Improving Algorithmic Decision–Making in the Presence of Untrustworthy Training Data

Wenting Qi
Department of Computer Science
University at Albany, SUNY
Albany, New York, USA
wqi@albany.edu

Charalampos Chelmis
Department of Computer Science
University at Albany, SUNY
Albany, New York, USA
cchelmis@albany.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671677

Abstract—**Although data quality is of paramount importance in algorithmic decision–making, most existing methods for supervised classification use training data without ever questioning their fidelity. At the same time, counterfactual explanation approaches widely used for post–hoc explanation of algorithmic decisions may result in unrealistic recommendations when left unconstrained. This work highlights a significant research problem, and introduces a novel framework to improve supervised classification in the presence of untrustworthy data, while offering actionable suggestions when an undesirable decision has been made (e.g., loan application rejection). Evaluation results spanning datasets from different domains demonstrate the superiority of the proposed approach, and its comparative advantage as the percentage of mislabeled instances increases.**

Index Terms—**counterfactual explanations, data quality, data science, supervised learning**

I. Introduction

Machine learning models are increasingly applied in high–stakes domains, including, but not limited to, health [1] and policing [2], lending [3] and job profiling [4]. The success of such models is intrinsically tied to the quality of data used to train them. Unfortunately, obtaining good quality data, particularly with respect to their labels in a supervised classification setting, is often impossible [5]. For instance, in homelessness service provision [6], individuals are assigned to shelters not necessarily based on their needs, but often due to availability and capacity constraints. In such cases, untrustworthy data are treated as reliable to compensate for the unavailability of high quality data. At the same time, untrustworthy data are often used to assess the accuracy of classification models (e.g., when good quality data are limited). This practice can result in potentially detrimental results [7], such as convicting an innocent person [8] or deciding to not treat a cancer patient risking irreversible health complications or even death [9].

Trustworthy automated classification and decision–making models require effective recognition and mitigation of “bad data” in the learning process. We specifically focus on supervised learning in the presence of mislabeled data instances. Our problem setting relates to that of adversarial learning [10], where counterfactual data with small feature perturbations are designed intentionally to push a machine learning model towards false predictions. The terms adversarial learning and counterfactual explanations have often been used interchangeably [11]. However, counterfactual explanations are more general object–class compared with feature–level adversarial examples [12]. In our work, we use counterfactual data for data instances associated with undesirable decisions.

At the same time, classification models must support their decisions by offering “explanations” [13]. For instance, upon declining a loan application, an algorithmic lending decision–making system should offer constructive feedback (e.g., *“Improve your credit score to more than* 750*”*) for the applicator to get a sense of what is needed for her to be awarded a loan in the future [14]. However, it may be unreasonable to suggest changing someones race or skin color so as to achieve the desirable outcome. To the best of our knowledge, existing machine learning models and counterfactual explanation frameworks treat all features equally.

The presence of untrustworthy labels in the training data hinders the ability to provide “correct” explanations of decisions made by an automated classification model. This is because mislabeled data directly influence the model training process by introducing noise in the trustworthy data distribution, thus perturbing the model decision boundary, which in turn influences classification output. Since “explanations” depend directly on prediction output and training data, if a prediction is wrong, the corresponding explanation can lead to inaccurate suggestions. Therefore, detecting potentially mislabeled data is crucial to provide correct explanations of decisions made by automated classification systems.

This work focuses on *improving algorithmic decision–making in the presence of untrustworthy training data, while offering actionable suggestions when an undesirable decision has been made* (e.g., loan application rejection).

In summary, the main contributions of this paper are:

- Highlighting an under–explored but significant research direction for the machine learning community.
- Conceptualizing and formulating the problem of learning trustworthy classification models in the presence of mislabeled training data.
- Presenting CGEP, a practical approach to (i) identify, in an *unsupervised* manner, potentially mislabeled data instances in the training set, (ii) perturb the feature vectors

This material is based upon work supported by the National Science Foundation under Grant No. ECCS–1737443.

of such instances to achieve a more *trustworthy* classification model, and (iii) offer *realistic counterfactual explanations* by considering real–world constraints.
- Demonstrating the superiority of the proposed solution against baselines using three real–world datasets of varying size and complexity. The comparative advantage of the proposed framework increases substantially as the percentage of mislabeled increases in the training data.

