



Truthful meta-explanations for local interpretability of machine learning models

Ioannis Mollas¹ · Nick Bassiliades¹ · Grigorios Tsoumakas¹

Accepted: 4 August 2023
© The Author(s) 2023

Abstract

Automated Machine Learning-based systems' integration into a wide range of tasks has expanded as a result of their performance and speed. Although there are numerous advantages to employing ML-based systems, if they are not interpretable, they should not be used in critical or high-risk applications. To address this issue, researchers and businesses have been focusing on finding ways to improve the explainability of complex ML systems, and several such methods have been developed. Indeed, there are so many developed techniques that it is difficult for practitioners to choose the best among them for their applications, even when using evaluation metrics. As a result, the demand for a selection tool, a meta-explanation technique based on a high-quality evaluation metric, is apparent. In this paper, we present a local meta-explanation technique which builds on top of the *truthfulness* metric, which is a faithfulness-based metric. We demonstrate the effectiveness of both the technique and the metric by concretely defining all the concepts and through experimentation.

Keywords Explainable artificial intelligence · Interpretable machine learning · Local interpretation · Meta-explanations · Evaluation · Argumentation

1 Introduction

Machine Learning (ML), a field of Artificial Intelligence, paves the way for emerging technologies in a wide variety of sectors, leading to technological advancements. ML provides solutions in manufacturing, such as predictive maintenance [1, 2], banking, including credit scoring [3, 4] and risk management [5], insurance, for fraud detection [6] and damage estimation [7], and healthcare, improving the efficiency of care delivery [8] and aiding in the diagnosis or prognosis of numerous diseases [9, 10]. Additionally, ML finds applications in education, specifically in predicting student learning performance [11], and social media, for the detection of online hate speech [12].

Even though ML elevates those sectors, societal and ethical issues may arise in high-risk scenarios. For example, credit score predicting algorithms are discriminating between minority and majority populations, leading minorities to poverty and homelessness [13]. As a result of worries about performance, biases and poor trust, an insurance company pulled the plug on an AI tool that was designed to detect fraud in claims through videos [14]. Reports of inappropriate patient treatments [15], as well as the use of biased risk prediction models [16], have raised concerns in both society and the research community. Thus, legal frameworks and regulations given by many sources, such as the *General Data Protection Regulation* (GDPR) [17] of the EU, the European AI ACT [18], and the *Equal Credit Opportunity Act* of the US¹, aim to establish requirements that every ML-powered system should satisfy.

One of the requirements is explainability, which led to the establishment of the Explainable AI (XAI) area [19, 20]. Interpretable ML (IML), a subfield focused on the interpretations of machine learning models, has attracted the attention of the research community [21, 22]. Not every machine learning model, especially deep learning models, can provide

✉ Ioannis Mollas
iamollas@csd.auth.gr

Nick Bassiliades
nbassili@csd.auth.gr

Grigorios Tsoumakas
greg@csd.auth.gr

¹ School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece

¹ ECOA 15 U.S. Code §1691 et seq.

explanations on its own. Since a lot of models are unable to provide interpretations intrinsically, IML has introduced explainability techniques to solve that issue. Such explanations can be in the form of *Feature Importance*, *Rules*, and *Counterfactual* explanations, among others. Particularly, *Feature Importance* (FI) techniques estimate the influence of each feature on the prediction. Each type of explanation has its own set of evaluation metrics, and for FI, they include *robustness* [23], *faithfulness* [24], *infidelity* [25], and *truthfulness* [26].

Nevertheless, techniques that generate feature importance explanations can only provide approximate information when applied to very complex models. Therefore, it is necessary to effectively evaluate the output of such techniques. Additionally, researchers and practitioners face challenges in selecting the most suitable explainability technique from the numerous available options. Consequently, an ensemble of explainability techniques or an automatic selection tool can be highly valuable. One approach to address the ensembling of explainability techniques is through aggregation, such as using averaging techniques or optimisation [27]. However, research in this area is limited, and these methods heavily rely on explainability metrics. Several metrics have been proposed to assess the quality of an explanation, depending on its form, including *fidelity* [28], *coverage* [29], and *stability* [30]. Nonetheless, most of these metrics are not useful for the end user.

Explaining the explanations is another interesting direction. Argumentation can be a first step towards this direction. In particular, argumentation is the study of how conclusions can be reached by a logical chain of reasoning, that is, claims based, soundly or not, on premises [31]. IML and argumentation both aim to persuade someone to accept the legitimacy of a decision. In the philosophy of science, it is debatable whether the explanations are arguments or not. An intriguing point of view distinguishes between arguments and explanations, stating that arguments are used to justify something in dispute, but explanations are used to provide a meaning of something incomprehensible [32].

In this work, based on our previous preliminary work [26], we aim to combine 3 concepts to create a meta-explanation ensembling multiple explanation techniques based on a complete and user-oriented explainability metric, called *truthfulness*, complemented with an argumentation framework.

Section 2 of the paper covers the necessary theoretic concepts, while Section 3 presents related studies. Section 4 introduces our technique, and Section 5 evaluates it through a series of experiments. Finally, in Section 6, we discuss the findings, and provide our concluding opinions and future plans.

2 Background

In this section, we introduce the basic notions that underlie our approach. We will discuss machine learning and interpretable machine learning concepts, as well as a few argumentation frameworks.

2.1 Machine learning

Machine Learning (ML) is a cutting-edge technology that forms the core of new and innovative products. We can use ML to solve both supervised and unsupervised problems. In this paper, we emphasise on supervised problems such as binary classification and regression. Thus, given a dataset D , containing instances $x_i \in X \subseteq \mathbb{R}^l$, where l is the size of the feature space $F = [f_1, f_2, \dots, f_l]$ and their predictions $y_i \in Y \subseteq \mathbb{R}$, we can train a model P to predict y given an instance x , $P(x) = y$.

Based on the data type x_i can have different shapes. In tabular data, x_i has l values according to the l different f_i features. Dealing with textual data, such as sentences, we can have multiple representations. The simplest representation is to use Bag-of-Words or TF-IDF vectors, which are one-dimensional and express each sentence x_i as a vector with l different values, where l is equal to the size of the vocabulary. We can also have more complex representations, such as word embeddings, which can be two-dimensional. These representations, given a fixed sentence length s , express each word of the sentence as a vector of size e . If x_i represents multivariate time-series data, then it contains $l \times m$ values, thus l values according to the l different f_i features across m time-steps. Finally, when dealing with images, we must handle with three-dimensional inputs. The first two dimensions represent the image's resolution, while the third expresses its colour channel. Therefore, we can deal with either 1D (tabular or textual data), 2D (textual or time-series data) or 3D (image data) inputs.

We can choose from a variety of ML models according to the task, data type, and size of the dataset, ranging from traditional algorithms (logistic regression and support vector machines) to ensemble algorithms (random forests and XGBoost), and deep neural networks such as CNNs, LSTMs, and Transformers. In this work, we will focus on neural networks and will employ three different types. The first type of neural network will be linear. This network has only an input layer and an output layer. The data, regardless of shape, are handled by the network via a flattening layer. The second type concerns network architectures that are designed specifically for the task, while the third type is a more complex version of them. These networks will contain feed-forward,

convolutional, recurrent, bidirectional, and attention layers to showcase our approach to a wide range of networks.

2.2 Interpretable machine learning

With the increasing adaptation of ML, there is a need for more transparent and understandable decision systems in a lot of sectors. IML, a subfield of XAI, aims to make ML models more accessible and transparent. There are intrinsically interpretable ML models, like linear models or decision trees, while others, like ensembles or neural networks, most of the time are more complex and uninterpretable. As a result, we require techniques to explain them.

IML approaches might be global, revealing an ML system's whole structure and working mechanism, or local, explaining a specific decision. We can also distinguish between techniques that are applicable to any ML model, known as model-agnostic techniques, and techniques that are limited to specific ML algorithms or architectures, known as model-specific techniques. For example, RuleFit is a global, model-specific technique [33], while LIME is a local, model-agnostic technique [34].

Another aspect could be the applicability of an explainability technique to different data types. There are algorithms that are applicable to specific data types, or there are data-type independent algorithms. For example, LionForests [35] is a data type specific algorithm applicable only to tabular data, while Anchors [36] is a data type independent algorithm. Furthermore, we can distinguish the difference of the explainability techniques based on how they provide explanations. There are numerous ways to present an explanation. Several techniques generate rule-based explanations, while others use weights to indicate the importance of input features.

The latter has been expressed using various terms, such as *attribution importance*, *saliency maps*, and *feature importance*, among others. In this work, we will use the last notation, *feature importance* (FI). Depending on the explainability technique, FI explanations can be global or local. Therefore, given a model P , an instance x_i (as presented in Section 2.1), and an FI explainability technique Z , the explanation will be $Z(P, x_i) = [z_1, z_2, \dots, z_l]$, where z_j corresponds to a weight – a.k.a. attribution or importance score – for the j^{th} value of instance x_i .

A variety of algorithms have been proposed in this field. LIME [34], SHAP [37], and Permutation Importance [38] are among the most well-known model-agnostic feature importance explainability algorithms. A plethora of model-specific algorithms, on the other hand, have also been proposed. In neural networks, algorithms exploiting back propagation operation, like Layer-wise Relevance Propagation (LRP) [39] and Integrated Gradients (IG) [40], are retrieving the influence of the input to the output.

2.3 Argumentation

Argumentation theory is a fundamental concept in AI with numerous applications, one of which is in the criminal justice field [41]. Argumentation procedures show step-by-step how they reached a decision. Therefore, argumentation is considered highly interpretable [42]. However, that's not always the case. Every argumentation procedure is based upon an argumentation framework. Regarding the argumentation framework employed, a few argumentation procedures are interpretable, but not explainable. Classic argumentation based on logic, as proposed by Hunter et al. [43], is a simple, yet explainable argumentation framework with many capabilities.

Argumentation based on Classical Logic (CL) concerns a framework defined exclusively with logic rules and terms. A sequence of inference to a claim is an argument in this framework. Specifically, an argument is a pair $\langle \Phi, \alpha \rangle$ such that Φ is consistent ($\Phi \not\vdash \perp$), $\Phi \vdash \alpha$, and Φ is a minimal subset of Δ (a knowledge base), which means that there is no $\Phi' \subset \Phi$ such that $\Phi' \vdash \alpha$. \vdash represents the classical consequence relation. In this framework counterarguments, the defeaters, are defined as well. $\langle \Psi, \beta \rangle$ is a counterargument for $\langle \Phi, \alpha \rangle$ when the claim β contradicts the support Φ . Furthermore, two more specific notions of a counterargument are defined as *undercut* and *rebuttal*. Some arguments specifically contradict other arguments' support, which leads to the undercut notion. An undercut for an argument $\langle \Phi, \alpha \rangle$ is an argument $\langle \Psi, \neg(\phi_1 \wedge \dots \wedge \phi_n) \rangle$ where $\{\phi_1, \dots, \phi_n\} \subseteq \Phi$. If there are two arguments in objection, we have the most direct form of dispute. This case is represented by the concept of a rebuttal. An argument $\langle \Psi, \beta \rangle$ is a rebuttal for an argument $\langle \Phi, \alpha \rangle$ if $\beta \leftrightarrow \neg\alpha$.

Argumentation begins when an initial argument is put forward, and some claim is made. This leads to an argumentation tree Tr with root node the initial argument. Objections can be posed in the form of a counterargument. In Tr , these are represented as children of the initial argument. The latter is addressed in turn, ultimately giving rise to a counterargument. Finally, a judge function decides if a Tr is rather *Warranted* or *Unwarranted*, based on marks assigned to each node as either undefeated U or defeated D . A Tr is judged as *Warranted*, $Judge(Tr) = Warranted$, if $Mark(A_r) = U$ where A_r is the root node of Tr is undefeated. For all nodes $A_i \in Tr$, if there is a child A_j of A_i such that $Mark(A_j) = U$, then $Mark(A_i) = D$, otherwise $Mark(A_i) = U$.

3 Related work

In this section, we will present feature importance evaluation metrics found in the literature, as well as a few meta-explanation techniques we identified.

3.1 Evaluation

One key evaluation metric in the IML research area is *fidelity*. It was first used to evaluate the performance of surrogate models and their ability to mimic the black box models they were explaining. We can define *fidelity* as the accuracy of a surrogate model on a test set in relation to the complex model's decisions. This metric, however, had several shortcomings because it was not user-centric and could not be used in non-surrogate explainability techniques. Influenced by *fidelity*, *faithfulness* and faithfulness-based metrics were therefore introduced [44].

