DOI: 10.17323/2587-814X.2024.3.24.40

Counterfactual explanations based on synthetic data generation

Yuri A. Zelenkov
E-mail: yzelenkov@hse.ru

Elizaveta V. Lashkevich
E-mail: evlashkevich@hse.ru

Graduate School of Business, HSE University, Moscow, Russia

Abstract

A counterfactual explanation is the generation for a particular sample of a set of instances that belong to the opposite class but are as close as possible in the feature space to the factual being explained. Existing algorithms that solve this problem are usually based on complicated models that require a large amount of training data and significant computational cost. We suggest here a method that involves two stages. First, a synthetic set of potential counterfactuals is generated based on simple statistical models (Gaussian copula, sequential model based on conditional distributions, Bayesian network, etc.), and second, instances satisfying constraints on probability, proximity, diversity, etc. are selected. Such an approach enables us to make the process transparent, manageable and to reuse the generative models. Experiments on three public datasets have demonstrated that the proposed method provides results at least comparable to known algorithms of counterfactual explanations, and superior to them in some cases, especially on low-sized datasets. The most effective generation model is a Bayesian network in this case.

Keywords: counterfactual explanations, synthetic data generation, multimodal distribution modelling, Bayesian network, credit scoring

Citation: Zelenkov Yu.A., Lashkevich E.V. (2024) Counterfactual explanations based on synthetic data generation. *Business Informatics*, vol. 18, no. 3, pp. 24–40. DOI: 10.17323/2587-814X.2024.3.24.40

Introduction

Recently, concern in interpretable AI (XAI) has grown rapidly, driven by the expanding use of machine learning algorithms in various fields of human endeavor [1, 2]. Moreover, many national and international regulators require transparency of algorithm-based decisions. In particular, the EU's General Data Protection Regulation (GDPR) provides citizens with the right to request "meaningful information about the logic involved and the meaning and intended consequences" of automated decisions, and US credit laws require that consumers be provided with reasons for unfavorable decisions [3]. Bank of Russia also follows the OECD recommendations on AI usage, whereby models should be transparent and interpretable to limit modeling risks and allow for independent external, internal and regulatory validation.

XAI methods can be categorized into two groups [4]. The first includes models where interpretability is a core property (e.g., decision trees or linear regression). The second group comprises methods that treat the model as a black box. In contrast to the models of the first group, they lack properties that provide a meaningful interpretation, so additional efforts must be made to explain the decision logic post facto (explainability). In the second group, in turn, we can distinguish the methods of model explanation, local result explanation and black-box examination [5].

This paper examines methods of counterfactual explanations [5–8]. A counterfactual explanation (CE) allows us, for a specified sample, to find a set of objects that belong to the opposite class but are as close as possible to the instance explained in the feature space. An example commonly cited in the literature is a borrower who was denied a loan based on the decision of an algorithm used at a bank. The objective of CE is to generate a profile for this borrower such that his application is approved (e.g., reducing the amount of the requested loan). An obvious constraint is the feasibility of the proposed changes, so the mandatory parameter minimized in this type of problem is the distance between the sample and the counterfactual. It follows from this example that, according to the above classification, CE belongs to the group of local ex post explanation methods, since it explains the solution of the trained model, treated as a black box, for a particular sample. In Russian, the concept of CE was first presented in a translated book [9].

A counterfactual is defined as a conditional statement in philosophy, the antecedent of which (a previous event that helps to understand the present) is false, while the consequent describes what the world would be like if the antecedent had occurred (an answer to the "what-if" question). According to the Great Russian Encyclopedia, counterfactual thinking is a type of thinking characterized by a person's tendency to imagine possible other variants of events that have already occurred, i.e. reflection contrary to facts.

While most XAI methods aim to answer the "why" question [4], counterfactual statements provide a means of interpretation by indicating what changes would be required to achieve a desired goal (prediction) rather than helping to understand why the current situation has a particular predicted outcome [8]. Therefore, many authors [5] state that CE corresponds to the third level of Pearl's causality models [10], which need to answer questions involving retrospective reasoning, e.g., "what is the probability of event y at x if there are x' and y' observed”. At the same time, CE also does not impose restrictions on model complexity and does not require disclosure of model information [3].

Obviously, CE methods are a powerful decision support tool in various fields such as finance [11, 12] and healthcare [13]. Several dozen CE algorithms are already known by now (see reviews [5, 6, 8] and other papers). Most of them are premised on optimizing some target function, and when this problem is solved each time a set of counterfactuals needs to be computed for a given sample. This imposes limitations on the performance and scalability of CE [6]. A possible alternative is to use methods that allow modeling the joint distribution of the features of the objects under study. In this case, a once-trained model can generate counterfactuals for different samples without significant computational costs.

Note that in this formulation the task can be viewed as generation of synthetic tabular data [14, 15]. Both statistical methods (copulas, Bayesian networks), and machine learning methods (variational autoencoders, generative adversarial networks, etc.) are used to create such models. [16]. Some scholars also adapt for this purpose oversampling methods that are designed to generate minor class objects in the case of imbalanced data [17].

Considering these circumstances, an approach to CE based on synthetic data generation principles is proposed here, involving two steps. In the first, a set of potential counterfactuals is generated, and in the second, a selection is made of those that satisfy the constraints of actionability, proximity, cost, etc. This organization allows to make the CE process transparent, manageable, reuse generation models and thus significantly reduce computational costs.

The rest part of the paper is organized as follows. After the review of literature, the proposed method is presented in section 2. Sections 3 and 4 set out the experimental results, comparing the proposed approach with other existing known CE methods. Finally, the limitations of the proposed method as well as directions for future research are discussed.

1. Literature review

1.1. Counterfactual generation

CE is based on several implicit assumptions [3]:

- the recommended variation of attribute values is unambiguously realized in the real world;
- the distribution of feature values can be reconstructed from available training data;
- the proposed modifications are relevant only to the decision being taken and do not affect other aspects;
- the model is stable in time, monotonic and restricted to binary outcomes.

As discussed above, CE is an actively growing area of research. The very term “counterfactual explanation” as applied to AI systems was first used in [18], However, papers using a similar approach have begun to appear since the mid-2010s [5].

Let’s give formal definitions. Consider a classifier $h\colon \mathcal{X} \rightarrow \mathcal{Y}$ trained on the data set

$$\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}, x_i \in \mathcal{X}, y_i \in \mathcal{Y},$$

where $\mathcal{X} \subset \mathbb{R}^m$ is a feature space, and $\mathcal{Y}$ is class label range.

