CHAPTER 13

User tests & techniques for the post-hoc explanation of deep learning

Eoin Delaney[a,b], **Eoin M. Kenny**[a,b], **Derek Greene**[a,b], **and Mark T. Keane**[a,b]
[a]School of Computer Science, University College Dublin, Dublin, Ireland
[b]VistaMilk SFI Research Centre, Dublin, Dublin, Ireland

13.1. Introduction

In recent years, following the significant breakthroughs in Deep Learning, the Artificial Intelligence (AI) community has turned to the problem of eXplainable AI (XAI), mainly because of rising public concern about the use of these technologies in people's everyday lives, jobs, and leisure time (Ala-Pietilä and Smuha, 2021). Indeed, for the research community, there is very real worry that issues of interpretability, trust, and ethical usage will limit or block the deployment of these AI technologies. For these reasons, the DARPA XAI program has specifically targeted XAI research with a strong user testing emphasis to overcome such impasses (Gunning and Aha, 2019). At the same time, governments have also woken up to the need for regulation in this space; for example, the European Commission has established the High Level Expert Group on Artificial Intelligence to define guidelines for Trustworthy AI in the European Union, as a precursor to further legal steps (Ala-Pietilä and Smuha, 2021). Indeed, in the EU, GDPR places requirements on the need to explain automated decisions (Wachter et al., 2017). This wave of activity around the notion of explainability and XAI is also spawning new subareas of research; for example, XAI methods to support users in reversing algorithmic decisions – so-called *algorithmic recourse* – have emerged as a vibrant research topic (Karimi et al., 2020b). In this chapter, we present some recent solutions to the XAI problem using several variants of post-hoc explanations-by-example (Kenny et al., 2020; Keane et al., 2021). In the remainder of this introduction, we first consider the concept of "explanation" in XAI before outlining some of the different strategies explored in the literature. Then, in the remaining sections of the chapter we present new empirical evidence on how these different methods perform in dealing with image and time series datasets, along with reviewing what has been learned from user studies on their application.

13.1.1 What is an explanation? Pre-hoc versus post-hoc

One of the key problems facing XAI research is that the notion of "explanation" is not well defined and is still debated in Philosophy and Psychology (Sørmo et al., 2005).

Explainable Deep Learning AI
https://doi.org/10.1016/B978-0-32-396098-4.00019-3

One response to this issue in XAI has been to use terms other than "explanation", such as "interpretability," "transparency," and "simulatability"; but these renamings do not circumvent the fundamental problem that "explanation" is a slippery, hard-to-define concept. However, one broad distinction from philosophy that has gained general acceptance is the distinction between "explanation proper" and "explanation as justification." Sørmo et al. (2005) cast this philosophical distinction as the difference between explaining *how* the system reached some answer (what they call transparency) and explaining *why* the system produced a given answer (post-hoc justification). Lipton (2018) echoes these ideas with a similar distinction between transparency (i.e., "How does the model work?"; what we call pre-hoc explanation) and post-hoc explanation (i.e., "What else can the model tell me?").

Pre-hoc explanations promise to, in some sense, explain the Deep Learning model directly. So, the user can understand how the whole model works given some representation of it (Frosst and Hinton, 2017) via simplified model that "behaves similarly to the original model, but in a way that is easier to explain" (Lipton, 2018) (e.g., Frosst and Hinton (2017)). The claim here is that the model is inherently "transparent," "simulatable," or "interpretable" by virtue of how it runs. Rudin (2019) argues that this use of inherently transparent models is the only appropriate solution to XAI in sensitive, high-stakes domains; pointing to her own use of prototypes (Chen et al., 2018). However, the literature is not replete with many examples of this type of solution; indeed, the idea that one could "show" the inner workings of a Deep Learner to an "ordinary" end-user seems somewhat implausible as a proposition. Furthermore, many of the solutions which claim to be pre-hoc "transparent machine learning" are, actually, post-hoc solutions. For example, some model "simplifications" are really mappings of the Deep Learner into another modeling method (e.g., decision trees), what some call *proxy or surrogate models* (Gilpin et al., 2018), rather than direct renderings of the original neural network. As such, most of the XAI literature really concerns itself with post-hoc methods.

Post-hoc explanations provide after-the-fact justifications for what the Deep Learner has done. The key idea here is that one can explain/justify how a model reached some decision with reference to other information (e.g., "the model did this because it used such-and-such data"). This approach involves a broad spectrum of approaches involving many different techniques that try to provide evidential justifications for why a black-box model did what it did. Almost by definition, this means that these approaches are approximate; often, they do not directly show what was done to reach a prediction, but provide some basis for understanding why a prediction arose. There are probably four main approaches taken in the post-hoc explanation sphere: proxy-models, example-based explanations, natural language accounts and visualizations (see also Lipton (2018); Keane et al. (2021); Kenny et al. (2020)).

13.1.2 Post-hoc explanations: four approaches

The four main solutions to post-hoc explanation – proxy-models, example-based explanations, natural language accounts and visualizations – present quite different alternatives to the XAI problem. Here, we sketch each in turn, before going on to consider example-based explanations in some detail.

Proxy-model solutions provide some post-hoc mapping of some aspect of the Deep Learner into a "more transparent" modeling framework; for example, the Deep Learner is rerendered as a decision tree or a rule-based system that is said to explain its functioning. In general, these solutions assume that the proxy model is inherently transparent, as an article of faith without any substantiation for whether end-users actually find the proxy model comprehensible (Doshi-Velez and Kim, 2017). As Lipton (2018) points out "neither linear models, rule-based systems, nor decision trees are intrinsically interpretable... Sufficiently high-dimensional models, unwieldy rule lists, and deep decision trees could all be considered less transparent than comparatively compact neural networks." This means that to some degree, the jury is still out on the status and success of these proxy-model approaches. However, to be positive, there are now more user studies on people's understanding of rule-based explanations being carried out (Lage et al., 2019).

Example-based explanations arise out of long-standing case-based reasoning approaches to explanation (Sørmo et al., 2005; Nugent et al., 2009), where a case/precedent/example is used to provide a justification for a prediction (e.g., my house is valued at $400k because it is very similar to your house which sold for $400k). However, traditionally, example-based explanations were only used for *k*-NNs with only a handful of papers attempting to extend them to explaining multilayer perceptrons (Caruana et al., 1999; Shin and Park, 1999). More recent work has extended example-based explanation to Deep Learning models for classification, regression and natural language processing (Kenny and Keane, 2019, 2021a). The latter have been described as *twin-systems*, in which a black-box model is paired with a white-box model with the functionality of the former being mapped into the latter to find explanations (Kenny et al., 2021a; Kenny and Keane, 2021a). This twinning notion is very similar to the proxy-model idea, but subtly differs in that, typically, the white-box's function is purely explanatory; its sole role is to elucidate the predictions of the black box. In proxy-model approaches, the proxy often takes over the predictive role (as well as having an explanatory one), with evaluations being directed at establishing the fidelity of the proxy's predictions to those of the black box (see, e.g., White and d'Avila Garcez (2019); Guidotti et al. (2019)). A second major development in this example-based approach has been the proposal of different types of example-based explanations. Traditional case-based explanations use *factual examples*; they use instances from the dataset to directly explain a prediction (e.g., the house-price example). Recently, researchers have proposed counterfactual and

semifactual example-based explanations, opening up whole new vistas for post-hoc explanations (see Section 13.1.3).

Natural language explanations are a third, post-hoc option reflecting a long history of attempts to turn AI model predictions or decision traces into natural language descriptions to be read by end-users (see, e.g., Camburu et al. (2020); Shortliffe et al. (1975); Nugent et al. (2009)). Traditionally, this approach tries to take some aspect of the model – such as its rules or outputs – and render it in a natural language description, on the assumption that users will then find the models workings more comprehensible. Obviously, this natural language processing step does not in itself guarantee that such an explanation will work, as it will also depend on what is being explained.

Visualizations are the final post-hoc XAI solution, one that has received significant attention in the literature. These methods attempt to surface significant aspects of a Deep Learner through visualizations using saliency maps, heat maps, and feature or class activation maps (Erhan et al., 2009; Simonyan et al., 2013; Zeiler and Fergus, 2014; Hohman et al., 2018; Zhou et al., 2016). As with the natural language solution, these methods to some extent depend on what is being highlighted for comprehension by end users.

13.1.3 Example-based explanations: factual, counterfactual, and semifactual

The present chapter focuses on recent advances in post-hoc example-based explanations and on the variety of solutions arising in this literature. Recently, the XAI literature has rapidly moved from traditional factual, example or case-based explanations to counterfactual (Byrne, 2019; Miller, 2019; Karimi et al., 2020a; Keane et al., 2021) and semifactual explanations (Kenny and Keane, 2021b). Here, we briefly sketch the ideas behind these explanation strategies, largely describing them using tabular-data content (see later sections for image and time series examples).

Factual Explanations. These explanations are the case-based examples discussed in hundreds, if not thousands of case-based reasoning (CBR) papers (Leake and McSherry, 2005; Sørmo et al., 2005); except that now the example-cases to explain Deep Learners are retrieved based on extracted feature-weighs from the Deep Learner (Kenny and Keane, 2019, 2021a). Imagine a SmartAg system, where a Deep Learning model for predicting crop growth tells a farmer that "in the next week, the grass yield on their farm will be 23 tons," and the farmer asks "Why?" (Kenny et al., 2021b). Using these techniques, a factual explanation could be found from historical instances in the dataset for this farm, to give the explanation "Well, next week is like week-12, two years ago, in terms of the weather and your use of fertilizer and that week yielded 22.5 tons of grass." This explanatory factual case comes from finding the nearest neighbor in the dataset (also known as the Deep Learner's training data) based on analyzing the feature weights contributing to the prediction made.

Counterfactual explanations. This explanation strategy is quite different to the factual option. It tells the end-user about how things would have to change for the model's predictions to change (hence, it can be used for algorithmic recourse; Karimi et al. (2020c)). Imagine the farmer thinks that the crop yield should be higher than 23 tons and asks, "Why not higher?"; now, the AI could provide advice for getting a better yield in the future, by explaining that "If you doubled your fertilizer use, then you could achieve a higher yield of 28 tons." So, unlike factual explanations which tend to merely justify the status quo, counterfactuals can provide a basis for actions that can change future outcomes (see, e.g., Byrne (2019); Miller (2019) on the psychology of counterfactual explanations for XAI).

