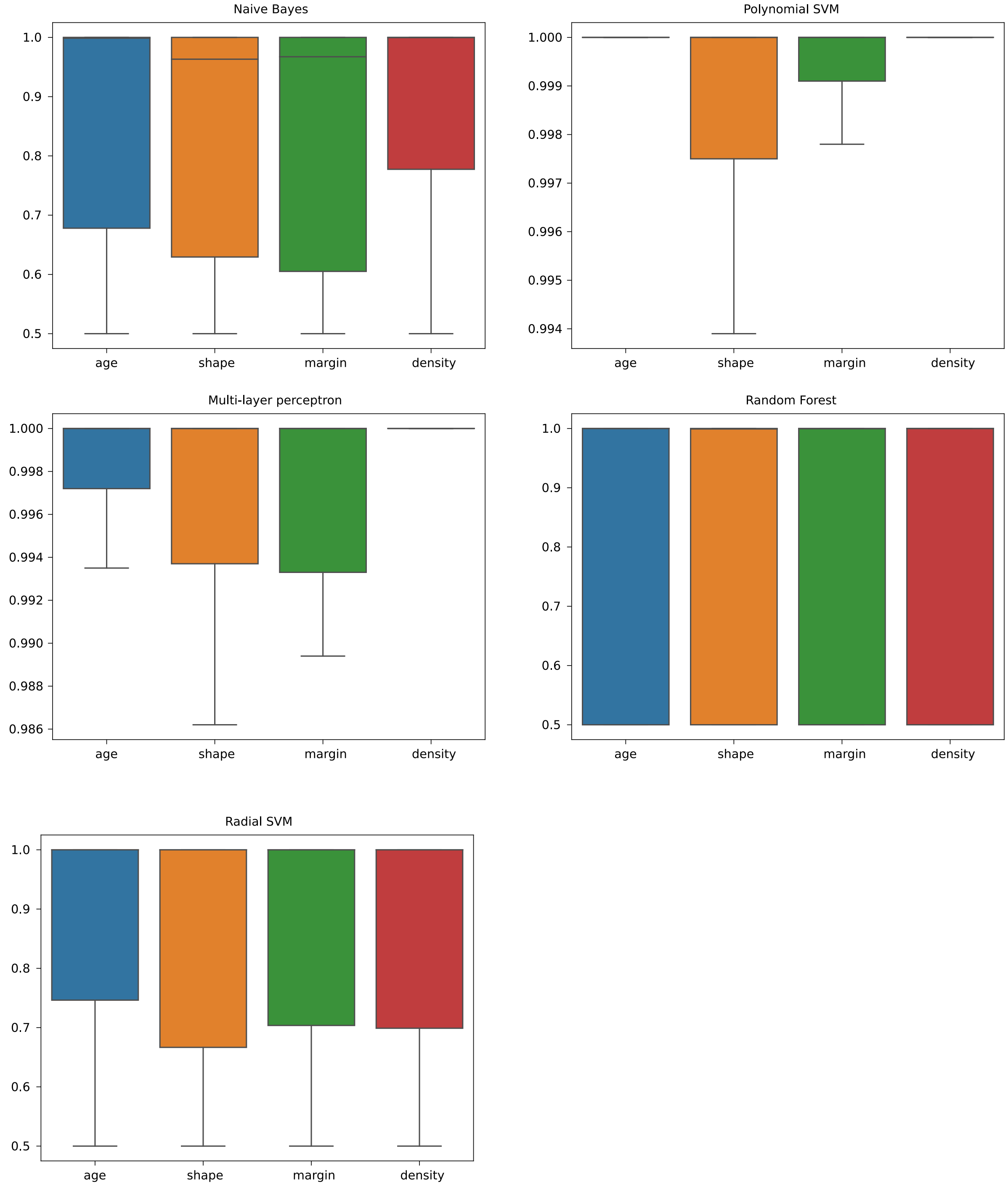

Fig. 5. Distribution of the values of the damping factor for the Mammographic Mass dataset.

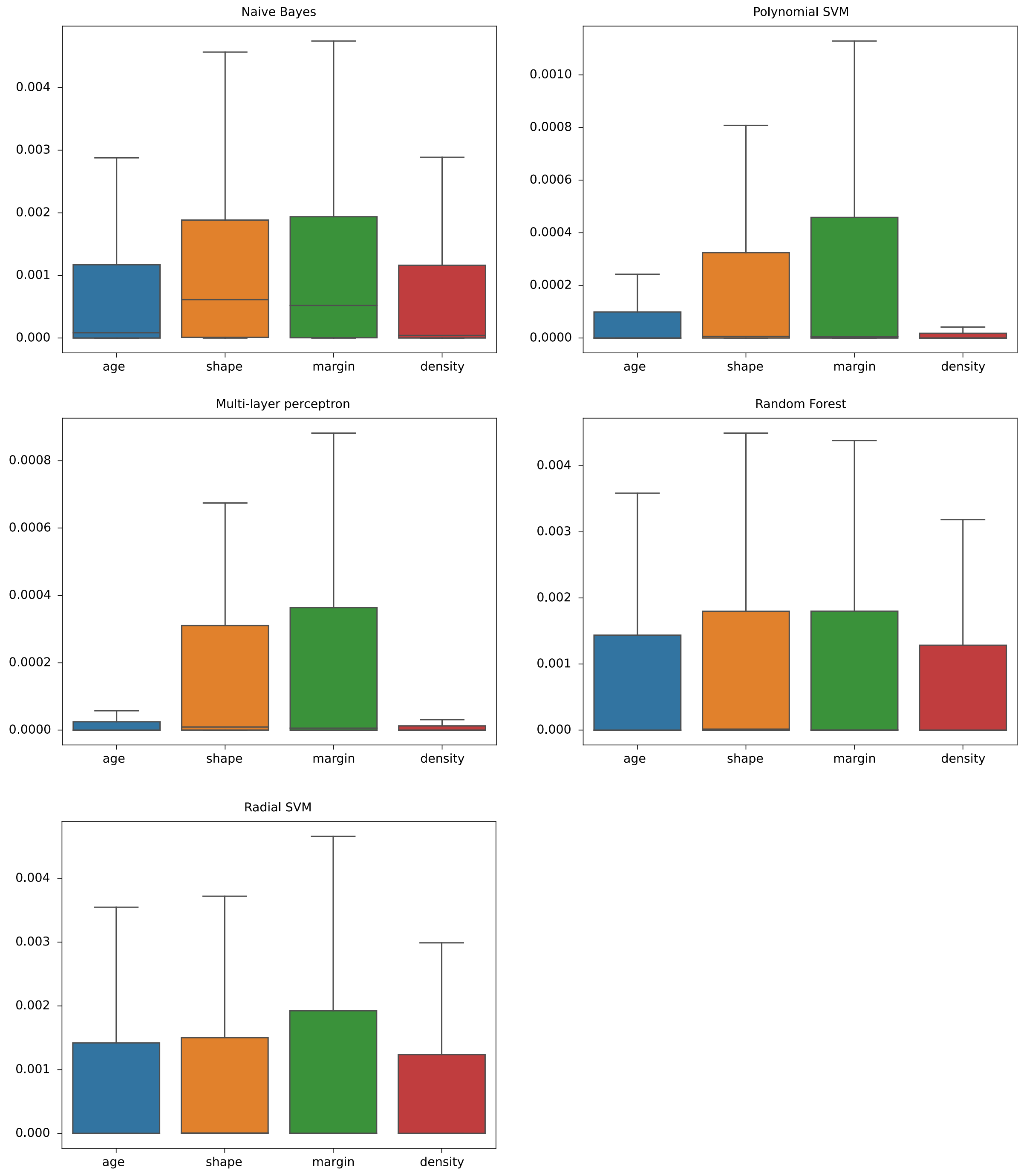

Fig. 6. Distribution of the relevance values for the Mammographic Mass dataset.

scenario than Iris. In this regard, a further interesting observation concerns the fact that the classifiers that perform better are different in different scenarios. This tells us that none of the classifiers we have chosen in our test campaign is useless; on the contrary, they, as a whole, provide a range of classifiers serving as underlying engines to our approach. It is exactly because of the variety of this range that our approach is able to return satisfactory results even in heterogeneous scenarios.

