Supplemental Material: Introducing User Feedback-based Counterfactual Explanations (UFCE)

Muhammad Suffian[1], Jose M. Alonso-Moral[2], Alessandro Bogliolo[1]

[1] Department of Pure and Applied Sciences, University of Urbino Carlo Bo, Urbino, Italy
m.suffian@campus.uniurb.it, alessandro.bogliolo@uniurb.it
[2] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela,
Santiago de Compostela, Spain
josemaria.alonso.moral@usc.es

Abstract. This supplemental material expands upon the experiments conducted in the manuscript "Introducing User Feedback-based Counterfactual Explanations (UFCE)" which is under consideration for publication in the International Journal of Computational Intelligence Systems by Springer Nature. Following the recommendations of Editor and Reviewer 2, the experiments presented in this document have been expanded to provide further insights and validation of the findings reported in the main manuscript. These additional experiments are conducted for the MLP model to validate and enhance the robustness of the proposed method, potentially contributing to the acceptance of the manuscript for publication.

1 Expanded Experiment for MLP Model

In this document, the experiments are expanded for the black-box model to validate the performance and robustness of UFCE. In this regard, the same three research questions (RQs) in the main manuscript are answered with the Multi-layer Perceptron (MLP) model. MLP black-box model is chosen due to its complexity and opaque nature. MLP model captures intricate data patterns, rendering them effective across diverse tasks. However, comprehending its internal mechanisms proves challenging due to the absence of transparent interpretative rules. Utilizing UFCE with the MLP black-box model will help to bridge this gap between complexity and human interpretability, thereby providing the model-agnostic validation for UFCE.
Let us recall the RQs:

- **(RQ1)** Does user feedback (user constraints) affect the quality and computations of CEs?
- **(RQ2)** How do randomly taken user constraints affect the generation of CEs?
- **(RQ3)** What is the behaviour of UFCE on multiple datasets?

We have performed three experiments, each to answer one research question. Experimental settings are the same as the main manuscript, however, in this experiment, UFCE is compared only with DiCE, and AR is skipped because it only works on logistic regression. We present the experiments in the following subsequent sections.

1.1 (RQ1) Effects of User-constraints on the Performance and Computation of Counterfactual Generations

This experiment entails the details of how the different levels of user constraints (user feedback) can affect the performance of the generation of counterfactual explanations (CEs). The different levels of user constraints are configured, and these constraints help to perform the perturbations that guide the sub-processes of UFCE to generate CEs. A specific percentage (absolute value) of median absolute deviation (MAD) from the actual data distribution is computed as a user-specified perturbation limit for each numeric feature. These configurations are divided into five levels and termed as *very limited, limited, medium, flexible, and more flexible.* These levels are assumed to simulate scenarios when different users can specify different choices. The different levels of choices simulate the behaviour of a user in the real scenario as follows:

- Very limited - This value is a 20% of the MAD of the relevant data.
- Limited - This value is a 40% of the MAD of the relevant data.
- Medium - This value is a 60% of the MAD of the relevant data.
- Flexible - This value is a 80% of the MAD of the relevant data.
- More flexible - This value is a 100% of the MAD of the relevant data.

The Bank Loan dataset is considered for this experiment. For example, the MAD of the feature 'Income' is 50.10. Accordingly, in this case, 'very limited' corresponds to 10.02, 'limited' is 20.04, 'medium' is 30.06, 'flexible' is 40.08, and 'more flexible' is 50.10.
This process is repeated for all the features taking part in perturbations. For categorical features, the feature values are reversed in all five levels of constraints.

Table 1: (RQ1) The performance comparison in terms of generation of feasible counterfactuals (%) for very limited (VL), limited (L), medium (M), flexible (F), and more flexible (MF) constraints. Plaus refers to number of plausible CEs, Act to number of actionable CEs, and Feas to number of feasible CEs.

-	UFCE-1			UFCE-2			UFCE-3			DiCE			DiCE-UF		
Levels	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas
VL	4	6	4 (8%)	18	23	18 (36%)	36	38	36 (72%)	23	4	4 (8%)	13	21	13 (26%)
L	11	14	11 (22%)	27	31	27 (54%)	41	42	41 (82%)	22	6	6 (12%)	18	20	18 (36%)
M	17	21	17 (34%)	39	39	39 (78%)	45	45	45 (90%)	21	10	10 (20%)	29	25	25 (50%)
F	22	29	22 (44%)	41	44	41 (82%)	47	47	47 (94%)	31	13	13 (26%)	33	29	29 (58%)
MF	27	31	27 (54%)	45	47	45 (90%)	49	50	49 (98%)	29	17	17 (34%)	41	35	35 (70%)
Average	-	-	32.4%	-	-	68%	-	-	87.2%	-	-	20%	-	-	48%