Semifactual explanations. Finally, semifactual explanations also have the potential to guide future actions. Imagine again, the farmer thinks that the crop yield should be higher than 23 tons and asks, "Why not higher?"; now, the AI could provide a semifactual "even-if" explanation that is also quite informative saying "Even if you doubled your fertilizer use, the yield would still be 23 tons." In this case, the farmer is potentially warned-off over-fertilizing and polluting the environment. Semifactuals have been examined occasionally in psychology (McCloy and Byrne, 2002), but hardly at all in AI (see discussion of a-fortiori reasoning for one notable exception in Nugent et al. (2009)).

13.1.4 Outline of chapter

In the remainder of this chapter, we focus on the different solutions we have found in the post-hoc example-based explanations across image and time series datasets. Most current research focuses on tabular datasets, but in this chapter we consider the, arguably more difficult, problem of XAI for image and time series datasets. This work introduces a suite of novel XAI methods for these domains, that have been supported by some user studies (though more are needed; see Keane et al. (2021)). The next three sections consider these different example-based solutions – factual, counterfactual, and semifactual – in which we describe the methods proposed, present some indicative results, and review the results from user testing. Section 13.2 considers factual example-based explanations for images and time series, before examining counterfactual and semifactual solutions for image data (Section 13.3) and for time series (Section 13.4). The latter two sections present novel extensions to our previous work. Finally, we conclude with a general discussion on the future directions for XAI and explainability in these domains.

13.2. Post-hoc explanations using factual examples

As we saw earlier, the use of factual explanations for neural networks emerged over 20 years ago in CBR, when the feature-weights of multilayered perceptrons (MLPs) were mapped into *k*-NN models to find nearest neighbors for a target query, to be used as

example-based explanations (Caruana et al. (1999); Shin and Park (1999); see Keane and Kenny (2019) for a review). More recent work has extended this approach to convolutional neural networks (CNNs) exploring tabular, image, and time series datasets, though the specific solutions proposed are somewhat different. These techniques share the common idea that example-based explanations can be found using nearest neighbors to explain the predictions of a novel, unseen test instances.

13.2.1 Factual explanations of images

Kenny and Keane (2019, 2021a) extended factual explanations for tabular data involving MLPs to image datasets and generalized the approach – the *twin-systems* framework – to Deep Learners (mainly, CNNs). This approach relies on twinning the Neural Network model with a *k*-NN, where the feature-weights for a test-instance in the Neural Network are applied to a *k*-NN model, operating over the same dataset, to retrieve factual explanations (see Figs. 13.1, 13.2, and 13.3); notably, the feature-weights are based on feature contributions to the local prediction made.

The twin-systems framework proposes that an ANN may be abstracted in its entirety into a single proxy CBR system that mimics the ANN's predictive logic. Most methods for post-hoc explanation-by-example use feature activations to locate similar training examples to a test instance (also known as neuron activations in the ANN) (Papernot and McDaniel, 2018; Jeyakumar et al., 2020). In contrast, the twin-systems solution uses feature contributions, which weight these neuron activations by their connection weights to the predicted class (the so-called COLE Hadamard Product, C-HP, method). This approach has the effect of finding nearest neighbors that (i) are predicted to be in the same class as the test case, and (ii) have similarly-important features used in the prediction. This solution has a notable advantage over other explanation methods as CBR is nonlinear (e.g., as opposed to LIME (Ribeiro et al., 2016)) and can thus more accurately abstract the nonlinear ANN function using only a single proxy model.

Kenny and Keane (2019, 2021a) have shown that this contributions-based feature-weighting method provides the most accurate analysis of black-box ANNs, with a view to finding factual example-based explanations. This feature-weighting method – Contributions Oriented Local Explanations (COLE) – can be applied to both multilayered perceptrons (MLPs) and convolutional neural networks (CNNs) to find explanatory cases from the twinned *k*-NN/CBR model (i.e., a CNN-CBR twin) applied to the same dataset. Negative weights in the C-HP indicate that a certain feature map is not important for retrieving informative explanatory cases. COLE fits a *k*-NN model with feature contributions to abstract the ANN function, that are calculated by multiplying a data-instance by weights it used in the final prediction. To implement this in a CNN there are two possible options. Firstly, the CNN may have several fully connected layers post feature-extraction, in which case we have shown how saliency map techniques can be used to implement COLE (Kenny and Keane, 2019). Secondly, there may be a linear

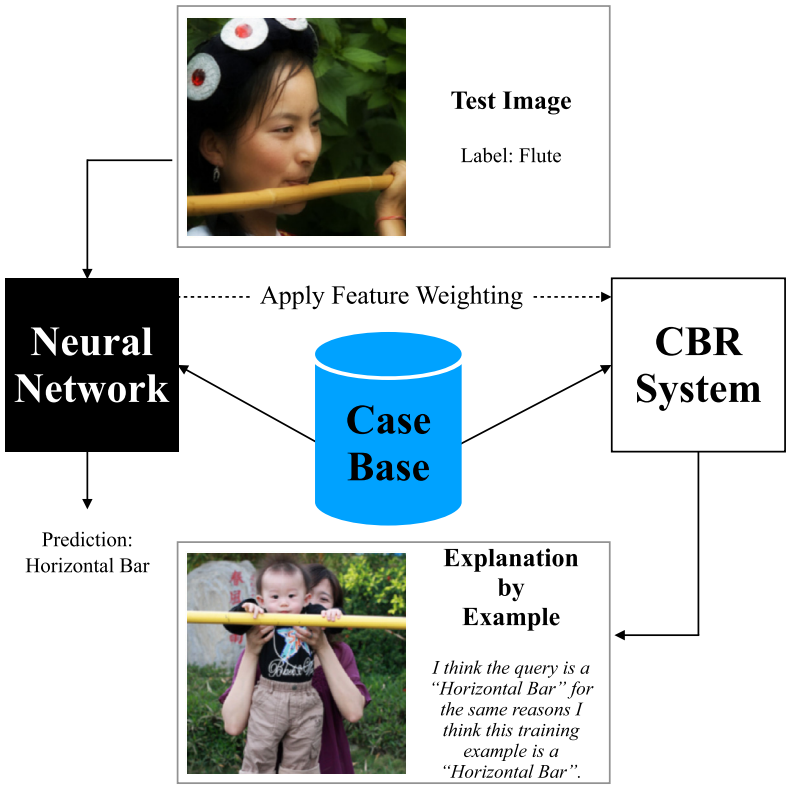

Figure 13.1 The Twin-Systems Explanation Framework: A deep learning model (Neural Network) produces a misclassification for an ImageNet test image, wrongly labeling a "Flute" as a "Horizontal Bar." This prediction is explained by analyzing the feature-weights of the network for that prediction and applying these to a twinned *k*-NN (Case Based Reasoner/CBR System) to retrieve a nearest neighbor to the test-image in the training set. This explanatory image shows that the model used an image of a "Horizontal Bar," where the bar looked very like the flute in the test image, to help make the classification. So, although the classification is wrong, it is somewhat understandable.

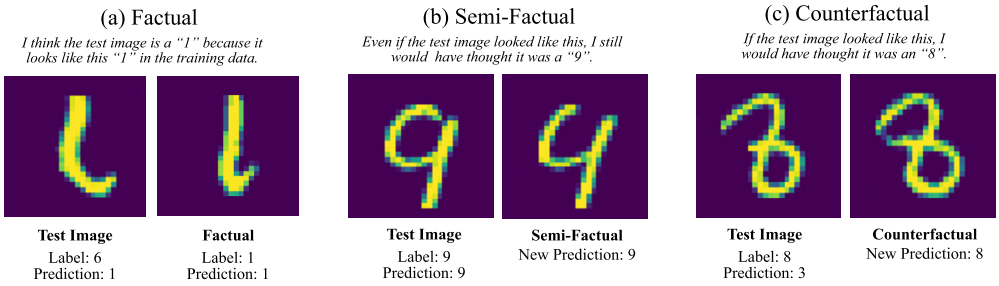

Figure 13.2 Factual, semifactual and counterfactual Explanations for a CNN's predictions applied on the MNIST dataset.

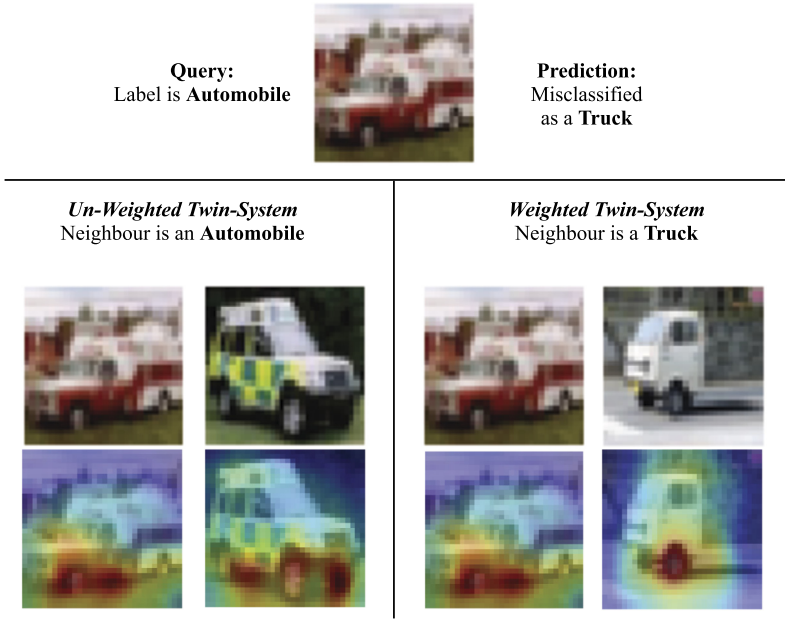

Figure 13.3 A CNN-CBR twin misclassifies an image of an automobile as a truck. Perhaps the main advantage of twin-systems is their ability to retrieve training examples predicted to be in the same class as the test instance. To illustrate this, in the unweighted twin-system, the explanation retrieved is an image of an automobile, which does not make sense since the test image was classified as a truck. To be explicit, this explanation is saying "I think the test image is a truck because it reminds me of this image I think is an automobile" (which makes no sense). In contrast, the weighted twin-system retrieves an image classified as a truck. This basic requirement of explanation-by-example (i.e., retrieving a training image predicted to be in the same class as the test image) is only guaranteed to be fulfilled if twin-systems (and their feature weighting) are used.

classifier post feature-extraction (e.g., the ResNet architectures), in which case contributions can be calculated by taking the Hadamard product of an instance's penultimate activations with the weight vector connected to its final classification (henceforth called C-HP). The saliency maps (i.e., FAMs here) are not used in the nearest neighbor search, they are a post-hoc addition after the neighbors are found (see Fig. 13.3).

