XAI for Time Series Classification: Evaluating the Benefits of Model Inspection for End-Users

Brigt Håvardstun[1⋆][0009−0007−1034−1422], Cèsar Ferri[2][0000−0002−8975−1120], Kristian Flikka[0009−0004−2343−9566][3], and Jan Arne Telle[1][0000−0002−9429−5377]

[1] Department of Informatics, University of Bergen, Norway
brigt.havardstun@uib.no,Jan.Arne.Telle@uib.no
[2] VRAIN, UPV-Polytechnic University of Valencia, Spain
cferri@dsic.upv.es
[3] Eviny AS, Norway
Kristian.Flikka@eviny.no

Abstract. We present an XAI tool for time series classification providing model-agnostic instance-based post-hoc explanations, by means of prototypes and counterfactuals. Additionally, our tool allows for model inspection on instances generated by the user, to navigate the boundary between classes. This will allow the user to test and improve their hypotheses when formulating a model of the black box classification. We perform a human-grounded evaluation with forward simulation, to contribute a quantitative end-user evaluation to the field of XAI for time series.

Keywords: Time series · Human-grounded evaluation · Instance-based explanations

1 Introduction

Temporal data is encountered in many real-world applications ranging from patient data in healthcare [15] to the field of cyber security [20]. Deep learning methods have been successful in time series classification [11,15,20], but such methods are not easily interpretable, and often viewed as black boxes, which limits their applications when user trust in the decision process is crucial. To enable the analysis of these black-box models we revert to post-hoc interpretability. Recent research has focused on adapting existing methods to time series, both specific methods like SHAP-LIME [8], Saliency Methods [10] and Counterfactuals [4], and also combinations of these [16].

However, compared to images and text, time series data are not intuitively understandable to humans [18]. This makes interpretability of time series extra demanding, both when it comes to understanding how users will react to the provided explanations and also to the evaluation of its usefulness. Nevertheless, as humans learn and reason by forming mental representations of concepts based

[⋆] Corresponding author.

on examples, and any machine learning model has been trained on data, then data e.g. in the form of prototypes and counterfactuals is indeed the natural common language between the user and this model. In addition, several studies have highlighted the need to rethink new ways of interaction with an XAI algorithm, to allow for a dialogue between explainer and explainee, and to enable model inspection at will [1,13,17].

Hence, we advance an XAI time series tool that on the one hand provides users with instances of both prototype and counterfactual time series, and on the other hand lets the user generate their own instances for classification. This enables active learning and allows the user to test and improve their hypotheses when formulating a mental model of the black box classification.

We perform a quantitative user evaluation, thereby meeting a demand from the XAI research community [4,21,22], to measure how prototypes, counterfactuals and interactivity increases the understanding that the user has of the black box classification. Using the taxonomy of interpretability evaluation [5] what we do is a Human-Grounded Evaluation with Forward Simulation.

We use 3 datasets having a binary classification (e.g. 24-hour power demand of a household in Winter versus Summer) and for each dataset we train an ML model. Note that it is generally more difficult to explain the classification of an AI model than the classification of a real-world dataset. The reason for this is that the dataset contains only real instances, whereas the AI model classifies *any* instance, also artifical ones. In this work, for prototypes we use real instances from the dataset, whereas we produce counterfactuals by combining real instances with artifical ones, based on the NativeGuide method [4]. For tests we have chosen to use real instances.

In the remainder of this paper, we first discuss related work, then we give definitions, followed by a description of the tool and a discussion of the user evaluation results.

2 Related work

Research on XAI for time series classification has progressed similarly to XAI in general, with earlier work focused on feature-importance rather than on instance-based methods using prototypes and counterfactuals. A comprehensive survey of XAI methods for time series can be found in [21]. Our own work deals with model-agnostic instance-based post-hoc explanations, by means of prototypes and counterfactuals. In [14], the author offers a survey of work on instance-based explanations for XAI mainly in the image domain, defining and classifying the various approaches in the litterature. In this paper we evaluate the use of instance-based explanations for time series by end-users. The work [7] studied how non-experts handled post-hoc example-based explanations, however not in the time series domain. They found that even though these do assist users with correct judgement, people have significant difficulties dealing with misclassifications in an unfamiliar domain. Thus we should maybe not expect very high accuracy from our own evaluations on end-users, as time series are notoriously hard for humans to interpret [18].