5.3. Sensitivity computation

5.3.1. Iris dataset

The purpose of this experiment is to evaluate the sensitivity of our approach on the Iris dataset. The reasons for adopting this performance indicator lie precisely in the role that sensitivity plays as a measure of the effects of changes in input data on our approach. This issue was discussed in detail in Section 3.4, where we introduced this parameter. Initially, we randomly selected 50% of the instances for building the starting network and coefficients (we call this task “first phase” in the following) and left the remaining 50% of them for the sensitivity computation (we call this task “second phase” in the following). At the end of the first phase, we computed the relevance of each feature. Next, we gave as input to our framework (and, therefore, also to the classifier within it) 10% (resp., 20%, 30%, 40%, 50%) of new instances from the remaining ones. These instances were randomly selected. We call $\tilde{\Delta}_r^{10}$ (resp., $\tilde{\Delta}_r^{20}$, $\tilde{\Delta}_r^{30}$, $\tilde{\Delta}_r^{40}$, $\tilde{\Delta}_r^{50}$) the corresponding values of $\tilde{\Delta}(F_k, IS_j)$ thus obtained (in this case, IS_j represents the set of instances added in the various cases). In Fig. 7, we report the results obtained for the

Table 13
Median relevance of each feature returned by the five classifiers for the Mammographic Mass dataset.

Model	Feature	Relevance
Naive Bayes (Zhang, 2004)	age	0.000085
	shape	0.000613
	margin	0.000521
	density	0.000042
SVM with polynomial kernel (Chang & Lin, 2011)	age	$2.402922 \cdot 10^{-9}$
	shape	$6.369760 \cdot 10^{-6}$
	margin	$4.756909 \cdot 10^{-6}$
	density	0
SVM with radial basis function (Chang & Lin, 2011)	age	$4.805844 \cdot 10^{-9}$
	shape	$7.030652 \cdot 10^{-6}$
	margin	$2.917475 \cdot 10^{-6}$
	density	$4.805844 \cdot 10^{-9}$
Multi-Layer Perceptron (He et al., 2015)	age	0
	shape	$9.2807971 \cdot 10^{-6}$
	margin	$5.727522 \cdot 10^{-6}$
	density	$5.848218 \cdot 10^{-8}$
Random Forest (Breiman, 2001)	age	$2.402922 \cdot 10^{-9}$
	shape	$1.542896 \cdot 10^{-5}$
	margin	$2.776558 \cdot 10^{-6}$
	density	$2.402922 \cdot 10^{-9}$

Table 14
Values of the function $\alpha(\cdot)$ for the classifiers into consideration and for the Mammographic Mass dataset.

Model	Value of $\alpha(\cdot)$
Naive Bayes (Zhang, 2004)	29.53%
SVM with polynomial kernel (Chang & Lin, 2011)	1.39%
SVM with radial basis function (Chang & Lin, 2011)	0.37%
Multi-Layer Perceptron (He et al., 2015)	2.55%
Random Forest (Breiman, 2001)	0.86%

various classifiers. From the analysis of this figure, we can see that our framework is very resilient regardless of the classifier used. In fact, for any feature, the relative variation in relevance is very low, even when the number of features introduced during the second phase is high.

In the previous test, the gradually inserted instances were randomly selected. In order to evaluate our framework even in presence of radical variations in a given direction, we decided to repeat the previous experiment again by giving 10% (resp., 20%, 30%, 40%, 50%) of new instances in input to the classifier. These instances were taken from the ones left for the second phase. However, instead of selecting them randomly, we chose them all belonging to the same class. When the testing instances were not sufficient to do this, we derived the necessary instances using the bootstrap technique (Bruce, Bruce, & Gedeck, 2020). We call $\tilde{\Delta}_0^{10}$ (resp., $\tilde{\Delta}_0^{20}$, $\tilde{\Delta}_0^{30}$, $\tilde{\Delta}_0^{40}$, $\tilde{\Delta}_0^{50}$) the values of $\tilde{\Delta}(F_k, IS_j)$ obtained when the added instances all belong to the class 0. Similarly, we can proceed with classes 1 and 2. These values are shown in Figs. 8, 9, and 10, respectively.

The analysis of these figures is extremely interesting. In fact, in each of these cases, the inserted instances simulate a change in the reference scenario in a specific direction. In this case, it was desirable for our framework not to be resilient but to be flexible enough to capture the change and adapt its behavior accordingly. And, indeed, the values of the relevance variation obtained in all these figures, for all features and for all classifiers, confirm that our framework exhibits the desired behavior.

5.3.2. Mammographic mass dataset

In this section, we repeat the previous experiment on Mammographic Mass. The way we conducted it is exactly the same as the one adopted for Iris. The only difference is that Mammographic Mass involves two classes and not three. In Fig. 11, we report the values of $\tilde{\Delta}_r^{10}$ $\tilde{\Delta}_r^{20}$, $\tilde{\Delta}_r^{30}$, $\tilde{\Delta}_r^{40}$ and $\tilde{\Delta}_r^{50}$ obtained in this experiment.

Instead, in Figs. 12 and 13 we show the values we obtained for $\tilde{\Delta}_i^{10}$ $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \leq i \leq 1$.

The analysis of Figs. 11, 12, and 13 confirms what we have already seen for Iris, namely that, again, our framework is extremely resilient to the presence of noise and outliers. However, at the same time, it is flexible and capable of adjusting its behavior in presence of significant and structural changes of the reference scenario towards a given direction.

5.4. Additional experiments on real-world case studies

We extended our experiments to real-world case studies, aiming to further validate the practical applicability of our framework. We leveraged publicly available datasets to simulate real-world scenarios and apply our framework to derive insights. In addition, for each case study, we compared the results obtained by our framework with those returned by SHAP (Lundberg & Lee, 2017), which, as we saw in Section 2, is a XAI approach widely used in the literature. We adopted the implementation of SHAP provided by the `shap` Python library.[2] For each case study, we applied SHAP on the corresponding dataset and, for each feature, we computed the average of the absolute values returned by SHAP for it when applied to all the instances of the dataset and we refer to them as SHAP values. Each average returned by SHAP can range from the minimum value of 0, indicating that the feature has no effect on the model's output (on average), to a maximum value depending on the data and the model itself, indicating a strong impact of the feature on the model's output.

5.4.1. Healthcare diagnosis

In this case study, we applied our framework to a dataset, derived from Chicco and Jurman (2020), which includes clinical records related to 12 key features pivotal in predicting heart failure. They are: *(i)* `age`: it indicates the age of the patient, given in years; *(ii)* `anaemia`: it is a binary feature indicating a decrease in red blood cells or hemoglobin; *(iii)* `creatinine_ phosphokinase`: it represents the level of the CPK enzyme in the blood, measured in mcg/L; *(iv)* `diabetes`: it is a binary feature indicating whether the patient has diabetes; *(v)* `ejection_fraction`: it denotes the percentage of blood leaving the heart during each contraction; *(vi)* `high_blood_pressure`: it is a binary feature indicating the presence of hypertension in the patient; *(vii)* `platelets`: it quantifies the platelets in the blood, given in kiloplatelets/mL; *(viii)* `sex`: it is a binary feature distinguishing between female and male patients; *(ix)* `serum_creatinine`: it indicates the level of serum creatinine in the blood, measured in mg/dL; *(x)* `serum_sodium`: it represents the level of serum sodium in the blood, given in mEq/L; *(xi)* `smoking`: it is a binary feature indicating whether the patient smokes; *(xii)* `time`: it specifies the follow-up period, measured in days. The target feature is `death_event`: it is a binary feature indicating whether the patient died during the follow-up period.

We applied our framework on this dataset using XGBoost as classifier. The framework computed both the relevance and sensitivity of features. Due to space limitations, in the following we consider only the relevance. Table 15 shows the median relevance calculated by our approach for each feature. Then, we applied SHAP using XGBoost as classifier and calculated the average of the absolute SHAP values for each feature. They are reported in Table 16.