C-HP has been extensively tested on 17 classification/regression datasets, which consistently showed C-HP to be the best for both MLPs and CNNs. Initially, twinning was demonstrated for classification (Kenny and Keane, 2019) but it has now been extended to regression problems and natural language domains (Kenny and Keane, 2021a). Kenny & Keane originally used feature-activation maps (FAMs) to show the important features in the explanatory images for the prediction (Kenny and Keane, 2019); however, FAMs can often produce unclear or unintuitive heat-maps for important features. So, KDK21-CCR has proposed a new method for finding *critical regions* in the explana-

tory image, to provide more information to users about the feature(s) that underlie the classifications.

Sample Factual Explanations for Images. Fig. 13.1 shows the general architecture for twinning between a Neural Network (CNN) and a *k*-NN (CBR) where the nearest neighboring image found hinges on finding critical regions in candidate images in the set of nearest neighbors; here, the original test image is misclassified but the explanation shows why the model failed because of the presence of a very similar image (with a Horizontal Bar instead of a Flute). Fig. 13.2(a) shows another factual explanation, again for a misclassification, but using the MNIST dataset. Here, a test image of a "1" is presented to a CNN and the model inaccurately labels it as a "6." When the feature-weights are abstracted from the CNN and mapped to the *k*-NN to retrieve nearest-neighbors of the test in the dataset, a similar instance is found showing a "1" that was annotated as a "1." Here, the example-based explanation shows the user why the model was in error; namely, that it has been presented with instances of ones that were very like badly-drawn sixes and, accordingly, misclassified the test instance. Fig. 13.3 shows a more complex case, using the CIFAR-10 dataset involving the misclassification of an automobile as a truck. This incorrect prediction is justified by the explanatory example from the weighted-twin, which essentially says to the user "I think this is a truck because it looks like the trucks I saw before." In addition, the FAMs highlight the most important (i.e., the most positively contributing) feature in the classification, which clearly focuses on the vehicle wheels in all images. Since these are a central aspect of both automobiles and trucks, it makes the misclassification more reasonable. In this case, the explanatory example found by the unweighted twin does not actually explain the misclassification, in a way that the weighed-twin does.

13.2.2 Factual explanations of time series

In the time series domain, factual explanations can be retrieved by identifying nearest neighbors to the to-be-explained test instance. Typically, either Euclidean distance or Dynamic Time Warping (if the instances are out of phase) is used in this nearest neighbor retrieval. However, it is generally agreed that comparing test instances with their nearest neighbors often yields little information about a classification decision (Ye and Keogh, 2011). Hence, factual explanations in this domain are often gained from retrieving class prototypes. *Class prototypes* are instances that are maximally representative of a class and have demonstrated promise in providing global explanations for time series classifiers in the healthcare domain (Gee et al., 2019). A simple method used to retrieve class prototypes is to extract medoids using the *k*-medoids clustering algorithm (Molnar, 2020).

However, one of the recognized problems with these techniques is that they typically fail to identify discriminative subsequences of the time series, that often contain semantically-meaningful information for both classification and explanation. In light

of these issues, a variety of techniques such as Shapelet mining (Ye and Keogh, 2011; Grabocka et al., 2014) and Class Activation Mapping (Zhou et al., 2016; Wang et al., 2017) have been proposed to identify discriminative regions of the time series. These solutions can be cast as a type of explanation by visualization (Lipton, 2018), a very popular family of explanation methods in the time series domain. However, recent research has drawn attention to the unreliability of saliency-based approaches, especially for multivariate time series data in combination with deep learning classifiers, motivating a need for more robust forms of explanation (Adebayo et al., 2018; Ismail et al., 2020; Nguyen et al., 2020; Jeyakumar et al., 2020).

The twin-systems approach is still a relatively untapped, yet promising solution, to the development of factual explanations for time series classification. The closest works to this approach are those of Leonardi et al. (2020) and Sani et al. (2017) who suggested mapping features from a Deep Learner (typically a CNN) to a CBR system for interpretable time series classification. For global explanations, Gee et al. demonstrated the promise of leveraging an autoencoder to learn prototypes from the latent space. This design also enabled the extraction of real-world and semantically-meaningful global features (e.g., bradycardia in electrocardiogram waveforms), highlighting the advantages of combining Deep Learning and CBR for global factual explanations (Gee et al., 2019; Li et al., 2017). As we shall see in the later section on counterfactual explanations, the leveraging of discriminative features and instances from the training data also has a role to play in generating informative contrastive explanations.

13.2.3 User studies of post-hoc factual explanations

Even though factual example-based explanations are one of the oldest XAI solutions in the AI literature (in CBR see Sørmo et al. (2005); Leake and McSherry (2005), and in Recommender Systems, see Tintarev and Masthoff (2007); Nunes and Jannach (2017)), there are few well-designed user studies that test them. Keane and Kenny (2019), in a survey of the CBR literature, found $< 1\%$ of papers reported user studies (many of which were loose surveys of expert users). Furthermore, this literature also focuses more on tabular data (see, e.g., Nugent and Cunningham (2005); Cunningham et al. (2003); Dodge et al. (2019)) than on image or time series data (the latter receiving really no attention for the reasons outlined earlier).

The few papers on factual explanations for images focus on two questions: how do explanations (i) change people's subjective assessments of a model (e.g., in task performance, trust, and other judgments), and (ii) impact people's negative assessments of a model's errors (so-called *algorithmic aversion*, see Dietvorst et al. (2015)). On the question users' perceptions of the model, the few relevant studies show somewhat modest impacts for these explanations. Buçinca et al. (2020) reported two experiments examining how example-based explanations influenced people's use of an AI-model making predictions about fatty-ingredients from pictures of food-dishes; they used multiple examples (i.e.,

four photos of similar food dishes) or a single example with highlighted features (i.e., photo of one food-dish with identified ingredients). They found that providing explanations improved performance on the fat-estimating task and impacted trust measures in varied ways. However, these experiments have several design flaws that mean the results need to be considered with care (e.g., imperfectly matched materials, statistical comparisons between experiments as if they were conditions). Another study by Yang et al. (2020) tested users' ($N = 33$) trust in example-based explanations for a classifier's predictions for images of tree-leaves and found that specific visual representations improved "appropriate trust" in the system; their classifier had an accuracy of 71% but, notably, their participants were perhaps less expert (i.e., not botanists). Cai et al. (2019) used drawings of common objects as explanations for misclassifications by a classifier; their users ($N = 1150$) reported a better understanding of the model and viewed it as a more capable when given an explanation. Finally, there is a smattering of other studies, some using MNIST, that either have low N values (< 12) or are not reported in sufficient detail from which to draw conclusions (Bäuerle et al., 2018; Glickenhaus et al., 2019; Ross and Doshi-Velez, 2018). A notable and worrying finding from this work is the lack of evidence on people's performance on a target task. Many of them show that people's subjective assessments of the model change, but they do not show that explanations improve their performance on a task (which is generally assumed to be one of the goals of good explanation).

Finally, in a significant sequence of studies (involving several 100 participants), Kenny et al. (2021a) and Ford et al. (2020) showed that people's judgments of correctness of a CNN's errors on MNIST were subtly influenced by example-based explanations. Specifically, people (perhaps without them being aware of it) came to view the errors as "less incorrect" when given an explanation; ironically, it was also found that they came to blame the model more than the data (i.e., poorly written numbers) when explanations were provided. This work also systematically addressed the second question about the impact of errors on people's algorithmic aversion, by presenting different groups with different levels of errors (between 3% and 60%). In general, they found that people's trust in the model linearly decreased with increasing error-levels and explanations did not mitigate this decreasing trust. Indeed, beyond about 3–4% errors there is a steep shift in trust levels.

Taken together, all of these results suggest three significant conclusions. First, the provision of factual explanations is not a silver bullet for remedying algorithmic aversion. Second, explanations can subtly affect people's perceptions of a model (e.g., perceptions of correctness) in ways that could be unethically exploited. Third, the evidence for changes in people's performance on a task (e.g., debugging a CNN or learning about an unfamiliar domain), as a function of explanation is, at best, weak. However, it should be said that these conclusions are made against a backdrop of a very poor programme of user testing.

13.3. Counterfactual & semifactual explanations: images

Although factual explanations have traditionally been the focus for example-based explanations, recently a huge research effort has focused on contrastive, example-based explanations (Kenny and Keane, 2021b; Miller, 2019; Byrne, 2019; Keane and Smyth, 2020). These developments have been partly motivated by the argument that contrastive explanations are much more causally-informative than factual ones, as well as being GDPR-compliant (Wachter et al., 2017). However, most current counterfactual methods only apply to tabular data (Wachter et al., 2017; Keane and Smyth, 2020), though recent work has begun to consider images (Goyal et al., 2019; Van Looveren and Klaise, 2019; Kenny and Keane, 2021b). Figs. 13.2(b) and 13.2(c) show some samples of explanations using contrastive, example-based explanations for a CNN's predictions on the MNIST dataset. In Fig. 13.2(c) the CNN misclassifies an image of an "8" as a "3" and the counterfactual explanation generated shows how the test instance would have to change to be correctly classified by the CNN as an "8". Fig. 13.2(b) shows the other type of contrastive explanation – a semifactual explanation – where a "9" is correctly classified by the CNN and the explanation shows a generated instance of a "9" essentially saying "even if the 9 changed to look like this, it would still be classified as a 9." One way to think about semifactuals is that they show users the "headroom" that exists just before the decision boundary is crossed, whereas the counterfactual shows users instances that occur after the decision boundary is crossed (see Fig. 13.4). In this section, we reprise our model of contrastive explanations – the PIECE model (Kenny and Keane, 2021b) – and propose some improvements to it, before testing it in a novel experiment.

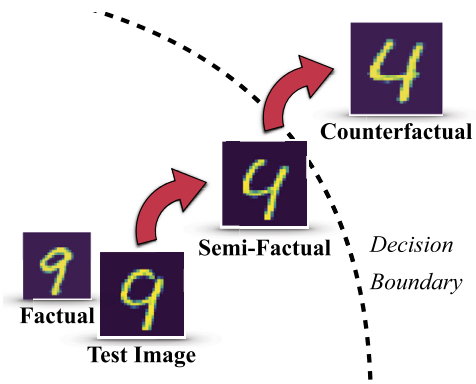

Figure 13.4 A test image from MNIST. A factual explanation could be presented (i.e., a nearest neighbor). Otherwise, a semifactual could be presented for an explanation which points towards the decision boundary (but does not cross it) to help convince the user the initial classification was correct. Lastly, a counterfactual could be presented for an explanation which explains how to modify the test image into a plausible example of the counterfactual class.