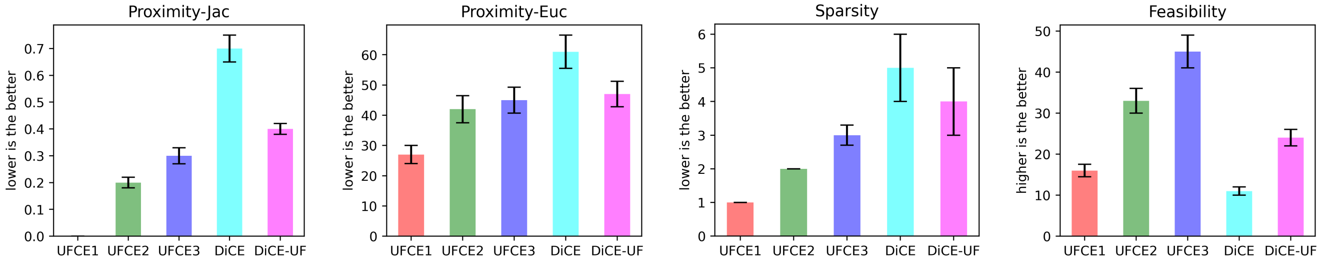

Fig. 1: (RQ1) Performance of CE methods for different evaluation metrics (with error bar of st.dev).

We run the experiment on a pool of 50 test instances, for each test instance, the counterfactuals are generated for all levels of user constraints with UFCE and DiCE (our approach includes its 3 variations). The DiCE was configured in two ways: (i) DiCE-UF takes as input the same user feedback as UFCE, and (ii) the basic DiCE does not take as input any specific user feedback, but after counterfactuals are generated we verify if they adhere or not to the desired user feedback ranges. All the features are assumed as the user-specified list of features to change for all methods. For each test instance, both counterfactual explainers were configured to give a chance to generate its 5 best counterfactuals. Then, costs of proximity were calculated for each CE, and the one nearest to the test instance was chosen (given that it is feasible) to consider for further evaluations. A CE is feasible if it is actionable and plausible. To fulfil this requirement, we had considered a CE as an actionable when it used at least 30% of the features from the user-specified list to its total changes (suggested feature changes) and it is not an outlier. Table 1 presents the consolidated results of feasible counterfactuals (%ge) by each method for all five levels of user feedback and Fig. 1 plots the average results for all evaluation metrics. In general, UFCE surpassed DiCE and DiCE-UF in all configurations of user constraints for a feasible counterfactual generation, for the MLP model too. The reasons behind the better performance of UFCE, in general, are the targeted perturbations to look for valid counterfactuals, plausible to the reference population and actionable to certain user-defined limits. Further, in Fig. 1, the average results for different evaluation metrics are plotted. For each plot, the CE methods are placed on the x-axis and the metric scores on the y-axis. The lower value is the better case for $Proximity - Jac$, $Proximity - Euc$, and $Sparsity$, while the higher value is the better case for $Feasibility$. Proximity-Jac represents the percentage of categorical features utilised. UFCE1 and UFCE2 did not consider any categorical features for generating CEs, and DiCE is the method utilising maximum categorical features for CEs. Proximity-Euc represents to Euclidean distance of generated CE from the test instance, DiCE turned out to be the most expensive method to suggest changes, whereas UFCE variations performed better than the other methods. Sparsity represents the number of features changed in the generated CE. DiCE has shown a higher sparsity value, therefore, it incurred multiple feature changes, leading to higher Proximity-Euc, while UFCE performed better, in general. Similarly, UFCE performed better in generating feasible counterfactuals than the other methods. This experiment has shown that the impact of user feedback on the generation of counterfactuals is influencing. It is evident that as the user constraints are flexible (at least equal to the MAD), the results are better for each method incorporating user feedback, in their capacity.

1.2 (RQ2) How do the randomly taken user-preferences affect the generation of CEs?

The second experiment is similar to the experiment previously described in Section 1.1, the only change is in the user feedback. In this experiment, we worked with randomly taken user preferences rather than any pre-suppositions. This is not an actual *Monte Carlo Simulation* but rather a random sampling of the upper bound for each feature to use in the generation of counterfactuals. The Bank loan dataset is utilised for this experiment. For a pool of 50 test instances, the randomly generated user feedback is utilised (in a real scenario, a user has to provide as an affordable recourse), and this process is repeated 10 times.