II. Related Work

Detecting mislabeled data in a training set has been relatively well explored. [7], [15], [16]. [7] formulated mislabeled data detection as an optimization problem, while accounting for similarity between data instances. Similarity, however, might be misleading if mislabeled data are not first identified and excluding. Ensemble–based methods [15], [17]–[19] assume that mislabeled data result to conflicting class labels when multiple independent classifiers are used for label prediction [7]. However, most such methods rely on base classifiers that are trained on data with potentially mislabeled data instances [7]. Local learning is based on the assumption that mislabeled data tend to be inconsistent with their surrounding data labels [20], [21]. [21] used nearest centroid neighborhood to detect mislabeled data, without considering possible changes incurred by decisions of other data. Our work leverages both local learning and ensemble learning to minimize the influence of mislabeled data instances. Specifically, we propose a cluster centroid–based method, and consider the label of a data instance to be inconsistent with the class of its nearest cluster centroid. Such data instance is treated as mislabeled. Cluster centroids are computed by trusted data, which are selected using majority voting.

As for the explainability of classification and automated decision–making models, multiple approaches have been recently explored [22]–[24]. Outcome explanations aim to explain a specific prediction from a given model, and are applicable to a broader class of learning models [25], [26]. One of the widely used methods generates explanations through feature importance [27]. Recently, a method to produce example–based explanations by feature perturbation was proposed [28]. Feature perturbation may lead to different prediction results given a learning model, and as such, are considered counterfactual data. [14] proposed to use multiple counterfactual explanations for a given data instance by minimizing the distance between the counterfactual data and original data and maximizing the distance between multiple counterfactual data. However, a smaller distance between counterfactual and actual data points cannot guarantee that unreasonable or offensive suggestions (e.g., changing race) will not be provided. In our work, we incorporate real–world constraints, such as immutable features (e.g., race) and directionality of the perturbation process for features such as education level, into the counterfactual data generation. The limiting factor of explainability methods is that explanations are indirectly generated from potentially mislabeled data that have been used to train the decision–making model [14]. To the best of our knowledge, this is the first work to bridge the gap between post–hoc explainability of classifciation models in the presence of untrustworthy training data.

III. Preliminaries and Problem Statement

A. Notation and Setting

Let (X, Y) denote the training dataset, and N be the total number of data instances. Each data instance $x \in X$ is associated with a d dimensional feature vector and label $y \in \{0, 1\}$. In many real–world cases (e.g., applying for a loan), one of the two labels (e.g., $y = 1$) is desirable (e.g., get the loan), and the other is undesirable. The task is to train a model C to predict the label of previously unseen data. Let $\bar{y}$ denote the predicted outcome. By comparing y with $\bar{y}$, the training data can be divided into four subsets, namely, the true positive set X_{11} (i.e., $y = 1$ and $\bar{y} = 1$), false positive set X_{01} (i.e., $y = 0$ and $\bar{y} = 1$), true negative set X_{00} (i.e., $y = 0$ and $\bar{y} = 0$), and false negative set X_{10} (i.e., $y = 1$ and $\bar{y} = 0$). For data instance $x \in X_{00}$, we wish to find its counterfactual data x_{cf} to achieve the desired outcome. Let the generated counterfactual data pair to be denoted as $(x_{cf}, y_{y=1})$. Different with standard classification, we inherently address the problem of mislabeled data instances in both the training dataset and subsequently generated counterfactual data. Therefore, original data and counterfactual data belong to one of two sets: correctly labeled set (X_r, Y_r) and mislabeled set (X_w, Y_w).

B. Problem Statement

The goal of this work is to improve algorithmic decision–making by minimizing the influence of mislabeled data instances in the training set, while at the same time providing actionable counterfactual explanations of classification results. To achieve this goal, we address three sub–problems: First, potentially mislabeled data instances must be detected and corrected. Second, actionable counterfactual data for each data instance with an undesirable prediction outcome is generated. Third, "invalid" counterfactual data are identified and replaced with "valid" counterfactual data in an iterative manner.

IV. Classification with Counterfactual Data Generation in the Presence of Mislabelled Data

We propose a framework for improved Classification with counterfactual data GEneration in the Presence of mislabelled data instances (CGEP). CGEP comprises three parts as follows. The **mislabel detection component** (Section IV-A) is designed to detect mislabeled data, and substitute the label for such data. The **classification model** (Section IV-B) is used to predict the label of previously unseen data. The **counterfactual data generation** component (Section IV-C) perturbs the feature vector of data instances with undesirable label, so as to bring them closer to the desirable outcome.