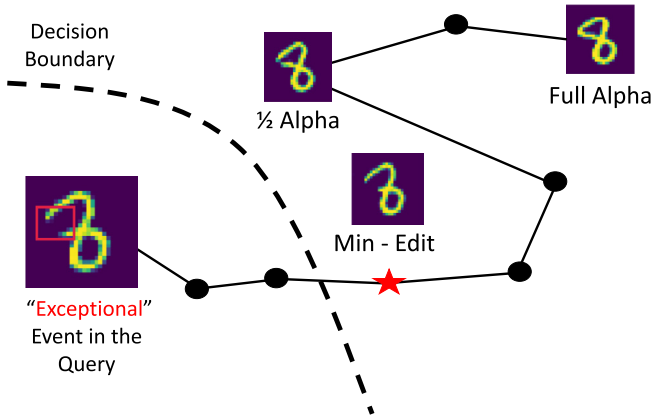

Figure 13.5 The PIECE counterfactual method. The alpha parameter controls the proportion of "exceptional" features that are modified to become "normal."

13.3.1 PIECE: generating contrastive explanations for images

Kenny and Keane (2021b) proposed the PlausIble Exceptionality-Based Contrastive Explanations (PIECE) method to generate counterfactuals for image datasets using a statistical technique combined with a generative model. PIECE generates counterfactual images by identifying "exceptional" features in the test image, and then perturbs these identified features in the test instance to be "normal." PIECE also generates semi-factuals, as a side effect of generating the counterfactual, as the latter is generated from perturbing exceptional features, just before crossing the decision boundary to generate the counterfactual.

PIECE works by identifying "exceptional features" in a test instance with reference to the training distribution; that is, features of a low probability in the counterfactual class are modified to be values that occur with a high probability in that class. For example, when a CNN has been trained on the MNIST dataset and a test image labeled as "8" is misclassified as "3," the exceptional features (i.e., low probability features in the counterfactual class 8) are identified in the extracted feature layer of the CNN via statistical modeling (i.e., a hurdle model to model ReLU activations) and modified to be their expected statistical values for the 8-counterfactual-class (see Fig. 13.5). PIECE has three main steps: (i) "exceptional" features are identified in the CNN for a test image from the perspective of the counterfactual class, (ii) these are then modified to be their expected values, and (iii) the resulting latent-feature representation of the explanatory counterfactual is visualized in the pixel-space with help from the GAN.

Fig. 13.5 illustrates how PIECE works in practice to generate a counterfactual image-explanation. Here, the counterfactuals to a test image I, in class c, with latent features x, are denoted as I', c' and x', respectively. Fig. 13.5 shows a test image labeled as class "8" (i.e., c) is misclassified as class "3" (i.e., c'). Exceptional features are identi-

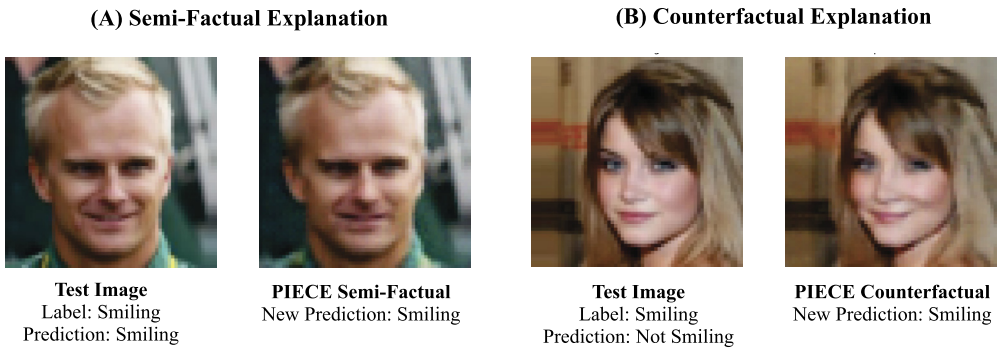

Figure 13.6 (A) A semifactual explanation justifying why the initial classification was definitely correct, in that, even if the image was smiling much less, it still would have classified it as "smiling." (B) A counterfactual explanation conveying to a user why the CNN made a mistake, and how the image would need to look for it to have classified it correctly (as computed by PIECE+ and Min-Edit).

fied using mathematical probability in the extracted feature layer X which have a low chance of occurrence in c'; these are then modified to be their expected feature values for class c' which modify the latent representation x to be x'. This new latent counterfactual representation x' is then visualized in the pixel space as the explanation I' using a GAN depending on the number of exceptional features changed, PIECE will produce a semifactual or counterfactual. To implement semifactual explanations for images, we used the PIECE algorithm, but stop the modification of exceptional features before the decision boundary is crossed. Figs. 13.6(A) and 13.6(B) illustrate other examples of semifactuals and counterfactuals for the CelebA dataset.

Reported experiments using PIECE have shown that it generates plausible counterfactuals and semifactuals, and is less likely than other models to generate out-of-distribution explanatory instances. It also may be unique in that it is (to our knowledge) the only method which produces counterfactuals for multiclass classification, without requiring human intervention to select the counterfactual class. However, as we shall see in the next subsection, PIECE is not as general as it could be and can be improved in several ways.

13.3.2 PIECE+: designing a better generative method

The generation of explanatory counterfactual images using deep learning hinges on finding key features in the image and then modifying these features in plausible, intuitive and informative ways. Several solutions to this problem have been proposed in the literature. He et al. (2019) proposed AttGAN, a method which produces very realistic-looking modifications to images. However, their method focuses only on a single dataset, and relies on class attribute labels, with no consideration given to counterfactual explanation or class modification. Liu et al. (2019) proposed that AttGAN

could be used for counterfactual generation; however, their approach cannot be applied to a pretrained network, and again relies on attribute-labeling to work. Mertes et al. (2020) pursued a different approach based on modifying CycleGAN (Almahairi et al., 2018); this method produces quite realistic image modifications in radiology, but is limited to binary classification problems. In contrast to these methods, PIECE adopts another approach, utilizing statistical hurdle models alongside a GAN. However, PIECE is limited in its requirement for a well-trained GAN for the domain in question, alongside the ability to recover latent representations of test images in the GAN. This limits PIECE to relatively simple datasets such as MNIST, as locating a test image's latent representation in a GAN is far from a solved problem in AI currently (Zhu et al., 2020). The current improvement on PIECE – PIECE+ – builds upon AttGAN with three important modifications: (i) a pretrained CNN is incorporated into the PIECE framework, so that any CNN may be explained post-hoc, (ii) AttGAN is modified to handle multiclass classification, and (iii) the more flexible architecture allows handling of more complex datasets beyond MNIST (such as CelebA) by virtue of avoiding the need for latent recovery of test images in a GAN. In the next subsection, we describe PIECE+ in more detail, before reporting some preliminary observations.

13.3.2.1 PIECE+: the method

The PIECE+ method is trained over several steps using three distinct losses – the Reconstruction, Adversarial, and Classification Losses – combined using multiobjective optimization, which all start from the initial encoder/generator component (see Fig. 13.7). During training, the test image is encoded by E into a latent representation $z \in \mathbb{R}^{(4,4,k)}$, where k is the number of feature kernels. During training, the encoding z firstly has a label vector $C \in \mathbb{R}^{(1,1,n)}$ appended to it, to generate a reconstruction, and again a counterfactual label vector of the same shape (which is randomly chosen), to generate a counterfactual image (i.e., there are two separate forward passes). Either way, this additional vector is expanded into a matrix $C \in \mathbb{R}^{(4,4,n)}$, where n is the number of classes in the domain. Hence, the vector z has a $4 \times 4 \times 10$ matrix appended to it in MNIST here (since there are 10 digit classes), where one of these 4×4 "slices" is filled in with 1's (representing the class we are generating), and the rest 0's. This final representation is decoded through the generator G and (depending on what vector c was appended to it) generates either a target counterfactual image, or a reconstruction of the original image.

The *Reconstruction Loss* involves the reconstruction of the image compared to the original image, and the loss is taken to train the encoder/generator (L_1-loss is used here). Using *Adversarial Loss*, the generated counterfactual image is then input to the discriminator network (also known as a "critic" network), and an adversarial loss is taken by comparing it to the dataset of "real" images. WGAN-GP is used here (Gulrajani et al., 2017), and the typical hyperparameters associated with it (e.g., using an Adam

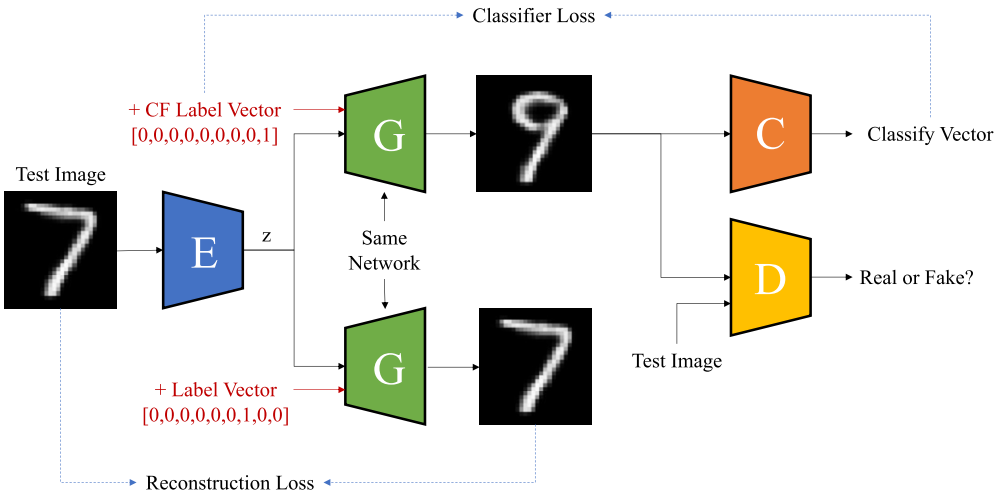

Figure 13.7 An overview of the CF Method: a combination of (1) a reconstruction loss, (2) an adversarial loss using the discriminator (D), and (3) a classifier (C) loss combine to train the encoder (E) and generator (G) architecture. Heavily adapted from He et al. (2019).

optimizer and training the generator every five steps). Then *Classification Loss* is used when the generated counterfactual image is passed into the CNN we are trying to explain, with the aim of classifying the image with a probability of 1.0 in the target counterfactual class, a cross-entropy loss is taken for multiclass classification as is typical. Following the typical WGAN-GP framework, we train the entire system as follows. The critic (also known as discriminator D) is trained for 4 iterations using the typical loss in addition to the gradient penalty, then on the 5th iteration the generator/encoder is trained using the Classification, Reconstruction, and Adversarial Losses (i.e., how good the generated images are measured by the critic). So, a combination of all these losses is backpropagated.