It is usually assumed $\mathcal{Y} = \{0, 1\}$ but all the proposed definitions can be simply generalized to the case of multiclass classification. Each instance x_i is a vector m feature pairs, $x_i = \{(a_j, v_{ij})\}_{j=1}^{m}$, where a_j is an attribute, v_{ij} is its value from the domain a_j. Attributes can be either categorical, ordinal or continuous.

Definition 1. If a classifier h assigns the label $y = h(x)$ to an instance x then the counterfactual explanation of x is an instance x^* whose label is different from y, i.e., $h(x^*) \neq y$, with the difference betweeny x and x^* being minimal.

The concept of minimal difference is not specified here as it depends on the context of the problem and will be discussed later.

Definition 2. The counterfactual explainer is a function f_k that, for a dataset $\mathcal{D}$, a classifier h and an instance x, returns a set $C = f_k(h, \mathcal{D}, x)$ of $l \leq k$ valid counterfactual examples $C = \{x_1^*, \ldots, x_l^*\}$, where k is the number of counterfactuals required.

Characteristics that enable us to evaluate the quality of the counterfactual generation algorithm:

1. Validity is measured by the ratio of the number of counterfactuals that have the required class label to the total number of generated objects [11]:

$$V = |C_v| / |C|,$$

where C_v is a set of valid counterfactuals generated by the model f_k;C is a set of samples generated f_k, $C_v \subset C$.

The validity of the generated sample is determined using a predictive model h, for a valid example. The following condition must be fulfilled $h(x^*) \neq h(x)$. As follows from the definition, the maximum validity value is $V = 1$; values less than 1 indicate insufficient efficiency of the model.

2. Proximity is the distance of a counterfactual from the sample for which the explanation is generated. The proximity of a set of counterfactuals is estimated by the average distance on this set [19]:

$$P = \frac{1}{|C_v|} \sum_{x^* \in C_v} dist\,(x^*, x).$$

To measure distance $dist(x^*, x)$ most used L_0, L_1, L_2 and L_∞ norms, $L_k = \left(\sum_i |x_i|^k\right)^{1/k}$, and its weighted combinations. The lower the value P, the closer the objects found are to the explained factual.

3. Sparsity estimates the number of features that need to be changed to move into the counterfactual class. It is preferable for counterfactuals to have the smallest possible changes in their features. This property allows for more efficient, human-understandable and interpretable counterfactuals [18].

$$S = \frac{1}{|C_v|} \sum_{x^* \in C_v} K(x^*).$$

$K(x^*)$ is the number of counterfactual attributes which value changes in comparison with the factual x. Thus, models with a lower value of S are preferred.

4. Diversity. Searching for the closest instances according to a distance function can lead to very alike counterfactual candidates with few differences between them. Diversity implies that the counterfactual generation process produces different explanations for the same sample. This leads to explanations that are more interpretable and more understandable to the user. The authors [19] propose to use the average distance between all pairs of valid counterfactuals as a measure of diversity:

$$D = \frac{1}{|C_v|^2} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} dist(x_i^*, x_j^*),$$

here $dist(x_i^*, x_j^*)$ is a measure of the distance between two counterfactuals x_i^* and x_j^*. The higher the diversity, the more efficient the CE algorithm.

5. Plausibility. This property emphasizes that the generated counterfactuals must be legitimate, and the search process must provide logically valid results. This means that the counterfactual found should never change immutable characteristics such as gender or race. Three categories of plausibility are distinguished in the literature [20]:

- domain consistency, which limits the range of acceptable values of counterfactual attributes;
- distributional consistency requires that the probabilities of counterfactual feature values are matched to the (empirical) distribution of the data. This property can be measured [6] as the average distance to k nearest neighbors e.g., local outlier factor (LOF) [21], as well as by kernel density estimation (KDE). In the last case, the density of distribution of each attribute is estimated based on KDE, and then the probability of the corresponding counterfactual attribute belonging to this distribution is calculated. This approach has obvious limitations – each attribute is considered separately, and it is applicable only to continuous attributes. The nearest neighbors method has no such limitations;
- prototype consistency selects counterfactual instances that are either directly present in the dataset or are close to the data object being explained.

Note that this property is close to the definition of proximity presented above.

In this paper, we will use a plausibility measure based on the LOF value, i.e.

$$U = \frac{1}{|C_v|}\sum_{x^* \in C_v} LOF(x^*).$$

Note that LOF values are difficult to interpret due to the local nature of the method. Values about 1, says that the point is interior; the higher the value, the more likely it is an outlier. Thus, from the point of view of evaluating the CE algorithm, values close to 1 are preferred.

6. Actionability / feasibility. Finding the closest counterfactual for a data instance does not necessarily result in a feasible change in characteristics. The feasibility of changing a particular variable is described by one of three categories:

- the attribute can be actionable and therefore the attribute is mutable, e.g., balance sheet data;
- the attribute is changeable, but the change is not feasible (e.g., credit rating);
- the attribute is unchangeable (e.g., place of birth).

Remark that the user cannot change the values of the variables of the last two categories, but these values can change because of their ancestors in the causal model [20]. Some authors assume that fulfilling the feasibility requirement automatically guarantees the plausibility of a counterfactual recommendation [22], but despite some overlap, these are different concepts [20]. Feasibility restricts the set of actions to those that can be performed, while plausibility requires that the resulting counterfactual be realistic.

Authors of review articles [5–8] use different taxonomies of CE methods. Here we propose a classification based on the architecture of the models used (*Fig. 1*).

The first group of techniques is based on the solution of an optimization problem in which some of the above properties are treated as a target function and the remaining properties are treated as constraints. For example, in [18], the distance is used as a target $dist(x^*, x)$ with a counterfactual label restriction $h(x^*) = y^*$. This task can be transformed into a problem described by a differentiable function without constraints:

$$x^* \in \arg\min_{x^*}\max_{\lambda} \lambda(h(x^*) - y^*)^2 + dist(x^*, x).$$

The $\lambda(h(x^*) - y^*)^2$ term ensures that the counterfactual label matches the desired class.