From the analysis of Tables 15 and 16, we can see that our framework and SHAP agree that the most important features in determining the death of a patient with hearth diseases are `anaemia`, `high_blood_pressure`, `smoking` and `diabetes`. This is reasonable since indeed cardiac deaths are mostly caused by these four factors.

[2] https://github.com/shap/shap.

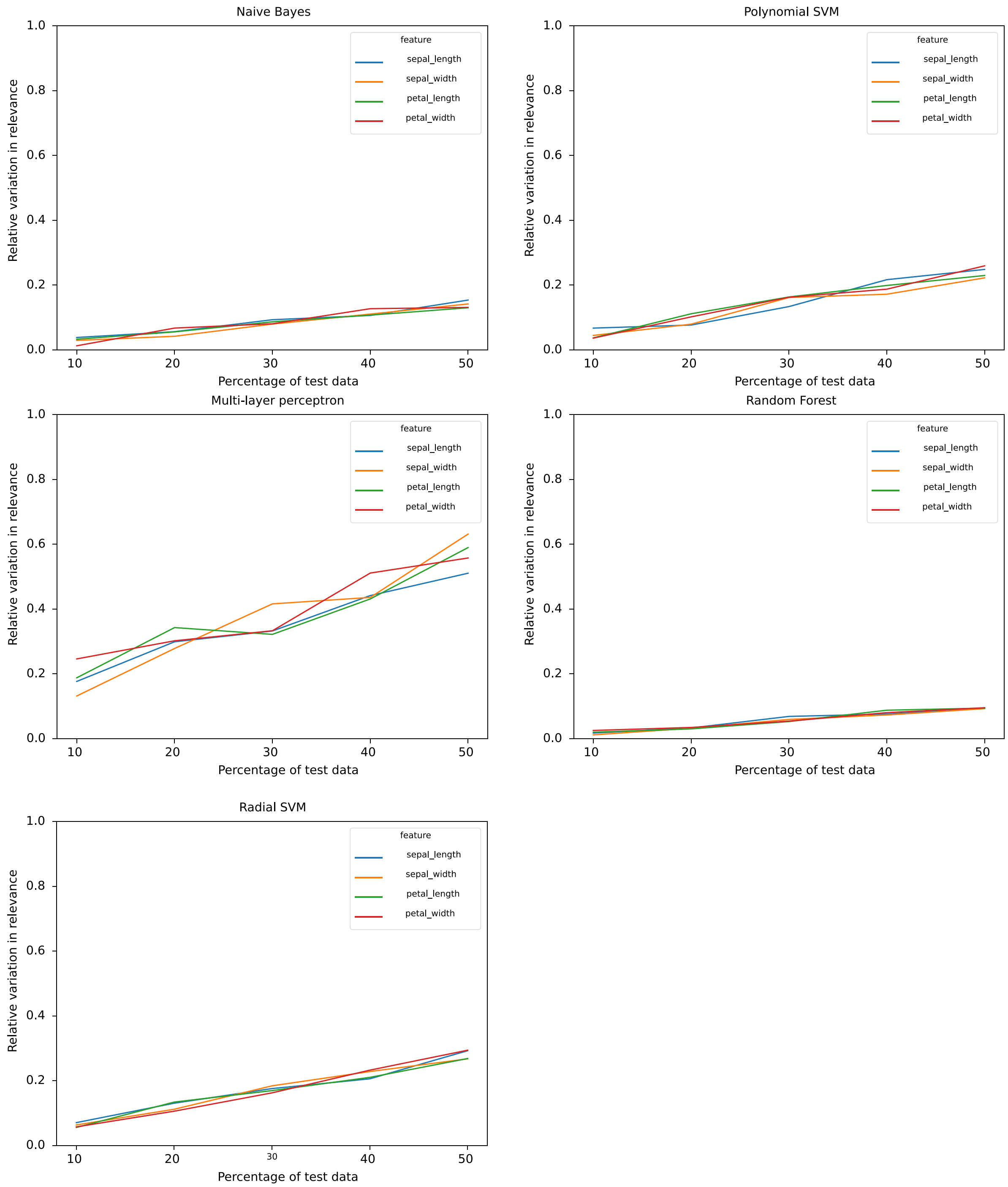

Fig. 7. Values of $\tilde{\Delta}_r^{10}$, $\tilde{\Delta}_r^{20}$, $\tilde{\Delta}_r^{30}$, $\tilde{\Delta}_r^{40}$ and $\tilde{\Delta}_r^{50}$ for the various features and classifiers — Iris dataset.

5.4.2. Bank fraud detection

In this case study, we applied our framework to the Paysim Synthetic Financial Dataset (Lopez-Rojas, Elmir, & Axelsson, 2016), which simulates mobile money transactions based on real transaction samples. The dataset encompasses records related to 10 key features instrumental in predicting fraudulent activities. They are: *(i)* `step`: it represents a unit of real-world time, where 1 step equates to 1 h; *(ii)* `type`: it indicates the transaction type, such as CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER; *(iii)* `amount`: it quantifies the transaction amount in local currency; *(iv)* `name_orig`: it denotes the initiator of the transaction; *(v)* `old_balance_orig`: it specifies the initiator's balance before the transaction; *(vi)* `new_balance_orig`: it specifies the initiator's balance after the transaction; *(vii)* `name_dest`: it denotes the recipient of the transaction; *(vii)* `old_balance_dest`: it specifies the recipient's balance before the transaction; *(ix)* `new_balance_dest`: it specifies the recipient's balance after the transaction; *(x)* `is_flagged_fraud`: it flags massive inter-account transfers that exceed a specified threshold. The target feature is `is_fraud`: it is a binary feature marking transactions executed by fraudulent agents.

We applied both our framework and SHAP on this dataset using GBM as classifier. The results obtained are reported in Tables 17 and 18, respectively.

From the analysis of Tables 17 and 18, we can see that our framework and SHAP agree that the most important features in determining a