While the origins of the initial Faithfulness-based measure are unclear, one of the first research to propose it aimed at evaluating sentence-level explanations in text classification tasks [24]. In this study, for a given instance, the sentence with the highest important score was removed and the change in the prediction was recorded. The higher the change in the prediction, the better the explanation. A different definition for *faithfulness* was also provided by a study [23], measuring the correlation between importance and prediction by continuously removing the most important elements from the input and observing the output.

Several variations on *faithfulness* were also introduced. *Decision Flip (most informative token)* removes the most informative token and awards the explanation if and only if the prediction is changing [45], whereas *Decision Flip (fraction of token)* identifies the number of important tokens that must be removed to flip the model decision [46].

Two other metrics, *comprehensiveness* and *sufficiency*, were introduced as *faithfulness* alternatives [47]. The former evaluates the explanation by deleting a set of elements from the input and observing the change in the prediction, whereas the latter does so by preserving only the important ones and removing the rest.

Monotonicity, also known as *PP Correlation*, is another similar metric [48]. It adds elements in descending order of priority, beginning with an empty input. The prediction should increase proportionally to the importance of the new elements. The correlation between the prediction and importance scores is then used to calculate *monotonicity*.

In a previous work of ours, *Truthfulness* was introduced as a faithfulness-based metric, which focuses only on the polarity of the feature importance weights [26]. It analyses every element of the input and making different alterations it monitors the model's behaviour. One additional metric, influenced by *faithfulness* and *truthfulness*, proposed to both consider importance correlation and polarity consistency, is *Faithfulness Violation Test* [49]. This metric captures both the correlation between the importance scores and the change in the probability, while it also examines if the sign of the explanation weights correctly indicates the polarity of input impact, similarly to *truthfulness*.

In addition to the metrics mentioned in this discussion, numerous other metrics are available. The Quantus GitHub repository² offers various variations of the faithfulness metric, as well as metrics related to robustness, complexity, randomization, and other related concepts [50].

A few studies introduced datasets with ground truth *rationales*, which are golden explanations. Rationales can be used to evaluate explainability techniques using traditional ML metrics like F1 score and area under the precision-recall curve (AUPRC). One work proposed ERASER, a benchmark for NLP models, which includes datasets containing both document labels and snippets of text recognized as explanations by annotators [47]. However, in real applications, most of the datasets do not contain ground truth information regarding the explanations, and as such these evaluation approaches cannot be applied. Furthermore, we can only assume that humans are capable of annotating unbiased *rationales* [51]. Nevertheless, the usefulness of such benchmarks is to enable comparison of newly proposed explainability techniques.

3.2 Meta-explanations / aggregation

Different aggregation procedures are initially introduced in a very interesting research [27]. Attempting to combine multiple explanation techniques, metrics such as *sensitivity*, *faithfulness*, and *complexity* [52], are used over different combination strategies. Among these combination strategies, *Mean* and *Median*, are presented. Through experimentation, it is suggested that aggregating leads to a smaller error compared to the error an explanation by one technique can have. Moreover, another combination strategy is presented. For a given instance, a set of near neighbours is identified. Extracting explanations for the predictions of these neighbours, the final explanation is the aggregation of the explanations, weighted by the distance of the neighbours to the original instance. The latter was designed to lower the *sensitivity* and *complexity*.

A recent research introduced a method, called *inXAI*, that enables the combination of explanations provided by multiple techniques, using specific evaluation metrics, to do so [53]. In their experiments, they use LIME, SHAP and Anchors to ensemble explanations. They select three metrics; *stability*, *consistency*, and *area under the loss curve*, to ensemble the weights produced by the techniques into one. One issue with this approach is that the *consistency* metric requires to create and use different ML models to produce explanations. One of the framework's shortcomings is that it only enables model weighting using comparative evaluation metrics across several models/explainers. It does not guarantee that the final explanations are correct or acceptable for

² <https://github.com/understandable-machine-intelligence-lab/Quantus>

the end user. This method was evaluated only in an image classification use case.

Finally, another work on ensembling explanations introduces EBEC, a method for correcting global explanations of non-differentiable ML models with a non-differentiable importance score [54]. The central idea of EBEC is to train multiple ML models on a dataset to identify different local minima, then produce global explanations using an explainability technique (in this work SHAP), and finally combine them by solving an optimization problem that guarantees certain qualitative properties. They conclude that EBEC works effectively in three different tabular datasets based on their evaluation.

4 Truthful meta-explanations supported by arguments

In this work, we are presenting a three-dimensional contribution to the IML community. Focusing exclusively on FI explainability techniques, we first formulate the definition of the *truthfulness* metric. Then, we present a meta-explanation technique for ensembling multiple explanations in an ensemble fashion. Finally, we also present how arguments can enhance the meta-explain process, which uses the *truthfulness* metric. All of these are visible in the workflow of Fig. 1.

To begin, we will state a few assumptions that must hold for our technique to be theoretically sound. Assumption 1 ensures that the ML model we are trying to apply our technique is able to provide continuous predictions. This is a necessary property for the metric we are going to formulate in the following section (Section 4.1).

Assumption 1 The machine learning model $P(x) = y$ can provide continuous predictions $y \in \mathbb{R}$. A classification model, for example, should be able to provide predictions in the form of probabilities of good quality (e.g. neural networks or probabilistic models). In our technique, a classification model that produces inadequate probability estimates, such

as decision trees, would not yield satisfactory results. A regression model, on the other hand, always produces continuous outputs.

The second assumption (Assumption 2) concerns the explainability techniques utilised in the approach. The amount and type of the technique to be used in the ensemble to produce one final explanation is not limited, with the only exception being to provide weights that represent a (local) monotonic relation to the prediction of a specific label or being perceived by end users as such. This is critical since a few explainability techniques, such as SHAP, provide the contribution of a feature to the prediction without assuming any local or global monotonicity. Nonetheless, based on this proposed contribution, still, end-users perceive the relationship between a feature and the output to have monotonic behaviour when altered.

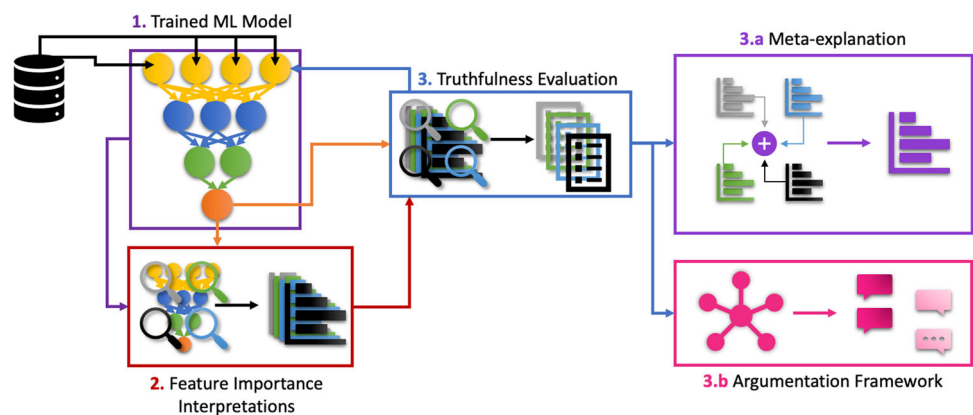
Assumption 2 FI's are producing, or they are perceived as producing, z_j weights with local or global monotonic notion.

4.1 Truthfulness metric

The first contribution of this paper concerns the *truthfulness* metric. Truthfulness is a user-inspired evaluation metric that simulates a user's behaviour with respect to an explanation. It addresses issues of other Faithfulness-based metrics by evaluating all feature importance elements and taking into account all signs (*Positive*, *Negative* and *Neutral*). But, before we get into the definition of the metric, we will present an example. The following explanation explains a prediction of a customer's loan disapproval in a bank.

The customer received the explanation shown in Fig. 2. Despite her friend's income of \$1,000 (Co-applicant Income), she observed that it had minimal impact on her decision, leading to a disapproval with a score below the minimum threshold. Consequently, she decided to involve her mother, who had a slightly higher income (\$1.2K), as a co-applicant. Surprisingly, this change had no effect on the outcome, as she received the same score and, therefore, the same decision

Fig. 1 Workflow of MetaLion



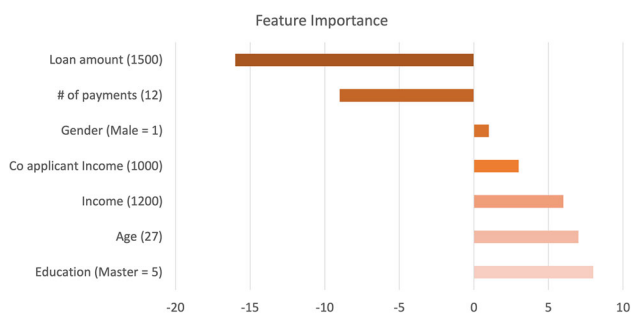


Fig. 2 Feature importance weights assigned to the features of the example

(disapproval), which she anticipated would improve slightly. As a result, the customer perceived the explanation as dishonest, while also lacking trust in the predictive system.

Influenced by this, we suggest a metric that, given an explanation, performs a few tests to ensure that the explanation provided to the end-user is truthful. This procedure shares similarities with counterfactual techniques [55], but it differs in its objective. While counterfactual techniques aim to switch the class in a classification problem, our goal here is to observe the change in the probability of the predicted class, which may not necessarily result in a switched prediction. Hence, for each feature importance score z_j assigned to the feature values v_i^j of x_i , we both increase and decrease the feature values, and we observe if the model behaves as expected with respect to the feature importance.

In this work, we will focus on four types of data: textual, image, tabular, and time-series. The words are the features in textual tasks, and the importance concerns a word and, in some cases, its position. When dealing with images, feature importance is used to describe either a single pixel or a group of pixels known as superpixels. Each feature in tabular data has its own importance score. Lastly, in time-series, feature importance can refer to either a sensor's time-step value or a sensor throughout the entire time-window. In all these cases, feature importance can be either *Positive*, *Negative*, or *Neutral*, as described in Definition 1.

Definition 1 The importance assigned to a feature can be $\text{IMP} \in \{1 = \text{Positive } (z_i > 0), -1 = \text{Negative } (z_i < 0), \text{ or } 0 = \text{Neutral } (z_i = 0)\}$.

Let's discuss now how we alter a feature value v_i^j . Given a set of samples X' , we measure each feature's distribution statistics, namely, min, max, mean and STD values. Then, as presented in Algorithm 1, we calculate a *noise* or the alternative values. This *noise* is small, and therefore these alterations are local. This procedure is different for the various data types. In textual datasets, this procedure replaces the word regarding the examined feature importance score with an empty string. In images, we compute a *noise* which makes lighter and darker a pixel or a superpixel. For superpixels,

Algorithm 1 Process of determining the alternative values for a feature

Require: Instance's feature value $value$, Distribution statistics of feature $feature_distribution$, Noise level $level$, Data type $type$

procedure DETERMINEALTVVALUES($value, feature_distribution, level, type$)

if $type$ is *Textual* **then**

$value^+ \leftarrow value$

$value^- \leftarrow 0$

return $value^-, value^+$

end if

$min, max, mean, std \leftarrow extract(feature_distribution)$

$noise \leftarrow abs(mean - gaussian_noise(mean, std))$

if $type$ is *Image* or *TimeSeries* **then**

return $-noise, +noise$

end if

$value^- \leftarrow value - noise$

$value^+ \leftarrow value + noise$

if $value^- < min$ **then** $value^- \leftarrow min$ **end if**

if $value^+ > max$ **then** $value^+ \leftarrow max$ **end if**

return $value^-, value^+$

end procedure




we have to also employ an image segmentation algorithm to identify the superpixels of an image. Regarding tabular data, we create a *noise* which both increases and decreases the feature value, while in time-series, we increase and decrease with the calculated *noise* either a specific time-step of a sensor, or the whole time-window of a sensor. We set three different *noise* levels; “weak”, “normal”, and “strong”, in the cases which are applicable (image, tabular and time-series).

Definition 2 The alteration of the value of a feature can be $\text{ALT} \in \{1 = \text{Increasing } (v'_{j,i} > v_{j,i}), -1 = \text{Decreasing } (v'_{j,i} < v_{j,i})\}$, where $v'_{j,i}$ the altered value.

In Fig. 3, we show an example of each data type. The textual example demonstrates how to remove (decrease) a feature, in this case, the word “John”. In the image example, we can see that the first alteration involves lightening the kitten's ear by increasing the values of the superpixel, and the second alteration involves darkening the ear by decreasing the values of the pixels. In the tabular example, we make two changes to the “Age” feature. We increase “Age” from 25 to 27, while also decreasing it to 23. Finally, in the time series, we increase the first sensor's readings by 0.1 at each time step and decrease them by the same value.