Such an approach can be extended to include constraints on actionability, sparsity, data manifold closeness and others, for example [6]:

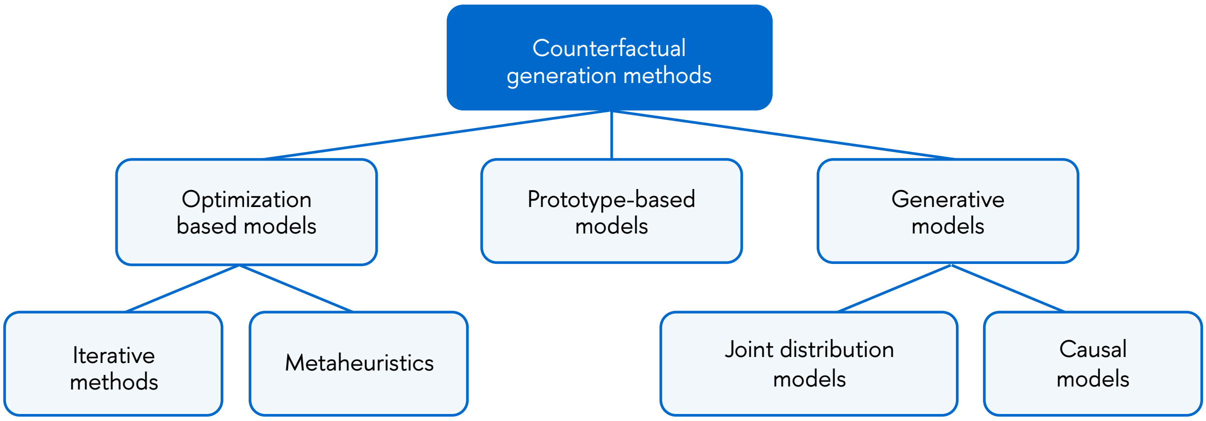
Counterfactual generation methods
Optimization based models
Prototype-based models
Generative models
Iterative methods
Metaheuristics
Joint distribution models
Causal models

Fig. 1. Classification of counterfactual generation algorithms.

$$x^{*} \in \arg \min_{x^{*} \in \mathcal{A}} \max_{\lambda} \lambda \left(h(x^{*}) - y^{*} \right)^{2} + dist(x^{*}, x) + \\ + g(x^{*} - x) + l(x^{*}; \mathcal{X}).$$

The condition $x^{*} \in \mathcal{A}$ restricts the list of attributes x^{*} to ones whose modification is feasible, $g(x^{*} - x)$ is a penalty function for the difference between the original instance and the counterfactual (e.g., L_0, L_1 norm), and $l(x^{*}; \mathcal{X})$ is a penalty function for deviation from the data manifold.

Authors of optimization-based CE methods focus first on defining a target function including various metrics for the above properties, and then on choosing an algorithm to find the optimum. It is usually not possible to guarantee the convexity of the target function in this process. Iterative methods of various orders are often used, and metaheuristics (e.g., genetic algorithms) are also widely used. However, this approach requires solving an optimization problem when generating counterfactuals for each new sample. Therefore, [6] recommends authors of such works to cite computation time as one of the algorithm characteristics.

The second group of methods is based on searching in $\mathcal{D}$ for prototypes that will be used to generate counterfactuals [23]. Conceptually, this approach is close to Case Based Reasoning (CBR) [24], which includes four steps: (1) retrieve – extract a case relevant to the problem to be solved, (2) reuse – map the solution found to the problem, (3) revise – test the solution and revise it, if necessary, (4) retain – save the successfully adapted solution.

In particular, an algorithm is proposed in [25], according to which a dataset $\mathcal{D}$ is considered as a set of pairs (x, x^{*}), where (x, x^{*}) are the closest objects for which $h(x^{*}) \neq h(x)$. For a given factual z the closest instances of x, belonging to the same class are found, $h(z) = h(x)$. The attribute values of the counterfactual z^{*} are initialized with the values from z, then those attributes that differ in x and x^{*} are changed until z^{*} is found such that $h(z^{*}) = h(x^{*})$. If this condition is not achieved, the following pair (x, x^{*}) is used. The idea is that z^{*} should differ from z in the same way that x^{*} differs from x.

The third group of CE methods (generative models) is based on modeling the process of data generation. Two types of models can be distinguished in this group: joint distribution modeling and causal models.

A model of the joint distribution $P(\mathcal{X})$ is trained from observations $\mathcal{D}$ and then used to find counterfactuals. As such a model, CE most commonly uses variational autoencoders (VAE), which consist of two parts – an encoder that maps the feature distribution $P(\mathcal{X})$ in $\mathbb{R}^{m}$ space into the distribution of latent variables $P(\mathcal{Z})$ in a space of lower dimensionality $\mathcal{Z} \subset \mathbb{R}^{k}$ $(k < m)$, and a decoder that generates the value x' corresponding to the point z' in $P(\mathcal{Z})$. The VAE-based approach offers the interesting prospect of searching for counterfactuals in latent space; in particular, some authors use gradient descent for this purpose [26, 27], but as shown in [28], this is associated with potential problems.

The authors of VAE-based CE methods must consider the above requirements for generating counterfactual explanations, so they introduce additional constraints to the latent representation model. Thus, [29] adapts the traditional scheme, where the encoder is only used to find $P(\mathcal{Z})$ and is not involved in data generation and includes it in the generation process. The encoder is used to find a point z in the latent space corresponding to a given factual x; the counterfactual is generated from the point $z^{*} = z + \delta$, where δ is a small perturbation. This should enforce the proximity requirement. In addition, the authors of this paper cluster the latent space based on a Gaussian mixture to obtain a conditional distribution $P(\mathcal{Z} \mid \mathcal{I})$, where $\mathcal{I}$ is the set of immutable features.

The authors [28] use a VAE model adapted to find latent variables correlated with class labels [30]. This divides the latent domain into two parts: one for training the representations that predict the labels, and the other for training the rest of the latent representations needed to generate the data. This allows counterfactuals to be generated by modifying only the relevant latent features. The generated examples are then filtered according to causal constraints (e.g., an increase in a borrower's education level must be matched by a corresponding increase in his age).

Note that besides VAE, other models of joint distribution of $P(\mathcal{X})$, can be used, for example statistical models such as copulas and Bayesian networks. However, these techniques are much less frequently used in CE tasks (see reviews of algorithms in [5, 8]). In addition, generative adversarial networks can be applied in some specific cases, such as image analysis tasks [13].

The causal model can be represented as a directed acyclic graph (DAG), which allows for a compact and visual representation of the structure of the system under study [10]. The ability of DAG to encode causal relationships is based on the criterion of d-separation, which corresponds to the conditional independence of variables in the data set. In other words, for any three non-overlapping subsets of variables (X, Y, Z), if nodes X and Y are conditionally independent given Z in the joint distribution $\mathcal{P}$, then they will be d-separated in graph $\mathcal{G}$ (Markov condition): $(X \perp\!\!\!\perp_{\mathcal{P}} Y)| Z \Rightarrow (X \perp\!\!\!\perp_{\mathcal{G}} Y)| Z$. DAG nodes correspond to variables, edges correspond to relationships between them, and the direction of edges corresponds to causal relationships.