Algorithm 1 summarizes the process. Initially, (X, Y) is provided as input to the mislabel detection component. Once mislabeled data instances are identified, their labels are adjusted, and the updated dataset is used to train a classification model. The outputs of the model are inspected, and data instances for which the predicted label is undesirable, are fed into the counterfactual data generator to generate counterfactual data with desirable prediction outcome. Note that this correspond to instances $x \in X_{00}$, as opposed to $x \in X_{00} \cup X_{10}$. The rationale behind this decision is that the goal of training a classification model is to maximize prediction accuracy, which is equivalent to maximizing the size of X_{11} and X_{00}, as well as minimizing the size of X_{10} and X_{01}. In general, our counterfactual generation component tries to convert X_{00} to X_{11}, whereas the classification model pushes X_{10} to X_{11} in an attempt to improve classification accuracy. Generated counterfactual data and their corresponding label (i.e., $(x_{cf}, \bar{y}_{y=1})$) are examined to ensure that they are not considered to be noise (i.e., mislabeled) with respect to the original data distribution. In the event that the counterfactual data is detected as mislabeled, the original data (i.e.,$(x, y_{y=0})$) is retained, and another attempt is made for counterfactual data, until the generated counterfactual data is detected as correctly labeled by the mislabel detection component, or the number of maximum iterations is reached.

Algorithm 1 CGEP Algorithm

```
Input: $(X, Y)$, $\varepsilon_0$, desired class $y_{y=1}$, and $\lambda_0$.
Initialize $t = 0$.
Initialize the cluster centroids $cen_{y=0}$ and $cen_{y=1}$.
for $x \in X$ do
    if $x$ satisfies Eq. (1) then
        Add $x$ in $X_r$.
    else if $x$ satisfies Eq. (2) then
        Add $x$ in $X_w$.
    end if
end for
Flip the label for all $x \in X_w$.
while $t = T$ or $X_{00}^t$ is empty do
    Train the classifier $C^t$ with updated dataset.
    Output the prediction result of $C^t$ and aggregate true negative data in $X_{00}^t$.
    Generate counterfactual data $x_{cf}$ for $X_{00}^t$.
    Update the dataset with counterfactual data $x_{cf}$ which satisfies Equation (3).
    $t = t + 1$
end while
Output: Trained learning model $C$, $x_{cf}$ for each $x \in X_{00}^{t=0}$.
```

A. Cluster–based Mislabeled Data Detection

To detect potentially mislabeled data, we assume the true cluster centroid of desirable class is $cen_{y=1}$, and the undesirable cluster centroid class is $cen_{y=0}$. If the data x is labeled as the desirable class, but the distance between x to desirable cluster centroid $cen_{y=1}$ is larger than the distance between x to undesirable cluster centroid $cen_{y=0}$. In such case, x is mislabeled. To get a reasonable estimate of the true cluster centroids for the desired and undesired classes, data points assumed to be correctly labeled must be identified.

We adopt majority voting [29] to discriminate between the true positive and true negative data. We use Random Forest (RF) [30], Decision Tree (DT) [31] and Support Vector Machine (SVM), mainly due to their simplicity and popularity. Specifically, if the original label is consistent with the majority of the three classifiers, the corresponding data is either true positive or true negative depending on its original label. After selecting and grouping true positive data set and true negative data set, the cluster centroids $cen_{y=1}$ and $cen_{y=0}$ are computed correspondingly using standard methods [32]. The above processes visualized in the Figure 1.

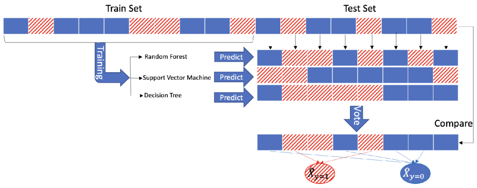


Fig. 1. Aggregation of trusted data for computing cluster centroids $cen_{y=0}$ and $cen_{y=1}$. Data instances are assigned to trusted label sets ($\hat{X}_{y=1}$ and $\hat{X}_{y=0}$) if the majority vote corresponds to their label in the original test data.

We use a 5−fold cross–validation, so that each data instance is examined for mislabeling. We group data instances with trusted labels in $\hat{X}$, and separate them into $\hat{X}_{y=1}$ and $\hat{X}_{y=0}$ according to their label value. Then, we initialize the cluster centroid $cen_{y=1}$ and $cen_{y=0}$ as follows: $cen_{y=1} = \frac{1}{|\hat{X}_{y=1}|} \sum_{i=1}^{N} x_i * \mathbb{1}(x_i \in \hat{X}_{y=1})$, $cen_{y=0} = \frac{1}{|\hat{X}_{y=0}|} \sum_{i=1}^{N} x_i * \mathbb{1}(x_i \in \hat{X}_{y=0})$, where $\mathbb{1}$ is the indicator function.

