IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 May 2022, accepted 29 June 2022, date of publication 7 July 2022, date of current version 14 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3189432

RESEARCH ARTICLE

FCE: Feedback Based Counterfactual Explanations for Explainable AI

MUHAMMAD SUFFIAN[1], PIERLUIGI GRAZIANI[1], JOSE M. ALONSO[2], (Member, IEEE), AND ALESSANDRO BOGLIOLO[1]

[1]Department of Pure and Applied Sciences, Università degli Studi di Urbino Carlo Bo, 61029 Urbino, Italy
[2]Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782 Santiago de Compostela, Spain

Corresponding author: Muhammad Suffian (m.suffian@campus.uniurb.it)

This work is funded by the Department of Pure and Applied Sciences, Universita degli Studi di Urbino Carlo Bo, Urbino, Italy.

ABSTRACT Artificial Intelligence can provide quite accurate predictions for critical applications (e.g., healthcare), but lacks the ability to explain its internal mechanism in most applications which require high interaction with humans. Even if many studies analyze machine learning models and their learning behavior and eventually provide an interpretation of the inner mechanics of these models, these studies often entail a simpler surrogate model, which generates explanations by producing a piece of interpretable information such as feature scores. The crucial caveat against these studies is the lack of human involvement in the design and evaluation of explanations, consequently giving rise to trust issues and lack of acceptance and understanding. To this end, we address this limitation by involving humans in the counterfactual explanation generation process which is enriched with user feedback, thus enhancing the automated explanations which are better aligned with user expectations. In this paper, we propose a user feedback based counterfactual explanation approach (FCE) for explainable Artificial Intelligence. In our work, we utilize feedback in two ways: first, to customize the explanations by providing the acceptable ranges in the feature space where to look for feasible counterfactuals, and second, to evaluate the generated explanations.

INDEX TERMS Counterfactual explanations, explainable AI, human-in-the-loop, interactive machine learning, user feedback.

I. INTRODUCTION

Over the last few decades, artificial intelligence (AI) has been overwhelmingly responsive to everyday applications and industries such as self-driving cars, financial services, and healthcare. The technological advancements (i.e., software, hardware) and algorithm updates are the main reasons behind the proliferation of AI. Digitization of multiple domains and industries has created big data that is difficult for humans to process to gain insights into decision-making tasks. It is fair to say that many AI algorithms have reached to human task performance in many domains. For example, in Go [1] and Poker games [2], these algorithms outperformed the professional players. Similarly, AI-based systems are more accurate in detecting breast cancer [3].

In critical decision-making systems, a number of machine learning (ML) models provide strikingly accurate decisions. Nevertheless, the lack of ability to explain the internal mechanism of decision-making undermines the users' trust and hence decreases acceptance and usability of the systems [4]–[6]. The explanation of ML models is of crucial importance, however, most of these models are an enigma because humans cannot inspect how they work and how they make decisions. This enigma is regarded as a *black box*, an AI system that doesn't provide any details about the internal working mechanisms to the user [7]–[10].

The need to transform black-box decisions into transparent decisions for human decision-makers has led to a new field of study known as explainable AI (XAI)[1] [11].

The associate editor coordinating the review of this manuscript and approving it for publication was Frederico Guimarães.

[1]XAI stands for eXplainable Artificial Intelligence. This acronym became popular when the USA Defense Advanced Research Projects Agency introduced the challenges of designing self-explanatory AI systems. (http://www.darpa.mil/program/explainable-artificial-intelligence)

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Many XAI techniques are found in the literature to make ML systems explainable. Most of the techniques involve a simpler *surrogate model* (modeling the changes in the prediction based on input changes) that interprets the internal mechanics of the underlying model in terms of feature scores, also employed by popular XAI methods such as LIME [12] and SHAP [13]. Nevertheless, these techniques compromise *fidelity* over *interpretability* and vice versa [14]. High fidelity explanations can be as complex as the original ML model and possibly less interpretable, but highly interpretable explanations can be too roughly approximated and possibly inconsistent with the original ML model (low fidelity). For example, in a binary classification setup, a model with two classes, P and Q, and an input sample x, the aforementioned techniques try to answer the question ''What evidence assures the model to predict class for sample instance?'' with a list of feature scores as essential factors in the input sample. This type of answer is only suitable and understandable by the audience familiar with ML models. For the rest of the audience, it could be better to answer a bit moulded question, ''Why didn't the model predict class Q?'', to get an answer, one can ask ''What need to be different in input sample x to predict class Q instead of P?''. This alternative way of inspecting outcomes furnishes a set of inputs akin to the original input and is regarded as a counterfactual mechanism. This mechanism is endorsed by counterfactual explanations proposed by Wachter *et al.* [15], which suggests changes in features and some human evaluation contradicts this. Counterfactual explanations are appealing because they provide a piece of actionable information, and have been deemed acceptable by some legislative bodies like GDPR[2] [16]. The new European regulation on AI also pushes in the same direction [17].

Counterfactual explanations are considered human-friendly and actionable; while their explanation format helps to interpret the underlying ML model, they have their own set of challenges. One of the significant challenges is to generate actionable counterfactual explanations. For example, in a bank loan application scenario, an applicant would be approved if the suggested counterfactual explanation became true: *"Increase your salary to double, increase your credit card usage, level up your education, and convert your regular account to deposit account"*. However, recommended actions could be impractical for the applicant. Also, such explanations should be realistic and follow some natural laws, i.e., age cannot be suggested to decrease, education cannot be suggested to level-up without an increase of age. Despite all the issues and challenges therein, the major limitation of these explanations is no user involvement in the process of explanation generation, which causes problems of understanding. The user involvement in providing his/her feedback to confine the feasible ranges of counterfactuals for understandable and actionable explanations is of paramount importance [18]–[20].