Interactivity and model inspection have recently been seen as important in XAI. The paper [2] argues that interactivity in XAI is a core value in the interface between the model and the user, and that a user study is needed for a qualitative evaluation. In [22], the authors develop an interactive XAI tool for loan applications that allows users to experiment with hypothetical input values and inspect their effect on model outcomes, and perform a user evaluation on MTurk.

User evaluations of XAI systems come in various forms. A taxonomy of interpretability evaluation, from the gold standard of Application-oriented evaluations, to Human-grounded evaluations as we perform in this work, to Functionally-grounded evaluations that do not require human experimentations is developed in [6]. For time series it seems the latter approach is the more common. the authors in [16] apply several XAI methods previously used on image and text domain to time series, and introduces verification techniques specific to times series, in particular a perturbation analysis and a sequence evaluation, but they do not include any user evaluation of their systems. Likewise, [9] presents a Python package to provide a unified interface to interpretation of time series classification, but no user evaluation.

The work of [4] provides a method for generating counterfactuals for time series classifiers, called Native Guide, that applies Class Activation Mappings [23] to select discriminative areas for modification. They end their paper by arguing that 'Given the ubiquitous nature of time series data and the frequent requirement for explanation, it is clear that experiments with human users and CBR solutions have much to offer in future work.' The survey [21] says about Native Guide that '...evaluating this promising approach involving end users could be promising for future work'. Indeed this is a part of what we do in the current paper as the counterfactuals we show users are exactly the ones generated by Native Guide.

3 Definitions

Let us present formal definitions for Time Series Classification (TSC) and recall basic notions.

Staying consistent with earlier notation [21,4] a time series $T = \{t_1, t_2, \ldots, t_m\}$ is an ordered set of m real-valued observations (or time steps). A time series dataset $D = \{T_1, T_2, ..., T_n\} \in R^{n \times m}$ is a collection of such time series where each time series has a class label c forming a vector of class labels. In this paper we consider only binary classification tasks. Given such a dataset, Time Series Classification is the task of training a mapping b from the space of possible inputs to a probability distribution over the class values. Thus, a black-box classifier $b(T)$ takes a time series T as input and predicts a probability output over the class values. Given a to-be-explained time series T, with predicted label $b(T) = c$ from the black-box classifier, a counterfactual explanation aims to find how T

needs to (minimally) change to some T' for the system to classify it alternatively, as $b(T') = c' \neq c$. We refer to T' as a counterfactual explanation for T, without having to specify the target class since we consider only binary classification tasks. The minimality criterion usually refers to a notion of distance (proximity) between time series, but another criteria property can be sparsity (that T and T' differ on few data points, or on few contiguous sequences of data points) and plausibility (that the instance is not an outlier).

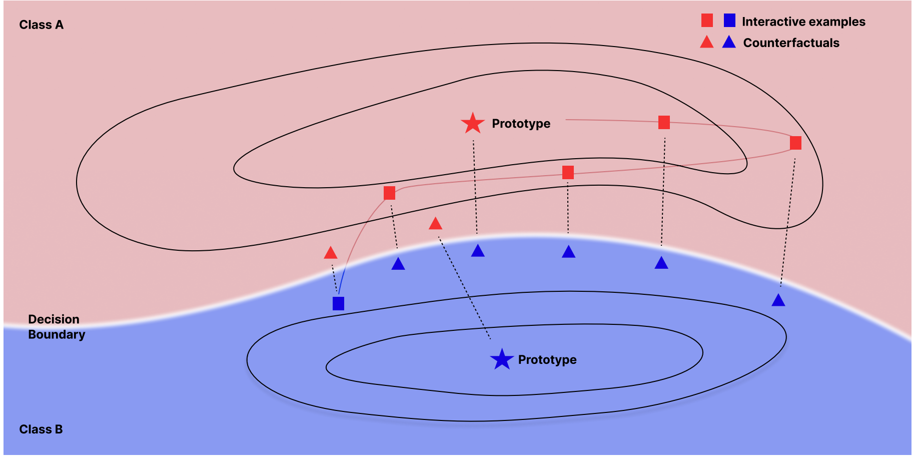


Fig. 1: Explanation types illustrated in a two-class decision space. The two prototypes are representatives of each class, positioned in the middle of the data density contour map. A path of interactive examples and their counterfactuals is shown to illustrate a possible user's journey.