We discussed a feature's feature importance score and introduced the concept of alterations in the values of features across different data types. We are now introducing the concept of expected behaviour. Given a feature importance score z_j for f_j , we have two alternative values for that feature for a specific instance x_i . We can request the ML model to predict the modified instance $x'_i \in \{x_i^{inc}, x_i^{dec}\}$, where x_i^{inc} is the same instance but with a higher value for the examined feature and x_i^{dec} has a lower value. Then, regarding the feature importance score z_j , we evaluate whether

Fig. 3 Example of altering a feature of an instance for the four different data types

Data Type	Textual	Image	Tabular	Time Series				
					T1	T2	T3	T4
Original	'My name is John and I'm happy'		Age = 25, Height = 170, Weight = 62, Income = \$2000	S1	0.1	0.2	0.4	0.5
				S2	0.2	0.5	0.8	1
				S3	0.9	0.9	0.8	0.9
ALT = Increasing	Not applicable		Age = 27, Height = 170, Weight = 62, Income = \$2000	S1	0.2	0.3	0.5	0.6
				S2	0.2	0.5	0.8	1
				S3	0.9	0.9	0.8	0.9
ALT = Decreasing	'My name is John and I'm happy'		Age = 23, Height = 170, Weight = 62, Income = \$2000	S1	0	0.2	0.3	0.4
				S2	0.2	0.5	0.8	1
				S3	0.9	0.9	0.8	0.9

the model's predictions $P(x_i^{inc})$ and $P(x_i^{dec})$ behave as expected.

Definition 3 The expected behaviour of an M component can be $EXP \in \{1 = \text{Increasing } (P_M(x_i) - P_M(x'_i) < \delta), -1 = \text{Decreasing } (P_M(x_i) - P_M(x'_i) > -\delta), 0 = \text{Remaining Stable } (|P_M(x'_i) - P_M(x_i)| < \delta)\}$, where x'_i the instance with the altered value, while tolerance δ is defined either manually by the user or is set to a default value (0.0001).

As presented in Definition 3, the model's prediction can behave in three ways. It can increase, decrease, or remain stable. If the feature importance score z_j is positive, we expect the prediction to increase for the x_i^{inc} modified instance while decreasing for the x_i^{dec} . If z_j is negative, we expect the prediction of the two changes, x_i^{inc} and x_i^{dec} , to decrease and increase, respectively. In the case of a neutral feature importance score z_j , we anticipate that the prediction will remain stable for both alterations. We also use a δ tolerance value. This will help evaluate an importance score as truthful in cases where the prediction will change, for example, from 0.75 to 0.7502, where the difference is extremely small. This will help to not punish small mistakes. However, setting $\delta = 0$ leads to a stricter evaluation. In the experiments section (Section 5), we examine different delta values. Table 1 summarises all of these.

As a result, we argue that a feature importance score is truthful if and only if the behaviour of the model's prediction regarding the alterations is as expected. This is also included in Definition 4. It is worthwhile to provide an example to demonstrate this. The ML model predicts $P_M(x_i) = 0.7$ for a random instance x_i , and the feature f_1 , with a value of $v_{1,i} = 1$, has an IMP $z_1 = 0.5$ (positive). We use Gaussian noise to increase and decrease the value of the feature based on its distribution. We change the value to $v_{1,i}^{inc} = 1.21$

and $v_{1,i}^{dec} = 0.85$, for x_i^{inc} and x_i^{dec} , respectively. Then, we observe the model's predictions by querying the ML model. In this example, the prediction for the $v_{1,i}^{inc}$ was increased to 0.85, while the prediction for the $v_{1,i}^{dec}$ remained stable. As a result, we can conclude that the behaviour in the second alteration was not as expected, and hence the feature importance score is untruthful.

Definition 4 [Truthfulness] The importance assigned to a feature can be defined as *truthful* when the expected changes to the output of the M model $P_M(x'_i)$ are correctly observed with respect to the alterations that occur in the value of this feature. Thus, for both values of ALT and a given IMP, the $IMP \times ALT = EXP$ must be in accordance with the *truthfulness* matrix (Table 1).

The *truthfulness* metric analyses the feature importance scores individually and penalizes those that are deemed untruthful. For an instance with $|F|$ features, we examine the importance scores assigned to each feature's value one by one. If a score is deemed truthful, we increment the *truthfulness* score by one. However, if a score is considered untruthful, we do not increment the score. Finally, we can optionally normalize the *truthfulness* score to the range of $[0, 1]$ by dividing it by the number of features. While in the experiments below we do not normalize the scores, normalization can help the comparison of multiple explana-

Table 1 Truthfulness matrix [(t)ruthful and (u)ntruthful states]

		1	1	-1	1	-1	ALT EXP
IMP	1	t	u	u	u	t	
	0	u	t	u	u	t	
	-1	u	u	t	t	u	

tion techniques across different models. This process can be mathematically formulated as follows:

$$Truthfulness(Z(P, x_i)) = T(Z(P, x_i)) = \frac{1}{|F|} \sum_{j=1}^{|F|} evaluate(z_j, x_i) \tag{1}$$

$$evaluate(z_j, x_i) = \begin{cases} 1 & \text{if } z_j \text{ truthful with respect to the} \\ & \text{alterations of } x_i, x'_i \in \{x_i^{inc}, x_i^{dec}\} \\ 0 & \text{else} \end{cases}$$

Before proceeding with the meta-explanation ensembling technique, we are arguing on why we only use two alternative values to evaluate a feature importance score of a feature for a given instance. We assume that if there is monotonicity between these two values, there will be monotonicity in the intermediate values as well. We make this assumption to minimise the computational cost, considering that many explainability techniques have high response times. We achieve three things with this choice:

Faster evaluation: Given a set of explainability approaches and their response times, *truthfulness* has a low computing cost when applying two alterations per feature, as opposed to more.

Integration to a meta-explanation technique: Being lighter computationally, we can use this metric in a meta-explanation technique to produce even better explanations. In the following section, we introduce a meta-explanation technique that makes use of the *truthfulness* metric.

Reduce the environmental impact: Given that *truthfulness* necessitates re-querying the ML model twice for each feature, utilising a set of alternative values rather than two will increase the cost exponentially in larger feature sets. As a result, we anticipate a lower environmental impact by selecting only two alternative values.

4.2 Meta-explanation technique

Based on the *truthfulness* metric, we introduce a meta-explanation technique, called *MetaLion*. Differently than the other recent research, we employ *truthfulness* metric to combine multiple explanation techniques, to provide a more accurate local explanation.

Given E explainability techniques, and an examined instance x_i whose prediction made by a model M , $P_M(x_i)$, to ensemble the different explanations $Z = [Z_0, Z_1, \dots, Z_E]$, we calculate the *truthfulness* of each explanation. We also measure the average change of the output given the two alter-

ations x'_i of each feature f_j , $ac_j = \frac{1}{2}(|P_M(x_i) - P_M(x_i^{inc})| + |P_M(x_i) - P_M(x_i^{dec})|)$.

The first step in performing the ensembling of the multiple explanations is to determine the candidate importance scores for each feature based on its *truthfulness*. Algorithm 2 illustrates this. For each feature, we verify the *truthfulness* of the importance scores assigned by the various explainability techniques and save them for use in the next step.

Algorithm 2 Identification of candidate truthful feature importance scores

Require: Explanations Z , Examined Instance x_i , Feature Set F

procedure CANDIDATE TRUTHFUL SCORES(Z, x_i, F)

```

CZ ← new Hash Map
for feature  $f_j \in F$  do
    temp ← []
    for Explanation  $Z_m \in Z$  do
         $z_j = Z_m[f_j]$ 
        if evaluate( $z_j, x_i$ ) = 1 then
            temp ← temp  $\cup z_j$  end if
    end for
    insert temp list in CZ with  $f_j$  as key
end for
return CZ
end procedure

```

By identifying the truthful importance scores from each explanation for each feature for an examined instance, we present our ensemble procedure in Algorithm 3. First, we sort the features by average change. Then, starting with the feature with the greatest absolute change, we examine the candidate importance scores and choose the one with the highest absolute value. We save the highest absolute value, which we will use in the following steps. We proceed on to the next feature, which has the second largest absolute change. Again, we choose the highest absolute importance score among the candidate scores, which has to be lower than the previous score by absolute value (line 13th). The sequence of handling features and their importance scores is vital. It is possible for a technique to have a truthful score but an incorrect magnitude (e.g., an importance score of 1 instead of 0.4). Therefore, by prioritizing the feature with the highest average change and adjusting the weights accordingly, we can better meet the user’s expectations. We assign a zero value when a feature has no candidate feature importance scores.

Let’s discuss an example using this algorithm. In Table 2, we have an instance $x_i = [0.27, 0.12, 1, 5, 6, -3]$, and three explanations $Z = [Z_0, Z_1, Z_2]$ regarding its prediction. We calculate the *truthfulness* of each explanation. The truthful importance scores of each technique are represented with a green check mark in Table 2, while untruthful scores with a red cross mark. The average change (AC) of the output for each feature based on two alterations is also provided. Based

Table 2 Example of evaluation of three techniques and the meta-explanation

	f_1	f_2	f_3	f_4	f_5
Z_0	0✓	1✓	0.5✗	0.3✗	-0.2✓
Z_1	0.1✓	0.23✓	0.7✗	0.3✗	0.3✗
Z_2	-0.1✗	0.2✓	-0.4✓	0.2✗	-0.1✓
AC	0.05	0.8	0.5	0.01	0.3
$MetaLion$	0.1	1	-0.4	0	-0.2

on the Algorithm 3, we select first the score of f_2 , which has the highest AC score ($ac_2 = 0.8$). Among the three truthful importance scores, we select the highest $z_2^0 = 1$, the one from Z_0 . Then, we proceed to the next feature f_3 . There is only one truthful importance score $z_3^2 = -0.4$ provided from Z_2 . In the same fashion, we are selecting the most appropriate importance scores for each feature, till we have a complete explanation.

Algorithm 3 Truthfulness-based meta-explanation algorithm

Require: Candidate Imp. Scores CZ , Average Change AC , Feature Set F

procedure TRUTHFULMETAEXPLANATION(CZ, AC, F)

$OF \leftarrow \text{sort}(F)$ based on AC , $temp_score \leftarrow None$

$meta_explanation \leftarrow$ new Hash Map

for feature $f_j \in OF$ **do**

if $CZ[f_j]$ is not empty **then**

if $temp_score$ is $None$ **then**

$ind \leftarrow$ index of $max_abs(CZ[f_j])$

$temp_score \leftarrow max_abs(CZ[f_j])$

else

if $CZ[f_j]$ has only one value **then**

$ind \leftarrow 0$

$temp_score \leftarrow \min(abs(CZ[f_j][ind]), temp_score)$

else

$index \leftarrow$ index of the most appropriate value

$temp_score \leftarrow \min(abs(CZ[f_j][index]), temp_score)$

end if

end if

 insert $CZ[f_j][ind]$ in $meta_explanation$ with f_j as key

else

 insert 0 in $meta_explanation$ with f_j as key

end if

end for

return $meta_explanation$

end procedure

With this ensembling procedure, we achieve two things. The first is that we gather more truthful importance scores in the final explanation. Moreover, we re-rank the importance scores of the features using the *truthfulness* evaluation and our ensembling algorithm. Later in the experiments, we will discuss the effectiveness of our approach.

4.3 Argumentation

The argumentation framework we designed to provide justifications for the *truthfulness* evaluation was a very useful component of Altruist, our earlier preliminary work [26]. While we do not re-formulate the entire argumentation system, in this section, we do re-formulate the atoms which form arguments, utilized in our system to make them more descriptive. More information about the theoretical formulation of the framework can be found in our earlier work [26]. The original available atoms were the following:

- a : The explanation is untrusted
- b : The explanation is trusted
- c_j : The importance z_j is untruthful
- d_j : The importance z_j is truthful
- $e_{j,ALT}$: The model's behaviour by altering f_j 's value is not according to its importance
- $f_{j,ALT}$: The evaluation of the alteration of f_j 's value was performed and the model's behaviour was as expected, according to its importance.

We are re-phrasing the last two atoms, $e_{j,ALT}$ and $f_{j,ALT}$, as seen below:

- $e_{j,ALT}$: The model's behaviour by altering f_j 's value from X to Y (ALT) is not according to its importance Z
- $f_{j,ALT}$: The evaluation of the alteration of f_j 's value X to Y (ALT) was performed and the model's behaviour was as expected EXP , according to its importance z_j .