DAG corresponds to the structural model $\mathcal{M}$:

$$\mathcal{M} = (\mathbf{S}, P_U), \;\; \mathbf{S} = \left\{X_j := f_j\left(X_{pa(j)}, U_j\right)\right\}_{j=1}^{m},$$

$$P_U = P_{U1} \times \ldots \times P_{Um}.$$

Here **S** are structural equations specifying the rules of generation of observed variables X_j as a deterministic function of their ancestors in the causal model $X_{pa(j)} \subseteq \mathrm{X} \backslash X_j$. The assumption of mutual independence of the noises U_j (full factorization of P_U) implies the absence of unobserved confounders, e.g. variables affecting cause and effect simultaneously. Note that many studies assume that noise is additive, i.e., $\mathbf{S} = \left\{X_j := f_j\left(X_{pa(j)} + U_j\right)\right\}_{j=1}^{m}$. This allows one to build efficient algorithms for model identification from data [31].

An important component of causal modeling is the apparatus of do-calculus [10]. For example, an intervention, i.e., the assignment of values $\boldsymbol{\theta}$ to a subset of variables $\mathbf{X}_K (K \subseteq |m|)$, is described using the $do(\mathbf{X}_K = \boldsymbol{\theta})$ operator. The distribution of the remaining variables $\mathbf{X}_{-k}$ can be obtained from the system $\mathbf{S}^{do(\mathbf{X}_K = \theta)}$, in which the equations for $\mathbf{X}_k$ are replaced by the corresponding values. Thus, the causal model can be used to find counterfactuals [20], for an instance x a counterfactual is defined as $x^* = \mathbf{X}(a)|x$ where $a = do(\mathbf{X}_K = \boldsymbol{\theta})$, $a \in A$, a is an action, and A is the set of admissible actions.

Causal models can be recovered from observed data or constructed from expert knowledge. However, a model $\mathcal{M}$ trained on data may be imperfect, for example, because of sample limitations or, more importantly, because of incorrect specification of the model (i.e., assuming the incorrect parametric form of the structural equations). On the other hand, although in many cases expert knowledge allows the construction of a causal model, assumptions about the form of the structural equations are usually not verifiable [32]. As a result, counterfactual explanations computed based on an ill-defined causal model may be inaccurate and recommend suboptimal or even worse, ineffective actions.

To circumvent these limitations, the authors of [20] propose two probabilistic approaches to selecting optimal actions when there is limited knowledge of causality (e.g., when only the DAG is known). The first one applies to models with additive Gaussian noise and uses Bayesian averaging to estimate the counterfactual distribution. The second excludes any assumptions about the structural equations and instead calculates the average effect of actions on objects that are similar to the factual under consideration.

1.2. Synthetic tabular data generation

Synthetic data generation (SDG) is a core element in solving several machine learning problems: data anonymization, augmentation of small datasets, class balancing in case of severe imbalance, etc. [14].

Definition 3. A synthetic generation model is a function $g \in \mathcal{G}$ that, for an observed data set $\mathcal{D} \sim \mathbb{P}$, returns a data set $\mathcal{D}^S = g(\mathcal{D}, \theta)$ of a given size, $\mathcal{D}^S \sim \mathbb{P}^S$, such that the condition $\mathbb{P}^S \approx \mathbb{P}$, $x_i \neq x_j$, $\forall x_i \in \mathcal{D} \wedge \forall x_j \in \mathcal{D}^S$ is fulfilled. Here θ is a vector of hyperparameters defining the generation policy and $\mathcal{G}$ is a generative function class family.

Mathematically, this can be represented as a Kullback–Leibler distance minimization problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_i \mathbb{P}(x_i) \log g(x_i, \theta).$$

Based on this definition, the key performance metric of a generative model is the fidelity of the synthetic data distribution $\mathbb{P}^S$ to the empirical distribution $\mathbb{P}$. Moreover, additional metrics can be introduced [33], such as diversity and generalization. The diversity requirement requires that synthetic instances should cover the entire range of variation $\mathcal{D}$. The generalization property requires that synthetic data should not be copies of real observations.

In this review, we confine our attention to synthetic table (cross-sectional) data generation (tSDG). The following classes of tSDG methods can be distinguished:

- Randomization models based on mixing, interpolation and geometric transformation of the original data and the addition of random noise.
- The probabilistic algorithms that generate data based on a multivariate distribution $\mathbb{P}^S$, modeling the real distribution $\mathbb{P}$. Several approaches can be distinguished here, as follows:
  - modeling of the joint distribution of $\mathbb{P}$, e.g., based on a Gaussian mixture or copulas [15];
  - sequential generation of $\mathcal{D}$ attributes based on conditional distributions $\mathbb{P}(x_i \,|\, \mathcal{D} \backslash \{x_1, ..., x_{i-1}\})$;
  - modeling $\mathbb{P}$ using factorization based on a graphical probability model (Bayesian network) [34].
- Models generating data from lower dimensional latent space.
- Sampling modeling based on generative adversarial networks (GAN).
- Models based on a priori known causal structure.

We remark that the conditional distributions model approach synthesizes the variables x_i sequentially using regression models $x_i = f(x_1, ..., x_{i-1})$, which can be constructed by both parametric (linear regression) and non-parametric (decision tree) methods [35, 36]. Thus, the conditional distributions $\mathbb{P}(x_i \,|\, \mathcal{D} \backslash \{x_1, ..., x_{i-1}\})$, from which the synthetic values of x_i, are derived, are defined for each variable separately and depend on the attributes $x_1, ..., x_{i-1}$, that are earlier in the synthesis sequence. The value of the very first variable in the sequence is generated based on its marginal distribution.

A comprehensive analysis of tSDG methods is presented in [14]. Several publications [16, 17] compare some of the approaches considered on real datasets. From the results presented we can conclude that there is no dominant method, and the quality of generation depends on the specific problem.

It can also be observed that conceptually synthetic data generation methods are close to CE algorithms: both are based on modeling the distribution of observed data but differ in the result. While the objective of CE is to find an instance as close as possible to the sample under study but with the opposite label (see Definition 1), the objective of tSDG is to generate a set of instances that belong to the distribution of the observed data (Definition 3). Accordingly, they are based on different performance metrics.

2. Proposed method

As can be deduced from the review presented above, the known CE algorithms have several limitations. Optimization-based methods require repeated model building for each factual; prototype-based approaches require "factual – counterfactual" pairs in the training set $\mathcal{D}$; generative model-based approaches introduce additional constraints into the algorithm, which also complicates the computation. At the same time, as noted above, synthetic data generation methods are conceptually close to CE, differing only in the result and its evaluation metrics.

Based on these considerations, we propose a two-stage method for generating counterfactual explanations (*Fig. 2*). In the first stage, a model $g(\mathcal{D}, \theta)$ of synthetic data generation is trained. According to Definition 3, this model emulates the empirical distribution $\mathbb{P}$ of real data. Using this model, a set of potential counterfactuals $\{x^*\}_g$ is generated for a given factual x.