One limitation of this method is that the CNN is required to have been trained using the same activation functions/normalization as the critic network. Here we used leaky ReLU and instance normalization in both networks. However, there is some leeway to be found, in that the usage of leaky ReLU in the critic and ReLU in the CNN appears to produce almost optimal results, but we empirically found the best results to be using leaky ReLU and instance normalization in both networks. Future research would do well to investigate how to train this system without any limitations on the pretrained CNN architecture.

13.3.2.2 Results: PIECE+

Here the results of two brief evaluations are shown, the first of which is a demonstration of the method on MNIST, and the second a quantitative evaluation. Lastly, we conclude

with a brief reflection on what improvements this new implementation of PIECE has brought, and what future challenges remain.

Sample explanations

In Fig. 13.8 we see an example of a straightforward transition from the digit 7 to the digit 9 in MNIST. Although the network is not trained to produce plausible images between the image reconstruction and the counterfactual image, it can nevertheless "fade" between both images by gradually adjusting the $C \in \mathbb{R}^{(4,4,n)}$ matrix accordingly, similar to the original AttGAN paper (He et al., 2019). During this transition, it is possible to generate either a semifactual image of a 7, or a counterfactual image of a 9; note, the red line (dark gray in print version) in Fig. 13.8 shows where the decision boundary falls, so semifactuals will occur before it and counterfactuals after it.

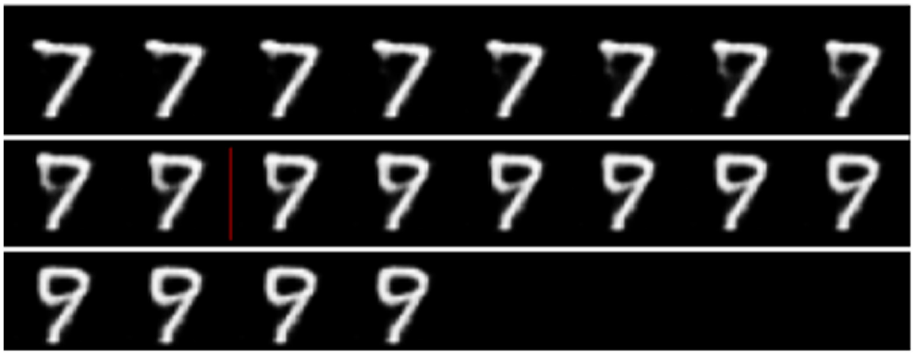

Figure 13.8 A test image of a 7 on MNIST is slowly transformed from a 7 to a 9. During the transition, it is possible to generate both a semifactual and a counterfactual. The red line (dark gray in print version) represents the model decision boundary.

Automatically selected counterfactuals

Perhaps the key novelty of PIECE is its ability to generate counterfactual images in multiclass classification problems, and being able to automatically select counterfactual classes in such an instance (most work in the area has focused on binary classification). Originally, PIECE did this by gradient ascent, were the full pipeline was able to automatically select the CF class (Kenny and Keane, 2021b), this was necessary as the GAN formed an integral part of the pipeline. Here however, that is not the case, and the CF class can be automatically selected simply by choosing the class of second highest probability in a classification. Hence, here we compare this against a baseline which randomly selects a CF class in 64 randomly selected test images. The main idea here is that the better the method, the closer the explanation should be in pixel-space. Hence, if allowing PIECE to automatically select the counterfactual is better, the generated explanations should be closer to the original test image.

For counterfactuals, the mean L_2 pixel-space distance between the original images and for PIECE selected CF classes was 14.65 ± 0.36, whilst it was 16.50 ± 0.51 for the randomly selected ones, showing a largely statistically significant difference with a two tailed, independent t-test ($p < 0.005$). For semifactuals, the mean L_2 pixel-space distance between the original images and for PIECE selected CF classes was 8.80 ± 0.23, whilst it was 17.44 ± 0.23 for the randomly selected ones, again showing a largely statistically significant difference with a two tailed, independent t-test ($p < 0.0001$). Overall, the results clearly show that PIECE's ability to automatically select the counterfactual class via simply choosing the class of second highest probability will result in better "closer" contrastive explanations than allowing users to manually select it, which should generally result in better explanations (Keane and Smyth, 2020).

Conclusion: PIECE improvements

The main problem with the original implementation of the PIECE framework by Kenny and Keane (2021b) is that it requires the recovery of a latent representation for a test image in a GAN. Whilst this works well on MNIST, it is still an open research area for more complex domains (Zhu et al., 2020). To solve these issues, PIECE+ has taken inspiration from AttGAN and designed a general framework for post-hoc contrastive explanations in image domains. Namely, to solve the issue of recovering a latent representation for a test image, PIECE+ has an encoder/decoder architecture which alleviates the issue completely by training the prior to encode these representations during training. This has the added benefit of allowing PIECE+ to work in more complex domains such as CelebA (see Fig. 13.6). Additionally, quantitative testing has suggested it is still beneficial in this new implementation of PIECE to allow automatic selection of the CF class, rather than trusting users to. Future work will examine the integration of modifying "exceptional features" within this new framework, which was shown to work well before (Kenny and Keane, 2021b). In the next section, we continue this examination of contrastive explanations by considering methods for time series data.

13.4. Contrastive explanations: time series

In the previous section, we reported on the explosion of research on counterfactual explanations and on how most of this research tends to focus on tabular rather than image datasets. Even less of this research has considered the time series domain. Interestingly, tabular methods for counterfactuals (Wachter et al., 2017; Keane and Smyth, 2020), quickly become intractable for time series data because of the massive number of possible feature dimensions and the utility of domain-specific distance measures (such as DTW) (Delaney et al., 2021). Indeed, much of the work reviewed here has only been published in the last two years (only the present work has even considered semifactuals). In this section, we reprise our model of contrastive explanations for time series – Native

Guide (Delaney et al., 2021) – and propose some improvements to it, before testing it in a novel experiment.

The current focus in XAI for time series mainly considers saliency-based approaches where important subsequences or features are highlighted (Wang et al., 2017; Fawaz et al., 2019b; Schlegel et al., 2019). However, it is quite unclear if these explanations are faithful to the underlying black-box model in providing informative explanations (Adebayo et al., 2018; Ismail et al., 2020; Nguyen et al., 2020)

Many have considered leveraging shapelets to generate contrastive explanations (Karlsson et al., 2018; Guidotti et al., 2020). However, issues have been raised about the interpretability of the shapelets produced by the frequently deployed shapelets learning algorithm (Grabocka et al., 2014), and many solutions are not model agnostic. By modifying the loss function proposed by Wachter et al. (2017) to generate counterfactuals, Ates et al. (2020) explored generating counterfactual explanations for multivariate time series classification problems. Labaien et al. (2020) have progressed contrastive explanations for the predictions of recurrent neural networks in time series prediction.

The Native Guide method (Delaney et al., 2020, 2021) adopts a different approach to these other methods and we demonstrate that it can work with any classifier (model-agnostic). In the next subsection, we sketch the essence of this method before proposing a novel extension to it using Gaussian noise.

13.4.1 Native guide: generating contrastive explanations for time series

Native Guide (Delaney et al., 2020, 2021) incorporates a strategy where the closest in-sample counterfactual instance to the test-instance is adapted to form a new counterfactual explanation (Keane and Smyth, 2020; Goyal et al., 2019). Here the "Native-Guide" is a counterfactual instance that already exists in the dataset (e.g., the nearest-neighbor time series to the test-instance that involves a class change; see Fig. 13.9). We can retrieve this in-sample counterfactual instance using a simple 1-NN search. Once this instance is found it is leveraged to guide the generation of the explanatory counterfactual T'. The generated counterfactual instance T' (the yellow square [light gray in print version] in Fig. 13.9), should offer better explanations than the original in-sample counterfactual as it is in closer to the test whilst still staying within the distribution of the data. As an aside, we note that Native Guide could also be used to compute plausible semifactual explanations by terminating the counterfactual generation process just before the explanatory instance enters the counterfactual class.

Native Guide generates counterfactual explanations for a to-be-explained query, T_q, by leveraging both (i) discriminative feature information and (ii) native counterfactual instances (e.g., the query's nearest unlike neighbor (T'_{NUN})). Blind perturbation techniques frequently fail to account for dependencies between features and leveraging information from native counterfactual instances in the training data can immensely aid the generation of meaningful explanations (Keane and Smyth, 2020; Delaney et al.,

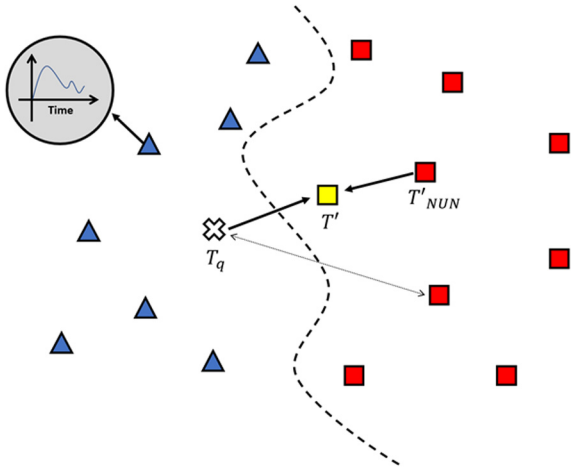

Figure 13.9 A query time series T_q (X with solid arrow) and a nearest-unlike neighbor T'_{NUN} (red square [dark gray in print version] with solid arrow) are used to guide the generation of counterfactual T' (see yellow square, light gray in print version) in a binary classification task. Another in-sample counterfactual (i.e., the *next* NUN; other red square [dark gray in print version] with dashed arrow) could also be used to generate another counterfactual for diverse explanations.