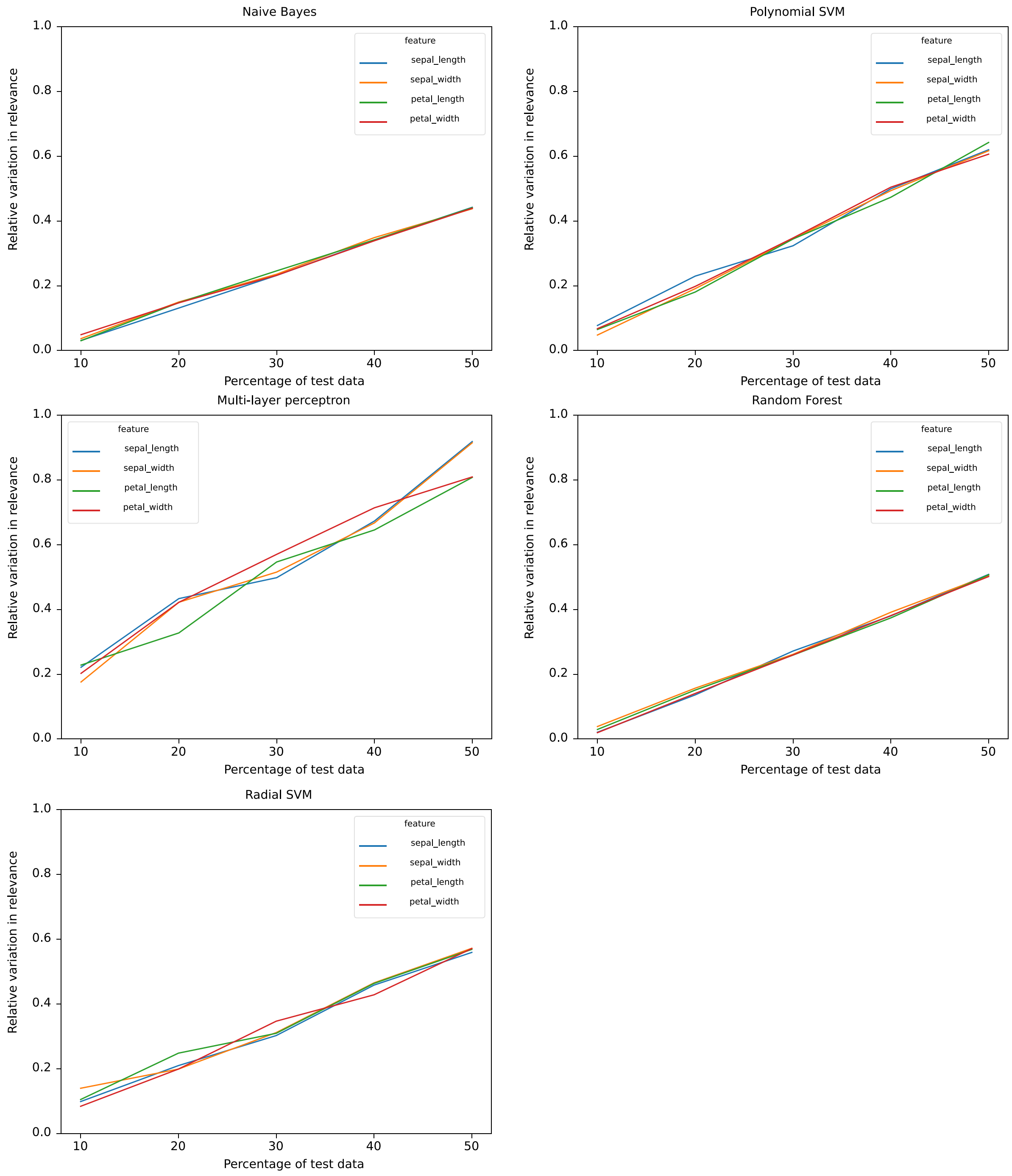

Fig. 8. Values of $\tilde{\Delta}_i^{10}$, $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \le i \le 2$, for the various features and classifiers — Iris dataset, class 0.

fraudulent activity are `amount`, `name_dest` and `is_flagged_fraud`, which is reasonable for a human expert.

5.4.3. *Classifying restaurants based on recommendations on yelp*

In this case study we applied our framework to a dataset of reviews on Yelp.[3] After some ETL operations on the dataset, we obtained 9 key features instrumental in predicting the star rating of the restaurant. They are: *(i)* `review_id`: it is a unique identifier for each review; *(ii)* `user_id`: it is an identifier for the user providing the review; *(iii)* `business_id`: it is an identifier for the restaurant being reviewed; *(iv)* `business_city`: it is the city where the restaurant is located; *(v)* `business_category`: it denotes the types of cuisines or themes of the restaurant; *(vi)* `text`: it is the textual content of the review; *(vii)* `useful`: it is the number of users who found the review useful; *(viii)* `funny`: it is the number of users who found the review funny; *(ix)* `cool`: it is the number of users who found the review cool. The target feature is `stars`: it represents the star rating given by the user to the restaurant.

We applied our framework and SHAP on this dataset using KNN as classifier. The results obtained are reported in Tables 19 and 20, respectively.

From the analysis of Tables 19 and 20, we can see that our framework and SHAP agree that the most important features in determining

[3] https://www.yelp.com/dataset.

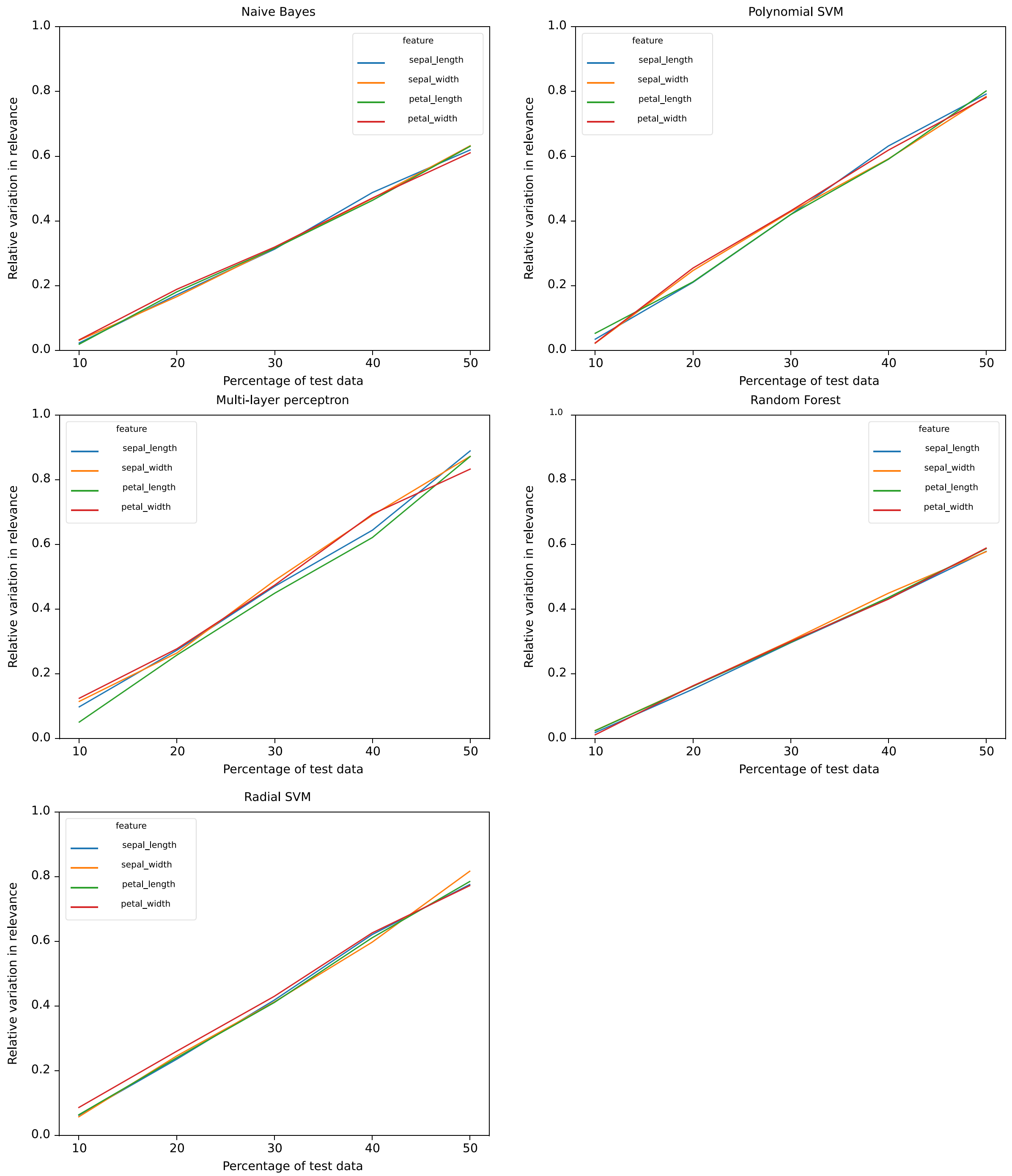

Fig. 9. Values of $\tilde{\Delta}_i^{10}$, $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \leq i \leq 2$, for the various features and classifiers — Iris dataset, class 1.

the star rating of the restaurant are `useful`, `cool` and `user_id`. This result seems reasonable since the success of a review depends primarily on the user who made it and how it was judged by other users.

6. Conclusion

In this paper, we have proposed a model-agnostic, network-based XAI framework to explain the behavior of any classifier. Our framework is based on network theory; therefore, it can benefit from the large amount of results that researchers in this area have found in the past. We have seen that our framework reaches its goal by evaluating the relevance of features in the behavior of a classifier. We have also completed our framework with a set of quantitative indicators aiming to support its sensitivity analysis when it is applied in a given scenario. We have also highlighted the similarities and differences between our approach and related ones already proposed in the literature. Finally, we have illustrated an experimental campaign to assess the adequacy of our framework.

The main contributions of this paper are as follows: *(i)* we propose a new model-agnostic, network-based XAI framework for classifiers; *(ii)* we present a new measure, called dyscrasia, which evaluates the consistency of the occurrences of a feature in supporting the classification of the corresponding instances; *(iii)* we define a new approach to compute the relevance of a feature in classifying the corresponding instances; *(iv)* we introduce some quantitative indicators to support the sensitivity analysis of our approach.