This way, we do not modify the theoretic argumentation framework supporting our system, but we are making the last two atoms used in the arguments more descriptive. While we are presenting a complete example in the qualitative experiments, we are showing an example below:

- $e_{2,INC}$: The model's behaviour by altering f_2 's value from 25 to 26 (increased) is not according to its importance Z
- $f_{2,INC}$: The evaluation of the alteration of f_2 's value 25 to 26 (increased) was performed and the model's behaviour was as expected (increased), according to its importance z_2 .

An example showcasing the enhanced argumentation framework is being presented in the qualitative experiments (Section 5.3).

5 Experiments

In this section, we will test the *truthfulness* metric on several types of datasets, as well as our meta-explanation technique, in a series of quantitative experiments. We will also conduct

Table 3 Information about the datasets incorporated in our experiments. *After preprocessing (R: Regression, BC: Binary Classification)

Name	# of Instances	Task	Sector	Data Type
TEDS	33.727*	R	Manufacturing	Time-Series
CCA	1.232*	BC	Banking	Tabular
HDE	23.000	BC	Insurance	Images
MedN.	3.204	BC	Health	Textual

a qualitative evaluation of the explanations produced by the meta-explanation technique.

5.1 Setup

We will begin by describing the datasets we used, the preprocessing procedures we utilised, the predictive models we employed, and the explainability techniques we included in our experiments.

5.1.1 Datasets

We included the following datasets in our experiments to cover a variety of the critical sectors that use ML in their workflows, as presented in Section 1. We incorporated the Turbofan Engine Degradation Simulation (TEDS) dataset [56, 57] for the manufacturing sector’s predictive maintenance scenario, which aims to predict the remaining useful lifetime (RUL) of engines using *time-series data*. The second dataset we are using, Credit Card Approval Prediction (CCA), contains information about bank customers (*tabular data*), as well as information regarding their debt payments (if any)³. The goal is to determine whether a client is eligible for a credit card. A dataset for Hurricane Damage Estimation (HDE)⁴ of properties using satellite *images* [58], in a classification manner, is incorporated in our experiments to cover the insurance sector. Finally, data for the identification of Acute Ischemic Strokes (MedN) through brain MRI reports (medical notes - *text*)⁵ [59] connects the experiments to the healthcare sector. More information about the datasets is visible in Table 3, while about their preprocessing in Section 5.1.2 and in the GitHub repository “MetaLion: Truthful Meta Explanations”⁶.

³ <https://cutt.ly/xQ1mqyo>

⁴ <https://cutt.ly/kQ1n3kE>

⁵ <https://cutt.ly/sQ1nM9c>

⁶ <https://github.com/iamollas/MetaLion-Truthful-Meta-Explanations.git>

5.1.2 Preprocessing

While extended preprocessing is accessible in our GitHub repository, here we mention few crucial preprocessing steps. We suggest other researchers and users to apply similar preprocessing towards more explainable end-to-end systems.

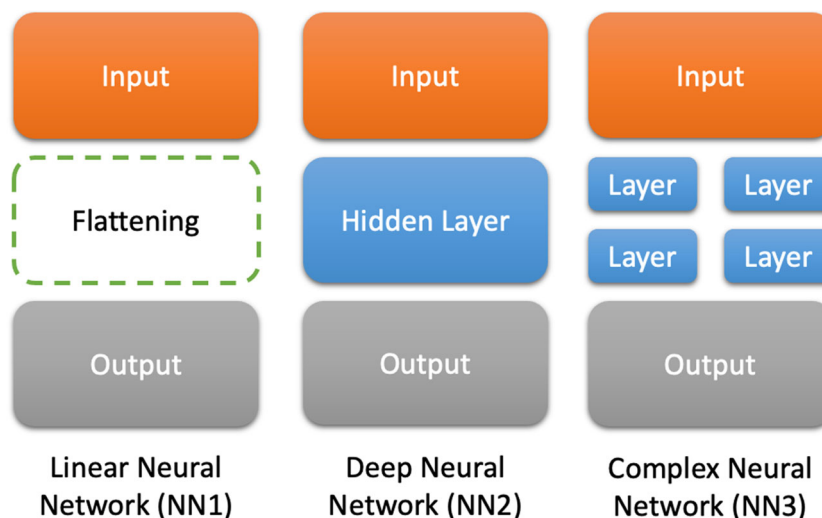
Starting with the time-series dataset, TEDS, we scale our data to $[0.1, 1]$ after reducing the available features and retaining only measurements from 14 sensors. We chose to scale our data to this range rather than $[0, 1]$ because none of the 14 sensors have measurements that are equal to 0, but only positive values. This is a critical decision in explainability. A lot of the time, the 0 value has a neutral notion in terms of explainability and, more precisely, feature importance. As a result, local techniques such as LIME can generate a positive weight that, when multiplied by the 0 value, is neutralized. Furthermore, we create examples of 14 measurements across 50 time steps, using the RUL value on the most recent time step as the goal variable, utilizing a time window of 50 time steps. One final preprocessing step we use is to scale the output, the RUL value, to $[0, 1]$, which is easier for a neural network to learn.

In our tabular data, on the other hand, there exist features with both positive, negative, and zero values. In this case, we’d like to scale them to $[-1, 1]$ while keeping the centre at zero. To address this issue, we use maximum absolute value scaling. In terms of image preprocessing, we did augmentation by randomly flipping and rotating the samples and scaling them from $[0, 255]$ to $[0, 1]$. In terms of the textual dataset, we used a symbol-removal process on each document and reduced the maximum number of words from 380 to 250, because relatively few documents were longer than 250 words. Finally, we used the BioBERT [60] pretrained transformer to obtain word-level embeddings for each document.

5.1.3 Model architectures

We utilized three distinct model architecture formats in our study. The first architecture, referred to as NN1, is a linear network consisting of input and output layers for all data types, with an additional layer for flattening 2D or 3D data. NN1 is similar to linear regression models. The second architecture, NN2, is a deep neural network specifically designed for each data type. For instance, we employed recurrent and feedforward layers for time series data, feedforward layers for tabular data, convolutional and feedforward layers for image data, and bidirectional recurrent, one-dimensional convolutional, and feedforward layers for textual data. The third architecture, NN3, is a more complex version of NN2 with additional layers, more neurons, and different activation functions. While NN3 networks may seem unnecessarily complex, they are essential for facilitating our research.

Fig. 4 The three different architecture formats used in our experiments



The various architecture formats mentioned above are depicted in Figure 4. For the classification datasets, the models in our study utilized a sigmoid activation function in the output layer. However, for the regression dataset (TEDS), we employed linear activation functions. For detailed information on the activation functions used in the hidden layers and specific layer configurations for each dataset, please refer to the implementation available in our GitHub repository⁷. In terms of the training process, a batch size of 32 was employed for CCA, HDE, MedN, and 512 for TEDS. The number of epochs varied for each dataset and model, and the specific values can be found in our GitHub page.

All models in our study were trained using separate training and validation sets, and their performance was evaluated on dedicated test sets. The performance of each model on the four individual tasks is presented in Table 4. It is noteworthy that the linear neural network (NN1) consistently performs worse than the other models.

Nevertheless, we included the linear neural network model in our study to assess the performance of our metric in the simplest case. This allows us to evaluate explanations for fully interpretable models accurately and serves as a baseline for comparison with more complex models. By including this model, we can ensure that our metric is capable of correctly evaluating explanations even in cases where the model's behaviour is transparent and easily interpretable. We conducted this test, influenced by recent research on evaluating explainability techniques using ground-truth synthetic explanations [61].

In all cases, NN2 performs equally well or better than NN3. NN2 is the most used architecture among the neural networks. It is important to mention that NN3 does not consistently outperform NN2 in all scenarios (evident in

datasets HDE and MedN). However, such complex models are necessary to stress the explainability techniques based on gradients.

5.1.4 Explainability techniques

In our experiments, we used three different explainability techniques, one model-agnostic and two model-specific (neural-specific). These are LIME, IG, and LRP, which we discussed in Section 2.2. In our study, we utilized the original Python library for LIME. For the IG and LRP techniques, we employed the iNNvestigate library. The default parameters for IG, LRP, and LIME were used for all datasets, except for HDE and MedN. Due to computational reasons, we adjusted the number of neighbours to 100 and 250 for LIME in the HDE and MedN datasets, respectively. Lastly, we employed a random explanation as a baseline in our study. This random explanation served as a reference point and allowed us to assess the impact of random noise on meta-explanation techniques.

One more explainability technique would be to exploit the real weights of the linear neural models (NN1). Those weights are ground truth interpretations. However, we will treat our NN1 as a black box model, and we will use these ground truth interpretations to evaluate our metric. Our metric should provide a perfect score for these interpretations, as they are correct and real.

We use the *Mean*, *Median* and *inXAI* for meta-explanation techniques as well as our proposed technique, *MetaLion*, as presented in Section 4. The *Mean* meta-explanation technique for each feature computes the mean value across the importance scores provided by the different explainability techniques, while *Median* takes the median value. For example, if LIME, IG, and LRP provided 0.2, 0.3, and 0.7 importance scores for the feature "age", *Mean* will assign

⁷ <https://github.com/iamollas/MetaLion-Truthful-Meta-Explanations.git>

Table 4 The three different architecture formats used in our experiments

Dataset	Metric	Avg # of Features	NN1	NN2	NN3
TEDS	RMSE	14	36.98	35.50	32.71
CCA	F_1	12	58.9%	70.71%	71.08%
HDE	F_1	29.71	56.55%	93.22%	85.29%
MedN	F_1	48.25	91.27%	96.64%	95.73%

a 0.4 importance score to “age”, while *Median* will assign a 0.3. For *inXAI*, we follow the original implementation, which builds upon three metrics to weight the seed explanations [53].

We will apply these explainability techniques in all datasets. Regarding TEDS, we will both apply them in time-step and sensor level, where in the former feature importance scores will be assigned to each time-step of each sensor, while in the latter, importance scores are assigned in the time-steps. For the CCA dataset, each feature is assigned a feature importance score, while in HDE, each pixel. Finally, in MedN dataset, the explainability techniques assigned importance scores to each term (word).

5.2 Quantitative experiments

The quantitative experiments we conducted include the evaluation of the *truthfulness* metric using ground truth information from the NN1, and a comparison of the different explainability techniques we chose, the meta-explanation techniques we used, and the one we designed, accompanied by an ablation study. Additionally, we study the influence of the *noise* and δ values on the evaluation, as well as we compare *truthfulness* to other metrics, *complexity*, *stability*, and *consistency*. All the metrics employed in our study were measured exclusively on the test set.

5.2.1 Truthfulness evaluation

The first part of the experiments focuses on the evaluation of explanations provided by linear neural models (NN1). Those are ground truth, and therefore we want to assess if the *truthfulness* metric presented in Section 4.1 correctly identifies these explanations.

In Table 5, we present the assessment of the inherent explanation of NN1s on the different datasets. We use three different levels of *noise* ($noise \in$ [“weak”, “normal”, “strong”]), and four different δ values ($\delta \in$ [0, 0.0001, 0.001, 0.01]). We know that NN1s are interpretable. Therefore, their explanations must be 100% truthful. In the results of the table, we can see that in TEDS (sensor level), CCA, and MedN, it perfectly evaluates the linear interpretations. However, in HDE, which is at the superpixel level, it assigns on average 1.29 wrong weights out of 29.71 superpixels.

This is reasonable as in our implementation, if the value of a pixel is increased (decreased) above (below) the maximum (minimum) value after the alteration, then the value is reset to the maximum (minimum) value. Then, considering that the superpixels are working by averaging the per-pixel importance scores, if a pixel does not get increased (decreased) as the others, due to such a limitation, the possibility of observing unexpected behaviours increases. On the other hand, if we were conducting the experiment at the pixel level, that issue would not have occurred.

Let’s see the following example. If we have 3 values [0.8, 0.6, 0.9] (3 pixels in a superpixel) all of them having a range of [0, 1], with the importance weights [0.2, 0.05, -0.35], and the prediction = $sigmoid(0.2 \times 0.8 + 0.05 \times 0.6 - 0.35 \times 0.9) = sigmoid(-0.125) = 0.469$, the average importance would be -0.04 (the weight of the superpixel). If we make a positive alteration by 0.2 the 3 values get [1, 0.8, 1]. Notice that 0.9 changed to 1 instead of 1.1, in order to not violate the range [0, 1]. The prediction will accordingly change to = $sigmoid(0.2 \times 1 + 0.05 \times 0.8 - 0.35 \times 1) = sigmoid(-0.125) = 0.472$. While a positive alteration, given a negative weight, should lead to decreased prediction, this did not happen. If we had allowed the value 0.9 to get to 1.1, the prediction would have been 0.444, hence decreased, as expected. We are aware of this limitation, but based on the experiments, it appears to be rare. Therefore, we have decided to maintain the restriction of keeping the alternative values within their respective ranges.