After determining the cluster centroids, potentially mislabeled data in the original dataset are identified using the following criterion: if $x \in X_r$, then x needs to satisfy Equation (1); if $x \in X_w$, then x needs to satisfy Equation (2) below.

$$(||x - cem_{y=0}|| - ||x - cen_{y=1}||)(\mathbb{1}(y = 1) - \mathbb{1}(y = 0)) > 0 \quad (1)$$

$$(||x - cen_{y=0}|| - ||x - cen_{y=1}||)(\mathbb{1}(y = 1) - \mathbb{1}(y = 0)) < 0 \quad (2)$$

Intuitively, Equation (1) checks for inconsistent data instances with respect to their nearest cluster centroid. The meaning of Equation (2) is opposite.

As mentioned in Section IV, the mislabel detection component is additionally used to check whether the generated counterfactual data are consistent with the data distribution of the original dataset. If x_{cf} satisfies Equation (3), then x_{cf} is corrected and retained, and is dropped otherwise.

$$||x_{cf} - cen_{y=0}|| - ||x_{cf} - cen_{y=1}|| > 0. \quad (3)$$

B. Classification Model

We opt for a simple feed–forward neural network (NN) [33] for classification. Compared with other types of widely used neural network structures, the feed–forward neural network is more explicable because the information in this network travels in an unidirectional manner, i.e., from the input to hidden layers to the output. We adopt the cross–entropy loss, defined as $L = -\sum_{i=1}^{N} \tilde{y}_i^T \log \mathbf{o}_i$, where $\tilde{y}_i$ is the one–hot encoding of label y_i and $\mathbf{o}_i$ is the output of the feed–forward neural network with x_i as input. The selection of cross–entropy ensures that the output (a probability vector over the two classes in our case) follows a binomial distribution.

C. Counterfactual Data Generation

We formulate the counterfactual generation process as an optimization problem that takes into account three constrains described in Subsections IV-C1, IV-C2, and IV-C3, accordingly below. Specifically, we define the loss function as:

$$dl = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{d}\frac{dist(x_{cf}^{d},x^{d})}{Z_i^d(p(k_o,k_r)-\mathbb{1}(k\in K_{od})dir(k_o,k_r))}, \quad (4)$$

where Z_i^d is a normalization factor computed as:

$$Z_i^d = \sum_{k=1}^{d}\frac{1}{(p(k_o,k_r)-\mathbb{1}(k\in K_{od})dir(k_o,k_r))}. \quad (5)$$

Then, the overall loss is defined as:

$$\arg\min_{x_{cf}}(1-\lambda)dl + \lambda(|C(x_{cf}) - y_{y=1}|), \quad (6)$$

where $p(k_o, k_r)$ and $dir(k_o, k_r)$ are defined in Eqs. (10) and (9), and hyperparamter λ controls the trade–off between the validity and actionability of the generated counterfactual data. The objective function in Eq. (6) is optimized iteratively, with hyperparameter λ being initialized as 0.5 (we use λ_0 to denote this initial value). In each counterfactual optimization iteration, the value of λ increases from λ_0 in increments of 0.01. The counterfactual data generation process terminates if the counterfactual data with a desirable outcome is generated, or the maximum number of iterations allowed (T_C) is reached. ε_0 is the minimum distance requirement for the counterfactual data prediction output to the desirable output.

1) Validity Constraint: Validity [28] is a critical constraint in the counterfactual data generation process. In our context, x_{cf} is valid if the prediction outcome is desirable (i.e., $C(x_{cf}) = 1$), and invalid otherwise. [14] defined validity constraint as a loss minimization over multiple counterfactual data for a given data instance. Different from [14], to reduce complexity, and at the same time improve the functionality of counterfactual explanations as suggestions for improvement, we generate one counterfactual data per instance. Therefore, we relax [14]'s validity constraint into:

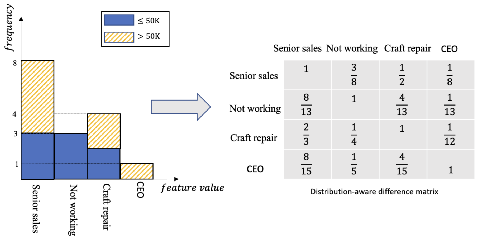
$$\arg\min_{x_{cf}} yloss(C(x_{cf}), y_{y=1}), \quad (7)$$

where $yloss(a, b) = \|a - b\|_2$.

2) Proximity Constraint: Proximity [14] evaluates the distance between counterfactual data x_{cf} and the original feature vector x for a given data instance. Intuitively, x_{cf} being close to x translates to either few changes or small changes in magnitude that must be made to achieve the desirable label. We use l_1 distance to quantify the distance between counterfactual data and original data as:

$$\arg\min_{x_{cf}} dist(x_{cf}, x) = \|x_{cf} - x\|_1. \quad (8)$$

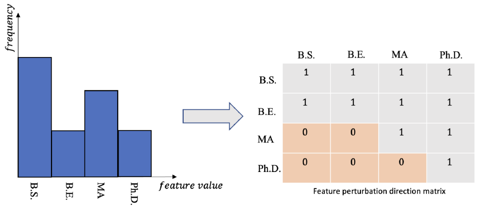
For categorical features, we assign a distance of 1 if the categorical feature value of counterfactual data is different than its original data, in line with [34].