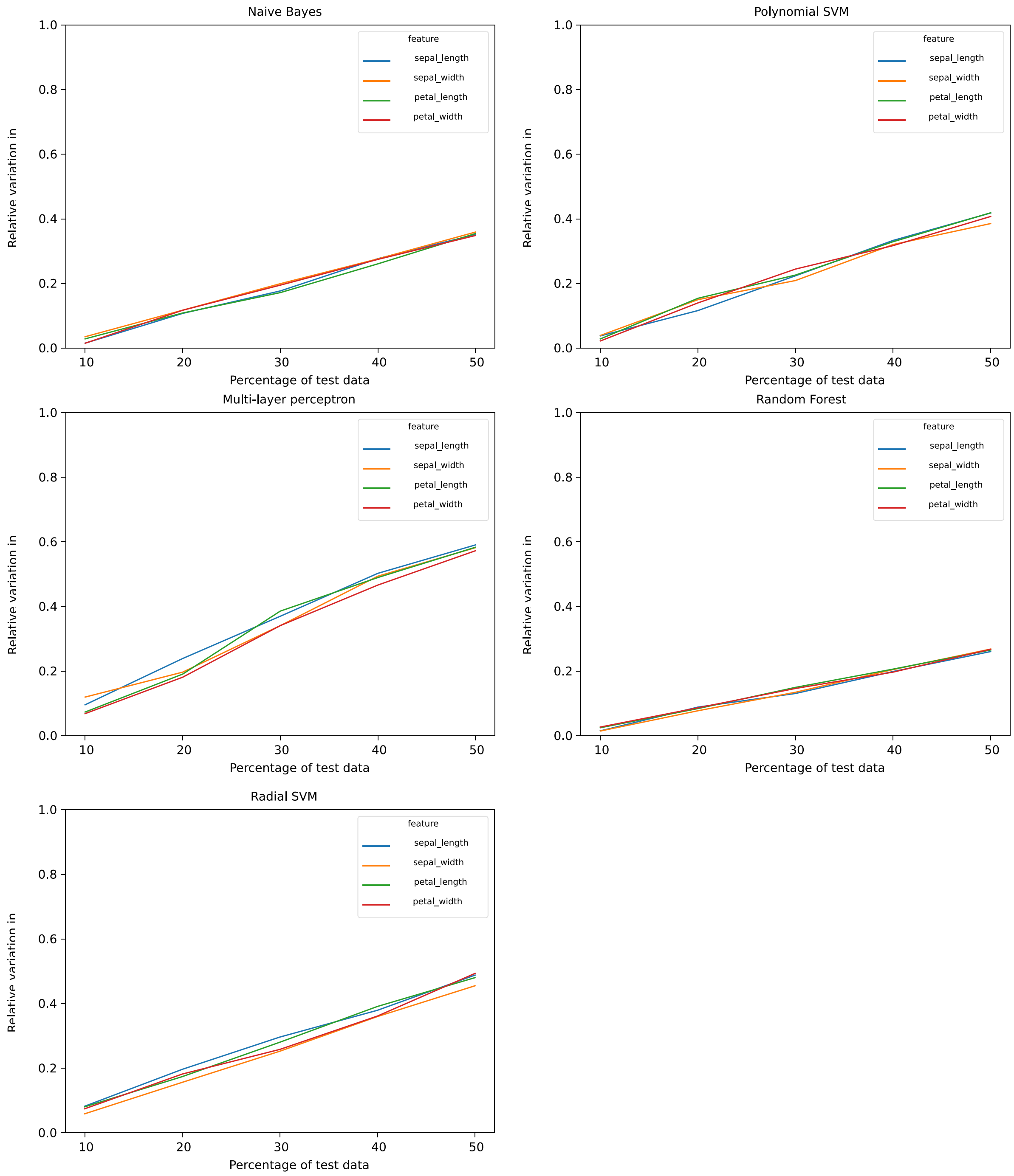

Fig. 10. Values of $\tilde{\Delta}_i^{10}$, $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \leq i \leq 2$, for the various features and classifiers — Iris dataset, class 2.

This paper should not be considered an ending point but rather a starting point for further research. In fact, it is possible to think of several developments of the research described here. First, we might consider latent structural properties. Indeed, the current framework operates as a solid foundation by considering instances as nodes in a network with the direction of the arcs capturing the confidence level of the classifier decisions. This modeling represents a macroscopic view of the classifier behavior. Analysis of latent structural properties would allow us to gain deeper insights. For instance, one might consider studying subnetworks, node centrality, and node influence. In this regard, some subnetworks might co-occur frequently or exhibit unique behaviors; in either case, the study of subnetworks might lead to the discovery of insights concerning classifier behavior. For example, the presence of certain sets of instances that consistently cluster together in subnetworks might lead to think about inherent biases or strong correlations of those instances. In addition, by examining the values of centrality measures in nodes we could identify which instances play a pivotal role in the overall behavior of the classifier. For example, instances with anomalous values of one or more features or instances carrying critical information for classification could be analyzed as potential hubs in the network.

Furthermore, it would be interesting to consider a totally different network-based model, such as one employing multilayer networks (Bonifazi, Breve, Cirillo, Corradini, & Virgili, 2022; Newman, 2018), to provide a new point of view and capture different properties than we did with the framework proposed in this paper. For example,