Generally, the user raises questions in response to ML outcomes such as *why*, *why-not*, and *what-if*. This paper focuses on *what-if* questions. For example, a *what-if* question ''What would be the outcome if the input values of some features get changed'' entails an inquiry that is solved by counterfactual explanations in the literature [18], [21]. In this paper, we consider the need for an explanation and propose an approach that involves (what-if) user feedback in its process. To demonstrate the working mechanism of our approach, we set a user-system interaction environment to record user feedback in two parts. In the first part, the user is prompted to provide a list of changeable features with their boundary ranges in the input sample. This feedback is used by our method to define a neighbourhood space to mine counterfactuals. This strategy is theoretically akin to LIME which samples the neighbourhood based on the instance of interest with specific proximity. We find this proximity constraint with user feedback, and the dependence (correlation) among the features is found based on highly correlated features in the user feedback. Our approach mines the neighborhood based on random search and heuristics (i.e., synthetic dataset) later used to generate the counterfactual explanations. The well-known method by Wachter *et al.* [15] is used to compute and minimize the distance between the instance of interest and the counterfactuals (perturbed samples). Here, it is important to note that for some cases user feedback constraints could end up with small neighbourhood that couldn't be enough to represent strong correlation among the features, nevertheless, our approach iteratively tries to find possibilities for counterfactuals by adopting the actual correlation among the features in training dataset. In the second part, the generated counterfactual explanations (counterfactuals') are presented to the user (domain expert) to evaluate their feasibility and actionability (acceptance).

The rest of the paper is organized as follows: Section II investigates the related work, including methods and frameworks. Section III introduces the proposed approach for XAI. Section IV presents some evaluation metrics and results. Finally, section V draws some conclusions and discusses future work.

II. BACKGROUND AND RELATED WORK

In the XAI research field, good research is being conducted to make the systems/models explainable [22]. Researchers have adopted different strategies to provide explanations, such as ante-hoc explanations (explainable by design also known as intrinsic interpretability) and post-hoc explanations (explanations extracted by doing analysis on outcomes of systems/models). Post-hoc explanations are further categorized into significant explanation stages, such as *model-specific* and *model-agnostic* reported by [4], [23]–[25]. Explanation techniques can be further categorized based on the scope of an explanation technique such as local and global [12], [13]. Local and global explanations answer to the local instance

[2]GDPR stands for General Data Protection Regulation, it is a regulation and law on data protection and privacy adopted in European Union.

of interest (single instance at a time) and general answer approximated for all instances of interest, respectively.

A. COUNTERFACTUAL EXPLANATIONS

Feasible and actionable counterfactual explanations (FACE) [26] is a post-hoc, model-agnostic, and local method for counterfactual explanation generation. It accounts for the actionable counterfactual explanations based on a reasonable degree of change in the feature values. It proposes how much change in the input sample could lead to the desired outcomes. It explores multiple paths and then follows some feasible paths achievable by the shortest distance metric defined on density-weighted metrics. It constructs a graph on the data points with weight, density, and condition function. It updates this graph and applies the shortest distance path algorithm (Dijkstra's Algorithm) to find the feasible data points to generate counterfactual explanations satisfying the user requirements. Another framework for Diverse Counterfactual Explanations (DiCE) proposed to generate and evaluate two characteristics: feasibility and diversity. It addresses the necessary tradeoffs and causal implications to optimize the generation of counterfactual explanations [27].

Another study about factual and counterfactual explanations computes the length of rules in a decision tree to estimate the conciseness of the generated explanations. It also calculates the number of counterfactuals and the best minimum distance to the explanation of the facts, to assess the relevance of the counterfactuals [28]. A framework GeCo [29] relies on a genetic algorithm tuned to prioritize the search for counterfactual explanations with the least number of changes. It introduces two new optimizations for the real-time performance of counterfactual generation: first, representation of candidate counterfactuals; and second, conditional clauses for the evaluation of the classifier. Similarly, some of the model-agnostic frameworks stress measuring how well the actual output of black boxes and output of interpretable models resemble the local neighbourhood of the instance of interest to be explained [30]. In most of the above-discussed approaches and frameworks, the notion of user involvement is not addressed. It motivates us to design and evaluate the user feedback into the generation of counterfactual explanations.

B. FACTUAL EXPLANATIONS

The explanation for learning models became highlighted when Trepan [31] extracted symbolic representation from trained neural networks. Trepan is a global but model-specific explainer; it extracts decision trees from neural networks and approximates the maximization of the ratio for fidelity. The extracted decision tree is then used as an explanation to the user. Single tree approximation (STA) [32] presents a post-hoc and global explanation method that builds a decision tree with the aim of explaining a black-box random forest (RF) model. STA was used to reduce the burden of filtering medical questionnaires, and its approximate decision tree was found valuable and relevant. RF models are reasonable in performance but do not provide enough space for interpretations of their decisions. Another method was proposed by Hara *et al.* [33] with a post-hoc strategy to better explain the output of RF models to users. They formalized tree ensembles as a simplification problem (model selection problem) to obtain the simplest representation that is essentially equivalent to the original one. To determine the model complexity they derived a Bayesian algorithm with the aim of approximating complex models to simple models for better interpretability.

Anchor [34] is a model-agnostic method that produces high precision rules as explanations. It performs a rigorous search for mining the best rules. Anchor works on all black-box prediction models, as it tries to learn the internal behavior of the model by perturbing the input features and recording those changes (edge, anchors). When a decision is changed, it develops the rules, which are later presented to the user as an explanation.

LOcal Rule-based Explainer (LORE) [35] is a model-agnostic and local method; it explains the reasons of the decision taken on a specific instance by employing rules and counterfactuals. First, it uses a genetic algorithm for sampling the neighbourhood records for the explanation. Then, it trains a decision tree on the sampled neighbourhood records of a given instance; then, it generates an explanation in a unit of two parts, i.e. decision rules and counterfactuals. LORE has proved empirically to outperform Anchor in a pool of benchmark datasets.

GoldenEye [36] is a model-agnostic and global algorithm that provides an explanation in the form of feature importance. It finds out the dependencies of interacting attributes iteratively by using multiple classifiers that learn differently and accumulates the overall importance of features in groups. In addition, Measure of Feature Importance (MFI) [37] is a model-agnostic, local and global explainer, which uses feature importance to explain predictions. It focuses on the interaction of features and detects the only features that impact each prediction. MFI could be applied to any model, such as non-linear algorithms and neural networks. Likewise, Local Interpretable Model-agnostic Explanation (LIME) [12] is a method that can be applied to any ML model. The primary working mechanism behind LIME is to perturb the neighbourhood of the instance of interest by creating a dataset with specific proximity and storing corresponding predictions to learn how the prediction values change on which features. It tries to interpret every individual model prediction by using the local inputs and estimating their predictions around the given actual prediction. It assigns weights to each feature by calculating and minimizing its underlying loss function.