The results for the feasibility metric are presented in Table 2. In Table 2, UFCE3 surpassed the other methods regarding the average reported results for randomly taken user constraints.

Table 2: (RQ2) The performance comparison in terms of generation of feasible CE (in $\%ge$) for Monte Carlo-like random generation of user feedback. Plaus refers to number of plausible CEs, Act to number of actionable CEs, and Feas to number of feasible CEs.

-	UFCE-1			UFCE-2			UFCE-3			DiCE			DiCE-UF		
No.	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas	Plaus	Act	Feas
1	9	14	9 (18%)	33	22	22 (44%)	33	41	33 (66%)	14	4	4 (8%)	25	13	13 (26%)
2	11	18	11 (22%)	21	17	17 (34%)	39	44	39 (78%)	17	7	7 (14%)	19	17	17 (34%)
3	10	13	10 (20%)	22	14	14 (28%)	29	33	29 (58%)	21	9	9 (18%)	23	15	15 (30%)
4	9	19	9 (18%)	29	19	19 (38%)	34	39	34 (68%)	22	6	6 (12%)	19	11	11 (22%)
5	12	15	12 (24%)	31	21	21 (42%)	41	41	41 (82%)	34	10	10 (20%)	28	18	18 (36%)
6	11	21	11 (22%)	20	16	16 (32%)	38	41	38 (76%)	29	9	9 (18%)	17	17	17 (34%)
7	13	17	13 (26%)	29	23	23 (46%)	40	41	40 (80%)	19	9	9 (18%)	16	14	14 (28%)
8	9	15	9 (18%)	20	26	20 (40%)	39	43	39 (78%)	23	8	8 (16%)	18	12	12 (24%)
9	13	18	13 (26%)	18	23	18 (36%)	31	35	31 (62%	16	7	7 (14%)	24	11	11 (22%)
10	12	14	12 (24%)	22	24	22 (44%)	41	41	41 (82%)	25	9	9 (18%)	15	13	13 (26%)
Avg.	-	-	22%	-	-	38.4%	-	-	73%)	-	-	15.6%	-	-	28.2%

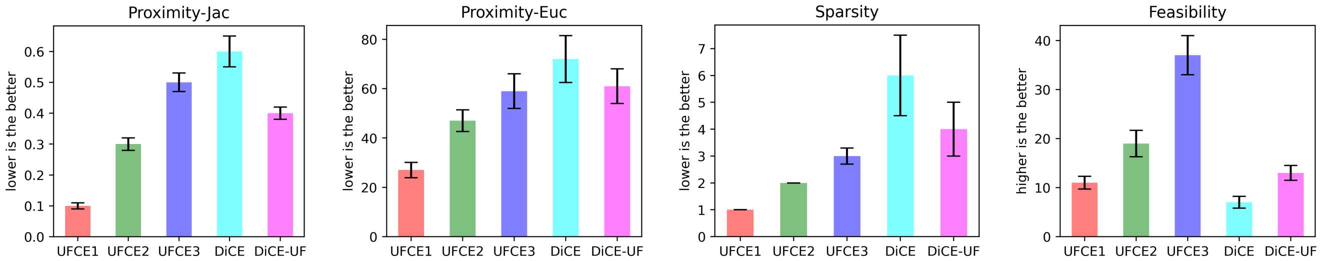

Fig. 2: (RQ2) Random user-preferences for CE generation: The bar plots depict the evaluation results for different evaluation metrics (with error bar of st.dev).

Figure 2 illustrates the average results for all evaluation metrics. We can observe that UFCE performed better for proximity, sparsity, and feasibility than DiCE. We have considered the average metric scores for all the generated counterfactuals. For example, in the case of sparsity, if a method generates only 22% (like UFCE1) feasible counterfactuals, then, we have computed the sparsity of the generated counterfactuals only, and taken the mean of those generated counterfactuals. In the case of UFCE1, the sparsity value is 1, which means UFCE1 has changed only 1 feature of the total features, whereas DiCE has changed around 6% of features on average. In sparsity, UFCE1 is better than DiCE as it suggests a smaller percentage of feature changes to get CE.

Furthermore, when DiCE was subjected to generate CEs on the same subset of features as for UFCE, it was only able to generate from 15.6% CEs; that being the case, DiCE was allowed to exploit the rest of the features in their search for counterfactuals. It is worth noting that DiCE-UF was able to generate around 28.2% feasible counterfactuals. To do so, DiCE-UF takes into account the given (randomly generated) user feedback similarly as UFCE does but without considering mutual information. In consequence, DiCE-UF achieved a smaller percentage of feasible counterfactuals than UFCE. In general, UFCE was taking into account the user feedback in the generation mechanism which restricted it to not making extreme changes for finding CE, hence, UFCE (and its variations) have shown better results than DiCE.