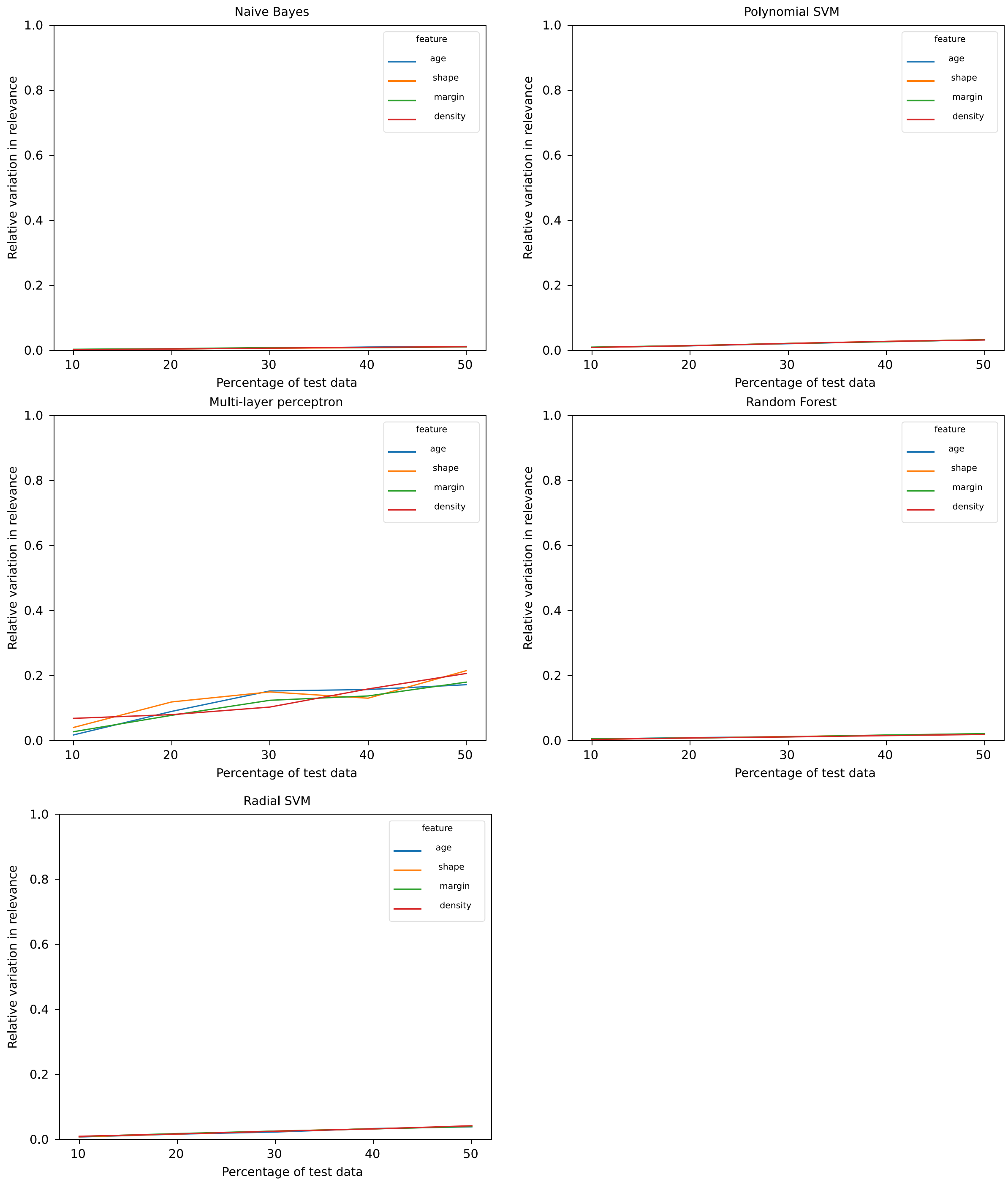

Fig. 11. Values of $\tilde{\Delta}_r^{10}$, $\tilde{\Delta}_r^{20}$, $\tilde{\Delta}_r^{30}$, $\tilde{\Delta}_r^{40}$ and $\tilde{\Delta}_r^{50}$ for the various features and classifiers — Mammographic Mass dataset.

each layer in the multilayer network could represent a subset of data features. This could allow our framework to analyze how different feature sets influence the classifier's decision together or alone. In addition, we would like to test the contribution of our framework in a responsible AI context by considering aspects like fairness, ethics, privacy and accountability. For instance, an extension to identify features that potentially lead to biased or unfair decisions could be designed. Recognizing these features can help in developing methods to either balance their influence or mitigate their impact. Last but not least, we would like to extend our approach so that it can work in Federated Learning scenarios (Barredo Arrieta et al., 2020) using local model knowledge. In particular, we might consider our framework to generate explanations locally, that is, on each decentralized device or server in the Federated Learning setup. These local explanations can be then aggregated in a privacy-preserving way to generate a global explanation that reflects the collective insights, which maintains a holistic view of the classifier while resorting to local data privacy.

Another particularly interesting research problem to be studied in the future involves the investigation of the interactions between our model-agnostic framework and neurosymbolic models (Garcez & Lamb, 2023). In this scenario, our framework can be adopted not only to interpret the "black-box" aspects of classifiers but also to decode the symbolic and logical rules employed in a neurosymbolic system. The measure of dyscrasia could be extended to assess the consistency of these symbolic rules across instances. At the same time, the network-based representation could be enriched by explicitly capturing symbolic

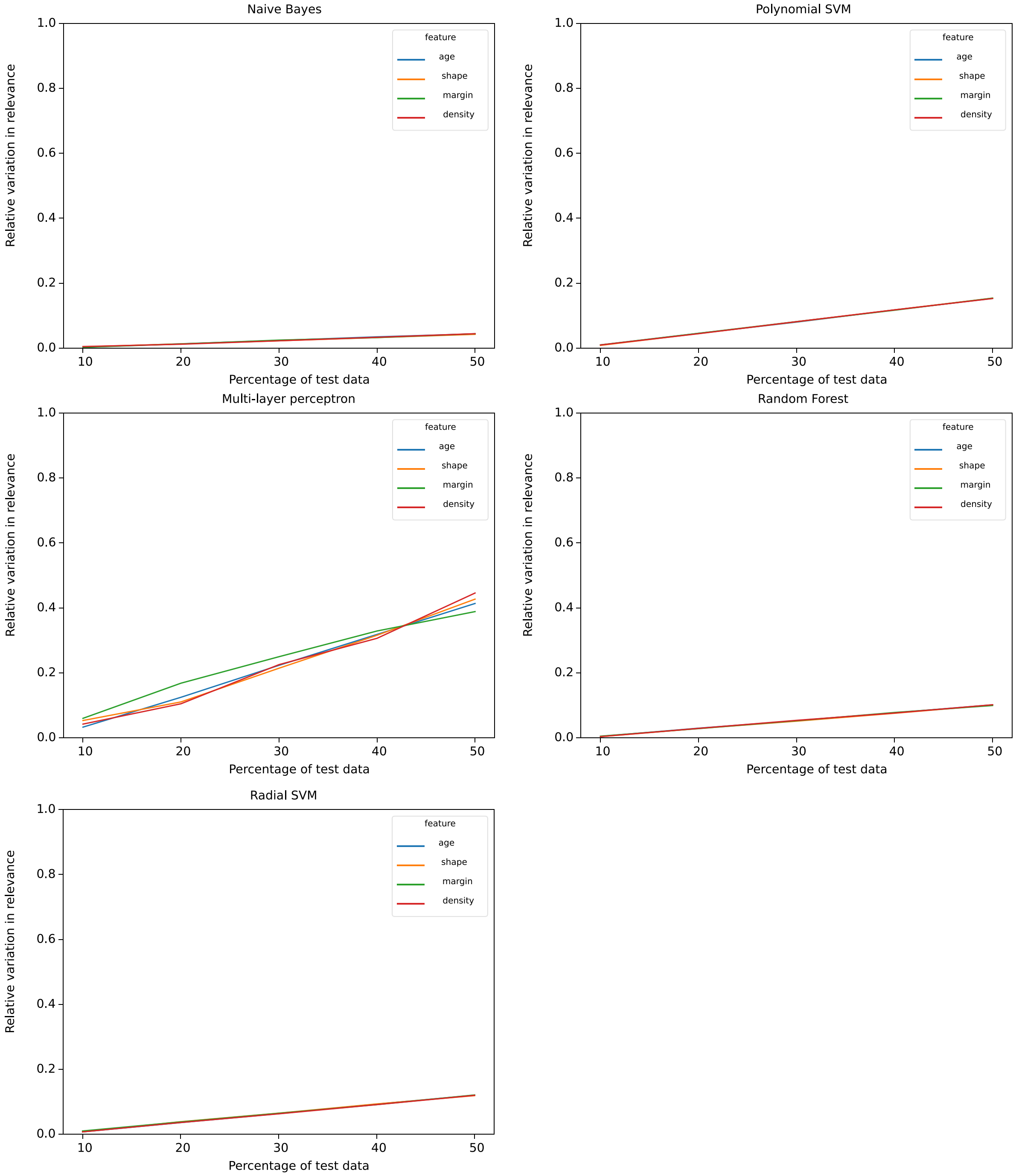

Fig. 12. Values of $\tilde{\Delta}_i^{10}$, $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \leq i \leq 1$, for the various features and classifiers — Mammographic Mass dataset, class 0.