	Senior sales	Not working	Craft repair	CEO
Senior sales	1	$\frac{3}{8}$	$\frac{1}{2}$	$\frac{1}{8}$
Not working	$\frac{8}{13}$	1	$\frac{4}{13}$	$\frac{1}{13}$
Craft repair	$\frac{2}{3}$	$\frac{1}{4}$	1	$\frac{1}{12}$
CEO	$\frac{8}{15}$	$\frac{1}{5}$	$\frac{4}{15}$	1

Distribution-aware difference matrix

Fig. 2. Example of distribution-aware difference matrix. In the matrix the y axis direction points original feature value k_o, the x-axis direction points to perturbation result k_o. The matrix based on Eq. (9).



	B.S.	B.E.	MA	Ph.D.
B.S.	1	1	1	1
B.E.	1	1	1	1
MA	0	0	1	1
Ph.D.	0	0	0	1

Feature perturbation direction matrix

Fig. 3. Illustrative example of feature perturbation direction matrix (PDM). In the matrix on the right, the y axis represents original feature value k_o, and the x axis the corresponding perturbation result k_r. The matrix is computed using Eq. (10). B.S., B.E., M.A., and Ph.D. are short for Bachelor of Science, Bachelor in Engineering, Masters, and Doctor of Philosophy degree.

3) Actionability Constraint: The proximity constraint imposes limits only to the magnitude of the changes, rather than what those changes can be. As a result, counterfactual data may make unreasonable suggestions. To avoid such fallacies, we propose additional constraints, as follows.

First, immutable features (e.g., race) are excluded from the feature perturbation process. Second, we consider the feature distribution into the counterfactual data generation process. For example, considering the task of predicting annual income with desirable class being "over $\$50,000$" and undesired label being "below $\$50,000$". Let also a feature value distribution of occupation as illustrated in Figure 2. Although a CEO achieves 100% desirable income, there is only one CEO instance. Since occupation is a categorical feature, the distance of "not working" to the rest of the feature values is the same. Unfortunately, transitioning to "CEO" may be significantly more challenging for a jobless individual. To address this challenge, we propose a distribution–aware distance matrix in the counterfactual data generating process. Specifically, we compute the distance between the original feature value, k_o, and the perturbed value, k_r, as:

$$p(k_o, k_r) = \frac{\sum_{i=1}^{N}\mathbb{1}(x_i^k = k_r)}{N - \sum_{i=1}^{N}\mathbb{1}(x_i^k = k_o)}, \quad (9)$$

where $k \in 1, 2, ..., d$ is a perturbed feature, and M denotes the total number of feature values in k. A toy example of the distribution–aware distance matrix is shown in Figure 2.

Last but not least, we impose directionality in the feature perturbation process of features that inherently allow for changes only in one direction. For instance, it is both counter intuitive for counterfactual explanations to suggest downgrading one's education level. To incorporate this intuition into the counterfactual data generation process, we group features that can only be perturbed in one direction in set K_{od}, and use a feature perturbation direction matrix (PDM) for those features:

$$dir(k_o, k_r) = \begin{cases} 1, & \text{if } k_o \text{ to } k_r \text{ is permissible,} \\ 0, & \text{if } k_o \text{ to } k_r \text{ is not permissible.} \end{cases} \tag{10}$$

Figure 3 shows a toy PDM for a hypothetical scenario involving a feature based on education level.

V. Experiments

We conduct experiments on three real–world datasets to explore the effectiveness of the proposed algorithm. All codes are implemented in Python 3.8 with Pytorch 1.9.0, and all experiments were conducted on a commodity laptop running macOS Big Sur with 3.8 GHz 8–core intel Core i7 processor. To ensure the *reproducibility* of our work, we will make our source code available on Github upon acceptance.

A. Datasets

We use the following datasets, which are widely used to evaluate counterfactual explanations for machine learning [34].

Adult–Income: This publicly dataset [35], comprises individual–level data, including educational, employment, and personal situation from 1994 Census database. The task is to determine whether a person earns over $\$50,000$ a year [14].

Bank–Marketing: This publicly dataset [35] is related to phone call based marketing campaigns of a Portuguese banking institution. A client is contacted more than once in order to access her willingness to subscribe to a product (i.e., bank term deposit) [36]. Therefore, the task for this dataset is to determine whether the client will subscribe a term deposit.