Given these findings, we can conclude that when the explanations are correct, *truthfulness* accurately evaluates a technique. We shall put it to the test in non-linear, complex models in the following experiments.

5.2.2 Explainability techniques evaluation

Let’s examine how truthful the different explainability techniques are on the four selected datasets. For the *noise* and δ parameters, we choose “normal” and 0.0001 (default parameters), respectively. In Table 6, for the four different datasets, we can observe how the different explainability techniques correctly assign weights to the predictions of NN2 and NN3. Let’s focus on the first four rows, which are the typical explainability techniques, and their evaluations.

Table 5 Truthfulness evaluation of linear models (NN1) on the different datasets

Dataset	Weak				Normal				Strong			
	0	0.0001	0.001	0.01	0	0.0001	0.001	0.01	0	0.0001	0.001	0.01
TEDS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
CCA	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
HDE	0.58	0.34	0.06	0.01	0.70	0.51	0.21	0.04	1.29	1.18	0.79	0.22
MedN	-	-	-	-	0.00	0.00	0.00	0.00	-	-	-	-

The two explanation strategies that perform best in these cases are IG and LRP. There is no apparent winner between the two, since IG outperforms LRP in TEDS while performing similarly to CCA, and LRP outperforms IG in HDE and MedN. LIME, on the other hand, is the worst explanation approach, doing worse than random explanations in three of the four cases.

5.2.3 Meta-explanation techniques comparison

The performance of meta-explanation techniques, also known as ensembles, is shown in the same table (Table 6). As we showed in the previous section, there is no clear winner among the many explainability techniques in several cases. Given the need for a meta-explanation, the ensembling technique appears to be a good choice. *Mean*, which averages the explanations, proves to be a promising approach based on the literature. Inspired by this, we included in the experiments a meta-explanation technique based on *Median* (selects the median weight among the suggested), as well as the *inXAI* technique, and the *MetaLion* technique as proposed in Section 4.2.

Checking the results from Table 6, we can see that our meta-explanation technique can drastically reduce the number of identified untruthful features, in all cases. All four meta-explanation techniques use the four explainability techniques as input. *MetaLion* reduces the untruthful features by 66% compared to the original techniques, and by 64% compared to the other three meta-explanation techniques.

Table 6 Avg. number of identified untruthful features per explainability technique

	TEDS		CCA		HDE		MedN	
	NN2	NN3	NN2	NN3	NN2	NN3	NN2	NN3
IG	<u>4.09</u>	<u>2.22</u>	4.45	4.59	18.09	18.09	4.31	5.16
LRP	7.41	5.82	4.26	4.59	<u>15.35</u>	<u>17.36</u>	4.17	<u>3.19</u>
LIME	11.46	11.67	6.49	6.56	20.73	23.13	4.33	4.50
Random	8.34	7.65	5.59	5.62	19.29	22.10	5.16	5.65
<i>Mean</i>	7.77	6.96	<u>4.25</u>	<u>4.34</u>	17.18	19.21	<u>3.54</u>	3.74
<i>Median</i>	7.41	6.29	4.45	4.59	17.12	18.45	3.74	3.83
<i>inXAI</i>	8.55	7.91	5.46	5.37	17.18	19.20	3.55	3.74
<i>MetaLion</i>	2.87	1.68	1.36	1.35	11.70	15.09	0.50	0.29

Best is in bold, second best is underlined

Ablation study We also conduct an ablation study on seed explanation techniques. We remove each seed technique one at a time, monitoring how the meta-explanation techniques perform. In Table 7, we can see the results. The performance of the meta-explanation techniques employing all four explanation techniques is shown in the first row. In the next rows, we omit the following approaches in this order: IG, LRP, LIME, and Random. The red up arrow (↑) indicates that omitting the specific explanation technique reduced the meta-explanation technique's performance compared to the original performance. The green down arrow (↓) indicates that performance improved as the average number of untruthful elements in meta-explanations decreased.

When LIME or Random are omitted, *MetaLion* appears to perform slightly worse, whereas the other three perform slightly better than their original performance. This occurs because of erroneous explanations introducing *noise* into these three meta-explanation techniques. *MetaLion*, on the other hand, detects the correct elements even in these noisy explanations and uses only them, discarding any potentially noisy ones. Based on this, we can conclude that our technique is resilient to noisy and even contradictory seed explanation techniques.

Another intriguing discovery is that the *Mean*, and *Median* meta-explanation techniques are susceptible to changes in the seed explainability techniques. On average, the performance of both techniques changes 0.60 and 0.77. Contrarily, *MetaLion* and *inXAI* appear to be more stable. On both neural networks, the average change across all datasets is 0.56 and

Table 7 Ablation study regarding the meta-explanation techniques

		TEDS		CCA		HDE		MedN	
		NN2	NN3	NN2	NN3	NN2	NN3	NN2	NN3
Original	<i>Mean</i>	7.77	6.96	4.25	4.34	17.18	19.21	3.54	3.74
	<i>Median</i>	7.41	6.29	4.45	4.59	17.12	18.45	3.74	3.83
	<i>inXAI</i>	8.55	7.91	5.46	5.37	17.18	19.20	3.55	3.74
	<i>MetaLion</i>	2.87	1.68	1.36	1.35	11.70	15.09	0.50	0.29
w/o IG	<i>Mean</i>	9.25↑	8.70↑	4.90↑	5.00↑	17.36↑	20.17↑	3.73↑	3.71↓
	<i>Median</i>	9.41↑	8.67↑	5.25↑	5.43↑	17.96↑	20.37↑	4.24↑	3.91↑
	<i>inXAI</i>	8.39↓	8.05↑	5.54↑	5.66↑	17.36↑	20.18↑	3.74↑	3.71↓
	<i>MetaLion</i>	4.67↑	3.31↑	1.53↑	1.52↑	12.13↑	15.37↑	0.90↑	0.59↑
w/o LRP	<i>Mean</i>	7.81↑	7.81↑	4.91↑	4.90↑	18.90↑	20.52↑	3.89↑	4.26↑
	<i>Median</i>	7.74↑	6.85↑	5.38↑	5.41↑	19.39↑	20.71↑	4.34↑	4.81↑
	<i>inXAI</i>	8.45↓	7.93↑	5.57↑	5.69↑	18.90↑	20.52↑	3.89↑	4.25↑
	<i>MetaLion</i>	3.05↑	1.77↑	1.51↑	1.56↑	12.90↑	15.63↑	1.07↑	1.21↑
w/o LIME	<i>Mean</i>	6.65↓	5.51↓	4.40↑	4.30↓	16.72↓	18.75↓	3.99↑	4.36↑
	<i>Median</i>	5.78↓	6.85↑	4.38↓	4.55↓	16.68↓	17.78↓	4.16↑	4.19↑
	<i>inXAI</i>	8.56↑	7.56↓	5.58↑	5.75↑	16.72↓	18.75↓	4.00↑	4.36↑
	<i>MetaLion</i>	2.95↑	1.73↑	2.60↑	2.48↑	12.32↑	15.47↑	1.02↑	0.64↑
w/o Random	<i>Mean</i>	7.52↓	6.53↓	4.05↓	3.98↓	16.85↓	18.51↓	3.45↓	3.57↓
	<i>Median</i>	7.08↓	5.36↓	4.48↑	4.66↑	17.22↑	17.92↓	3.92↑	3.64↓
	<i>inXAI</i>	8.71↑	8.10↑	6.00↑	5.95↑	16.84↓	18.51↓	3.44↓	3.56↓
	<i>MetaLion</i>	3.00↑	1.75↑	2.37↑	2.39↑	12.66↑	15.60↑	0.92↑	0.58↑

0.38. We assume that adding more seed explainability techniques will significantly enhance these observations.

5.2.4 Parameters impact

In the prior experiments, we identified IG as the best explainability technique, and our meta-explanation technique as the best across the other meta-explanation techniques. We select these two, as well as inXAI and LIME, which the former is one possible competitor and the latter probably the most popular explainability technique, to analyse how they perform given different *noise* and δ values in our cases, as tested in the NN2 model.

In Table 8, we can see the performance of the techniques given the different *noise* and δ values changes on the NN2 neural network. In every case, “weak” *noise* and higher δ values (e.g., 0.01) produce higher *truthfulness* scores and do not allow to easily distinguish between techniques. On

the other hand, “strong” *noise* and $\delta = 0$ is very strict and punitive. Therefore, we suggest the use of “normal” *noise* with a small $\delta = 0.0001$, but not 0.

5.2.5 Comparison with other metrics

One last quantitative experiment compares the *truthfulness* metric to another well-known explainability metrics, including as *complexity*, *stability*, and *consistency*. *Complexity* measures the number of non-zero weights included in an explanation. Lower scores in this metric suggest lower complexity, which means more comprehensible explanations. We can include a *complexity* threshold. The importance scores that fall below that threshold are then regarded zero, and the number of non-zero elements is reduced. In the results shown in Table 9, where we use “normal” *noise*, we use the δ values as the *complexity* threshold as well. For $\delta = 0.0001$, the evaluation of the different techniques is unclear, and we can-

Table 8 Performance of IG, LIME, *inXAI*, and *MetaLion* technique on *truthfulness* based on different *noise* and δ values

		Weak				Normal				Strong			
		0	0.0001	0.001	0.01	0	0.0001	0.001	0.01	0	0.0001	0.001	0.01
TEDS	IG	3.5	3.4	2.9	1.1	4.2	4.1	3.8	2.3	4.8	4.8	4.5	3.6
	LIME	11.0	10.9	10.0	6.0	11.5	11.5	11.0	8.6	12.0	12.0	11.8	10.4
	<i>inXAI</i>	8.2	8.1	7.3	4.1	8.6	8.6	8.2	6.1	9.1	9.0	8.8	7.5
	<i>MetaLion</i>	2.2	2.1	1.8	0.6	2.9	2.8	2.6	1.5	3.8	3.7	3.6	2.6
CCA	IG	4.3	4.2	3.9	2.6	4.5	4.5	4.3	3.2	4.8	4.8	4.7	3.9
	LIME	6.2	6.3	5.9	3.8	6.5	6.5	6.3	4.8	6.6	6.6	6.5	5.5
	<i>inXAI</i>	5.5	5.5	5.2	3.5	5.6	5.6	5.5	4.3	5.8	5.8	5.7	4.9
	<i>MetaLion</i>	1.1	1.0	0.9	0.5	1.4	1.4	1.2	0.8	1.8	1.7	1.7	1.2
HDE	IG	17.6	15.3	10.9	4.0	19.7	18.1	14.5	7.4	20.6	19.6	16.9	10.2
	LIME	20.7	18.4	13.2	4.8	22.3	20.7	16.8	8.4	23.2	22.2	19.3	11.7
	<i>inXAI</i>	16.2	14.1	9.8	3.3	18.7	17.2	13.6	6.7	20.4	19.4	16.7	10.0
	<i>MetaLion</i>	9.4	7.8	4.9	1.3	13.0	11.7	8.9	3.7	15.3	14.3	11.8	6.2
MedN	IG	-	-	-	-	21.2	4.3	1.9	0.6	-	-	-	-
	LIME	-	-	-	-	22.9	4.3	1.8	0.5	-	-	-	-
	<i>inXAI</i>	-	-	-	-	19.6	3.6	1.5	0.4	-	-	-	-
	<i>MetaLion</i>	-	-	-	-	2.6	0.5	0.2	0.1	-	-	-	-

not easily choose the best technique. For example, in TEDS dataset, both IG and LIME have the same *complexity*, but they have very different *truthfulness* scores, with LIME making twice the mistakes as IG.

Another intriguing finding from the *complexity* metric is that *MetaLion*, which always shows the highest *truthfulness* score, also has the lowest *complexity* scores in three of the four test cases. This means that the meta-explanation technique can reduce the number of elements that appear

to the end user, making the explanations shorter and easier to understand, while guaranteeing that the remaining importance scores are truthful. This is due to the meta-explanation technique just replacing the incorrect components with zero. Given that the *truthfulness* score is always lower, this replacement is most likely correct. As a result, it not only ensembles, but also corrects the seed explainability techniques.