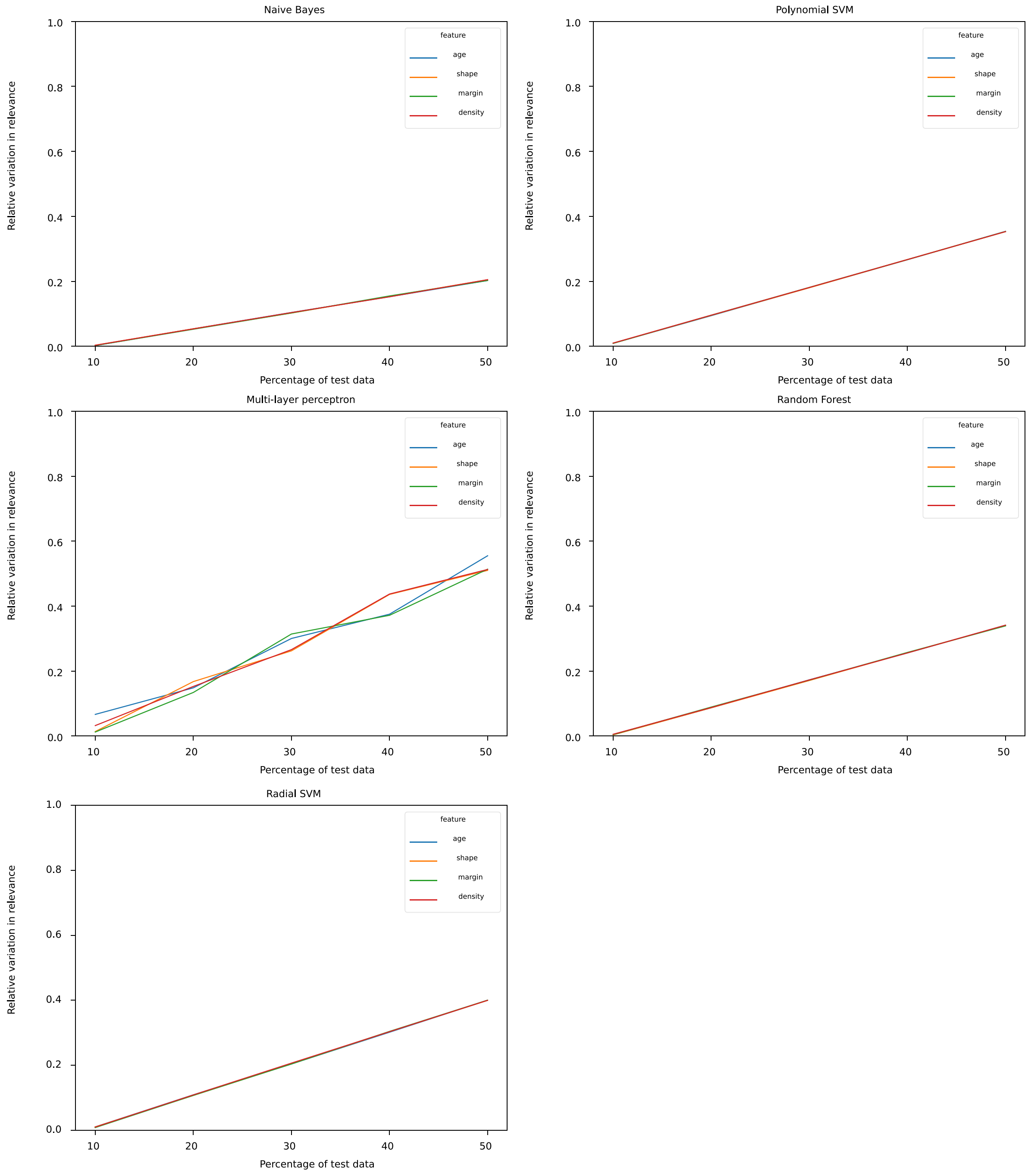

Fig. 13. Values of $\tilde{\Delta}_i^{10}$, $\tilde{\Delta}_i^{20}$, $\tilde{\Delta}_i^{30}$, $\tilde{\Delta}_i^{40}$ and $\tilde{\Delta}_i^{50}$, $0 \le i \le 1$, for the various features and classifiers — Mammographic Mass dataset, class 1.

Table 15
Median relevance of each feature returned by our framework for healthcare diagnosis.

Feature	Relevance
age	0.006332
anaemia	0.006757
creatine_phosphokinase	0.006110
diabetes	0.006677
ejection_fraction	0.006294
high_blood_pressure	0.006708
platelets	0.006147
serum_creatine	0.006147
serum_sodium	0.006245
sex	0.006549
smoking	0.006694
time	0.006594

Table 16
Average of the absolute SHAP values for each feature returned by SHAP for healthcare diagnosis.

Feature	Mean of absolute SHAP values
age	0.2387
anaemia	1.7350
creatine_phosphokinase	0.0607
diabetes	0.6396
ejection_fraction	0
high_blood_pressure	0.6905
platelets	0.4374
serum_creatine	0.3960
serum_sodium	0
sex	0
smoking	0.6128
time	0

Table 17
Median relevance of each feature returned by our framework for bank fraud detection.

Feature	Relevance
step	0.00005492
type	0.00005648
amount	0.00005732
name_orig	0.00005489
old_balance_orig	0.00005478
new_balance_orig	0.00005479
name_dest	0.00005745
old_balance_dest	0.00005687
new_balance_dest	0.00005645
is_flagged_fraud	0.00005685

Table 18
Average of the absolute SHAP values for each feature returned by SHAP for bank fraud detection.

Feature	Mean of absolute SHAP values
step	0.2542
type	0.8254
amount	0.8563
name_orig	0.0974
old_balance_orig	0
new_balance_orig	0
name_dest	1.8902
old_balance_dest	0.7465
new_balance_dest	0.5685
is_flagged_fraud	0.7842

Table 19
Median relevance of each feature returned by our framework when applied on our Yelp dataset.

Feature	Relevance
review_id	0.0007523
user_id	0.0007690
business_id	0.0007301
business_city	0.0007538
text	0.0007612
useful	0.0007877
funny	0.0007634
cool	0.0007756

Table 20
Average of the absolute SHAP values for each feature returned by SHAP when applied on our Yelp dataset.

Feature	Mean of absolute SHAP values
review_id	0.0251
user_id	0.7453
business_id	0
business_city	0.4893
text	0.6842
useful	1.7523
funny	0.7358
cool	0.8569

relationships. This would result in networks whose nodes could represent not only instances, but also symbolic rules, and whose arcs could model logical dependencies. Finally, an additional challenging context that we believe is worthy of future investigation concerns the exploration of the temporal dynamics of black-box classifiers. In this context, time-series analysis could be incorporated into our framework to allow us to study how feature relevance and dyscrasia evolve over time, especially for classifiers dealing with dynamic data sources. Understanding the temporal dynamics can provide insights into how classifiers adapt and change their decisions over time, thus allowing for a deeper understanding of their behavior.

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.

Data availability

Data will be made available on request.

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, *6*, 52138–52160, IEEE.

Ahern, I., Noack, A., Guzman-Nateras, L., Dou, D., Li, B., & Huan, J. (2019). NormLIME: A new feature importance metric for explaining deep neural networks. http://dx.doi.org/10.48550/arXiv.1909.04200, arXiv preprint arXiv:1909.04200.

Ahsan, M., Mahmud, M., Saha, P., Gupta, K., & Siddique, Z. (2021). Effect of data scaling methods on machine learning algorithms and model performance. *Technologies*, *9*(3), 52, MDPI.

Akhiat, Y., Asnaoui, Y., Chahhou, M., & Zinedine, A. (2021). A new graph feature selection approach. In *Proc. of the international IEEE congress on information science and technology (CIST'20)* (pp. 156–161). Agadit - Essaouira, Morocco: IEEE.

Asuncion, A., & Newman, D. (2007). *UCI machine learning repository*. Irvine, CA, USA: available online at: https://archive.ics.uci.edu/ml/index.php.

Banerjee, P., & Barnwal, R. (2023). Methods and metrics for explaining artificial intelligence models: A review. In *Explainable AI: Foundations, methodologies and applications* (pp. 61–88). Springer.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., et al. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115, Elsevier.

Bonifazi, G., Breve, B., Cirillo, S., Corradini, E., & Virgili, L. (2022). Investigating the COVID-19 vaccine discussions on Twitter through a multilayer network-based approach. *Information Processing & Management*, *59*(6), Article 103095, Elsevier.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32, Springer.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN systems*, *30*(1–7), 107–117.

Bruce, P., Bruce, A., & Gedeck, P. (2020). *Practical statistics for data scientist* (2nd ed.). Sebastopol, CA, USA: O'Reilly.

Burkart, N., & Huber, M. (2021). A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, *70*, 245–317, AI Access Foundation.

Camacho, D., Panizo-LLedot, A., Bello-Orgaz, G., Gonzalez-Pardo, A., & Cambria, E. (2020). The four dimensions of social network analysis: An overview of research methods, applications, and software tools. *Information Fusion*, *63*, 88–120, Elsevier.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(3), 1–27, ACM.

Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, *20*(16), 1–16, BioMed Central.

Chinu, J., & Bansal, U. (2022). Explainable AI: To reveal the logic of black-box models. *New Generation Computing*, 1–35. http://dx.doi.org/10.1007/s00354-022-00201-2, Springer.

Chung, F. (2014). A brief survey of PageRank algorithms. *IEEE Transactions on Network Science and Engineering*, *1*(1), 38–42, IEEE.