1.3 (RQ3) What is the behaviour of UFCE on multiple datasets?

We have conducted a third experiment to compare UFCE with DiCE on five datasets. This experiment follows the same setup as the one described in Section 1.1. Nevertheless, the user constraints are now fixed to a threshold equal to 50% of the MAD of features in the actual data distribution for each test fold. Each dataset was split into 5-test folds, and the mean results of CE generation for all folds are reported.
In the main manuscript, the details about the different datasets, their features, and the ML model's 5-fold mean accuracy with cross-validation have already been presented. The comparative results (mean of different folds of test set) on five datasets are presented for proximity-Jac, proximity-Euc, sparsity, actionability, plausibility, and feasibility in Table 3. Fig. 3 provides the readers with complementary bar plots to facilitate the interpretation of numbers reported in Table 3.

Table 3: (RQ3) Comparative results on multiple datasets for different evaluation metrics. The evaluation metrics are provided with up-arrow ↑ to show that higher is better and down-arrow ↓ for lower is better. The *na* denotes not applicable (in datasets where categorical features are not present).

dataset	CE Method	prox-Jac ↓	prox-Euc ↓	sparsity ↓	actionability ↑	plausibility ↑	feasibility ↑
Graduate	DiCE	0.7	23.10	4.00	8.00	6.00	6.00
	DiCE-UF	0.4	15.2	3.00	12.00	**19.00**	12.00
	UFCE1	**0.0**	**5.00**	**1.00**	12.00	11.00	11.00
	UFCE2	0.2	7.90	2.00	17.00	14.00	14.00
	UFCE3	0.5	14.10	3.00	**23.00**	18.00	**18.00**
BankLoan	DiCE	0.70	70.60	5.00	23.00	19.00	19.00
	DiCE-UF	0.60	37.50	4.00	33.00	27.00	27.00
	UFCE1	0.60	**21.45**	**1.00**	15.00	15.00	15.00
	UFCE2	**0.00**	39.70	2.00	37.00	32.00	32.00
	UFCE3	0.30	44.50	3.00	**46.00**	**46.00**	**46.00**
Movie	DiCE	0.50	76.50	8.00	4.00	7.00	4.00
	DiCE-UF	0.30	53.60	5.00	6.00	9.00	6.00
	UFCE1	0.00	**12.10**	**1.00**	7.00	5.00	5.00
	UFCE2	**0.00**	30.30	2.00	17.00	9.00	9.00
	UFCE3	0.20	41.90	3.00	**20.00**	**14.00**	**14.00**
Wine	DiCE	na	41.90	6.00	10.00	17.00	10.00
	DiCE-UF	na	35.20	4.00	17.00	25.00	17.00
	UFCE1	na	20.90	**1.00**	21.00	15.00	15.00
	UFCE2	na	**14.90**	2.00	39.00	32.00	32.00
	UFCE3	na	30.70	3.00	**40.00**	**35.00**	**35.00**
Bupa	DiCE	na	41.30	4.00	4.00	7.00	4.00
	DiCE-UF	na	31.40	3.00	2.00	5.00	2.00
	UFCE1	na	**10.80**	**1.00**	13.00	11.00	11.00
	UFCE2	na	16.85	2.00	**24.00**	**17.00**	**17.00**
	UFCE3	na	24.90	3.00	22.00	16.00	16.00

We can observe that UFCE performed better in most of the evaluation metrics on multiple datasets. The better result from any of the methods on any dataset for each specific evaluation metric is highlighted in bold in Table 3. Regarding proximity-Jac, there are three datasets which have categorical features, UFCE1 utilised 0.6(60%) of the categorical features for the Bank loan dataset. The UFCE variations utilised fewer categorical features across the datasets. This is positive in the sense that the user does not have to change the category of the features which in some cases is not viable like changing the *gender* feature in some real-world dataset. Regarding the proximity-Euc, UFCE1 performed better than others on the Graduate, Bank Loan, Movie, and Bupa datasets, while UFCE2 performed better than others on the Wine dataset. Regarding sparsity, UFCE1 performed better than all other methods on all datasets by suggesting only one feature change. Regarding actionability, UFCE3 performed better than others on the Graduate, Bank Loan, Movie, and Wine datasets, while UFCE2 performed better than others on the Bupa dataset. Regarding plausibility, UFCE3 performed better than others on the Bank Loan, Movie, and Wine datasets; DiCE-UF performed better than others on the Graduate dataset while UFCE2 performed better than others on the Bupa dataset. Regarding feasibility, UFCE3 performed better than others in the Graduate, Bank Loan, Movie, and Wine datasets, while UFCE2 performed better than others in the Bupa dataset.
UFCE consistently exhibited better results across all the five datasets under study. These positive outcomes suggested robustness and effectiveness in various scenarios with the MLP model too. Finally, we can conclude that UFCE has shown better performance than DiCE across the experiments.