SHapley Additive exPlanations (SHAP) [13] can be seen as an extension of LIME. On the one hand, LIME approximates the feature importance scores by doing regression, on the other hand, SHAP defines feature scores with so-called shapley values for computing the contribution of features in the prediction, a strategy given by Lloyd Shapley for cooperative games to fairly distribute the payoff among players [38].

SHAP defines three properties for feature attribution i.e., local accuracy (the model should be accurate on a local instance of interest), missingness (this ensures if a specific feature value is zero, then it earns zero shares in terms of importance), and consistency (this ensures the model changes and then the marginal contribution of a feature should remain consistent). SHAP is coming from cooperative games in economics, where the additivity of the monetary gain/loss between players has an intrinsic meaning, in contrast with ML explainability [39], [40]. In general, SHAP does not guarantee to be suitable for feature selection, for this reason, it is not used in counterfactual explanation methods for filtering out important features.

Finally, NeuroX [10] is a toolkit to analyze the individual neurons in neural networks. It facilitates understanding the interpretation of neurons and models visually. One of the utilities of this toolkit is that the user can select a specific neuron from the neural network; then changing the values associated with the target neuron, it is possible to figure out its effect on the model accuracy.

III. METHODOLOGY

In this section, we detail the key parts of our method. Our local model-agnostic explanation method generates a counterfactual explanation based on user feedback. It contains two key components: (i) generation of counterfactual explanations, and (ii) evaluation of generated explanations. The main novelty in our approach is to integrate the user feedback in explanation generation. As a starting point, we extend the concept of neighbourhood space around the instance of interest given in the LIME method. LIME performs local regression on a neighbourhood space around the instance of interest and attempts to explain why some outcome y is predicted by model m given an input instance x. We use this concept of neighbourhood space, but in our approach, it is defined by user feedback to mine the candidate counterfactuals.[3] Then, we compute the distance between the actual instance and candidate counterfactuals (i.e., synthetically perturbed instances) to discern the counterfactual explanations.

The rest of this section is organized as follows. Section III-A provides the reader with answers to the following questions:

- How does the user feedback define a neighbourhood space?
- How are the candidate counterfactuals sampled?
- How are the counterfactual explanations generated?

Then, section III-B introduces the experimental setup for the empirical evaluation to be discussed in section IV. For a better understanding about the proposed approach, an interaction diagram is drawn in Fig.1. This diagram illustrates the user interaction and feedback to compute the candidate counterfactuals with a use case of the bank loan application.

[3]In the rest of this paper, the terms candidate counterfactuals and counterfactuals are used interchangeably.

A. FEEDBACK BASED COUNTERFACTUAL EXPLANATIONS (FCE)

Throughout this paper, we adopt the notation for ML classifiers as $m : X \rightarrow Y$ where m is the model, X is a set of instances (dataset), and Y is the set of corresponding outcomes (classes). Taking a single instance $x \in X$ as an instance of interest to be classified as $y \in Y$ that is a categorical output for x could be mapped as $m : x \rightarrow y$. Another way to write it is, $m(x) = y$, where the test instance includes n features as $x = \{x_1, x_2, x_3, \ldots, x_n\}$.

The goal of counterfactual explanations is to devise and suggest the smallest change in the test instance to flip the current outcome to the desired outcome. To do so, the test instance is changed to *a candidate counterfactual* instance $x^{cf} = \left\{x_1^{cf}, x_2^{cf}, x_3^{cf}, \ldots, x_n^{cf}\right\}$, for the prediction function $m(.)$ to obtain the desired outcome y' as $m(x^{cf}) = y'$.

Now, we formulate x^{cf} by using user feedback. User feedback is split into two parts: (i) a list of those features that the user admits to being changed (in other words, a set of feasible input fields), and (ii) feedback providing the ranges of those desired features. Also, the instance of interest could contain features belonging to different data formats (specifically, numerical and categorical features in our case). We can write

$$u = u^{num} + u^{cat} \tag{1}$$

where u^{num} is feedback about the numerical features and u^{cat} is feedback about categorical features. Further, the numerical part of the feedback is expanded as $u^{num} = [u_{f_1}^{num}, u_{f_2}^{num}, \ldots u_{f_k}^{num}]$, where $u_{f_1}^{num}$ represents to the first numerical feature on which user feedback is provided, and k represents the total number of numerical features with feedback in the instance of interest. For numerical features, the user can specify a feasible range for the selected features. For example, for the first numerical feature it takes the form, $u_{f_1}^{num} = [val_{min}^1, val_{max}^1]$, where val_{min}^1 is the minimum value and val_{max}^1 is the maximum value in the user-provided range. Hence, the first part of user feedback takes the following form,

$$u^{num} = \left\{u_{f_1 \ldots f_k}^{num} : [val_{min}^{1 \ldots k}, val_{max}^{1 \ldots k}]\right\} \tag{2}$$

Equation (2) represents the user feedback about each numerical feature. Similarly, user feedback about the categorical features is recorded. User-defined feature values are supposed to be flipped with value 1 and not with value 0. It takes the form (3)

$$u^{cat} = \left\{u_{f_1 \ldots f_k}^{cat} : val \in \{0, 1\}\right\} \tag{3}$$

The next target is to compute the neighbourhood space (i.e., a synthetic dataset) around the instance of interest for searching *candidate counterfactuals*. Candidate counterfactuals x^{cf}, are those that will participate in an evaluation (where the distance from actual instance and validity of outcome will be analyzed with the approach proposed