Additionally, we plan to evaluate all seed and meta-explanation techniques using *stability* and *consistency* met-

Table 9 Comparison of *truthfulness* and *complexity* between IG, LIME, *inXAI*, and *MetaLion* with different δ values

		0		0.0001		0.001		0.01	
		Tr	Co	Tr	Co	Tr	Co	Tr	Co
TEDS	IG	4.2	14.0	4.1	13.98	3.8	13.72	2.3	11.43
	LIME	11.5	14.0	11.5	13.98	11.0	13.78	8.6	11.89
	<i>inXAI</i>	8.6	14.0	8.6	14.0	8.2	13.58	6.1	10.12
	<i>MetaLion</i>	2.9	11.08	2.8	11.11	2.6	11.12	1.5	9.19
CCA	IG	4.5	10.28	4.5	10.28	4.3	10.26	3.2	10.03
	LIME	6.5	11.0	6.5	10.99	6.3	10.97	4.8	10.71
	<i>inXAI</i>	5.6	12.0	5.6	12.0	5.5	12.0	4.3	12.0
	<i>MetaLion</i>	1.4	10.62	1.4	10.62	1.2	10.71	0.8	10.64
HDE	IG	19.7	29.71	18.1	29.54	14.5	28.03	7.4	15.40
	LIME	22.3	26.05	20.7	25.98	16.8	25.37	8.4	19.08
	<i>inXAI</i>	18.7	29.71	17.2	29.53	13.6	27.75	6.7	12.46
	<i>MetaLion</i>	13.0	16.67	11.7	17.74	8.9	18.45	3.7	7.63
MedN	IG	21.2	48.25	4.3	48.13	1.9	47.17	0.6	38.27
	LIME	22.9	44.75	4.3	44.74	1.8	44.62	0.5	43.32
	<i>inXAI</i>	19.6	48.99	3.6	48.91	1.5	48.33	0.4	43.33
	<i>MetaLion</i>	2.6	45.41	0.5	41.63	0.2	32.88	0.1	18.01

rics. By examining the *stability* and *consistency* of the explanations, we can assess the robustness and reliability of the techniques. This evaluation will enable us to compare how these metrics align with the scores of *truthfulness*, providing a comprehensive analysis of the explanation quality across multiple dimensions.

Stability, also known as robustness, ensures consistent explanations for similar inputs. To evaluate *stability*, we utilize Lipschitz continuity, as discussed in [53], a modified concept of continuity. It measures the maximum difference in explanations between points within a defined neighbourhood. The neighbourhood is determined by a distance criterion denoted as ϵ , which ensures proximity between points.

For the CCA dataset, we set ϵ to the default value of 0.3. However, for TEDS, we found that 0.3 was inadequate as it resulted in the same score for all techniques. Therefore, we selected a higher value of 3. For datasets with higher-dimensional feature spaces like HDE and MedN, we searched for an ϵ value that could differentiate the techniques to some extent. We set ϵ to 50 in these cases. Although we would have preferred to increase this value further, the computational cost associated with increasing ϵ limited our search.

Table 10 presents the performance of each technique in terms of *stability*. While there is no clear winner, several interesting findings can be observed. In the case of the TEDS dataset, LIME demonstrates superior stability compared to all other techniques. Among the meta-explanation techniques, median shows promising results. For the CCA dataset, Random explanations exhibit the highest level of stability, while among the meta-explanation techniques, *inXAI* performs well. In the HDE dataset, all techniques achieve perfect stability, indicating the need for an increased value in the ϵ parameter. In the MedN dataset, IG and LRP achieve perfect *stability* scores, while all meta-explanation techniques demonstrate adequate performance. Lastly, we can observe that *MetaLion* is highly influenced by the instability of the seed techniques; thus, by removing instable ones, for exam-

ple, Random on TEDS, the stability of the technique is expected to be increased.

The last metric we will discuss is *consistency*, which measures the degree of variation in explanations generated by a technique for different models. Specifically, we consider our three models: NN1, NN2, and NN3. *Consistency* evaluates how different the explanations provided by a technique for these models are for each instance. It is important to note that the Random explainability technique produces the same explanation for an instance across all three models, leading to inflated *consistency* scores.

As seen in Table 11, in the case of TEDS, the seed techniques exhibit challenges with *consistency*, while the meta-explanation techniques perform better. For CCA, both the seed and meta-explanation techniques perform similarly, except for *inXAI*, which outperforms the others. A similar pattern is observed in HDE, while in MedN, IG demonstrates perfect *consistency* compared to the other techniques. Overall, *inXAI* produces the most consistent explanations across all datasets, with IG, *Mean*, and *MetaLion* following suit.

However, both metrics fail to identify a single technique as the best. It is worth noting that the *inXAI* meta-explanation technique, which aggregates the seed techniques by optimizing these two metrics, was included in our study. In contrast, the *truthfulness* metric achieves to identify the best technique, which is *MetaLion* which attempts to optimise this metric.

Furthermore, when comparing *inXAI* and *MetaLion*, the former optimizes metrics such as *stability*, *consistency*, and *area under the loss curve*. Thus, it is expected to perform slightly better.

5.3 Qualitative experiments

In this section, we will present two examples comparing explanations provided by IG, LRP, and *Mean*, with our meta-explanation technique from the textual (MedN) and image (HDE) datasets. In both cases, we use the NN3 neural model. Moreover, we will showcase how the argumentation framework can be employed to provide richer explanations.

Table 10 *Stability* per explainability technique

	TEDS			CCA			HDE			MedN		
	NN1	NN2	NN3	NN1	NN2	NN3	NN1	NN2	NN3	NN1	NN2	NN3
IG	0.78	0.74	0.80	0.84	0.73	0.70	0.98	<u>0.99</u>	<u>0.99</u>	1.00	1.00	1.00
LRP	0.78	0.56	0.68	0.84	0.74	0.71	0.98	<u>0.99</u>	<u>0.99</u>	1.00	1.00	1.00
LIME	0.98	0.87	0.88	0.68	0.69	0.68	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>	0.87	0.87	0.88
Random	0.62	0.62	0.62	0.83	0.83	0.83	<u>0.99</u>	<u>0.99</u>	<u>0.99</u>	0.98	<u>0.98</u>	<u>0.98</u>
<i>Mean</i>	0.87	0.79	<u>0.84</u>	0.87	0.79	0.77	<u>0.99</u>	1.00	1.00	0.93	0.92	0.93
<i>Median</i>	<u>0.88</u>	<u>0.80</u>	0.83	0.86	0.76	0.74	<u>0.99</u>	1.00	1.00	<u>0.99</u>	0.95	0.95
<i>inXAI</i>	0.80	0.76	0.80	<u>0.93</u>	<u>0.80</u>	<u>0.79</u>	1.00	1.00	1.00	0.95	0.95	0.97
<i>MetaLion</i>	0.66	0.72	0.79	0.78	0.74	0.74	<u>0.99</u>	<u>0.99</u>	1.00	0.95	0.91	0.93

Best is in bold, second best is underlined

Table 11 Consistency per explainability technique

	TEDS	CCA	HDE	MedN
IG	0.11	0.39	0.76	1.00
LRP	0.09	0.40	0.75	0.28
LIME	0.11	0.47	0.83	0.26
Random	1.00	1.00	1.00	1.00
<i>Mean</i>	<u>0.77</u>	<u>0.51</u>	<u>0.84</u>	0.49
<i>Median</i>	0.63	0.40	0.79	0.44
<i>inXAI</i>	0.89	0.88	0.86	<u>0.50</u>
<i>MetaLion</i>	0.64	0.36	0.82	0.34

Best is in bold, second best is underlined

We will start with the first example regarding an instance from the MedN dataset. In the first row in Fig. 5, we can see the examined instance. The prediction from the neural network regarding this medical report was 93% probability to concern acute ischemic stroke. Focusing on few specific words, “Diffusion”, “Restriction” (appearing 1st), “Restriction” (appearing 2nd), all of them have been assigned with a truthful weight from our meta-explanation technique, and

the corresponding arguments are presented below. For example, regarding the word “Diffusion”, which according to IG, LRP, *Mean*, and *inXAI* should have a negative weight, when it is removed, the probability drops. Therefore, the weight should have been positive, as our meta-explanation technique correctly assigned. The corresponding argument is the $f_{Diffusion,DEC}$.

$f_{Diffusion,DEC}$: The evaluation of the alteration of *Diffusion*’s value 1 to 0 (*DEC*) was performed and the model’s behaviour was as expected *DEC* (93% to 90%), according to its importance $z_{Diffusion} = 0.75$.

$f_{Restriction-1st,DEC}$: The evaluation of the alteration of *Restriction – 1st*’s value 1 to 0 (*DEC*) was performed and the model’s behaviour was as expected *DEC* (93% to 86%), according to its importance $z_{Restriction-1st} = 0.92$.

$f_{Restriction-2nd,INC}$: The evaluation of the alteration of *Restriction – 2nd*’s value 1 to 0 (*INC*) was performed and the model’s behaviour was as expected *INC* (93% to 95%), according to its importance $z_{Restriction-2nd} = -0.29$.

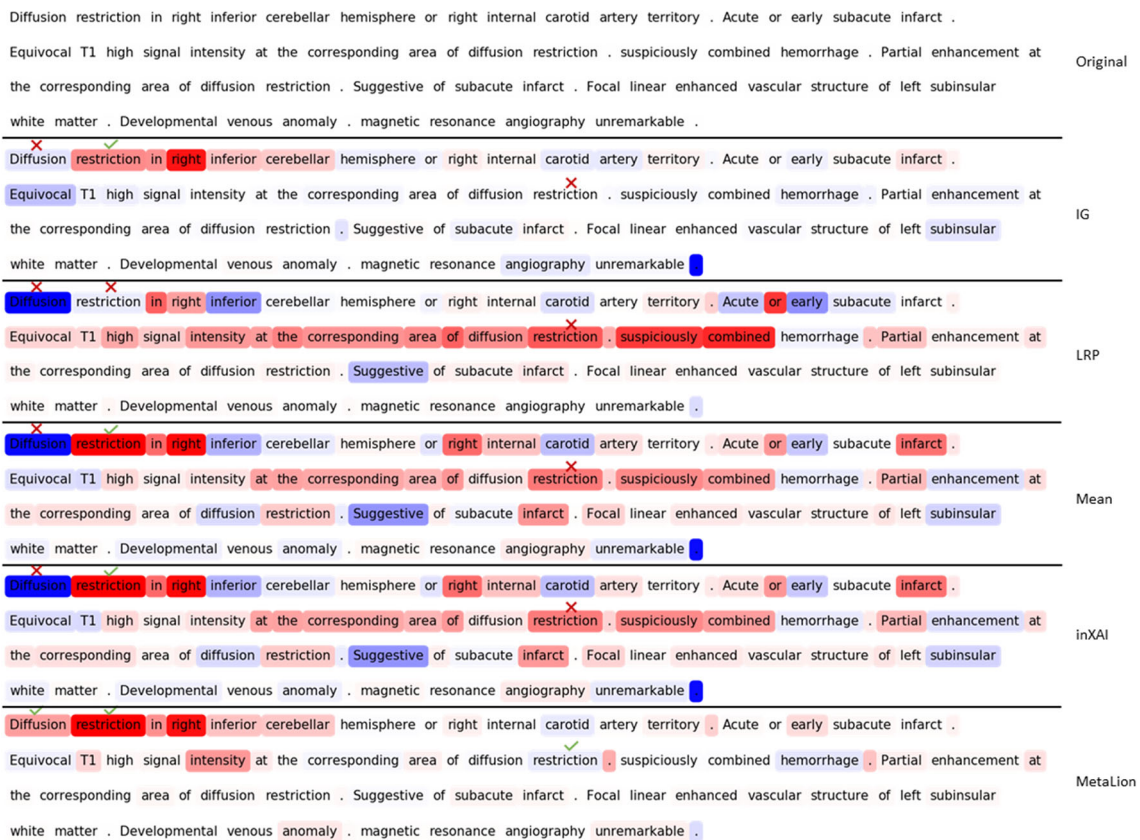


Fig. 5 MedN Example: Explanation provided by different explainability techniques for a specific instance. The colour red indicates that the word has a positive influence on the prediction, whereas the colour blue

suggests that it has a negative effect. Green mark indicates that the influence is correct, while red cross incorrect

In the second example (Fig. 6), we can see an examined instance, which is classified as “damaged” with a 60% probability. We can see the different explanations from the techniques, at the segment (superpixel) level. Red highlight indicates that the segment is very crucial to the prediction of this class, while blue for the other class. Our meta-explanation method achieves to both highlight fewer parts of the image, while it contains less untruthful weights. The segments, with number 1, 9 and 27, are presented. The weights assigned to all of them is correct in *MetaLion*, in contrast to the others. Segment 1 is negative to IG, LRP, *Mean* and *inXAI*, while it should have been positive. The arguments supporting this are $f_{Segment_1,DEC}$ and $f_{Segment_1,INC}$. Similarly, to segments 9 and 27.