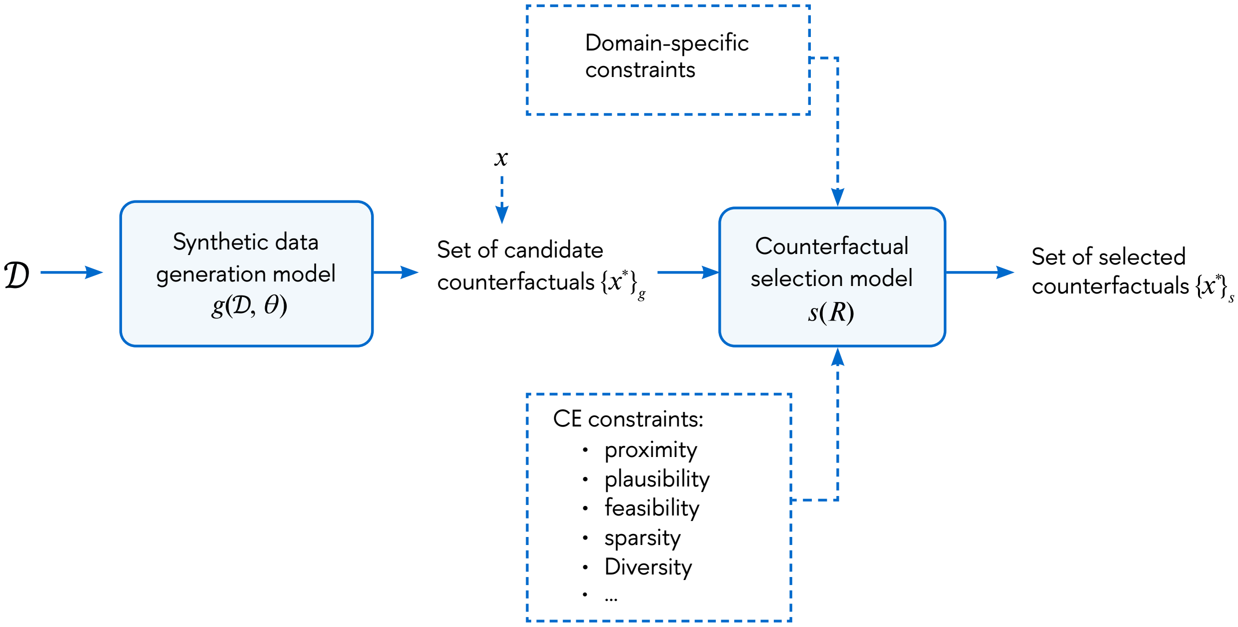
Domain-specific constraints
x
$\mathcal{D}$
Synthetic data generation model $g(\mathcal{D}, \theta)$
Set of candidate counterfactuals $\{x^*\}_g$
Counterfactual selection model $s(R)$
Set of selected counterfactuals $\{x^*\}_s$
CE constraints:
proximity
plausibility
feasibility
sparsity
Diversity
...

Fig. 2. A two-stage method for generating counterfactual explanations.

In the second step, using the selection model $s(R)$ a set $\{x^*\}_g$ is selected from $\{x^*\}_s$, the elements of which meet the constraints of R. The set $\{x^*\}_s$, is the solution to the CE problem. The selection model can include any constraints formulated as equations of the form $r(c) = m(c) \leq v(c),\ \ r \in R$. Here c is a requirement for the result (e.g., validity, proximity, sparsity for CE, or implementation cost), $m(c)$ is the corresponding metric, and $v(c)$ is the boundary of acceptable values. Note that requirements may also include constraints of a particular subject area.

The proposed approach has the following advantages:

- the generative model is built once and allows computing counterfactuals for any new observations without re-training;
- splitting the process into two steps allows the use of simple, easily modifiable selection rules;
- the selection model can include not only the requirements of CE tasks, but also any constraints specific to the subject area under consideration.

3. Experiment

To validate the proposed method, it is first necessary to verify that tSGD methods can generate counterfactuals that satisfy the requirements listed in Section 1.1, and to compare the results with existing known CE methods.

The $g(\mathcal{D}, \theta)$ generation models to be used in the experiment are summarized in *Table* 1. We have selected the simplest statistical models since our goal is to propose an efficient method for generating counterfactuals with low computational cost. These models include the Gaussian copula (GC), a sequential nonparametric model based on conditional distributions (CD), and the Bayesian network (BN), which models the distribution $\mathbb{P}$ as a multiplication of conditional distributions of factors (features). For comparison, we also include a model that generates data based on marginal distributions of features (MD). It can be regarded as a degenerate case of BN in which the relationships between features are not considered. As

Table 1.

Methods for generating synthetic tabular data

ID	Model type	Description	Source
GC	Joint distribution $\mathbb{P}$	Gaussian copula	[15][1]
CD	Conditional distributions $\mathbb{P}(x_i \mid \mathcal{D} \backslash x_i)$	Non-parametric method / decision tree	[36][2]
BN	Factorisation $\mathbb{P} = \prod p(x_i)$	Bayesian network	[34][3]
MD	Marginal distributions x_i	Sampling based on marginal distributions	[38][4]
GAN	Deep learning	Generative adversarial network	[37][5]

mentioned above, such simple models are hardly used in CE tasks, however, we suggest that their potential can be utilized much more effectively using the two-stage approach proposed here.

In addition, as the literature review suggests, most researchers using generative models to solve the CE problem focus on complex algorithms based on deep neural networks, so we also included GAN. We also investigated the possibility of applying VAE, but in our experiments these models did not achieve robust generation of $\{x^*\}_g$. This is most likely due to the insufficient amount of data for training (cf. *Table 2*).

The selection model $s(R)$ is given in the form of a rule:

$$R: h(x^*) \neq h(x) \wedge x_i^* \in \left\{ \overline{x_i} \mp 1.5 \cdot IQR(\mathcal{D}) \right\} \wedge dist(x^*; x) \rightarrow$$
$$\rightarrow \min \wedge \left| \{x^*\}_s \right| = k.$$

It means that for a particular x, k instances will be selected from the generated set $\{x^*\}_g$ whose label $h(x^*)$ is not equal to the label $h(x)$, the attribute values of x^* are within the range of three $IQR(\mathcal{D})$ interquartile intervals with respect to the mean x_i (Tukey Outlier Definitions), and the distance between x and x^* is minimal.

Table 2 presents the three datasets used in the experiments, their general characteristics and the classification of features in terms of change feasibility (feature changeable, change not feasible, feature not changeable). These public datasets are widely used in machine learning work and, in particular, CE research. The use of public datasets ensures the repeatability of the results.