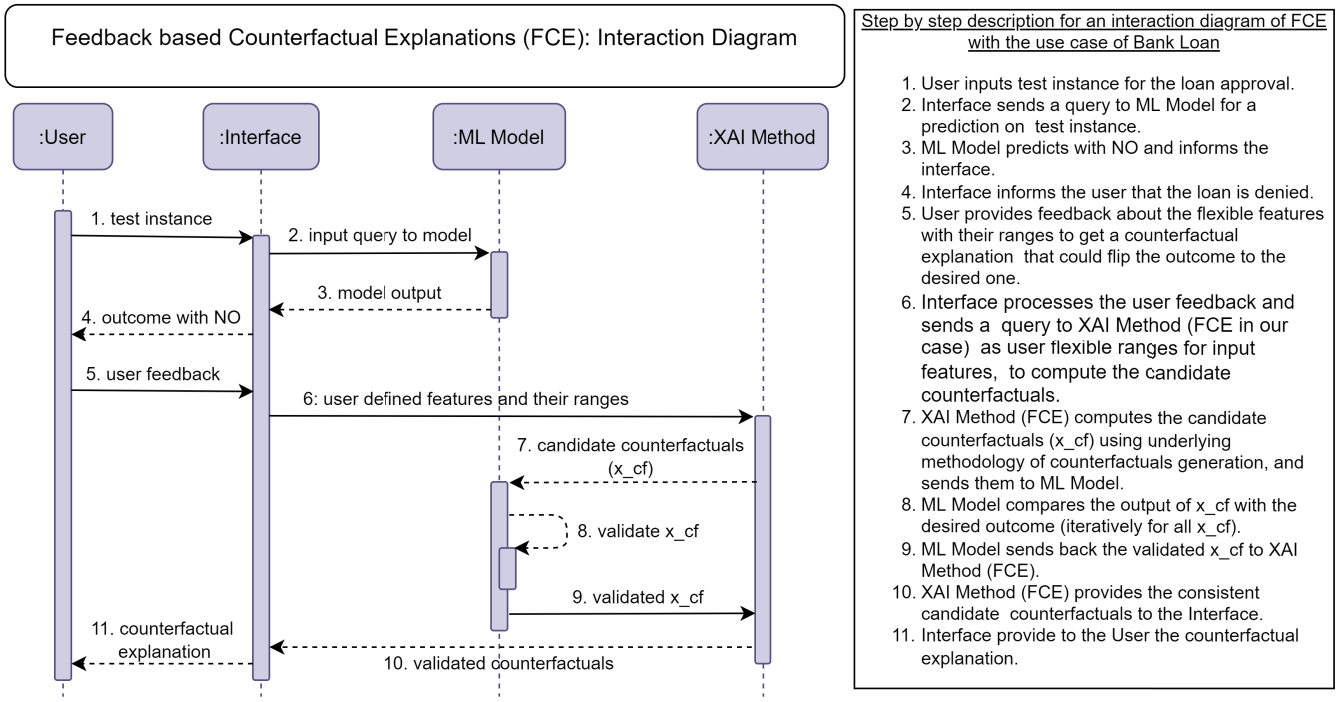

FIGURE 1. An interaction diagram of the proposed approach FCE is illustrated and its process flow is described step by step with a use of bank loan (loan denied).

in [15]), upon positive evaluation, each instance in x^{cf} could be recognized as a plausible counterfactual explanation. Here, we are also focusing on the feature-dependence, realisticness and intrinsic aspect of the counterfactual explanations, we design some heuristics based on domain knowledge for a neighbourhood space to mine the candidate counterfactuals. For example, in the case of a bank loan application, the counterfactuals should not suggest to decrease Age to get accepted for the loan. We have restricted it at the level of user feedback, by $u^{num}_{Age} \geq x_{Age}$, where u^{num}_{Age} holds the user feedback for feature Age, and x_{Age} is the actual age of the user. Similarly, the feature-dependence is calculated among the user-provided features (for those features appearing in user specified range), we accounted for a positive correlation to address the feature-dependence. Also, some other constraints are applied to ensure the realisticness and feasibility of counterfactuals at the initial level of explanation generation. The user-provided ranges help to perturb the instance of interest within the proximity of defined neighbourhood space. For each feature, the marginal distribution is estimated from (2) to generate the data entries. Eventually, synthetic data for all features are generated, changing one feature at a time, changing correlated features and their combinations ensuring that data for only one feature or combination is generated at a time, keeping others constant. Subsequently, (to harness the generated data with categorical features), (3) helps to identify the values of categorical features to be used, either to keep them the same or to flip in the candidate counterfactuals. The user feedback $u = u^{num} + u^{cat}$ is essential to generate candidate counterfactuals x^{cf}, thus, we can write it as $x^{cf} = u$.

Generally, good counterfactuals are scrutinized on many characteristics, we are using two of them. First, they should produce outcomes as close as possible to the defined or actual outcomes. Second, they should be as similar as possible to the actual instance regarding feature values. In the former, the distance between the outcome of the counterfactual instance and the actual instance (instance of interest) is pointed out. For a classification task, it is a distance of 0 or 1 (whether the predicted class is the same or not). In the latter, the distance between the feature values of the counterfactual instance and the actual instance is indicated. Here, we define and compute a loss function L that accounts for both of the above aforementioned characteristics, we minimized it using an optimization algorithm proposed by [15] in (4). The loss L takes the input parameters: the model m, an actual instance x, a counterfactual instance x^{cf} (sampled with (2) & (3)), and the user desired outcome y'. We adopted this loss function with the motivation that it optimizes (minimizes) the objective function using gradient descent to find counterfactuals that are close to the actual instance. We customized the distance function $d(.)$ defined in L as follows:

$$L(m, x, x^{cf}, y') = (m(x^{cf}) - y')^2 + d(x^{cf}, x) \tag{4}$$

Then, we used the Manhattan distance weighted with median absolute deviation for each feature and it is preferred over other distance metrics, because it provides an absolute value of distance. The limitation of (4) is not to handle categorical features. It is useful for various distance measurements where just the attributes that get change in counterfactuals

are counted to determine proximity to the real instance. It is easy to interpret the distance $d(.)$ which is represented in (5)

$$d(x^{cf}, x) = \sum_{f=1}^{n} (|x_f - x_f^{cf}|)/MAD_f \quad (5)$$

where f represents an individual feature, n total features, MAD_f is a median absolute deviation of f feature. The distance function in (5) calculates the sum of all features (feature-wise), that gives an absolute value between the actual instance x and a candidate counterfactual x^{cf}. This process is repeated for all candidate counterfactuals, and the top-k candidates with minimal distance are selected as counterfactual explanations. Also, the two characteristics of a good counterfactual explanation are significant from (4) as undertaken above, answering questions: (i) how far the desired result of the counterfactual instance is from the predefined result; and (ii) how far the counterfactual instance is from the actual instance (instance of interest).