Prototypes are time series exemplifying the main aspects responsible for a classifier's specific decision outcome. It can be a real instance (which is what we opt for in our tool) sampled from the dataset that is important and meaningful because it summarizes the shape of many other similar instances, or a synthetic one, for example a cluster centroid or an instance generated by following some ad-hoc processes. See Figure 1.

4 Prototypes, counterfactuals, and user-model inspection

In this section we introduce our tool for XAI on time series classification providing model-agnostic instance-based post-hoc explanation, by means of prototypes and counterfactuals. Additionally, one of our contributions is to allow for model inspection on instances generated by the explainee. This interactive tool allows the user to themselves navigate the boundary between classes. Starting from a prototype, while simultaneously seeing a counterfactual (see Figure 1) the user can change individual time points at will, and see the resulting classification. This will allow the user to test and improve their hypotheses when formulating a model of the black box classification.

In the rest of this section we describe our tool and start by giving an argument for its motivation, coming from the industry side. We then give a description of the algorithms for generating prototypes plus counterfactuals, and a discussion of how to present these to the user on-screen. We end by reporting on the design choices related to the interactive part, that allows the user to change individual time points and do model inspection.

4.1 The need for trust

When presenting output from black box ML systems to non-trained users, the need for explanations arises from at least two angles: The user must trust the results, and the user must, to a sufficient degree understand the results. To use a real-world example from energy companies [19], if a system based on charging data tells a car owner that their car seems faulty, or tells a home owner that their power consumption deviates from expectations - they would need to trust their correctness, otherwise the message will be ignored. When trust is established, the next step is usually to fix the situation. If the car owner understands the reason for their car charging being classified as faulty, he/she can bring it to the vendor or repair shop with concrete information that can be used to fix it. Similarly, if the home owner understands why their consumption is classified as deviating - they could perhaps fix a broken appliance - or adjust their consumption to a more favourable price pattern. These are situations encountered at Eviny, a Norwegian energy company collaborating on the present tool. The more complicated the data and classifications are (for example consisting of several series of data measuring power, temperature, etc. - and/or having temporal patterns like slowly decreasing trends), the more challenging will the explanation be. Thus - exploring different approaches and tools for explanations for non-trained users is of great use when companies plan on applying machine learning on complex data. Without trust the models will be ignored, and without an understanding of what the actual problem is no appropriate action can be taken. Of course, one can ask how much benefit non-trained users can gather from example-based explanations in the realm of time series, and this is indeed one of the questions guiding the user evaluations done in this work.

4.2 Generating and presenting prototypes and counterfactuals

We have opted to use cluster centers as our prototypes. To ensure the prototypes we selected have high plausibility, we use a $k-$medoids algorithm to find the prototypes, see [12]. Specifically we used KMedoids from sklearn_extra.cluster package. We used default configuration of KMedoids, with Euclidean distance as the distance metric.

There is a growing consensus that counterfactuals provide robust and informative explanations to a query time series whose classification is to be explained.

In the time series domain the visualization of counterfactuals is straightforward. We have opted to use a novel method for generating counterfactuals for time series called Native Guide, developed recently by Delaney et al [4]. This method extracts counterfactual time series, named Native Guides, starting from initial training data. Starting from the query time series whose classification is to be explained, the Native Guide method starts by finding a counterfactual time series belonging to the dataset that is close to the query. This Native series are then adapted in a Guided way to generate novel counterfactuals, following four identified key properties for good counterfactuals: proximity, sparsity, plausibility, and diversity. The Native Guide counterfactual generation method uses Class Activation Mappings to Guide the counterfactual generation from the Native series. The use of CAM in itself puts some limitations on the AI model used, e.g. having the last layer be global average pooling, so in that sense is not completely model-agnostic. To ensure compatibility between the model doing the classification and the Native Guide counterfactual generator, we therefore closely follow the time series classification model implementation of Delaney et al available here.

When presenting time series to the user we have opted to use two colours for the two classes, namely Blue and Pink. Since a counterfactual, say Blue, will be used to explain the classification of a given query Pink time series, we have chosen to present both at the same time to the user. Thus, we plot the query with Pink lines, and then show only the deviation of the counterfactual by Blue dotted lines. See Figures 2-5 which show also that the user is allowed to make interactive changes, as described in the next subsection. Note that the y-values in the figures represents the normalized values for each dataset.