Table 2 also presents the training results of the classifier h used in the CE finding process: the ROC AUC metric obtained using 10-fold cross-validation and

[1] https://sdv.dev

[2] https://www.synthpop.org.uk/

[3] https://github.com/DataResponsibly/DataSynthesizer

[4] https://github.com/vanderschaarlab/synthcity

[5] https://github.com/NextBrain-ai/nbsynthetic

Table 2.

Data sets

$\mathcal{D}$	$n \times m$	Features			Classifier		Description
		Immutable	The change is not implementable	Mutable	Model $h(x)$	AUC	
German Credit[6]	1000 × 20	3	14	3	RF	0.79 (0.03)	700 approved and 300 rejected loan applications
Adult[7]	48842 × 14	8	3	3	CB	0.93 (0.002)	Income levels based on census data
Loan Default[8]	255347 × 16	8	3	5	CB	0.76 (0.002)	29 653 rejected and 225 694 approved loan applications

the best performing model. In one case it is Random Forest (RF), in all other cases – CatBoost (CB).

One of the most popular libraries implementing CE methods is DiCE [19], which supports three counterfactual search methods. In addition to random search, they are genetic algorithm-based optimization and a method for searching and then adapting prototypes in a training sample [23]. We used these models to comparatively evaluate the results obtained. For each dataset, all three types of models were trained, and the best one was selected. It should be noted that the prototype-based approach failed to find counterfactuals for any dataset. This is obviously due to the limitation noted above: there must be a set of pairs (x, x^*) in $\mathcal{D}$ for a wide range of factuals.

To assess the results of calculating counterfactuals we will use the metrics of validity (V), proximity (P), sparsity (S), diversity (D), and plausibility (U), described above. Specifying the features whose variation is possible is carried out at the level of the generation model $g(\mathcal{D}, \theta)$.

4. Analysis of experimental results

Consider the process of applying the proposed method on the example of the German Credit dataset. This dataset contains records of 1 000 credit applications, 700 of which were approved. The attributes include the amount and term of the loan, as well as indicators of the borrower's social and financial status (credit rating, duration of employment, proportion of loan payments in the borrower's total income, etc.). Most attributes are either categorical or ordinal.

The task of CE in this case is to generate counterfactuals for borrowers who have been denied a loan. The data analysis shows that the attributes that are modifiable are *laufzeit* – loan term in months, *hoehe* – loan amount and *buerge* – presence of a co-borrower or guarantor. All other attributes are either not changeable (gender, citizenship) or cannot be changed by direct influence (credit rating).

[6] South German Credit (2019) UCI Machine Learning Repository. https://doi.org/10.24432/C5X89F

[7] Becker B., Kohavi R. (1996) Adult. UCI Machine Learning Repository. https://doi.org/10.24432/C5XW20

[8] Loan Default Dataset. https://www.kaggle.com/datasets/nikhil1e9/loan-default/data

The computation is performed according to the method presented in *Fig. 2.* At the first stage, the generation model $g(\mathcal{D}, \theta)$ is trained, which is used to generate 200 synthetic instances for the investigated factual. From this set, instances are selected according to the rule given by equation (1).

Table 3 shows an example of the data generated for a rejected application of DM 2348 for a term of 36 months. As can be seen from the data presented, the loan for this borrower can be approved if the term is reduced to 8 months and the amount to DM 1 956. If the borrower presents a co-borrower (*buerge* = 2), the amount can be increased to DM 2234 for a period of 14 months. If there is a guarantor (*buerge* = 3), the loan can be DM 4276 for a period of 26 months. As we can see, even the 3 presented counterfactuals allow us to describe the situation for a particular borrower and suggest a possible way for the borrower to achieve his goal.

Table 3.

An example of the generated data (attributes are explained in the body of the text)

	laufzeit	*hoehe*	*buerge*	Class label
Factual	36	2384	1	0
Counterfactuals	8	1956	1	1
	14	2234	2	1
	26	4276	3	1

Table 4.

Average values and standard deviations of model quality metrics for the three datasets

Model	*V*	*P*	*S*	*D*	*U*
BN	**1.000 (0.000)**	**1.230 (0.805)**	3.790 (1.251)	1.310 (0.340)	**2.053 (0.855)**
CD	**1.000 (0.000)**	1.400 (1.155)	3.303 (1.036)	1.135 (0.228)	2.128 (0.836)
DiCE	0.869 (0.273)	3.138 (1.519)	**1.962 (0.556)**	1.159 (0.206)	2.797 (0.734)
GAN	0.966 (0.186)	1.984 (1.045)	4.190 (1.640)	1.285 (0.372)	2.298 (0.742)
GC	0.967 (0.183)	1.333 (0.851)	3.887 (1.521)	1.284 (0.402)	2.089 (0.828)
MD	**1.000 (0.000)**	3.198 (2.094)	4.177 (1.506)	**1.314 (0.299)**	2.851 (0.760)

