APPENDIX

The Appendix supplies additional materials regarding the evaluation of *TSEvo* and the implementation. Appendix A show the datasetwise results for the mutation ablation study and appendix B describes the implementation and code basis.

A. Mutation Types

This section contains the remaining results. Table VII, VI, and VIII show the results for sparsity ($\mathbf{R_2}$), proximity ($\mathbf{R_1}$), and plausibility ($\mathbf{R_3}$) on dataset level. Figure 5 visualizes the counterfactuals achieved with the different mutation types on the first test image.

Table VII shows the sparsity per dataset and mutation type. Except for GunPoint, Authentic Opposing Information produces the counterfactuals with the lowest number of changed time steps. The generally much larger sparsity for frequency mutation results from the exchange of frequency bands, leading to a changed frequency and, therefore, more changed time steps. *TSEvo* achieves the best proximity scores on authentic opposing information for most datasets, followed by combination mutation. The best plausibility values for each dataset are distributed between the three basic mutation types Opposing, Frequency, and Gaussian. Note that on $\mathbf{R_3}$ frequency mutation performs exceptionally well on sensor-related datasets, indicating that authentic opposing information might not provide a good solution for all types of datasets. Combination Mutation is ranked second for most datasets and metrics, indicating that different mutations might be necessary for different dataset types to achieve plausible results.

	Opposing	**Frequency**	**Gaussian**	**Combination**
CBF	$\mathbf{0.22 \pm 0.13}$	0.28 ± 0.14	0.59 ± 0.17	0.29 ± 0.14
CharacterTrajectories	$\mathbf{0.21 \pm 0.07}$	1.0 ± 0.01	0.66 ± 0.14	0.26 ± 0.13
Coffee	$\mathbf{0.03 \pm 0.02}$	0.04 ± 0.01	0.05 ± 0.02	$\mathbf{0.03 \pm 0.02}$
ECG5000	0.34 ± 0.12	0.39 ± 0.16	0.45 ± 0.14	$\mathbf{0.33 \pm 0.11}$
ElectricDevices	0.12 ± 0.09	$\mathbf{0.2 \pm 0.17}$	0.58 ± 0.16	$\mathbf{0.2 \pm 0.14}$
FordA	$\mathbf{0.12 \pm 0.11}$	0.15 ± 0.13	0.24 ± 0.27	0.14 ± 0.12
GunPoint	0.25 ± 0.3	0.22 ± 0.28	$\mathbf{0.16 \pm 0.12}$	0.25 ± 0.3
Heartbeat	$\mathbf{0.0 \pm 0.01}$	0.01 ± 0.03	0.02 ± 0.03	0.01 ± 0.03
NATOPS	$\mathbf{0.15 \pm 0.09}$	0.17 ± 0.07	0.19 ± 0.1	0.17 ± 0.1
UWaveGestureLibrary	$\mathbf{0.35 \pm 0.15}$	0.43 ± 0.19	0.83 ± 0.12	0.57 ± 0.24

TABLE VI: Proximity ($\mathbf{R_1}$) for each dataset and mutation type. The lower, the smaller the changes made to the original instance.

	Opposing	**Frequency**	**Gaussian**	**Combination**
CBF	$\mathbf{0.26 \pm 0.15}$	0.88 ± 0.14	0.62 ± 0.16	0.36 ± 0.15
CharacterTrajectories	$\mathbf{0.21 \pm 0.07}$	1.0 ± 0.01	0.66 ± 0.14	0.26 ± 0.13
Coffee	$\mathbf{0.12 \pm 0.08}$	0.78 ± 0.21	0.37 ± 0.1	0.15 ± 0.1
ECG5000	$\mathbf{0.38 \pm 0.13}$	0.98 ± 0.05	0.56 ± 0.14	0.45 ± 0.19
ElectricDevices	$\mathbf{0.32 \pm 0.18}$	0.8 ± 0.21	0.64 ± 0.17	0.4 ± 0.24
FordA	$\mathbf{0.21 \pm 0.19}$	0.82 ± 0.25	0.22 ± 0.24	0.31 ± 0.29
Heartbeat	$\mathbf{0.03 \pm 0.03}$	0.88 ± 0.18	0.06 ± 0.08	0.11 ± 0.21
GunPoint	0.35 ± 0.38	0.66 ± 0.39	$\mathbf{0.33 \pm 0.2}$	0.37 ± 0.38
NATOPS	$\mathbf{0.25 \pm 0.15}$	0.96 ± 0.04	0.32 ± 0.15	0.57 ± 0.33
UWaveGestureLibrary	$\mathbf{0.29 \pm 0.12}$	0.96 ± 0.1	0.72 ± 0.1	0.85 ± 0.24

TABLE VII: Sparsity ($\mathbf{R_2}$) for each dataset and mutation type. The lower, the smaller the number of changes made to the original instance.

	Opposing	**Frequency**	**Gaussian**	**Combination**
CBF	0.9688	**0.9719**	0.9688	0.9703
CharacterTrajectories	**0.9813**	0.9791	0.9791	0.9791
Coffee	**0.9902**	0.9881	0.9881	0.9874
ECG5000	0.9714	**0.9743**	0.9729	0.9729
ElectricDevices	0.9625	**0.9708**	0.9646	0.9667
FordA	0.9924	0.9932	**0.996**	0.9924
GunPoint	0.9813	0.984	0.976	0.9813
Heartbeat	0.9906	**0.9916**	0.9911	0.9906
NATOPS	**0.9333**	0.9294	0.9294	0.9294
UWaveGestureLibrary	0.9873	0.9873	**0.9905**	0.9873

TABLE VIII: Plausibility ($\mathbf{R_3}$) of the generated counterfactuals for each dataset and mutation type. The higher, the better - i.e., more data support from the training data.