German–Credit: This small dataset [37] includes information about clients who took a loan from a bank. The classification task in this context is to determine whether the client has good credit confidence based on their past history.

Summary statistics for these datasets are provided in Table I. To encode categorical features, we adopt the numerical mapping method [38], which has been shown to resolve some of the issues found by simple one–hot encoding. As numerical mapping is known to induce numerical bias, however, we assign a distance of 1 if the counterfactual categorical feature value is different from the original in the loss calculation process [14] to avoid influencing the proximity loss.

B. Baselines

We compare CGEP with the following baselines:

- **SingleCF**: Proposed in [28], this SOTA method is designed to optimize validity and proximity constraints.
- **CGEP–NN–NoCF**: The NN classifier used in CGEP without counterfactual data generation. This baseline is used to quantify the ineffectiveness of a NN approach without counterfactual data, as compared against CGEP.
- **CGEP–RF–NoCF**: The Random Forest classifier [30] used in CGEP without counterfactual generation. This baseline is used to demonstrate the benefit of feed–forward neural network for classification as opposed to a simpler model, such as random forest. Nevertheless, our framework can be used with any classification model.

C. Experiment Setup

We split each dataset into a training and testing set with a ratio of 3:1. We use cross–validation on the training set to select the hyperparameters of the feed–forward neural network and choose parameters mentioned in Algorithm 1. The structure of the feed–forward neural network used in SingleCF, CGEP–NN–NoCF, and CGEP are same. The chosen structure has three hidden layers, and the number of neurons for each layer is 64, 32, 2. For SingleCF and CGEP, we use the Adamw optimizer [39], which is implemented in PyTorch [40], with a learning rate of 0.01, to minimize the loss function. We set $\varepsilon_0 = 0.1$ and $T_C = 30$.

D. Evaluation Metrics

We use accuracy, true positive rate, proximity, and average number of perturbed features to evaluate the proposed solution. Specfcially, we evaluate classification **accuracy** while accounting for counterfactual data, as $\frac{\sum_{i=1}^{N}((y_i=\bar{y}_i=1)+(y_i=\bar{y}_i=0))}{N}$. We use **true positive rate** (TPR), computed as $TPR = \frac{\sum_{i=1}^{N} y_i=\bar{y}_i=1}{N}$, to evaluate both the classification model and counterfactual generator. The higher the true positive rate, the better the performance. When counterfactual data are used, data instances belonging to TNR (i.e., true negative rate which is defined as: $TNR = Accuracy - TPR$) according to the original training dataset must be included in the computation of TPR. In line with [14], we calculate the proximity of continuous features separately and scale continuous features to [0,1]. Finally, we evaluate the number of changed features for the counterfactual data compared with the original data, by calculating the **average number of perturbed features**. Generally, the counterfactual data with fewer changed features are more actionable.

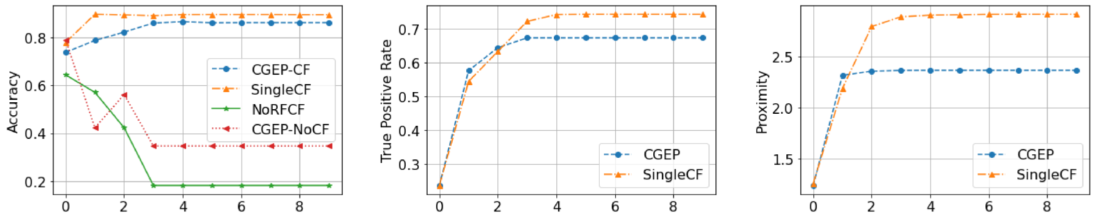
E. Evaluation Results

Figure 4(a) shows that the accuracy results on Adult–Income of CGEP with counterfactual data is much higher than the accuracy of an equivalent model that does not consider counterfactual data. This illustrates the benefit of counterfactual data generation in the proposed solution model. By further comparing CGEP–RF–NoCF and CGEP–NN–NoCF, it becomes evident that the feed–forward neural network performs better than random forest in all three datasets. However, in the presence of mislabeled data, the accuracy of SingleCF drops quickly, as illustrated in Figure 5. The comparative advantage of CGEP becomes clearer as the number of mislabeled data instances in the training data increases for all three datasets. In fact, even when more than half of the data being untrustworthy), classification accuracy can be maintained at a level much higher than that achieved by SingleCF.