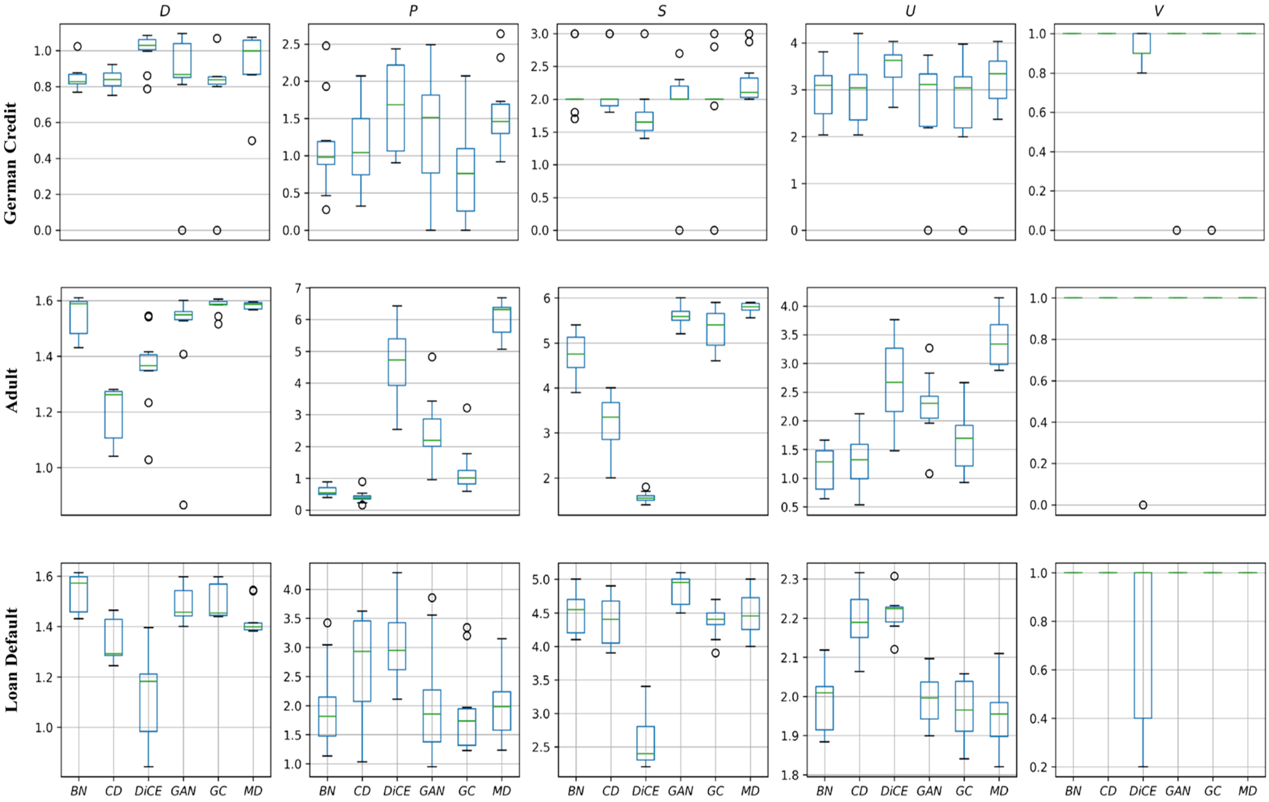

Fig. 3. Performance metrics of the models considered by datasets.

Table 4 lists the averages and standard deviations of the quality metrics for the considered methods computed over all three datasets, and *Fig. 3* presents distributions of metrics across datasets. The best values of the metrics in *Table 4* are in bold type.

It should be mentioned that for most metrics (validity, closeness, and plausibility), the best results are demonstrated by the Bayesian network (BN) model, which generates samples based on conditional distributions of features, i.e., considering the dependencies between them. Considering that this model is only slightly inferior to MD in terms of diversity, the choice of BN for CE seems quite justifiable. The high diversity of counterfactuals generated by MD is because this model considers only marginal distributions of features and does not consider the relationships between them. This model should work well in the case of uncorrelated features but may generate challenges when such correlations are present (see distribution *D* for the Loan Default dataset in *Fig. 3*). On the contrary, BN performs the best in terms of diversity among all models.

In our experimentation, the most sophisticated GAN model lost out to other models, possibly because there was not enough data to train, although the authors of the implementation we used [37] emphasize that it focuses specifically on small training samples. *Figure 3* shows that as the sample size increases, the GAN results improve but do not outperform the other models.

The methods developed directly for the CE problem (DiCE) showed the best result on the sparsity metric (*Table 4*), but *Fig. 3* shows that this was achieved by performing well on the largest dataset

(Loan Default). On smaller data, this method is inferior to simpler models, in particular GC and BN. Furthermore, it should be noted that DiCE on all datasets fails to find the required number of counterfactuals ($V = 1$) and in this sense this is the worst of the methods considered.

Conclusion

Therefore, we can conclude that the proposed method of counterfactual search based on synthetic data generation can achieve results at least comparable to the “standard” CE methods, and in some cases, it outperforms them, especially on small datasets. According to our results, the most obvious choice in this case is a generation model based on a Bayesian network that considers the interconnections between attributes.

This result reveals new possible research directions. The Bayesian network is a statistical model because it is built on associations measured by correlations. Therefore, it is of interest to study causal models that capture causal relationships in a dataset.

It should be noted, however, that to the best of our knowledge, the direction related to the use of causal models for CE is only beginning to be explored [20], and there are no works devoted to their application to the generation of synthetic data. ■

References

1. Samek W., Muller K.-R. (2019) Towards explainable artificial intelligence. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Lecture Notes in Computer Science*, vol. 11700, pp. 5–22. https://doi.org/10.1007/978-3-030-28954-6_1

2. Giuste F., Shi W., Zhu Y., Naren T., Isgut M., Sha Y., Tong L., Gupte M., Wang M.D. (2023) Explainable artificial intelligence methods in combating pandemics: A systematic review. *IEEE Reviews in Biomedical Engineering*, vol. 16, pp. 5–21. https://doi.org/10.1109/RBME.2022.3185953

3. Barocas S., Selbst A. D., Raghavan M. (2020) The hidden assumptions behind counterfactual explanations and principal reasons. Proceedings of the *2020 Conference on Fairness, Accountability, and Transparency (FAT*'20)*, pp. 80–89. https://doi.org/10.1145/3351095.3372830

4. Murdoch W.J., Singh C., Kumbier K., Abbasi-Asl R., Yu B. (2019) Definitions, methods, and applications in interpretable machine learning. *National Academy of Sciences*, vol. 116(44), pp. 22071–22080. https://doi.org/10.1073/pnas.1900654116

5. Guidotti R. (2022) Counterfactual explanations and how to find them: Literature review and benchmarking. *Data Mining and Knowledge Discovery*. https://doi.org/10.1007/s10618-022-00831-6

6. Verma S., Boonsanong V., Hoang M., Hines K. E., Dickerson J. P., Shah C. (2020) Counterfactual explanations and algorithmic recourses for machine learning: A review. *arXiv:2010.10596*. https://doi.org/10.4550/arxiv.2010.10596

7. Stepin I., Alonso J.M., Catala A., Pereira-Fariña M. (2021) A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access*, vol. 9, pp. 11974–12001. https://doi.org/10.1109/ACCESS.2021.3051315

8. Chou Y.L., Moreira C., Bruza P., Ouyang C., Jorge J. (2022) Counterfactuals and causability in explainable artificial intelligence: Theory, algorithms, and applications. *Information Fusion*, vol. 81, pp. 59–83. https://doi.org/10.1016/j.inffus.2021.11.003

9. Mishra P. (2022) *Practical explainable AI using Python: Artificial Intelligence model explanations using python-based libraries, extensions, and frameworks*. Apress.