4.3 Allowing interaction and model inspection

A central aspect of our tool is that it allows the user to alter individual data points and do model inspection. In real-time the tool will update the model classification by changing the color of the time series if the classification changes, showing the model confidence in this classification. It will also update to a new counterfactual. This enables active learning and allows the user to test and improve their hypotheses when formulating a mental model of the black box classification. In Figures 2-5, we see 4 screen shots of an actual session with the tool: Figure 2) User starts with a Pink prototype and a counterfactual. Figure 3) Makes changes to left end of series so that confidence (top bar) of model classification drops, plus new counterfactual, but same color/class. Figure 4) Makes further changes and now the model classification switches. Figure 5) Last changes made by user and confidence of model classification increases. Note how the user can explore their understanding of the classification by progressive changes to the current time series. Compare also with Figure 1.

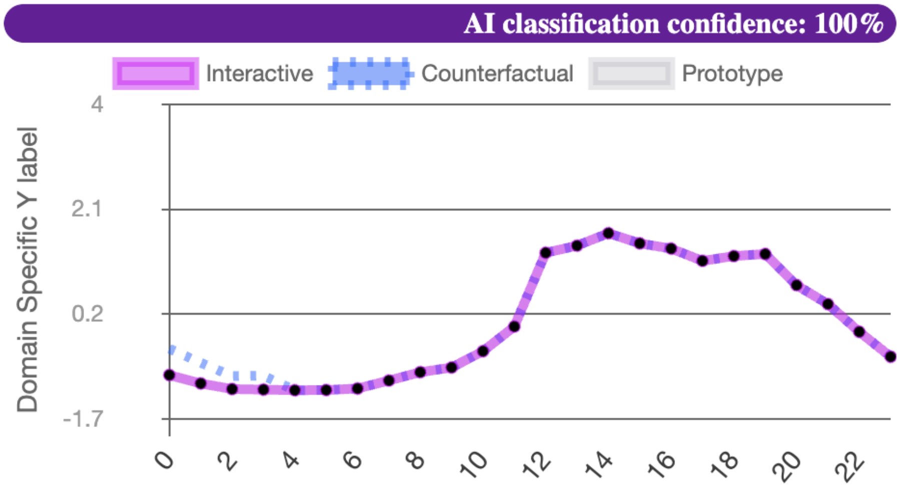


Fig. 2: Start in Pink prototype.

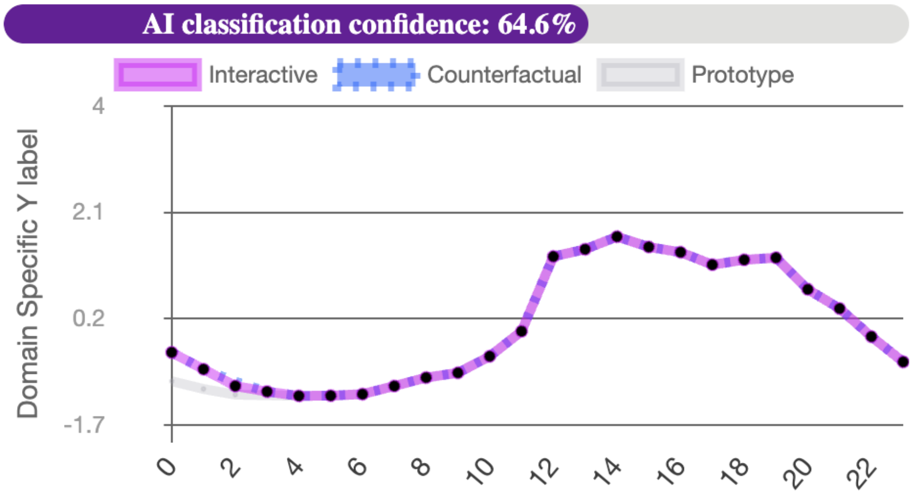


Fig. 3: Changes at left end, still Pink.

5 End-user evaluation with forward simulation

We have not before seen such an interactive tool for time series in the research literature. As time series are notoriously difficult for human users, most of the human evaluations of XAI methods done so far have been qualitative and involving domain experts. Our goal in this work has been to add the new dimension