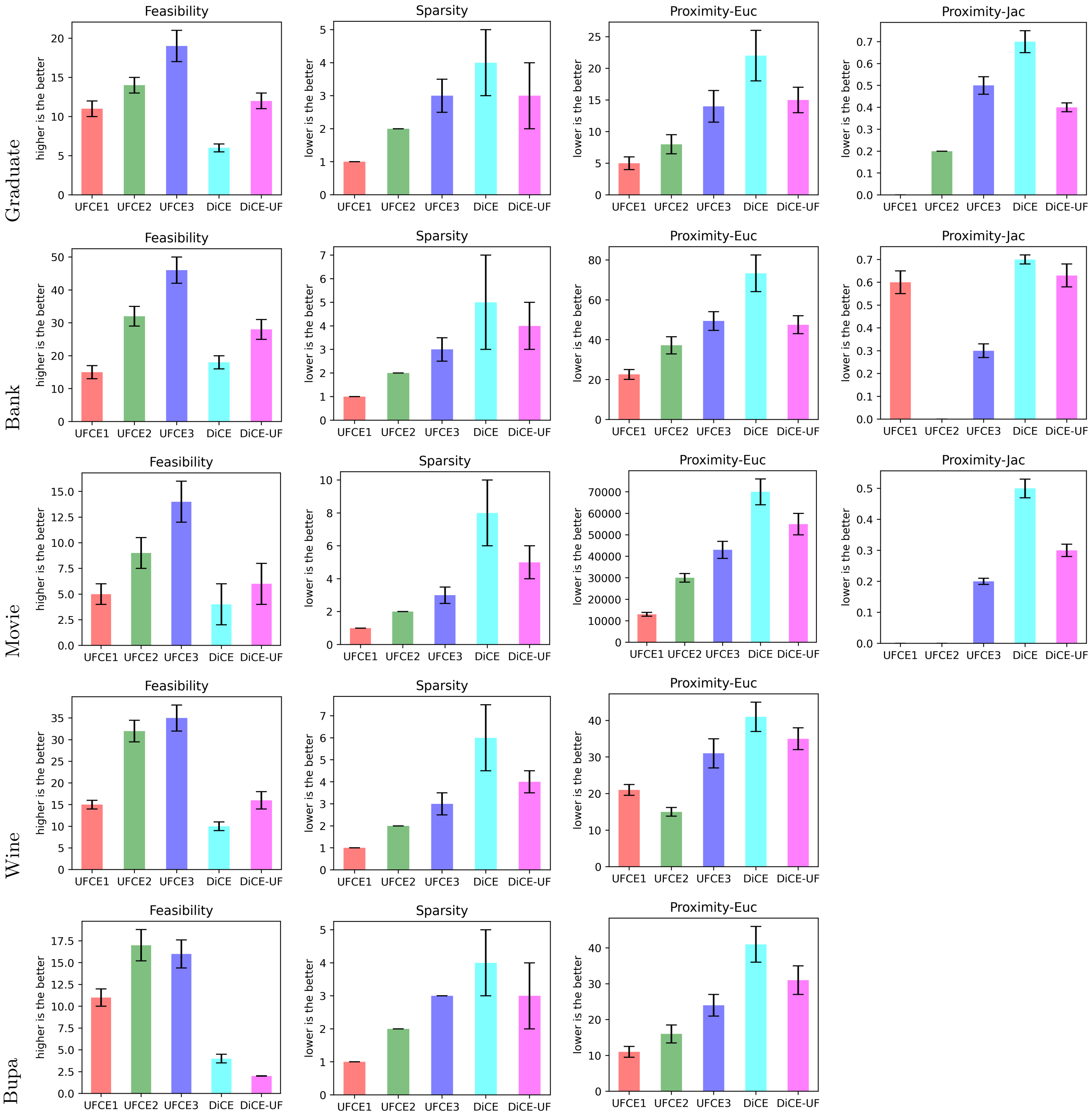

Fig. 3: (RQ3) Comparative results for CE generation on multiple datasets: The bar plots depict the evaluation results for different evaluation metrics (with error bar of st.dev).