2021). However, gaining informative information from neighbors can be quite difficult when working with noisy time series data (Le Nguyen et al., 2019; Schäfer, 2016) and techniques such as dynamic time warping and weighted barycenter averaging (Petitjean et al., 2011; Forestier et al., 2017) are often required to generate meaningful explanations when instances are out of phase (Delaney et al., 2021). Moreover, access to neighbors from the training dataset may not always be available when generating explanations. So, a technique that purely utilizes discriminative feature information in generating counterfactual explanations bears practical utility. Hence, we considered an adjustment to Native Guide in a scenario where information from training instances is not readily available, greatly improving the flexibility of the technique.

13.4.2 Extending native guide: using Gaussian noise

Nguyen et al. (2020) analyzed the informativeness of different explanation techniques by adding Gaussian noise to discriminative subsequences and monitoring the degradation of classification performance. In this work, we demonstrate that Gaussian noise can be leveraged to generate counterfactual explanations without the need to access native counterfactual instances in the training data. In Native Guide, the counterfactual explanations are generated by modifying a contiguous subsequence of the to-be-explained test instance. As before, the most influential contiguous subsequence, T_{Sub}, according to a feature-weight vector, ω, is identified. This feature-weight vector can be retrieved from class activation maps (CAMs) when using convolutional architectures (Zhou et al.,

2016; Wang et al., 2017), or alternatively, from model agnostic techniques such as LIME (Ribeiro et al., 2016) or SHAP (Lundberg and Lee, 2017). However, instead of replacing the values of T_q with the corresponding region in T'_{Native}, Gaussian noise is added to the discriminative subsequence, $T_{Sub} + \mathcal{N}(\mu, \sigma^2)$, to generate a counterfactual, T'. This is an iterative process, initializing with a small subsequence and iteratively increasing the size of the perturbed subsequence and/or the magnitude of the Gaussian noise until a counterfactual is generated.

Experiment: extending native guide

In this experiment we demonstrate the model agnostic flexibility of Native Guide through implementing both Mr-SEQL (Le Nguyen et al., 2019) and a pretrained ResNet architecture (Fawaz et al., 2019b) as the base classifiers on two popular UCR datasets (Dau et al., 2019) (Coffee & Gunpoint). We compare the counterfactual generated across architectures when injecting Gaussian noise onto discriminative subsequences of the time series. We also compare the explanations to the counterfactuals generated by the original variant where access to instances from the training data are available. For the ResNet architecture, we use Class Activation Maps (CAMs) to extract the feature weight vector ω. MR-SEQL converts the time series to a symbolic representation and extracts discriminative feature information using a symbolic sequence learning algorithm. Following Keane and Smyth (2020); Delaney et al. (2021), we use a relative counterfactual distance measure to monitor the proximity and sparsity of the generated counterfactual with respect to existing in-sample counterfactual solutions.

Results and discussion

Native Guide consistently generates sparse counterfactuals that are closer to the to-be-explained query than in sample counterfactual instances when access to the training data available. Interestingly, when adding Gaussian noise into the input time series, the counterfactuals for the ResNet classifier typically required much fewer feature changes than the Mr-SEQL classifier. For example, in the coffee dataset the mean relative L_1 distance between the query and the counterfactual was 0.47 ± 0.08 for the ResNet architecture and significantly larger, 1.12 ± 0.12, for the MR-SEQL classifier (independent two tailed t-test $p < 0.01$). One possible explanation for this is that Deep learning architectures are very sensitive to slight perturbations on the input time series and prone to adversarial attack (Fawaz et al., 2019a). Access to neighbors in the training data guarantees the generation of plausible counterfactual instances (perfect coverage) (Delaney et al., 2021). Adding Gaussian noise to a discriminative subsequence sometimes failed to generate a counterfactual explanation for instances in the gunpoint dataset, especially for the more robust MR-SEQL classifier, indicating the importance of leveraging information from the training data in generating informative explanations. Different feature weight vectors often resulted in different counterfactual explanations (see Fig. 13.10).

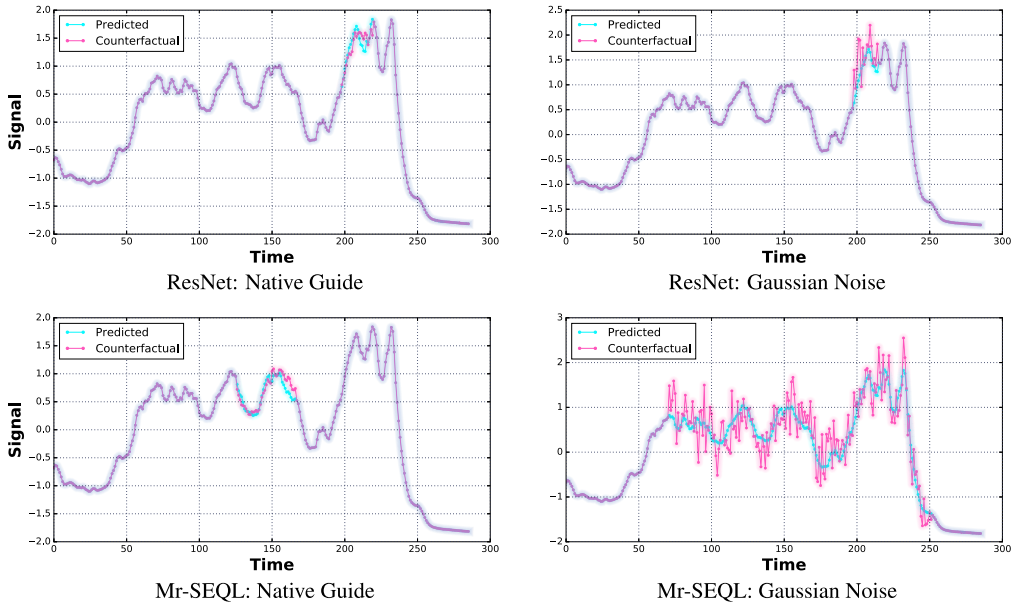

Figure 13.10 Counterfactual explanations for the predictions of both a ResNet architecture (Fawaz et al., 2019a) using Class Activation Mapping (Zhou et al., 2016; Wang et al., 2017), and Mr-SEQL (Le Nguyen et al., 2019) on an instance from the coffee dataset, where the task is to distinguish between Arabica and Robusta coffee beans from spectrographs. The ResNet architecture is more sensitive to slight perturbations on the input signal and the counterfactuals produced typically focus on a discriminative area of the time series that contains information about the caffeine content of the beans. Mr-SEQL is more robust to noise and the produced counterfactual explanations typically focus on an area containing information about the acid and lipid content of the coffee beans (Briandet et al., 1996).

For example on the coffee dataset, the counterfactuals generated from in symbolic sequence learning algorithm in Mr-SEQL typically focus on an area of the time series that contain information about the chlorogenic acid and lipid content of the coffee beans whist the class activation maps (CAMs) from the ResNet architecture often focused on a contiguous subsequence that contained discriminative information about the caffeine content of the beans (Briandet et al., 1996; Dau et al., 2019). As feature weighting techniques often highlight different regions in the input time series, the availability of ground truth domain knowledge expertise is crucial in assessing if the produced explanations are plausible as computational proxies are an imperfect proxy measure, further instilling the need for user studies with domain experts in the time series domain.

13.5. User studies on contrastive explanations

In the previous sections, we considered a variety of methods for computing contrastive explanations for images and time series datasets. For the most part, the focus of user

studies in this area has been almost wholly on counterfactual explanations with most of these studies focusing on tabular data. Though semifactuals have been researched in psychology and philosophy (e.g., see McCloy and Byrne (2002)), they have yet to be explicitly tested in XAI (for a solitary exception see Doyle et al. (2004)). Indeed, arguably, even the user studies on counterfactuals seem to be behind the curve of computational advances in the area.

Keane et al. (2021) reviewed the user studies on counterfactual explanations based on a survey of > 100 distinct counterfactual methods in the recent literature. They found that only 31% of papers perform user studies (36 out of 117) and fewer (21%) directly test a specific method on users. This means that few of the features that are discussed in the AI literature have been explicitly tested with users. Furthermore, as was the case with tests of factual explanations, many of these studies are methodologically questionable (e.g., use low *N*s, poor or inappropriate statistics, nonreproducible designs, inadequate materials).

The user-tests that have been done provide moderate support for the efficacy of counterfactual explanations under some conditions. Some studies show counterfactual explanations to be useful and preferred by end users (e.g., Lim et al. (2009); Dodge et al. (2019)). For instance, Lim et al. (2009) tested What-if, Why-not, How-to, and Why explanations, and found that they all improved performance relative to no-explanation controls. Dodge et al. (2019) assessed four different explanation strategies (e.g., case based, counterfactual, factual) on biased/unbiased classifiers and found counterfactual explanations to be the most impactful (though for a very limited set of problems). However, other studies show that counterfactual explanations often require greater cognitive effort and do not always outperform other methods (Lim et al., 2009; van der Waa et al., 2021; Lage et al., 2019; Dodge et al., 2019). Notably, however, few of these studies directly test specific facets of counterfactual algorithms (e.g., sparsity, plausibility, diversity) or compare competing methods (Goyal et al., 2019; Singla et al., 2019; Lucic et al., 2020; Akula et al., 2020; Förster et al., 2020a,b, 2021). This means that there is quite limited support for the specific properties of most counterfactual methods in the AI literature. Indeed, with respect to image and time series data, the literature is ever thinner and, as such, we await evidence to support the efficacy of these methods.

13.6. Conclusions

This chapter has considered state-of-the-art contributions to the rapidly evolving and increasingly important field of XAI; namely, the use of post-hoc explanations involving factual, counterfactual, and semifactual examples to elucidate a variety of Deep Learning models. There are many future directions in which this work can be taken. For instance, with respect to image data, we have begun to look at combining example-based explanations with visualizations of critical regions, reflecting important features

impacting a Deep Learner's predictions. Parallel opportunities exist for similar developments in the time series domain. However, if we were pressed on what is the most important direction to pursue, it would have to be that of user testing. In many respects, XAI is starting to exhibit quite a dysfunctional research program, where 100s or 1000s of models are being developed without any consideration of their psychological validity. Recently, there has been a growing commentary on this deficit in XAI, on the failure to properly address user requirements (Anjomshoae et al., 2019; Hoffman and Zhao, 2020), on the "over-reliance on intuition-based approaches" (Leavitt and Morcos, 2020), and on the increasing disconnect between the features of models and actions in the real-world (Barocas et al., 2020; Keane et al., 2021). In short, it is our view that a significant program of carefully-controlled and properly-designed user studies needs to be carried out as a matter of urgency, if XAI is to avoid drowning in a sea of irrelevant models.