B. Implementation

The implementation of *TSEvo* is based on DEAP[5], a highly flexible python framework for evolutionary computation. The classification models for the evaluation of the counterfactual methods are written in PyTorch[6]. Although not used in this paper, *TSEvo* also supports tensorflow. For usage with tensorflow please refer to our github repository.

For Ates et al. [1], we used their python implementation[7] and only adapted the prediction function to enable the usage with pytorch. We reimplemented the approach of Wachter et al. [32] in PyTorch, as most library implementations use tensorflow as Basis (e.g. Alibi[8]) or only cope with univariate classification (e.g. CARLA [23]). NUN-CF and NUN-CF GradCam were adapted from the authors github implementation[9] to enable the usage in a multiclass classification on PyTorch.

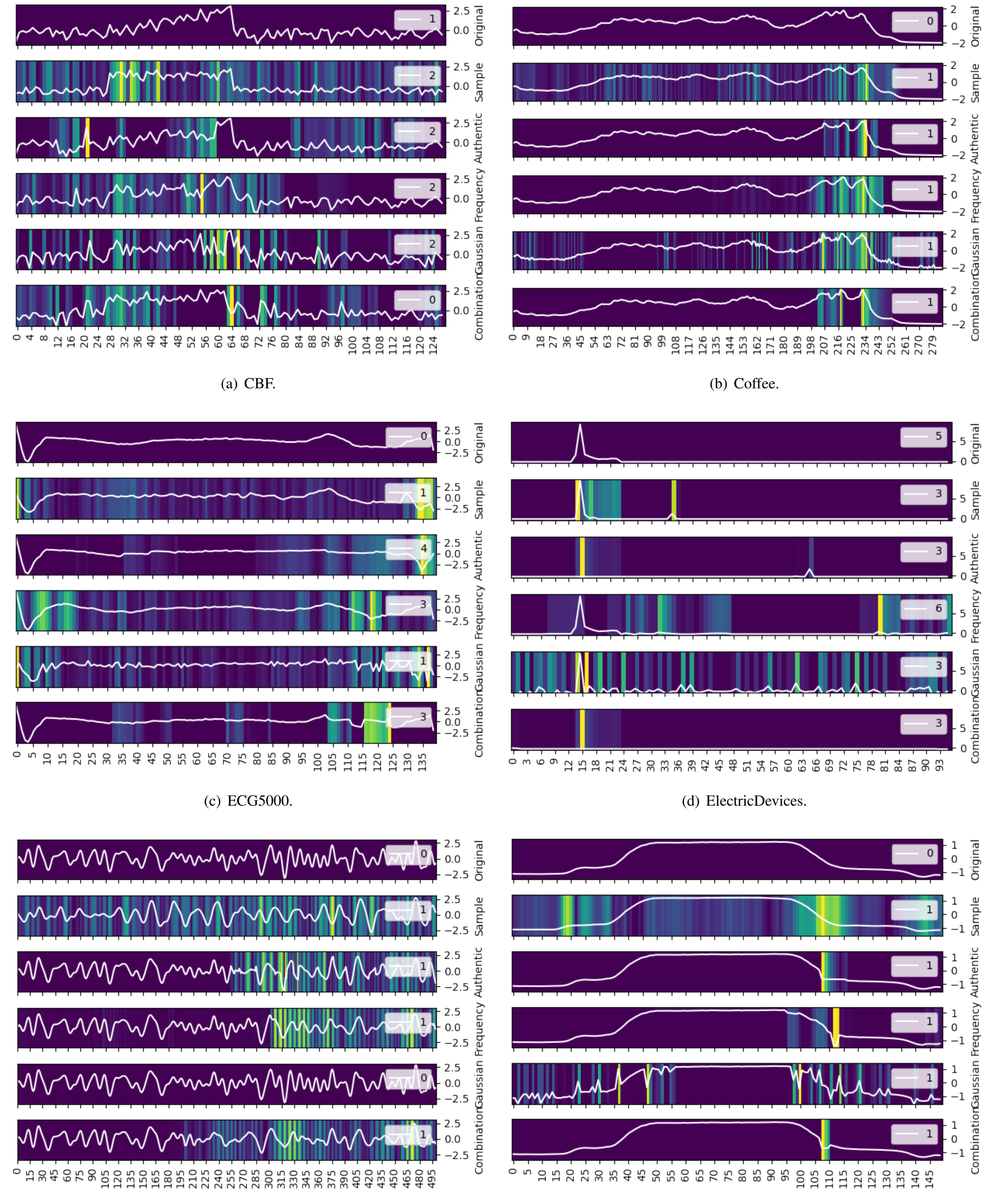
[5]https://deap.readthedocs.io/en/master/, last access July 8, 2022

[6]https://pytorch.org/, last access July 8, 2022

[7]https://github.com/peaclab/CoMTE, last access July 8, 2022

[8]https://docs.seldon.io/projects/alibi/en/latest/, last access July 8, 2022

[9]https://github.com/e-delaney/Instance-Based_CFE_TSC (commit: 19f87a1), last access July 8, 2022



(a) CBF.

(b) Coffee.

(c) ECG5000.

(d) ElectricDevices.

(e) FordA.

(f) GunPoint.

Fig. 5: Counterfactual for the first timeseries of the test set from CBR, Coffe, ECG5000, Electric Devices, GunPoint and FordA obtained with the different mutation types. If the labels are consistent with the original classification, the method failed to generate a true counterfactual.