Figure 4(b) shows how the proposed solution and the baselines fair with respect to true positive rate on Adult–Income. True positive rate alone is not enough to justify the superiority of CGEP. However, when the proximity metric is also considered (see Figure 4(c)), the advantage of CGEP becomes more evident. Similarly, Figure 6 shows that CGEP

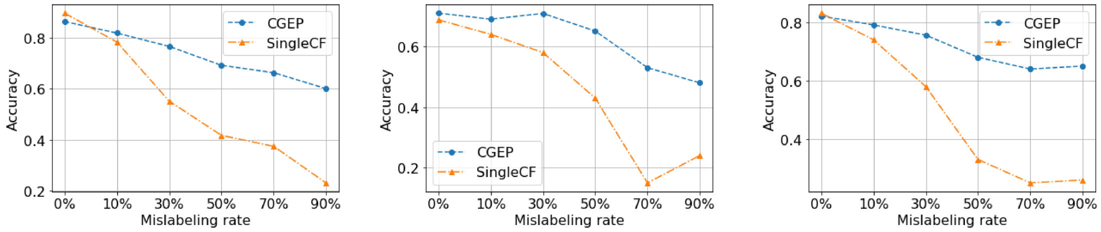
TABLE I
DATASETS STATISTICAL INFORMATION.

Dataset	# of data instances	# of mislabeled data instances	# of features	# of categorical features	# of continues features	Imutable features	One–directional feature
Adult-Income	48,842	127	14	6	8	gender, age, race, country	education
Bank–Marketing	45,211	236	20	10	10	age	education
German–Credit	1,000	20	15	5	5	age, credit history	credit score



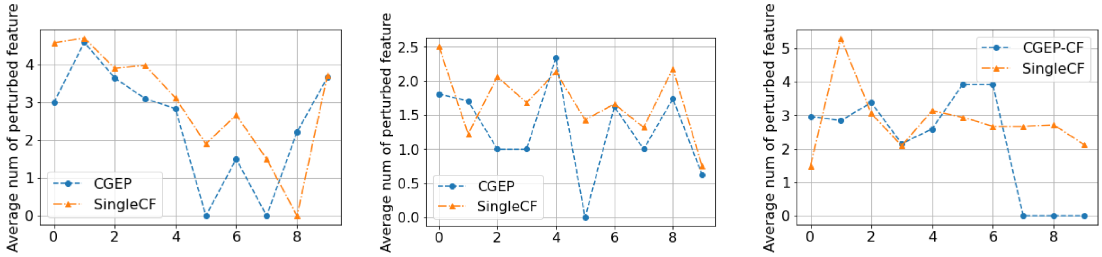
(a) (b) (c)

Fig. 4. Per epoch ($x-$axis) (a) Accuracy ($y-$axis), (b) True positive rate ($y-$axis), and (c) Proximity ($y-$axis), accordingly, achieved by CGEP on the Adult–Income dataset.



(a) (b) (c)

Fig. 5. Accuracy ($y-$axis) as a function of increasing percent of mislabeled data ($x-$axis) for CGEP and the baselines using the (a) Adult–Income, (b) German–Credit, and (c) Bank–Marketing datasets, accordingly.



(a) (b) (c)

Fig. 6. Average number of changed features ($y-$axis) per epoch ($x-$axis) for CGEP and the baselines over (a) Adult–Income, (b) German–Credit, and (c) Bank–Marketing datasets, accordingly.

is better than SingleCF, in the sense that fewer features have to be perturbed to achieve comparable or superior accuracy, even in when the number of mislabeled data is small (i.e., when the original dataset is used without synthetic mislabeled data).

VI. Conclusion

As automated classification and algorithmic decision–making become part of everyday life, the trustworthiness of data used to train such models becomes critical. This work studied the problem of improving supervised learning in the presence of mislabeled data instances in the training data. A new framework was proposed to identify, in an unsupervised manner, mislabeled data instances, perturb the feature vectors of such instances to train a trustworthy classification model, and incorporate real–world constraints so as to offer reasonable and realistic suggestions for improvement, when an undesirable decision has been made by the classification model. Experimental evaluation spanning three datasets demonstrated the benefit of identifying and correcting mislabaled data, while leveraging counterfactual explanations to improve classification. We hope that by highlighting this under–explored problem, more effective methods will be developed to address the challenges associated with quality issues in the datasets used to train machine learning models.

References

[1] M. M. Ahamad, S. Aktar, M. Rashed-Al-Mahfuz, S. Uddin, P. Liò, H. Xu, M. A. Summers, J. M. Quinn, and M. A. Moni, “A machine

learning model to identify early stage symptoms of sars-cov-2 infected patients," *Expert systems with applications*, vol. 160, p. 113661, 2020.

[2] N. Kreif and K. DiazOrdaz, "Machine learning in policy evaluation: new tools for causal inference," *arXiv preprint arXiv:1903.00402*, 2019.

[3] A. K. Tiwari, "Machine learning application in loan default prediction," *Machine Learning*, vol. 4, no. 5, 2018.