B. EXPERIMENTAL SETUP

This section summarizes the main implementation tasks for data pre-processing, training the models, and generating counterfactuals.

1) DATA PRE-PROCESSING

In our proposed approach, we have undertaken the Loan data for a classification problem from Kaggle.[4] This dataset consists of 5000 instances with 14 features (including ID, ZIP Code). The classification task consists of predicting one out of two classes: (i) loan granted and (ii) loan denied. The dataset was found to be unbalanced (9.6% loan granted and 91.4% loan denied). We first balanced the dataset by using the class balancing technique SMOTE. The data for imbalanced class "loan granted" was increased from 10% to 35% of the whole data. We applied few heuristics to substantiate the separation line of the generated cases from the opposite class cases. Still, there could be a bit of chance of closeness between the generated and opposite class cases, and further discussion on this aspect is not included considering out of the scope of this work. We drop the ID and ZIP Code features. The remaining numerical features are Age, Experience, Income, Family, Education, CCAvg (Credit card avg. usage), Mortgage; and categorical features are Securities account, CD account (current deposit account), Online (account is online or not) and Creditcard (either have credit card or not); and the Personal Loan is the output class (either loan granted or denied). The accuracy of classification models was improved after balancing the data.

2) IMPLEMENTATION DETAILS

We implemented the proposed approach in Python language and used *Google Colab's*[5] useful utility of Forms that allows the user to interact at runtime. We employed two classifiers: Support vector machines (SVM) and Random Forests (RF); we applied default configuration parameters for each model. Then, we trained and tested both classifiers to measure their goodness. We used 10-fold cross validation. The training, average cross-validation for 10-folds, and testing scores are presented in Table 1, where RF is more accurate as compared to SVM.

TABLE 1. Accuracy and F-measure scores of support vector machines (SVM) and random forest (RF) on loan data.

Model	Training	Avg. Cross validation	Testing	F1-Score
RF	0.95	0.95	0.94	0.95
SVM	0.93	0.93	0.92	0.91

We have designed the form to record the user feedback as a response to the model prediction. The form includes a *what-if* question that is handled with counterfactual explanations. The layout of the form includes selecting features and respective feature ranges that users can modify to obtain desired results. This user feedback is filtered through some constraints, in u^{cat}, user-selected features are converted into a binary format, and we store 1 against the selected features and 0 for the rest of categorical features (meaning 1 to be changed and 0 to keep constant). Similarly, in the case of u^{num}, we saved it as a key-value mapping, keeping user-selected features as keys and their respective range list as values.

3) EXAMPLE CASE WITH PROPOSED METHOD

Here, we provide readers with an illustrative example regarding how our proposal works in practice. Let us suppose the actual instance x is,

$$\begin{aligned} x = \{&Age = 62, Experience = 36, Income = 183,\\ &Family = 2, CCAvg = 3.4, Education = 3,\\ &Mortgage = 0, SecuritiesAccount = 0,\\ &CD - Account = 0, Online = 0, Creditcard = 0\} \end{aligned}$$

and u^{num} is,

$$\begin{aligned} u^{num} = \{&u^{num}_{Experience} : [36, 45], u^{num}_{Income} : [183 : 200],\\ &u^{num}_{CCAvg} : [3.4 : 7], u^{num}_{Mortgage} : [0 : 60]\} \end{aligned}$$

where u^{cat} is,

$$u^{cat} = \left\{u^{cat}_{CD-Account} : 1, u^{cat}_{CreditCard} : 1\right\}$$

After getting user feedback, candidate counterfactuals are generated (synthetic data) by perturbing x within the provided space. One example of a candidate counterfactual is represented as

$$\begin{aligned} x^{cf} = \{&Age = 62, Experience = 40, Income = 195,\\ &Family = 2, CCAvg = 6.39, Education = 3,\\ &Mortgage = 55, SecuritiesAccount = 0,\\ &CD - Account = 1, Online = 0, Creditcard = 1\} \end{aligned}$$

[4] https://www.kaggle.com/sriharipramod/bank-loan-classification/metadata

[5] https://github.com/msnizami/FCE/blob/main/FCE.ipynb

In this way, the distance is computed between x and x^{cf} by minimizing the loss function presented in (4). We recorded user feedback on the generated explanations, and then the user is provided with generated counterfactual explanations. It is done dynamically on runtime and only those candidate counterfactuals are selected as counterfactual explanations that fulfill the criterion of good explanations mentioned in section III-A. Multiple counterfactual explanations are generated for a specific instance and the quality of generated counterfactual explanations is evaluated and discussed in section IV.

IV. RESULTS AND EVALUATION

Our approach has reported encouraging results with the experimental setup previously described. Unfortunately, there is no universal metric to measure the quality and effectiveness of explanations. In the recent literature regarding counterfactual explanations, different evaluation metrics are used. We have employed two evaluation metrics, i.e., *target-class-validity* and *continuous-proximity*, proposed by [41]. Also, for evaluation purposes, we collected and carefully analyzed user feedback for $N = 100$ test instances. Then, the proposed approach generates counterfactuals for the test instances under study, and the related explanations were evaluated on the aforesaid evaluation metrics, also analyzed with the authors' domain knowledge.

Target-Class-Validity: The percentage of the generated counterfactuals whose predicted class by ML classifier is the same as the target class also represented as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{K}\left[m(x_{i,j}^{cf}) = y'\right]/N * K \tag{6}$$

where we took $N = 100$ test instances and generated K counterfactuals for each instance (K depends on user feedback), y' is target class, X^{cf} is the set of candidate counterfactuals. The target-class-validity was recorded 84% for the above N user feedback-based test instances. Equation (6) is advantageous because it helps to determine the percentage of valid counterfactual explanations. In this work, the focus was on the generation of counterfactuals based on the user feedback, and we attempted to compute those counterfactuals which are valid counterfactuals (good explanations).