Acknowledgments

This chapter emanated from research funded by (i) Science Foundation Ireland (SFI) to the Insight Centre for Data Analytics (12/RC/2289-P2), (ii) SFI and DAFM on behalf of the Government of Ireland to the VistaMilk SFI Research Centre (16/RC/3835).

References

Adebayo, Julius, Gilmer, Justin, Muelly, Michael, Goodfellow, Ian, Hardt, Moritz, Kim, Been, 2018. Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems, pp. 9505–9515.

Akula, Arjun, Wang, Shuai, Zhu, Song-Chun, 2020. CoCoX: Generating conceptual and counterfactual explanations via fault-lines. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 2594–2601.

Ala-Pietilä, Pekka, Smuha, Nathalie A., 2021. A framework for global cooperation on artificial intelligence and its governance. In: Reflections on Artificial Intelligence for Humanity. Springer, pp. 237–265.

Almahairi, Amjad, Rajeshwar, Sai, Sordoni, Alessandro, Bachman, Philip, Courville, Aaron, 2018. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In: International Conference on Machine Learning. PMLR, pp. 195–204.

Anjomshoae, Sule, Najjar, Amro, Calvaresi, Davide, Främling, Kary, 2019. Explainable agents and robots: Results from a systematic literature review. In: 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019). Montreal, Canada, May 13–17, 2019. International Foundation for Autonomous Agents and Multiagent Systems, pp. 1078–1088.

Ates, Emre, Aksar, Burak, Leung, Vitus J., Coskun, Ayse K., 2020. Counterfactual explanations for machine learning on multivariate time series data. arXiv:2008.10781.

Barocas, Solon, Selbst, Andrew D., Raghavan, Manish, 2020. The hidden assumptions behind counterfactual explanations and principal reasons. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 80–89.

Bäuerle, Alex, Neumann, Heiko, Ropinski, Timo, 2018. Training de-confusion: An interactive, network-supported visual analysis system for resolving errors in image classification training data. arXiv:1808.03114.

Briandet, Romain, Kemsley, E. Katherine, Wilson, Reginald H., 1996. Discrimination of Arabica and Robusta in instant coffee by Fourier transform infrared spectroscopy and chemometrics. Journal of Agricultural and Food Chemistry 44 (1), 170–174.

Buçinca, Zana, Lin, Phoebe, Gajos, Krzysztof Z., Glassman, Elena L., 2020. Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems. In: Proceedings of the 25th International Conference on Intelligent User Interfaces, pp. 454–464.

Byrne, Ruth M.J., 2019. Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning. In: IJCAI-19, pp. 6276–6282.

Cai, Carrie J., Jongejan, Jonas, Holbrook, Jess, 2019. The effects of example-based explanations in a machine learning interface. In: Proceedings of the 24th International Conference on Intelligent User Interfaces, pp. 258–262.

Camburu, Oana-Maria, Shillingford, Brendan, Minervini, Pasquale, Lukasiewicz, Thomas, Blunsom, Phil, Xie, Linhai, Miao, Yishu, Wang, Sen, Blunsom, Phil, Wang, Zhihua, et al., 2020. Make up your mind! Adversarial generation of inconsistent natural language explanations. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020. Seattle, Washington, USA, July 5–10, 2020. Association for Computational Linguistics, pp. 116–125.

Caruana, Rich, Kangarloo, Hooshang, Dionisio, J.D., Sinha, Usha, Johnson, David, 1999. Case-based explanation of non-case-based learning methods. In: Proceedings of the AMIA Symposium. American Medical Informatics Association, p. 212.

Chen, Chaofan, Li, Oscar, Barnett, Alina, Su, Jonathan, Rudin, Cynthia, 2018. This looks like that: Deep learning for interpretable image recognition. arXiv:1806.10574.

Cunningham, Pádraig, Doyle, Dónal, Loughrey, John, 2003. An evaluation of the usefulness of case-based explanation. In: International Conference on Case-Based Reasoning. Springer, pp. 122–130.

Dau, Hoang Anh, Bagnall, Anthony, Kamgar, Kaveh, Yeh, Chin-Chia Michael, Zhu, Yan, Gharghabi, Shaghayegh, Ratanamahatana, Chotirat Annh, Keogh, Eamonn, 2019. The UCR time series archive. IEEE/CAA Journal of Automatica Sinica 6 (6), 1293–1305.

Delaney, Eoin, Greene, Derek, Keane, Mark T., 2020. Instance-based counterfactual explanations for time series classification. arXiv:2009.13211.

Delaney, Eoin, Greene, Derek, Keane, Mark T., 2021. Instance-based counterfactual explanations for time series classification. In: International Conference on Case-Based Reasoning. Springer, pp. 32–47.

Dietvorst, Berkeley J., Simmons, Joseph P., Massey, Cade, 2015. Algorithm aversion: People erroneously avoid algorithms after seeing them err. Journal of Experimental Psychology: General 144 (1), 114.

Dodge, Jonathan, Liao, Q. Vera, Zhang, Yunfeng, Bellamy, Rachel K.E., Dugan, Casey, 2019. Explaining models: an empirical study of how explanations impact fairness judgment. In: Proceedings of the 24th International Conference on Intelligent User Interfaces, pp. 275–285.

Doshi-Velez, Finale, Kim, Been, 2017. Towards a rigorous science of interpretable machine learning. arXiv: 1702.08608.

Doyle, Dónal, Cunningham, Pádraig, Bridge, Derek, Rahman, Yusof, 2004. Explanation oriented retrieval. In: European Conference on Case-Based Reasoning. Springer, pp. 157–168.

Erhan, Dumitru, Bengio, Yoshua, Courville, Aaron, Vincent, Pascal, 2009. Visualizing higher-layer features of a deep network. University of Montreal 1341 (3), 1.

Fawaz, Hassan Ismail, Forestier, Germain, Weber, Jonathan, Idoumghar, Lhassane, Muller, Pierre-Alain, 2019a. Adversarial attacks on deep neural networks for time series classification. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–8.

Fawaz, Hassan Ismail, Forestier, Germain, Weber, Jonathan, Idoumghar, Lhassane, Muller, Pierre-Alain, 2019b. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery 33 (4), 917–963.

Ford, Courtney, Kenny, Eoin M., Keane, Mark T., 2020. Play MNIST for me! User studies on the effects of post-hoc, example-based explanations & error rates on debugging a deep learning, black-box classifier. arXiv:2009.06349.

Forestier, Germain, Petitjean, François, Dau, Hoang Anh, Webb, Geoffrey I., Keogh, Eamonn, 2017. Generating synthetic time series to augment sparse datasets. In: ICDM. IEEE, pp. 865–870.

Förster, Maximilian, Hühn, Philipp, Klier, Mathias, Kluge, Kilian, 2021. Capturing users' reality: A novel approach to generate coherent counterfactual explanations. In: Proceedings of the 54th Hawaii International Conference on System Sciences, p. 1274.

Förster, Maximilian, Klier, Mathias, Kluge, Kilian, Sigler, Irina, 2020a. Evaluating explainable artificial intelligence–what users really appreciate. In: Proceedings of the 28th European Conference on Information Systems.

Förster, Maximilian, Klier, Mathias, Kluge, Kilian, Sigler, Irina, 2020b. Fostering human agency: A process for the design of user-centric XAI systems. In: Proceedings ICIS.
Frosst, Nicholas, Hinton, Geoffrey, 2017. Distilling a neural network into a soft decision tree. arXiv:1711.09784.
Gee, Alan H., Garcia-Olano, Diego, Ghosh, Joydeep, Paydarfar, David, 2019. Explaining deep classification of time-series data with learned prototypes. CEUR Workshop Proceedings 2429, 15–22.
Gilpin, Leilani H., Bau, David, Yuan, Ben Z., Bajwa, Ayesha, Specter, Michael, Kagal, Lalana, 2018. Explaining explanations: An approach to evaluating interpretability of machine learning. arXiv:1806.00069.
Glickenhaus, B., Karneeb, J., Aha, D.W., 2019. DARPA XAI phase 1 evaluations report. In: DARPA XAI Program. Report.
Goyal, Yash, Wu, Ziyan, Ernst, Jan, Batra, Dhruv, Parikh, Devi, Lee, Stefan, 2019. Counterfactual visual explanations. arXiv:1904.07451.
Grabocka, Josif, Schilling, Nicolas, Wistuba, Martin, Schmidt-Thieme, Lars, 2014. Learning time-series shapelets. In: ACM SIGKDD, pp. 392–401.
Guidotti, Riccardo, Monreale, Anna, Giannotti, Fosca, Pedreschi, Dino, Ruggieri, Salvatore, Turini, Franco, 2019. Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems 34 (6), 14–23.
Guidotti, Riccardo, Monreale, Anna, Spinnato, Francesco, Pedreschi, Dino, Giannotti, Fosca, 2020. Explaining any time series classifier. In: IEEE Second International Conference on Cognitive Machine Intelligence (CogMI). IEEE, pp. 167–176.
Gulrajani, Ishaan, Ahmed, Faruk, 2017, Arjovsky, Martin, Dumoulin, Vincent, Courville, Aaron C. Improved training of wasserstein gans. In: Advances in Neural Information Processing Systems.
Gunning, David, Aha, David W., 2019. DARPA's explainable artificial intelligence program. AI Magazine 40 (2), 44–58.
He, Zhenliang, Zuo, Wangmeng, Kan, Meina, Shan, Shiguang, Chen, Xilin, 2019. AttGAN: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing 28 (11), 5464–5478.
Hoffman, Guy, Zhao, Xuan, 2020. A primer for conducting experiments in human–robot interaction. ACM Transactions on Human–Robot Interaction (THRI) 10 (1), 1–31.
Hohman, Fred, Kahng, Minsuk, Pienta, Robert, Chau, Duen Horng, 2018. Visual analytics in deep learning: An interrogative survey for the next frontiers. IEEE Transactions on Visualization and Computer Graphics 25 (8), 2674–2693.
Ismail, Aya Abdelsalam, Gunady, Mohamed, Bravo, Héctor Corrada, Feizi, Soheil, 2020. Benchmarking deep learning interpretability in time series predictions. arXiv:2010.13924.
Jeyakumar, Jeya Vikranth, Noor, Joseph, Cheng, Yu-Hsi, Garcia, Luis, Srivastava, Mani, 2020. How can I explain this to you? An empirical study of deep neural network explanation methods. Advances in Neural Information Processing Systems 33.
Karimi, Amir-Hossein, Barthe, Gilles, Balle, Borja, Valera, Isabel, 2020a. Model-agnostic counterfactual explanations for consequential decisions. In: International Conference on Artificial Intelligence and Statistics, pp. 895–905.
Karimi, Amir-Hossein, Schölkopf, Bernhard, Valera, Isabel, 2020b. Algorithmic recourse: from counterfactual explanations to interventions. arXiv:2002.06278.
Karimi, Amir-Hossein, von Kügelgen, Julius, Schölkopf, Bernhard, Valera, Isabel, 2020c. Algorithmic recourse under imperfect causal knowledge: a probabilistic approach. Advances in Neural Information Processing Systems 33.
Karlsson, Isak, Rebane, Jonathan, Papapetrou, Panagiotis, Gionis, Aristides, 2018. Explainable time series tweaking via irreversible and reversible temporal transformations. In: ICDM.
Keane, Mark T., Kenny, Eoin M., 2019. How case-based reasoning explains neural networks: A theoretical analysis of XAI using post-hoc explanation-by-example from a survey of ANN-CBR twin-systems. In: Proceedings of the 27th International Conference on Case-Based Reasoning (ICCBR-19). Springer, pp. 155–171.
Keane, Mark T., Kenny, Eoin M., Delaney, Eoin, Smyth, Barry, 2021. If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual XAI techniques. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21).