[4] I. Paparrizos, B. B. Cambazoglu, and A. Gionis, "Machine learned job recommendation," in *Proceedings of the fifth ACM Conference on Recommender Systems*, 2011, pp. 325–328.

[5] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[6] C. Chelmis, W. Qi, and W. Lee, "Challenges and opportunities in using data science for homelessness service provision," in *Companion Proceedings of the Web Conference 2021*, 2021, pp. 128–135.

[7] H. Valizadegan and P.-N. Tan, "Kernel based detection of mislabeled training examples," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 309–319.

[8] D. Wilkins and K. Pillaipakkamnatt, "The effectiveness of machine learning techniques for predicting time to case disposition," in *Proceedings of the 6th international conference on Artificial intelligence and law*, 1997, pp. 106–113.

[9] K. Huang, H.-G. Stratigopoulos, and S. Mir, "Fault diagnosis of analog circuits based on machine learning," in *2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010)*. IEEE, 2010, pp. 1761–1766.

[10] D. Lowd and C. Meek, "Adversarial learning," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 2005, pp. 641–647.

[11] S. Sharma, J. Henderson, and J. Ghosh, "Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," *arXiv preprint arXiv:1905.07857*, 2019.

[12] T. Freiesleben, "Counterfactual explanations & adversarial examples–common grounds, essential differences, and potential transfers," *arXiv preprint arXiv:2009.05487*, 2020.

[13] N. Burkart and M. F. Huber, "A survey on the explainability of supervised machine learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, 2021.

[14] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 607–617.

[15] C. E. Brodley and M. A. Friedl, "Identifying mislabeled training data," *Journal of artificial intelligence research*, vol. 11, pp. 131–167, 1999.

[16] X. Kang, P. Duan, X. Xiang, S. Li, and J. A. Benediktsson, "Detection and correction of mislabeled training samples for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 10, pp. 5673–5686, 2018.

[17] C. E. Brodley and M. A. Friedl, "Improving automated land cover mapping by identifying and eliminating mislabeled observations from training data," in *IGARSS'96. 1996 International Geoscience and Remote Sensing Symposium*, vol. 2. IEEE, 1996, pp. 1379–1381.

[18] X. Zhu, X. Wu, and Q. Chen, "Eliminating class noise in large datasets," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 920–927.

[19] Y. Jiang and Z.-H. Zhou, "Editing training data for knn classifiers with neural network ensemble," in *International symposium on neural networks*. Springer, 2004, pp. 356–361.

[20] A. M. R. A. J. Badenas, J. Sanchez, and R. Barandela, "Decontamination of training data for supevised pattern recognition," *Advances in Pattern Recognition Lecture Notes in Computer Science*, vol. 1876, pp. 621–630, 2000.

[21] J. S. Sánchez, R. Barandela, A. I. Marqués, R. Alejo, and J. Badenas, "Analysis of new techniques to obtain quality training sets," *Pattern Recognition Letters*, vol. 24, no. 7, pp. 1015–1022, 2003.

[22] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1885–1894.

[23] A. Henelius, K. Puolamäki, H. Boström, L. Asker, and P. Papapetrou, "A peek into the black box: exploring classifiers by randomization," *Data mining and knowledge discovery*, vol. 28, no. 5, pp. 1503–1529, 2014.

[24] S. Krishnan and E. Wu, "Palm: Machine learning explanations for iterative debugging," in *Proceedings of the 2nd Workshop on Human-In-the-Loop Data Analytics*, 2017, pp. 1–6.

[25] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[26] ——, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st international conference on neural information processing systems*, 2017, pp. 4768–4777.

[28] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the gdpr," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[29] S. Verbaeten and A. Van Assche, "Ensemble methods for noise elimination in classification problems," in *International workshop on multiple classifier systems*. Springer, 2003, pp. 317–325.

[30] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble machine learning*. Springer, 2012, pp. 157–175.

[31] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[32] M. G. Omran, A. P. Engelbrecht, and A. Salman, "An overview of clustering methods," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 583–605, 2007.

[33] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," *IEEE Potentials*, vol. 13, no. 4, pp. 27–31, 1994.

[34] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," *arXiv preprint arXiv:2010.10596*, 2020.

[35] R. Kohavi and B. Becker, 1996. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/adult

[36] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.

[37] D. H. Hofmann, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

[38] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," *arXiv preprint arXiv:1907.02584*, 2019.

[39] S. Gugger and J. Howard, "Adamw and super-convergence is now the fastest way to train neural nets," *last accessed*, vol. 19, 2018.

[40] N. Ketkar, "Introduction to pytorch," in *Deep learning with python*. Springer, 2017, pp. 195–208.