10. Pearl J. (2009) *Causality: models, reasoning, and inference. 2nd ed.* New York: Cambridge University Press.

11. Cho S.H., Shin K.S. (2023) Feature-weighted counterfactual-based explanation for bankruptcy prediction. *Expert Systems with Applications*, vol. 216, article 119390. https://doi.org/10.1016/j.eswa.2022.119390

12. Wang D., Chen Z., Florescu I., Wen B. (2023) A sparsity algorithm for finding optimal counterfactual explanations: Application to corporate credit rating. *Research in International Business and Finance*, vol. 64, article 101869. https://doi.org/10.1016/j.ribaf.2022.101869

13. Mertes S., Huber T., Weitz K., Heimerl A., André E. (2022) Ganterfactual – counterfactual explanations for medical non-experts using generative adversarial learning. *Frontiers in Artificial Intelligence*, vol. 5, article 825565. https://doi.org/10.3389/frai.2022.825565

14. Fonseca J., Bacao F. (2023) Tabular and latent space synthetic data generation: A literature review. *Journal of Big Data*, vol. 10(1), article 115. https://doi.org/10.1186/s40537-023-00792-7

15. Patki N., Wedge R., Veeramachaneni K. (2016) The synthetic data vault. Proceedings of the *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 399–410. https://doi.org/10.1109/DSAA.2016.49

16. Dankar F., Ibrahim M., Ismail L. (2022) A multi-dimensional evaluation of synthetic data generators. *IEEE Access*, vol. 10, pp. 11147–11158. https://doi.org/10.1109/ACCESS.2022.3144765

17. Endres M., Mannarapotta Venugopal A., Tran T.S. (2022) Synthetic data generation: A comparative study. Proceedings of the *26th International Database Engineered Applications Symposium*, pp. 94–102. https://doi.org/10.1145/3548785.3548793

18. Wachter S., Mittelstadt B., Russell C. (2017) Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology (Harvard JOLT)*, vol. 31, article 841.

19. Mothilal R.K., Sharma A., Tan C. (2020) Explaining machine learning classifiers through diverse counterfactual explanations. Proceedings of the *2020 Conference on Fairness, Accountability, and Transparency (FAT* '20)*, pp. 607–617. https://doi.org/10.1145/3351095.3372850

20. Karimi A.H., Barthe G., Schölkopf B., Valera I. (2023) A survey of algorithmic recourse: Contrastive explanations and consequential recommendations. *ACM Computing Surveys*, vol. 55(5), article 95. https://doi.org/10.1145/3527848

21. Breunig M.M., Kriegel H.-P., Ng R.T., Sander J. (2000) LOF: identifying density-based local outliers. Proceedings of the *2000 ACM SIGMOD International Conference on Management of Data (ICDM)*, pp. 93–104. https://doi.org/10.1145/335191.335388

22. Poyiadzi K., Sokol K., Santos-Rodriguez R., De Bie T., Flach P. (2020) FACE: feasible and actionable counterfactual explanations. Proceedings of the *2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES 2020)*, pp. 344–350. https://doi.org/10.1145/3351095.3372850

23. van Looveren A., Klaise J. (2021) Interpretable counterfactual explanations guided by prototypes. Proceedings of the *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2021)*, pp. 650–665. https://doi.org/10.1007/978-3-030-86520-7_40

24. Aamodt A., Plaza E. (1994) Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications*, vol. 7(1), pp. 39–59.

25. Keane M.T., Smyth B. (2020) Good counterfactuals and where to find them: A case-based technique for generating counterfactuals for explainable AI (XAI). Proceedings of the *28th International Conference on Case-Based Reasoning Research and Development (ICCBR)*, pp. 163–178. https://doi.org/10.1007/978-3-030-58342-2_11

26. Joshi S., Koyejo O., Vijitbenjaronk W., Kim B., Ghosh J. (2019) Towards realistic individual recourse and actionable explanations in black-box decision making systems. *arXiv:1907.09615*. https://doi.org/10.48550/arXiv.1907.09615

27. Guyomard V., Fessant F., Bouadi T., Guyet T. (2021) Post-hoc counterfactual generation with supervised autoencoder. Proceedings of the *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2021)*, pp. 105–114. https://doi.org/10.1007/978-3-030-93736-2_10

28. Downs M., Chu J.L., Yacoby Y., Doshi-Velez F., Pan W. (2020) CRUDS: Counterfactual recourse using disentangled subspaces. Proceedings of the *2020 ICML Workshop on Human Interpretability in Machine Learning (WHI 2020)*, pp. 1–23.

29. Pawelczyk M., Broelemann K., Kasneci G. (2020) Learning model-agnostic counterfactual explanations for tabular data. Proceedings of the *Web Conference 2020 (WWW'20)*, pp. 3126–3132. https://doi.org/10.1145/3366423.3380087

30. Klys J., Snell J., Zemel R. (2018) Learning latent subspaces in variational autoencoders. *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*.

31. Hoyer P., Janzing D., Mooij J.M., Peters J., Schölkopf B. (2008) Nonlinear causal discovery with additive noise models. *Advances in Neural Information Processing Systems 21 (NIPS 2008)*.

32. Peters J., Janzing D., Schölkopf B. (2017) *Elements of causal inference: foundations and learning algorithms*. MIT press.

33. Alaa A., van Breugel B., Saveliev E.S., van der Schaar M. (2022) How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. Proceedings of the *39th International Conference on Machine Learning*, pp. 290–306.

34. Ping P., Stoyanovich J., Howe D. (2017) DataSynthesizer: Privacy-preserving synthetic datasets. Proceedings of the *29th International Conference on Scientific and Statistical Database Management (SSDBM'17)*. https://doi.org/10.1145/3085504.3091117

35. Drechsler J., Reiter J.P. (2011) An empirical evaluation of easily implemented nonparametric methods for generating synthetic datasets. *Computational Statistics & Data Analysis*, vol. 55(12), pp. 3232–3243. https://doi.org/10.1016/j.csda.2011.06.006

36. Nowok B., Raab G.M., Dibben C. (2016) Synthpop: Bespoke creation of synthetic data in R. *Journal f Statistical Software*, vol. 74, pp. 1–26. https://doi.org/10.18637/jss.v074.i11

37. Marin J. (2022) Evaluating synthetically generated data from small sample sizes: An experimental study. *arXiv:2211.10760*. https://doi.org/10.48550/arXiv.2211.10760

38. Qian Z., Cebere B.C., van der Schaar M. (2023) Synthcity: facilitating innovative use cases of synthetic data in different data modalities. *arXiv:2301.07573.* https://doi.org/10.48550/arxiv.2301.07573

About the authors

Yuri A. Zelenkov

Dr. Sci. (Tech.);

Professor, Department of Business Informatics, Graduate School of Business, HSE University, 28/11, Shabolovka Str., Moscow 119049, Russia;

E-mail: yzelenkov@hse.ru

ORCID: 0000-0002-2248-1023

Elizaveta V. Lashkevich

Doctoral Student, Department of Business Informatics, Graduate School of Business, HSE University, 28/11, Shabolovka Str., Moscow 119049, Russia;

E-mail: evlashkevich@hse.ru

ORCID: 0000-0002-3241-2291