Keane, Mark T., Smyth, Barry, 2020. Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). In: ICCBR. Springer, pp. 163–178.

Kenny, Eoin M., Delaney, Eoin D., Greene, Derek, Keane, Mark T., 2020. Post-hoc explanation options for XAI in deep learning: The insight centre for data analytics perspective. In: International Conference on Pattern Recognition. Springer.

Kenny, Eoin M., Ford, Courtney, Quinn, Molly, Keane, Mark T., 2021a. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. Artificial Intelligence 294, 1–25.

Kenny, Eoin M., Keane, Mark T., 2019. Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In: Proceedings of the 28th International Joint Conferences on Artificial Intelligence (IJCAI-19), pp. 2708–2715.

Kenny, Eoin M., Keane, Mark T., 2021a. Explaining deep learning using examples: Optimal feature weighting methods for twin systems using post-hoc, explanation-by-example in XAI. Knowledge-Based Systems 233, 107530.

Kenny, Eoin M., Keane, Mark T., 2021b. On generating plausible counterfactual and semi-factual explanations for deep learning. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21), pp. 11575–11585.

Kenny, Eoin M., Ruelle, Elodie, Geoghegan, Anne, Shalloo, Laurence, O'Leary, Micheál, O'Donovan, Michael, Temraz, Mohammed, Keane, Mark T., 2021b. Bayesian case-exclusion and personalized explanations for sustainable dairy farming. In: International Joint Conference on Artificial Intelligence.

Labaien, Jokin, Zugasti, Ekhi, De Carlos, Xabier, 2020. Contrastive explanations for a deep learning model on time-series data. In: International Conference on Big Data Analytics and Knowledge Discovery. Springer, pp. 235–244.

Lage, Isaac, Chen, Emily, He, Jeffrey, Narayanan, Menaka, Kim, Been, Gershman, Sam, Doshi-Velez, Finale, 2019. An evaluation of the human-interpretability of explanation. arXiv:1902.00006.

Le Nguyen, Thach, Gsponer, Severin, Ilie, Iulia, O'Reilly, Martin, Ifrim, Georgiana, 2019. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. Data Mining and Knowledge Discovery 33 (4), 1183–1222.

Leake, David, McSherry, David, 2005. Introduction to the special issue on explanation in case-based reasoning. Artificial Intelligence Review 24 (2), 103.

Leavitt, Matthew L., Morcos, Ari, 2020. Towards falsifiable interpretability research. arXiv:2010.12016.

Leonardi, Giorgio, Montani, Stefania, Striani, Manuel, 2020. Deep feature extraction for representing and classifying time series cases: Towards an interpretable approach in haemodialysis. In: Flairs-2020. AAAI Press.

Li, Oscar, Liu, Hao, Chen, Chaofan, Rudin, Cynthia, 2017. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. arXiv:1710.04806.

Lim, Brian Y., Dey, Anind K., Avrahami, Daniel, 2009. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2119–2128.

Lipton, Zach C., 2018. The mythos of model interpretability. Queue 16 (3), 30.

Liu, Shusen, Kailkhura, Bhavya, Loveland, Donald, Han, Yong, 2019. Generative counterfactual introspection for explainable deep learning. In: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP). IEEE, pp. 1–5.

Lucic, Ana, Haned, Hinda, de Rijke, Maarten, 2020. Why does my model fail? Contrastive local explanations for retail forecasting. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 90–98.

Lundberg, Scott M., Lee, Su-In, 2017. A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems, pp. 4765–4774.

McCloy, Rachel, Byrne, Ruth M.J., 2002. Semifactual "even if" thinking. Thinking & Reasoning 8 (1), 41–67.

Mertes, Silvan, Huber, Tobias, Weitz, Katharina, Heimerl, Alexander, André, Elisabeth, 2020. This is not the texture you are looking for! Introducing novel counterfactual explanations for non-experts using generative adversarial learning. arXiv:2012.11905.

Miller, Tim, 2019. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 267, 1–38.
Molnar, Christoph, 2020. Interpretable Machine Learning. Lulu.com.
Nguyen, Thu Trang, Le Nguyen, Thach, Ifrim, Georgiana, 2020. A model-agnostic approach to quantifying the informativeness of explanation methods for time series classification. In: Proceedings of the 5th Workshop on Advanced Analytics and Learning on Temporal Data at ECML 2020. Springer.
Nugent, Conor, Cunningham, Pádraig, 2005. A case-based explanation system for black-box systems. Artificial Intelligence Review 24 (2), 163–178.
Nugent, Conor, Doyle, Dónal, Cunningham, Pádraig, 2009. Gaining insight through case-based explanation. Journal of Intelligent Information Systems 32 (3), 267–295.
Nunes, Ingrid, Jannach, Dietmar, 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. User Modeling and User-Adapted Interaction 27 (3–5), 393–444.
Papernot, Nicolas, McDaniel, Patrick, 2018. Deep *k*-nearest neighbors: Towards confident, interpretable and robust deep learning. arXiv:1803.04765.
Petitjean, François, Ketterlin, Alain, Gançarski, Pierre, 2011. A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognition 44 (3), 678–693.
Ribeiro, Marco Tulio, Singh, Sameer, Guestrin, Carlos, 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. ACM, pp. 1135–1144.
Ross, Andrew Slavin, Doshi-Velez, Finale, 2018. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In: Thirty-Second AAAI Conference on Artificial Intelligence.
Rudin, Cynthia, 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1 (5), 206–215.
Sani, Sadiq, Wiratunga, Nirmalie, Massie, Stewart, 2017. Learning deep features for *k*-NN-based human activity recognition. In: Proceedings of the ICCBR-17 Workshop. ICCBR (Organisers).
Schäfer, Patrick, 2016. Scalable time series classification. Data Mining and Knowledge Discovery 30 (5), 1273–1298.
Schlegel, Udo, Arnout, Hiba, El-Assady, Mennatallah, Oelke, Daniela, Keim, Daniel A., 2019. Towards a rigorous evaluation of XAI methods on time series. arXiv:1909.07082.
Shin, Chung Kwan, Park, Sang Chan, 1999. Memory and neural network based expert system. Expert Systems with Applications 16 (2), 145–155.
Shortliffe, Edward H., Davis, Randall, Axline, Stanton G., Buchanan, Bruce G., Green, C. Cordell, Cohen, Stanley N., 1975. Computer-based consultations in clinical therapeutics: Explanation and rule acquisition capabilities of the MYCIN system. Computers and Biomedical Research 8 (4), 303–320.
Simonyan, Karen, Vedaldi, Andrea, Zisserman, Andrew, 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv:1312.6034.
Singla, Sumedha, Pollack, Brian, Chen, Junxiang, Batmanghelich, Kayhan, 2019. Explanation by progressive exaggeration. arXiv:1911.00483.
Sørmo, Frode, Cassens, Jörg, Aamodt, Agnar, 2005. Explanation in case-based reasoning–perspectives and goals. Artificial Intelligence Review 24 (2), 109–143.
Tintarev, Nava, Masthoff, Judith, 2007. A survey of explanations in recommender systems. In: 2007 IEEE 23rd International Conference on Data Engineering Workshop. IEEE, Istanbul, Turkey, pp. 801–810.
van der Waa, Jasper, Nieuwburg, Elisabeth, Cremers, Anita, Neerincx, Mark, 2021. Evaluating XAI: A comparison of rule-based and example-based explanations. Artificial Intelligence 291, 103404.
Van Looveren, Arnaud, Klaise, Janis, 2019. Interpretable counterfactual explanations guided by prototypes. arXiv:1907.02584.
Wachter, Sandra, Mittelstadt, Brent, Russell, Chris, 2017. Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard Journal of Law & Technology 31, 841.
Wang, Zhiguang, Yan, Weizhong, Oates, Tim, 2017. Time series classification from scratch with deep neural networks: A strong baseline. In: IJCNN. IEEE, pp. 1578–1585.
White, Adam, d'Avila Garcez, Artur, 2019. Measurable counterfactual local explanations for any classifier. arXiv:1908.03020.

Yang, Linyi, Kenny, Eoin, Lok, Tin, Ng, James, Yang, Yi, Smyth, Barry, Dong, Ruihai, 2020. Generating plausible counterfactual explanations for deep transformers in financial text classification. In: Proceedings of the 28th International Conference on Computational Linguistics, pp. 6150–6160.

Ye, Lexiang, Keogh, Eamonn, 2011. Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. Data Mining and Knowledge Discovery 22 (1–2), 149–182.

Zeiler, Matthew D., Fergus, Rob, 2014. Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer, pp. 818–833.

Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, Torralba, Antonio, 2016. Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929.

Zhu, Jiapeng, Shen, Yujun, Zhao, Deli, Zhou, Bolei, 2020. In-domain GAN inversion for real image editing. In: European Conference on Computer Vision. Springer, pp. 592–608.