Continuous Proximity: The proximity for numerical features is calculated as the average distance between x^{cf} and test instance x in units of median absolute deviation for each feature, represented mathematically as follows

$$\sum_{i=1}^{N}\sum_{j=1}^{K}\sum_{f=1}^{M}(x_{i,j,f}^{cf} - x_{i,j,f}^{f}/MAD_f)/N * K * M \tag{7}$$

where f is a specific feature, M represents the total number of numerical features, K represents the number of generated counterfactuals, and N represents the number of test instances taken for evaluation. This measure of proximity could be compared from Table 2, where the median absolute deviation (MAD) for the actual instance and the generated counterfactuals are presented feature-wise. It could be observed that the generated counterfactuals exhibit reasonable proximity. The lower the distance, the better the proximity and feasibility. By looking deeply with domain knowledge to cases under study, the values of income and mortgage reflected significant difference between the actual test instances and the generated counterfactuals as compared to other features. The behavior of MAD difference is regular for most of the features, despite the income and mortgage. The income and mortgage show a marginal difference between the actual data and the generated counterfactuals data. The reason behind this behavior could be the numerical value threshold and the user-chosen values. Since, in the real dataset, these feature values deviated largely, such as high income (a businessman who was granted the loan), but in the user feedback, the provided values were normal (moderate income, the users who provided feedback belong to average income class of population).

The example test instance (of bank loan) exemplified in section III-B3 is provided with a counterfactual explanation in Table 3. The suggested changes are confined to only user-selected features and within the user-defined ranges. Similarly, another test instance is presented in Table 4, for which three counterfactual explanations are generated with user feedback. Our results are encouraging to complement the proposed approach of counterfactual explanations with user feedback. Still, many aspects of good counterfactuals need to be explored in the future work, as we will discuss in the next section.

A. EXPERT ANALYSIS FOR COUNTERFACTUAL EXPLANATIONS

The procedure of analysis for selecting and validating the generated counterfactuals was done manually: (i) the generated counterfactuals should be in agreement with user feedback, and (ii) the difference in MAD between generated counterfactual and actual instance having the desired label. The first criteria for the validation of actionability ensures the counterfactuals should be in the range of user-provided ranges and produce the desired results. This analysis resulted in 84% of the success rate for the generated counterfactuals with the domain expert's manual validation for the N test instances, also depicted mathematically in (6). The second criteria was to select the most suitable and realistic counterfactual by the domain expert, choosing the one having the lowest MAD (proximity). For example, if the user specifies to generate 10 counterfactuals for a single test instance then the one having the lowest MAD is chosen, also, presented mathematically in (7). The numeric results of evaluation metrics for proximity are presented in Table 2. If the calculated MAD for the generated counterfactuals was in the affordable range (an average of test and train instances all having the same desired class label), then it was accepted, otherwise rejected. This evaluation on a sample set of test instances was satisfactory in agreement with the computation of results of (6) and (7) for the whole test set. The manual evaluation

TABLE 2. Median absolute deviation between actual test instances and generated counterfactuals.

Type of Instance	Age	Experience	Income	Family	CCAvg	Education	Mortgage
Actual Instances	9.6	9.7	34.0	1.02	1.22	0.73	69.79
Generated Counterfactuals (continuous features)	5.6	4.54	12.16	0.40	1.12	0.52	89.49

TABLE 3. Counterfactual explanation for the test instance of loan data described in section III-B3.

Type of Instance	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Secruties-Account	CD Account	Online	CreditCard
Actual Instance	62	36	183	2	3.4	3	0	0	0	0	0
Counterfactual	62	40	195	2	6.39	3	55	0	1	0	1

TABLE 4. Feedback based counterfactual explanations for a test instance with three counterfactuals.

Type of Instance	Age	Experience	Income	Family	CCAvg	Education	Mortgage	Secruties-Account	CD Account	Online	CreditCard
Actual Instance	51	25	80	1	4.9	1	0	0	0	0	0
Counterfactual-1	51	25	125	2	4.9	2	146	0	1	1	0
Counterfactual-2	51	26	182	4	4.7	2	111	1	1	1	0
Counterfactual-3	51	36	195	1	4.4	2	64	0	0	1	1

is a time consuming effort. All the N test samples were evaluated over a span of couple of days by dedicating few hours daily. The 16% of counterfactuals couldn't be qualified as actionable, in other words these counterfactuals were not in agreement with user feedback. This depends on fine-tuning of the parameters of user feedback, and improvement could be replicated with flexible parameters and their values.

V. DISCUSSION AND CONCLUSION

The approach for counterfactual generation that is presented in this paper has both benefits and drawbacks. The main added-value of using user feedback around the instance of interest is that the user always goes with less number of features to be changed what reduces computational cost (complexity). In addition, other counterfactuals and feature attribution methods do not care about causation (correlation in case of LIME) during the feature perturbations. However, in the case of user feedback, chances are higher that a causal relation will hold because it is human nature to follow some correlation in actions. For example, it is intuitive that if the user can increase the income, then he/she can increase the average credit card usage and increase the mortgage too. Also, if he/she cannot increase the income, he will indeed not select CCAvg and Mortgage features in the user feedback to be changed.

Counterfactual explanations give flexibility to the user by having a contrast in prediction. Furthermore, they provide the user with some actionable suggestions to obtain the desired outcomes. However, counterfactuals may also mislead the user sometimes. For example, in the case of a loan application, presenting some counterfactuals, such as ''double the income and increase the mortgage value to very high'', is not realistic. This is an issue associated with all approaches/methods devised for counterfactual explanations. The advantage of our approach is that it minimizes the risk of misleading counterfactuals by taking user feedback into account and focusing on the defined criteria. In other words, we can deem our approach to be user-centric as the counterfactual explanations are actionable and human-friendly. In addition, we analyzed the generated counterfactual explanations with domain knowledge and, in some cases, it has been observed no counterfactuals qualified as counterfactual explanations. This is due to the fact that the user feedback may be too constrained and (may) not provide enough space to compute the candidate counterfactuals that can flip the outcome. The issue of no counterfactuals highlighted above for counterfactual approaches with user feedback needs to be further investigated by devising learning-related constraints of the underlying prediction model, ensuring the generation of valid and effective counterfactual explanations.

In this paper, we have spotlighted the key aspect of an explanation (i.e., user-involvement) that has not been investigated in the recent literature. We presented an approach to counterfactual explanations with user feedback for XAI systems. The current state-of-the-art methods in the field of XAI have been described, and the problems therein are identified with a potential solution to enrich counterfactual generation with user involvement. The presented approach is unique and novel, as it contributes to the body of knowledge with user feedback-based counterfactual explanation method. As far as we know, no work in the literature has addressed this issue yet. This work paves the way towards user-feedback-based XAI that will assist researchers of XAI systems to explore and exploit different techniques based on user feedback.