$f_{Segment_1,DEC}$ The evaluation of the alteration of *Segment*₁'s value by -0.07 (*DEC*) was performed and the model's behaviour was as expected *DEC* (60.5% to 60.1%), according to its importance $z_{Segment_1} = 0.20$.

$f_{Segment_1,INC}$ The evaluation of the alteration of *Segment*₁'s value by $+0.07$ (*INC*) was performed and the model's behaviour was as expected *INC* (60.5% to 60.9%), according to its importance $z_{Segment_1} = 0.20$.

$f_{Segment_9,DEC}$ The evaluation of the alteration of *Segment*₉'s value by -0.26 (*DEC*) was performed and the model's behaviour was as expected *DEC* (60.5% to 39.2%), according to its importance $z_{Segment_9} = 0.24$.

$f_{Segment_9,INC}$ The evaluation of the alteration of *Segment*₉'s value by $+0.26$ (*INC*) was performed and the model's behaviour was as expected *INC* (60.5% to 62.7%), according to its importance $z_{Segment_9} = 0.24$.

$f_{Segment_{27},DEC}$ The evaluation of the alteration of *Segment*₂₇'s value by -0.14 (*DEC*) was performed and the model's behaviour was as expected *INC* (60.5% to 62.5%), according to its importance $z_{Segment_{27}} = -0.75$.

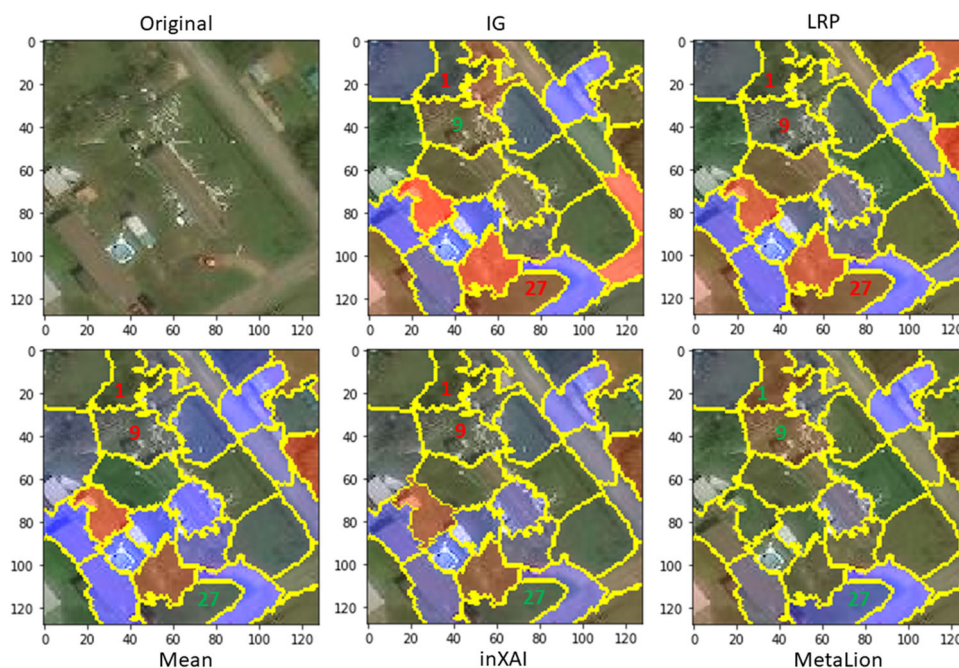
$f_{Segment_{27},INC}$ The evaluation of the alteration of *Segment*₂₇'s value by $+0.14$ (*INC*) was performed and the model's behaviour was as expected *DEC* (60.5% to 52.3%), according to its importance $z_{Segment_{27}} = -0.75$.

We believe that by presenting such arguments to the user, we may help them evaluate and choose the optimal method from among multiple options, as well as enhance their trust in the system. Specifying the reasons why an importance score is regarded truthful or untruthful can considerably increase trust in the explanation.

6 Conclusions

Machine learning models must be interpretable when used in high-risk applications. There are several techniques for explaining a model's decisions, as well as evaluation methods for determining the quality of the explanations. However, because there are so many alternatives, determining the best explanation technique for a given application can be challenging. While evaluation can assist in this process, it is not always the case. It would be extremely beneficial to give a user-friendly evaluation metric that would allow the combination of various explanation techniques via a meta-explanation technique.

Fig. 6 CCA Example: Explanation provided by different explainability techniques for a specific instance. The colour red indicates that the word has a positive influence on the prediction, whereas the colour blue suggests that it has a negative effect. Green numbers indicate that the weight of the segment is correct, while red numbers incorrect



The metric we introduced, truthfulness, is a good choice for a meta-explanation technique, whereas alternative faithfulness-based metrics produce outputs that would be difficult to incorporate into a local ensembling/meta-explanation technique. Truthfulness is appropriate since it filters the important scores of each explanation if they have the incorrect polarity based on the model's behaviour with a few alterations. Depending on this filtering, the meta-explanation technique may readily select the most appropriate truthful score among the scores based on the change in the output. As a result, with fully unsupervised methods, both explanation techniques and metrics, the meta-explanation we provide seems to be an appropriate choice.

Through large-scale experimentation, we explored the ability of *truthfulness* to accurately assess explanations, intrinsic or not, in four datasets of varied data types. Then, while conducting an ablation study, we discussed the performance of meta-explanation techniques, demonstrating that *MetaLion* always performs better when the number of input (seed) explanations is increased, even when noisy, contradicting explanations are included, whereas other methods perform better when noisy or erroneous explanations are omitted. This allows us to freely add explanations as seeds in our meta-explanation, considering solely the additional computational overhead as it improves efficiency while overlooking potential noise.

Although we used IG, LIME, and LRP in our experiments, our technique is not limited to them. To improve the performance of the meta-explanation technique, the end user can easily replace or add new ones. Nonetheless, they must always keep in mind that the additional explanation techniques must be consistent with the two assumptions (Assumptions 1 and 2).

A discussion about the two *truthfulness* parameters, *noise* and δ , took place as well, to allow users to select which values are most appropriate for their applications. A “strong” *noise* with a δ value of 0 is recommended for a more stringent evaluation. However, in most circumstances, “normal” *noise* with a δ value of 0.0001 would suffice. In most situations, such difference in prediction is almost minor, hence we recommend this setting. We also compared *truthfulness* to the *complexity*, *stability*, and *consistency* metrics. Based on this comparison, we wanted to highlight that our meta-explanation technique almost always delivers both the most truthful and the least complex explanations, especially as the δ value increases, while achieving decent performance in other metrics as well.

Lastly, we demonstrated a qualitative experiment using two examples from two distinct datasets. We contrast our meta-explanation technique, *MetaLion*, with various explainability techniques. *MetaLion* is less complex and more truthful. The *truthfulness* of each important score is additionally supported by a few arguments, which assist the end

user trust the explanation. Those arguments are re-phrased compared to the originals, as presented in our preliminary work [26].

6.1 Limitations

While our work offers several advantages, it is important to acknowledge its limitations. Firstly, concerning the *truthfulness* metric, the alterations made may not fully capture the behaviour of the underlying model accurately. This limitation could potentially impact the evaluation of explanations.

Additionally, the *truthfulness* metric is dependent on models that produce continuous outputs in regression tasks or probabilities in classification tasks. As a result, explanations for models like decision trees may not be suitable candidates for evaluation using this metric or other faithfulness-based metrics. Conversely, explanations for neural networks, where logits or output of activation functions (such as softmax/sigmoid) can be used directly, as well as explanations for probabilistic models and ensembles, can be appropriately evaluated using this metric.

Furthermore, it is important to note that *MetaLion*, like the seed explanation techniques it builds upon, does not consider feature dependencies. This limitation arises from the nature of the seed techniques themselves. It is an area that requires further exploration and consideration in future research.

6.2 Future directions

Our future objectives encompass several aspects. Firstly, we aim to conduct a large-scale experiment to compare the *truthfulness* metric with other metrics, investigating any potential correlations between them. Additionally, we seek to explore methods to enhance the *truthfulness* metric for even more accurate evaluations. One approach is to investigate alternative techniques, such as interpolation, to capture the polarity more accurately within a given range of values, replacing the current two alterations performed for each feature value.

Moreover, we plan to extend our analysis to incorporate different machine learning models, such as Transformers, and explore their compatibility with *MetaLion*. This expansion would also involve considering various tasks, including multi-class and multi-label classification, to assess the applicability of the approach across different domains. Furthermore, we intend to enhance both the argumentation framework and *MetaLion* itself. By refining these components, we aim to improve the overall effectiveness and usability of the approach. In addition, we aim to investigate the feasibility of incorporating alternative metrics, such as stability, consistency, and others, in *MetaLion*'s ensembling procedure.

Another potential future direction is to incorporate explanation techniques that consider feature dependencies and

investigate the applicability of *MetaLion* in such scenarios. By exploring the use of *MetaLion* in conjunction with explanation techniques that account for feature dependencies, we can assess its usability and effectiveness in capturing and presenting explanations in more complex and interdependent feature spaces. Finally, we aim to conduct a human-centred experiment to demonstrate the preference of end users for meta-explanation techniques. We also plan to carry out experiments involving domain experts, like those presented in recent studies [62].

Acknowledgements This paper is supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 825619 (AI4EU Project).

Funding Open access funding provided by HEAL-Link Greece. This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 825619 (AI4EU). All the authors are beneficiaries of this funding.

Availability of Data and Materials All datasets used in these research are public and freely available. Reference for all of them are provided throughout the manuscript.

Code Availability Experiments' code is available in the GitHub repository: <https://github.com/iamollas/MetaLion-Truthful-Meta-Explanation.git>

Declarations

Competing of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Carvalho TP, Soares FAAMN, Vita R, da P Francisco R, Basto JP, Alcalá SGS (2019) A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering* 137:106024. <https://doi.org/10.1016/j.cie.2019.106024>
- Paraschos, S, Mollas, I, Bassiliades, N, Tsoumakas, G (2021) Visioed: A visualisation tool for interpretable predictive maintenance. In: Proceedings of the 30th international joint conference on artificial intelligence, IJCAI-21. International Joint Conferences on Artificial Intelligence Organization
- Dastile X, Celik T, Potsane M (2020) Statistical and machine learning models in credit scoring: A systematic literature survey. *Appl Soft Comput* 91:106263. <https://doi.org/10.1016/j.asoc.2020.106263>
- Dumitrescu E, Hué S, Hurlin C, Tokpavi S (2021) Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*. <https://doi.org/10.1016/j.ejor.2021.06.053>
- Leo, M, Sharma, S, Maddulety, K (2019) Machine learning in banking risk management: A literature review. *Risks* 7(1). <https://doi.org/10.3390/risks7010029>
- Roy, R, George, KT (2017) Detecting insurance claims fraud using machine learning techniques. In: 2017 International conference on circuit, power and computing technologies (ICCPCT), pp 1–6. <https://doi.org/10.1109/ICCPCT.2017.8074258>
- Imaam, F, Subasinghe, A, Kasthuriarachchi, H, Fernando, S, Had-dela, P, Pemadasa, N: Moderate automobile accident claim process automation using machine learning. In: 2021 International conference on computer communication and informatics (ICCCI), pp 1–6 (2021). <https://doi.org/10.1109/ICCCI50826.2021.9457017>
- Ngiam KY, Khor IW (2019) Big data and machine learning algorithms for health-care delivery. *The Lancet Oncology* 20(5):262–273. [https://doi.org/10.1016/S1470-2045\(19\)30149-4](https://doi.org/10.1016/S1470-2045(19)30149-4)
- Thomsen K, Iversen L, Titlestad TL, Winther O (2020) Systematic review of machine learning for diagnosis and prognosis in dermatology. *J Dermatol Treat* 31(5):496–510. <https://doi.org/10.1080/09546634.2019.1682500>
- Yue, W, Wang, Z, Chen, H, Payne, A, Liu, X (2018) Machine learning with applications in breast cancer diagnosis and prognosis. *Designs* 2(2). <https://doi.org/10.3390/designs2020013>
- Zhang Y, An R, Liu S, Cui J, Shang X (2023) Predicting and understanding student learning performance using multi-source sparse attention convolutional neural networks. *IEEE Transactions on Big Data* 9(1):118–132. <https://doi.org/10.1109/TBDDATA.2021.3125204>
- Mollas I, Chrysopoulou Z, Karlos S, Tsoumakas G (2022) Ethos: a multi-label hate speech detection dataset. *Complex & Intelligent Systems* 8(6):4663–4678
- Hao, K (2020) The coming war on the hidden algorithms that trap people in poverty. <https://cutt.ly/2QtXHHt>. Accessed: 2021-07-27
- Rose, J (2021) An Insurance Startup Bragged It Uses AI to Detect Fraud. It Didn't Go Well. <https://cutt.ly/3QtDt0i>. Accessed: 2021-07-27
- Chen, A (2018) IBM's Watson gave unsafe recommendations for treating cancer. <https://cutt.ly/keHQDma>. Accessed: 2021-07-27
- Reuter, E (2021) In scramble to respond to Covid-19, hospitals turned to models with high risk of bias. <https://cutt.ly/hQtJra0>. Accessed: 2021-07-27
- Regulation GDP (2016) Regulation (eu) 2016/679 of the european parliament and of the council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *Official Journal of the European Union (OJ)* 59(1–88):294
- European Commission (2021) Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts. <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206>
- Dosilovic, FK, Brcic, M, Hlupic, N (2018) Explainable artificial intelligence: A survey. In: 2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO), pp 0210–0215. *IEEE*. <https://doi.org/10.23919/MIPRO.2018.8400040>
- Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access* 6:52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>

21. Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B (2019) Interpretable machine learning: definitions, methods, and applications. *Proceedings of the National Academy of Sciences* 116(44):22071–22080. <https://doi.org/10.1073/pnas.1900654116>
22. Linardatos P, Papastefanopoulos V, Kotsiantis S: Explainable AI, (2020) A review of machine learning interpretability methods. *Entropy* 23(1):18. <https://doi.org/10.3390/e23010018>
23. Melis, DA, Jaakkola, T (2018) Towards robust interpretability with self-explaining neural networks. In: *Advances in neural information processing systems*, pp 7775–7784
24. Du, M, Liu, N, Yang, F, Ji, S, Hu, X (2019) On attribution of recurrent neural network predictions via additive decomposition. In: *The world wide web conference*, pp 383–393
25. Yeh, C, Hsieh, C, Suggala, AS, Inouye, DI, Ravikumar, P (2019) On the (in)fidelity and sensitivity of explanations. In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, Vancouver, BC, Canada*, pp 10965–10976. <https://proceedings.neurips.cc/paper/2019/hash/a7471fdc77b3435276507cc8f2dc2569-Abstract.html>
26. Mollas, I, Bassiliades, N, Tsoumakas, G (2022) Altruist: Argumentative explanations through local interpretations of predictive models. In: *Proceedings of the 12th hellenic conference on artificial intelligence. SETN '22. Association for Computing Machinery*. <https://doi.org/10.1145/3549737.3549762>
27. Bhatt, U, Weller, A, Moura, JMF (2020) Evaluating and aggregating feature-based model explanations. In: Bessiere, C. (ed.) *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI 2020, Online*, pp 3016–3022. <https://doi.org/10.24963/ijcai.2020/417>
28. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51(5):1–42
29. Guidotti R, Monreale A, Giannotti F, Pedreschi D, Ruggieri S, Turini F (2019) Factual and counterfactual explanations for black box decision making. *IEEE Intell Syst* 34(6):14–23
30. Guidotti, R, Ruggieri, S (2019) On the stability of interpretable models. In: *2019 International joint conference on neural networks (IJCNN)*, pp 1–8. IEEE
31. Simari GR, Rahwan I (2009) *Argumentation in Artificial Intelligence* vol 47. Springer. <https://doi.org/10.1007/978-0-387-98197-0>
32. Bex F, Walton D (2016) Combining explanation and argumentation in dialogue. *Argument & Computation* 7(1):55–68
33. Friedman JH, Popescu BE (2008) Predictive learning via rule ensembles. *The Annals of Applied Statistics* 2(3):916–954
34. Ribeiro, MT, Singh, S, Guestrin, C (2016) Why should i trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp 1135–1144. ACM
35. Mollas, I, Bassiliades, N, Tsoumakas, G (2022) Conclusive local interpretation rules for random forests. *Data Mining and Knowledge Discovery*, pp 1–54
36. Ribeiro, MT, Singh, S, Guestrin, C (2018) Anchors: High-precision model-agnostic explanations. In: McIlraith, SA, Weinberger, KQ (eds.) *Proceedings of the thirty-second aaii conference on artificial intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), February 2–7, 2018*, pp 1527–1535. AAAI Press. <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982>
37. Lundberg, SM, Lee, S (2017) A unified approach to interpreting model predictions. In: Guyon, I, von Luxburg, U, Bengio, S, Wallach, HM, Fergus, R, Vishwanathan, SVN, Garnett, R (edn.) *Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, Long Beach, CA, USA*, pp 4765–4774. <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>
38. Breiman L (2001) Random forests. *Machine Learning* 45(1):5–32. <https://doi.org/10.1023/A:1010933404324>
39. Bach S, Binder A, Montavon G, Klauschen F, Müller K-R, Samek W (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE* 10(7):1–46. <https://doi.org/10.1371/journal.pone.0130140>
40. Sundararajan, M, Taly, A, Yan, Q (2017) Axiomatic attribution for deep networks. In: Precup, D, Teh, YW (eds.) *Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017. Proceedings of Machine Learning Research, vol 70*, pp 3319–3328. PMLR, Sydney, Australia <http://proceedings.mlr.press/v70/sundararajan17a.html>
41. Atkinson K, Bench-Capon T, Bollegala D (2020) Explanation in ai and law: Past, present and future. *Artif Intell* 289:103387. <https://doi.org/10.1016/j.artint.2020.103387>
42. Vassiliades, A, Bassiliades, N, Patkos, T (2021) Argumentation and explainable artificial intelligence: a survey. *The Knowledge Engineering Review* 36
43. Besnard, P, Hunter, A (2009) In: Simari, G, Rahwan, I (eds.) *Argumentation based on classical logic*, pp 133–152. Springer. https://doi.org/10.1007/978-0-387-98197-0_7
44. Chan, CS, Kong, H, Guanqing, L (2022) A comparative study of faithfulness metrics for model interpretability methods. In: *Proceedings of the 60th annual meeting of the association for computational linguistics (Volume 1: Long Papers)*, pp 5029–5038. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.acl-long.345>
45. Chrysostomou, G, Aletras, N (2021) Improving the faithfulness of attention-based explanations with task-specific information for text classification. In: *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (Volume 1: Long Papers)*, pp 477–488
46. Serrano, S, Smith, NA (2019) Is attention interpretable? In: *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp 2931–2951. Association for Computational Linguistics. <https://doi.org/10.18653/v1/P19-1282>
47. DeYoung, J, Jain, S, Rajani, NF, Lehman, E, Xiong, C, Socher, R, Wallace, BC (2020) ERASER: A benchmark to evaluate rationalized NLP models. In: Jurafsky, D, Chai, J, Schluter, N, Tetreault, J.R. (eds.) *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pp 4443–4458. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.408>
48. Luss, R, Chen, P-Y, Dhurandhar, A, Sattigeri, P, Zhang, Y, Shanmugam, K, Tu, C-C (2021) Leveraging latent features for local explanations. In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp 1139–1149
49. Liu, Y, Li, H, Guo, Y, Kong, C, Li, J, Wang, S (2022) Rethinking attention-model explainability through faithfulness violation test. In: *ICML. Proceedings of machine learning research, vol 162*, pp 13807–13824. PMLR
50. Hedström A, Weber L, Krakowczyk D, Bareeva D, Motzkus F, Samek W, Lapuschkin S, Höhne MM-C (2023) Quantus: An explainable ai toolkit for responsible evaluation of neural network explanations and beyond. *J Mach Learn Res* 24(34):1–11
51. Tan, C (2022) On the diversity and limits of human explanations. In: Carpuat, M, de Marneffe, M, Ruíz, I.V.M. (eds.) *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: human language technologies, NAACL 2022, July 10–15, pp 2173–2188. Association for Computational Linguistics*. <https://aclanthology.org/2022.naacl-main.158>

52. Robnik-Sikonja, M, Bohanec, M (2018) Perturbation-based explanations of prediction models. In: Human and machine learning - visible, explainable, trustworthy and transparent, pp 159–175. Springer. https://doi.org/10.1007/978-3-319-90403-0_9
53. Bobek, S, Bałaga, P, Nalepa, GJ (2021) Towards model-agnostic ensemble explanations. In: International conference on computational science, pp 39–51. Springer
54. Hamamoto, M, Egi, M (2021) Model-agnostic ensemble-based explanation correction leveraging rashomon effect. In: 2021 IEEE symposium series on computational intelligence (SSCI), pp 01–08. IEEE
55. Guidotti, R (2022) Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery, pp 1–55
56. Saxena, A, Goebel, K (2008) Turbofan engine degradation simulation data set. NASA Ames Prognostics Data Repository
57. Saxena, A, Goebel, K, Simon, D, Eklund, N (2008) Damage propagation modeling for aircraft engine run-to-failure simulation. In: 2008 International conference on prognostics and health management, pp 1–9. IEEE
58. Cao, QD, Choe, Y (2018) Detecting Damaged buildings on post-hurricane satellite imagery based on customized convolutional neural networks. IEEE Dataport. <https://doi.org/10.21227/sdad-1e56>
59. Kim C, Zhu V, Obeid J, Lenert L (2019) Natural language processing and machine learning algorithm to identify brain mri reports with acute ischemic stroke. PLOS ONE 14(2):1–13. <https://doi.org/10.1371/journal.pone.0212778>
60. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36(4):1234–1240
61. Guidotti R (2021) Evaluating local explanation methods on ground truth. Artif Intell 291:103428. <https://doi.org/10.1016/j.artint.2020.103428>
62. Yun, Y, Dai, H, Zhang, Y, Wei, S, Shang, X (2022) Interpretable educational recommendation: An open framework based on bayesian principal component analysis. In: 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp 3409–3414. <https://doi.org/10.1109/SMC53654.2022.9945498>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Ioannis Mollas holds a Ph.D. from the Aristotle University of Thessaloniki. He conducted extensive research in the field of explainable artificial intelligence, focusing particularly on interpretable machine learning.



Nick Bassiliades received his MSc in Applied Artificial Intelligence from the Computing Science Department of Aberdeen University, in 1992, and his PhD degree in parallel knowledge base systems from the School of Informatics at the Aristotle University of Thessaloniki (AUTH), Greece, in 1998, where he is currently a Professor. His research interests include knowledge-based and rule systems, multiagent systems, ontologies / knowledge graphs / Semantic Web, Electric Vehicles charging scheduling and explainable AI. He has published more than 250 papers at journals, conferences, and books, has coauthored 5 books and co-edited 11 volumes. His published research has received over 5400 citations (h-index 35), while 7 of his papers have received awards. He was the Program Chair of 11 conferences / workshops, he has been member of the Program Committee of more than 150 and on the Organizational Committee of 9 conferences / workshops. He has been involved in 40 R&D projects leading 11 of them. He has been the general secretary of the Board of the Greek Artificial Intelligence Society; he is a director of RuleML, Inc., and also a member of the Greek Computer Society, the IEEE, and the ACM. He is currently serving as the head of the IT Center of AUTH.



Grigorios Tsoumakas is an Assoc. Professor of Machine Learning and Knowledge Discovery at the School of Informatics of the Aristotle University of Thessaloniki (AUTH) in Greece. He received a degree in Computer Science from AUTH in 1999, an MSc in Artificial Intelligence from the University of Edinburgh, United Kingdom, in 2000 and a Ph.D. in computer science from AUTH in 2005. His research expertise focuses on supervised learning techniques (ensemble methods, multi-target prediction) and natural language processing (semantic indexing, keyphrase extraction, summarization). He has published more than 150 research papers and according to Google Scholar he has more than 16,000 citations and an h-index of 50. Dr. Tsoumakas is a senior member of ACM and IEEE. His honors include receiving the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD) 10- Year Test of Time Award in 2017 and the Marco Ramoni best paper award at the 19th International Conference on Artificial Intelligence in Medicine (AIME 2021). He is an advocate of applied research that matters and has worked as a machine learning engineer, researcher and consultant in several national, international, and private sector funded R&D projects. In February 2019, he co-founded Medoid AI, a spin-off company of AUTH developing AI solutions based on machine learning technology.