Cormen, T., Leiserson, C., Rivest, R., & Stein, C. (2001). *Introduction to algorithms*. The MIT Press.

Dabkowski, P., & Gal, Y. (2017). Real time image saliency for black box classifiers. In *Proc. of the international conference on neural information processing systems (NIPS'17)* (pp. 6970–6979). Long Beach, CA, USA: Curran Associates Inc..

Datta, A., Sen, S., & Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *Proc. of the international symposium on security and privacy (SP'16)* (pp. 598–617). Fairmont, San Jose, CA, USA: IEEE.

Di Vaio, A., Palladino, R., Hassan, R., & Escobar, O. (2020). Artificial intelligence and business models in the sustainable development goals perspective: A systematic literature review. *Journal of Business Research*, *121*, 283–314, Elsevier.

Dong, S., Wang, P., & Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, *40*, Article 100379, Elsevier.

Elter, M., Schulz-Wendtland, R., & Wittenberg, T. (2007). The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical Physics*, *34*(11), 4164–4172, Wiley Online Library.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188, Wiley Online Library.

Fong, R., & Vedaldi, A. (2017). Interpretable explanations of black boxes by meaningful perturbation. In *Proc. of the international IEEE conference on computer vision (ICCV'17)* (pp. 3449–3457). Venice, Italy: IEEE.

Garcez, A., & Lamb, L. (2023). Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, *56*, 12387–12406, Springer.

Gosak, M., Markovič, R., Dolenšek, J., Rupnik, M., Marhl, M., Stožer, A., et al. (2018). Network science of biological systems at different scales: A review. *Physics of Life Reviews*, *24*, 118–135, Elsevier.

Gunning, D., & Aha, D. (2019). DARPA's explainable artificial intelligence (XAI) program. *AI Magazine*, *40*(2), 44–58, AAAI.

Han, J., Kamber, M., & Pei, J. (2011). *Data mining: concepts and techniques - third edition*. Morgan Kaufmann notes.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. of the IEEE international conference on computer vision (ICCV 2015)* (pp. 1026–1034). Santiago, Chile: IEEE.

Henelius, A., Puolamäki, K., Boström, H., Asker, L., & Papapetrou, P. (2014). A peek into the black box: exploring classifiers by randomization. *Data Mining and Knowledge Discovery*, *28*(5), 1503–1529, Springer.

Henelius, A., Puolamäki, K., & Ukkonen, A. (2017). Interpreting classifiers through attribute interactions in datasets. http://dx.doi.org/10.48550/arXiv.1707.07576, arXiv preprint arXiv:1707.07576.

Ienco, D., Meo, R., & Botta, M. (2008). Using PageRank in feature selection. In *Proc. of the symposium on advanced database systems (SEBD'08)* (pp. 93–100). Mondello, Palermo, Italy.

Jan, Z., Ahamed, F., Mayer, W., Patel, N., Grossmann, G., Stumptner, M., et al. (2023). Artificial intelligence for industry 4.0: Systematic review of applications, challenges, and opportunities. *Expert Systems with Applications*, *216*, Article 119456, Elsevier.

Kaur, D., Uslu, S., Rittichier, K. J., & Durresi, A. (2022). Trustworthy artificial intelligence: A review. *ACM Computing Surveys*, *55*(2–39), 1–38, ACM.

Kumar, H., & Martin, A. (2023). Artificial emotional intelligence: Conventional and deep learning approach. *Expert Systems with Applications*, *212*, Article 118651, Elsevier.

Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., et al. (2022). Trustworthy AI: From principles to practices. *ACM Computing Surveys*, *55*(9–177), 1–46, ACM.

Lopez-Rojas, E., Elmir, A., & Axelsson, S. (2016). Paysim: A financial mobile money simulator for fraud detection. In *Proc. of the European Modeling and Simulation Symposium (EMSS'16)* (pp. 249–255). Larnaca, Cyprus.

Lundberg, S., & Lee, S. (2017). A unified approach to interpreting model predictions. In *Proc. of the international conference on neural information processing systems (NIPS'17)* (pp. 4768–4777). Long Beach, CA, USA.

Moradi, M., & Samwald, M. (2021). Post-hoc explanation of black-box classifiers using confident itemsets. *Expert Systems with Applications*, *165*, Article 113941, Elsevier.

Nagahisarchoghaei, M., Nur, N., Cummins, L., Nur, N., Karimi, M., Nandanwar, S., et al. (2023). An empirical survey on explainable AI technologies: Recent trends, use-cases, and categories from technical and application perspectives. *Electronics*, *12*(5), 1092, MDPI.

Newman, M. (2018). *Networks*. Oxford University Press.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M., et al. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, *51*(5–9), 1–36, ACM.

Razmjoo, A., Xanthopoulos, P., & Zheng, Q. (2017). Online feature importance ranking based on sensitivity analysis. *Expert Systems with Applications*, *85*, 397–406, Elsevier.

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proc. of the international conference on knowledge discovery and data mining (KDD'16)* (pp. 1135–1144). San Francisco, CA, USA.

Roffo, G., Melzi, S., Castellani, U., & Vinciarelli, A. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *Proc. of the international IEEE conference on computer vision (ICCV'17)* (pp. 1398–1406). Venice, Italy.

Semantic scholar. (2022). (n.d.). https://www.semanticscholar.org, Accessed: 2022-15-12.

Song, X., Song, Y., Stojanovic, V., & Song, S. (2023). Improved dynamic event-triggered security control for T–S fuzzy LPV-PDE systems via pointwise measurements and point control. *International Journal of Fuzzy Systems*, 1–16. http://dx.doi.org/10.1007/s40815-023-01563-5, Springer.

Song, X., Sun, P., Song, S., & Stojanovic, V. (2023). Quantized neural adaptive finite-time preassigned performance control for interconnected nonlinear systems. *Neural Computing and Applications*, *35*, 15429–15446, Springer.

Sporns, O. (2022). Graph theory methods: applications in brain networks. *Dialogues in Clinical Neuroscience*, *20*(2), 111–121, Taylor & Francis.

Strumbelj, E., & Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, *11*, 1–18, JMLR.org.

Štrumbelj, E., Kononenko, I., & Šikonja, M. R. (2009). Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, *68*(10), 886–904, Elsevier.

Sun, P., Song, X., Song, S., & Stojanovic, V. (2022). Composite adaptive finite-time fuzzy control for switched nonlinear systems with preassigned performance. *International Journal of Adaptive Control and Signal Processing*, *37*(3), 771–789, John Wiley & Sons.

Tunyasuvunakool, K., Adler, J., Wu, Z., Green, T., Zielinski, M., Žídek, A., et al. (2021). Highly accurate protein structure prediction for the human proteome. *Nature*, *596*(7873), 590–596, Nature Publishing Group.

Ucer, S., Ozyer, T., & Alhajj, R. (2022). Explainable Artificial Intelligence through graph theory by generalized social network analysis-based classifier. *Scientific Reports*, *12*(1), 15210:1–17, Nature Publishing Group.

Ullah, Z., Al-Turjman, F., Mostarda, L., & Gagliardi, R. (2020). Applications of Artificial Intelligence and machine learning in smart cities. *Computer Communications*, *154*, 313–323, Elsevier.

Wei, G., Zhao, J., Feng, Y., He, A., & Yu, J. (2020). A novel hybrid feature selection method based on dynamic feature importance. *Applied Soft Computing*, *93*, Article 106337, Elsevier.

Yoo, S., & Kang, N. (2021). Explainable Artificial Intelligence for manufacturing cost estimation and machining feature visualization. *Expert Systems with Applications*, *183*, Article 115430, Elsevier.

Yu, K., Beam, A., & Kohane, I. (2018). Artificial intelligence in healthcare. *Nature Biomedical Engineering*, *2*(10), 719–731, Nature Publishing Group.

Zhang, H. (2004). The optimality of Naive Bayes. In *Proc. of the seventeenth international Florida artificial intelligence research society conference (FLAIRS 2004)* (pp. 562–567). Miami Beach, Florida, USA: AAAI Press.

Zini, J., & Awad, M. (2022). On the explainability of natural language processing deep models. *ACM Computing Surveys*, *55*(5–103), 1–31, ACM.