The future work will extend the user involvement in the explanation generation process by focusing on the research question—to what extent does the user appreciate the presented explanation? In the current work, as the presented features/facts in the explanation fragment are relevant and known to the user, it is assumed that the user would perceive

the explanation to an acceptable extent. If new knowledge is needed to be present in the explanations, there must be an evaluation criterion to measure it, and we envisage a framework as a future line of action. Regarding how to increase the understanding of the user about the explanation, it could be targeted with the help of a gaming environment, where an agent can initiate a dialogue with the user to answer the user questions [42], to evaluate the understanding (under defined and designed parameters of evaluation). Moreover, to measure the goodness of explanations, a formal new evaluation criterion needs to be developed and verified with the help of interacting agents. For example, one of the evaluation criteria for the agents may be supported by a model checking framework to account for the trustworthiness of explanation on the grounds of computational tree logic [43]. Another perspective of our work where research could be conducted is to develop a criterion for the extent of *information to reveal* in the explanations. It is worth noting that some researchers have already highlighted the potential threats to those explanation systems in which the algorithms or models are subjected to be made transparent, providing enough space for intruders to manipulate the systems [44]. Hence, there is a need for comprehensive research keeping in view the privacy/security measures of such explanatory systems.

ACKNOWLEDGMENT

Muhammad Suffian is a Ph.D. Researcher (Matricola N.309445). Pierluigi Graziani's work was supported by the Italian Ministry of Education, University and Research through the PRIN 2017 project "The Manifest Image and the Scientific Image" prot.2017ZNWW7F_004. Jose M. Alonso is a Ramon y Cajal Researcher (RYC-2016-19802). Alessandro Bogliolo is a Full Professor and Coordinator of Europe CodeWeek. This work is funded by the Spanish Ministry for Science, Innovation and Universities (grant PID2021-123152OB-C21) and by the Galician Ministry of Culture, Education, Professional Training and University (grants ED431F2018/02, ED431C2018/29, ED431G/08, ED431G2019/04, ED431C2022/19). All grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

REFERENCES

[1] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, Oct. 2017.

[2] A. Blair and A. Saffidine, "AI surpasses humans at six-player poker," *Science*, vol. 365, no. 6456, pp. 864–865, Aug. 2019.

[3] S. M. McKinney *et al.*, "International evaluation of an AI system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, Jan. 2020.

[4] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[5] C. Miruna-Adriana and H. Hastie, "A survey of explainable AI terminology," in *Proc. 1st Workshop Interact. Natural Lang. Technol. Explainable Artif. Intell. (NLXAI)*, 2019, pp. 8–13.

[6] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Nov. 2021.

[7] J. A. Glomsrud, A. Ødegårdstuen, A. L. S. Clair, and O. Smogeli, "Trustworthy versus explainable ai in autonomous vessels," in *Proc. Int. Seminar Saf. Secur. Auto. Vessels (ISSAV) Eur. STAMP Workshop Conf. (ESWC)*, 2019, pp. 37–47.

[8] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.

[9] A. Alishahi, Y. Belinkov, G. Chrupała, D. Hupkes, Y. Pinter, and H. Sajjad. (Nov. 2020). *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*. [Online]. Available: https://aclanthology.org/2020.blackboxnlp-1.0

[10] F. Dalvi, A. Nortonsmith, A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, and J. Glass, "NeuroX: A toolkit for analyzing individual neurons in neural networks," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 9851–9852, Jul. 2019.

[11] D. Gunning, E. Vorm, J. Y. Wang, and M. Turek, "DARPA 's explainable AI (XAI) program: A retrospective," *Appl. AI Lett.*, vol. 2, no. 4, p. e61, Dec. 2021.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.

[13] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NIPS*, Long Beach, CA, USA. Red Hook, NY, USA: Curran Associates, 2017, pp. 4768–4777.

[14] R. Marcinkevičs and J. E. Vogt, "Interpretability and explainability: A machine learning zoo mini-tour," 2020, *arXiv:2012.01805*.

[15] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harv. J. Law Technol.*, vol. 31, no. 2, p. 841, 2017.

[16] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," in *A Practical Guide*, vol. 10, no. 3152676, 1st, ed. Cham, Switzerland: Springer, 2017, pp. 1–383.

[17] 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*. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52021PC0206

[18] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing design practices for explainable AI user experiences," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–15.

[19] K. Weitz, D. Schiller, R. Schlagowski, T. Huber, and E. André, "'Let me explain!': Exploring the potential of virtual agents in explainable AI interaction design," *J. Multimodal User Interfaces*, vol. 15, no. 2, pp. 87–98, Jun. 2021.

[20] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable AI systems," *ACM Trans. Interact. Intell. Syst.*, vol. 11, nos. 3–4, pp. 1–45, Dec. 2021.

[21] M. Cashmore, A. Collins, B. Krarup, S. Krivic, D. Magazzeni, and D. Smith, "Towards explainable ai planning as a service," in *Proc. 2nd ICAPS Workshop Explainable Planning (XAIP)*, Jul. 2019, pp. 1–9.

[22] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Farina, "A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence," *IEEE Access*, vol. 9, pp. 11974–12001, 2021.

[23] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *Proc. Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2018, pp. 210–215.

[24] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2018.

[25] D. V. Carvalho, M. E. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, Jul. 2019.

[26] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, "FACE: Feasible and actionable counterfactual explanations," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, Feb. 2020, pp. 344–350.

[27] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 607–617.

[28] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Fariña, "Generation and evaluation of factual and counterfactual explanations for decision trees and fuzzy rule-based classifiers," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2020, pp. 1–8.

[29] M. Schleich, Z. Geng, Y. Zhang, and D. Suciu, "GeCo: Quality counterfactual explanations in real time," *Proc. VLDB Endowment*, vol. 14, no. 9, pp. 1681–1693, May 2021.
[30] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and counterfactual explanations for black box decision making," *IEEE Intell. Syst.*, vol. 34, no. 6, pp. 14–23, Nov. 2019.
[31] M. Craven and J. Shavlik, "Extracting tree-structured representations of trained networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 8, 1995, pp. 24–30.
[32] Y. Zhou and G. Hooker, "Interpreting models via single tree approximation," 2016, *arXiv:1610.09036*.
[33] S. Hara and K. Hayashi, "Making tree ensembles interpretable: A Bayesian model selection approach," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2018, pp. 77–85.
[34] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1–9.
[35] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," 2018, *arXiv:1805.10820*.
[36] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou, "A peek into the black box: Exploring classifiers by randomization," *Data Mining Knowl. Discovery*, vol. 28, nos. 5–6, pp. 1503–1529, Sep. 2014.
[37] M. M.-C. Vidovic, N. Görnitz, K.-R. Müller, and M. Kloft, "Feature importance measure for non-linear learning algorithms," in *Proc. Interpretable ML Complex Syst. (NIPS)*, 2016, pp. 1–5.
[38] L. S. Shapley, *17. A Value for n-Person Games*. Princeton, NJ, USA: Princeton Univ. Press, 2016, pp. 307–318, doi: 10.1515/9781400881970-018.
[39] D. Fryer, I. Strumke, and H. Nguyen, "Shapley values for feature selection: The good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144352–144360, 2021.
[40] B. Kovalerchuk, M. A. Ahmad, and A. Teredesai, "Survey of explainable machine learning with visual and granular methods beyond quasi-explanations," in *Interpretable Artificial Intelligence: A Perspective of Granular Computing* (Studies in Computational Intelligence), vol. 937, W. Pedrycz and S. M. Chen, Eds. Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-64949-4_8.
[41] D. Mahajan, C. Tan, and A. Sharma, "Preserving causal constraints in counterfactual explanations for machine learning classifiers," 2019, *arXiv:1912.03277*.
[42] D. S. Watson and L. Floridi, "The explanation game: A formal framework for interpretable machine learning," *Synthese*, vol. 198, pp. 9211–9242, Oct. 2021, doi: 10.1007/s11229-020-02629-9.
[43] A. Termine, G. Primiero, and F. A. D'Asaro, "Modelling accuracy and trustworthiness of explaining agents," in *Logic, Rationality, and Interaction*, S. Ghosh and T. Icard, Eds. Cham, Switzerland: Springer, 2021, pp. 232–245.
[44] K. Gade, S. Geyik, K. Kenthapadi, V. Mithal, and A. Taly, "Explainable AI in industry: Practical challenges and lessons learned," in *Proc. Companion Web Conf.*, Apr. 2020, pp. 303–304.

MUHAMMAD SUFFIAN received the B.S. degree in computer science from Sukkur IBA University, in 2016, and the M.S. degree from Mohammad Ali Jinnah University (MAJU), Karachi, Pakistan, in 2018. He is currently pursuing the Ph.D. degree with the Universita Degli Studi di Urbino "Carlo Bo," Italy. From 2016 to 2020, he was with the Department of Computer Science, MAJU, as a Lecturer, for the same period, he was the part of the Center for Language Computing, MAJU. From January 2020 to November 2020, he was part of the School of Computing, NU-FAST University Chiniot-Faisalabad. His research interests include artificial intelligence, explainable artificial intelligence, machine learning, human-centered explainable AI, natural language generation, and computational linguistics.

PIERLUIGI GRAZIANI received the master's degree in philosophy from the University of Urbino, Urbino, Italy, in 2001, and the Ph.D. degree in logic and epistemology from the University of Rome La Sapienza, Rome, Italy, in 2007. From 2010 to 2013, he collaborated with the Department of Philosophy holding the position of a Postdoctoral Fellow in the history of mathematics, logic, and philosophy of science at the University of Urbino. From July 2014 to August 2017, he worked as a Postdoctoral Fellow in logic and philosophy of science at the University of Chieti-Pescara. From December 2018 to August 2021, he worked at the University of Urbino as a Postdoctoral Fellow in logic and philosophy of science. He is currently an Assistant Professor in logic and philosophy of science at the University of Urbino "Carlo Bo." His research focuses mainly on the foundation of geometry, logic and computer science, history/philosophy of logic and mathematics, and social robotics. The main results of his research appeared in journals and books published by Italian and international publishers.

JOSE M. ALONSO (Member, IEEE) received the M.Sc. and Ph.D. degrees in telecommunication engineering from the Technical University of Madrid (UPM), Spain, in 2003 and 2007, respectively. Since June 2016, he has been with the Research Centre in Intelligent Technologies (CiTIUS), University of Santiago de Compostela (USC). He is currently "Ramon y Cajal" Researcher funded by the Spanish Government under Project RYC-2016-19802, affiliated to CiTIUS-USC, the President of the Executive Board and the Deputy Coordinator of the H2020-MSCA-ITN-2019 Project titled "Interactive Natural Language Technology for Explainable Artificial Intelligence" (NL4XAI), the Chair of the IEEE-CIS Task Force on Explainable Fuzzy Systems, a member of the IEEE-CIS Task Force on Explainable Machine Learning, a member of the IEEE-CIS Working Group on eXplainable AI (P2976), a member of the IEEE-CIS Task Force on Fuzzy Systems Software, and a Board Member of the ACL Special Interest Group on Natural Language Generation (SIGGEN). He has published more than 150 papers in international journals, book chapters and conferences; being coauthor of the book *Explainable Fuzzy Systems—Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems* Studies in *Computational Intelligence* (Springer, 2021). His research interests include explainable artificial intelligence, interpretable fuzzy systems, natural language generation, and development of free software tools.

ALESSANDRO BOGLIOLO received the Laurea degree in electrical engineering and the Ph.D. degree in electrical engineering and computer science from the University of Bologna, Italy, in 1992 and 1998, respectively. From 1992 to 1999, he was with the Department of Electronics and Computer Science (DEIS), University of Bologna. In 1995 and 1996, he was with the Computer Systems Laboratory (CSL), Stanford University, CA, USA. From 1999 to 2002, he was with the Department of Engineering (DI), University of Ferrara, Italy. He joined the Università degli Studi di Urbino Carlo Bo, Italy, in 2002, where he is a Full Professor in computer systems. In 2019, he co-founded Digit srl, benefit corporation for digital social innovation. His research interests include mobile crowd-sensing, sensor networks, computational linguistic, and digital platforms for sustainability and participatory social innovation.