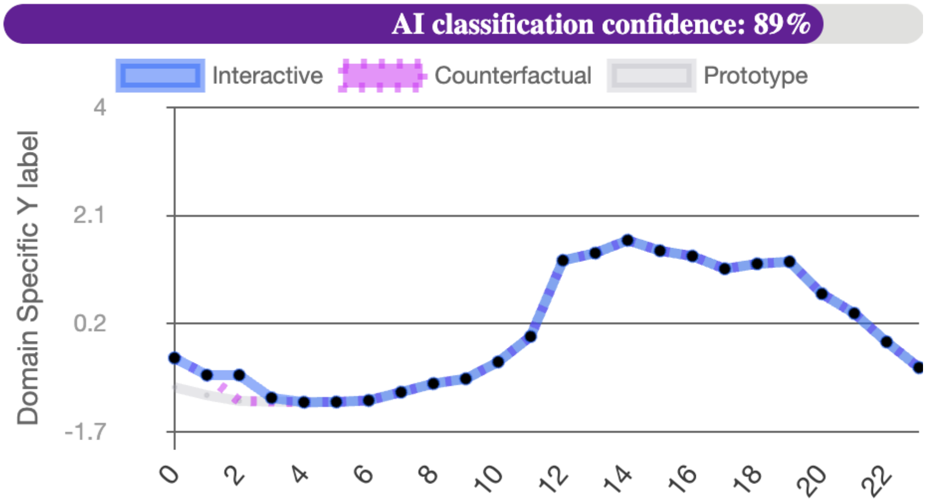


Fig. 4: Further changes, now Blue.

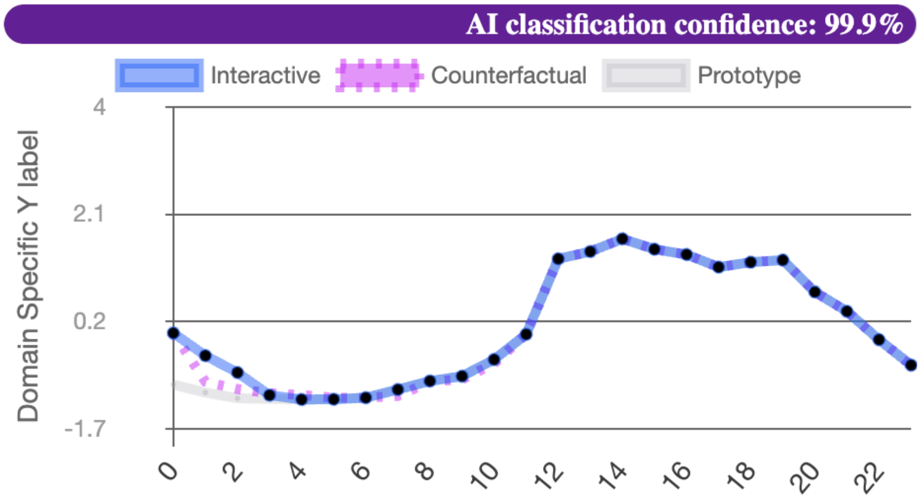


Fig. 5: Last changes, still Blue.

of quantitative user evaluations involving end-users, i.e. non-experts, on XAI for time series. In this section we present this user evaluation. We start by presenting the discussion leading up to the 3 time series datasets chosen for the evaluation. We then discuss the 3 distinct survey groups that will be given different abilities in the training stage. We end by giving the results of the evaluation, on each pairing of dataset and survey group, and a discussion of their significance.

Our tool is designed for univariate datasets. Moreover, since our test users are not domain experts and time series are notoriously non-intuitive to us humans we had some desiderata when choosing datasets for our survey. Firstly, we wanted univariate datasets with a binary classification, Secondly, we wanted datasets on time series with not too many data points. Thirdly, we wanted datasets where it is fairly easy for a non-expert to understand the domain. Datasets satisfying these criteria will increase the prospective of providing constructive quantitative feedback, and also they form the more common situation where an explanation would potentially be provided for end-users. However, it is important to note that apart from the above constraints we did not want simple datasets where the binary classification is particularly easy or could be described (e.g. by ourselves) in any straightforward way. We chose the following three datasets satisfying the above criteria:

- From UCR [3]: Italy Power Demand. This dataset shows power demand in Italian households over a 24-hour time period, and classifies these into Winter (October-March) and Summer (April-September).
- From UCR [3]: Chinatown. This dataset shows the number of pedestrians on a particular street corner of Chinatown in Melbourne over a 24-hour time period, and classifies these into Weekend (Sat-Sun) and Weekdays (Mon-Fri).
- From Eviny: Car Charging. This dataset shows the power demand at a particular charging station for electric vehicles over a 24-hour period, and classifies these into Weekend (Sat-Sun) and Weekdays (Mon-Fri).

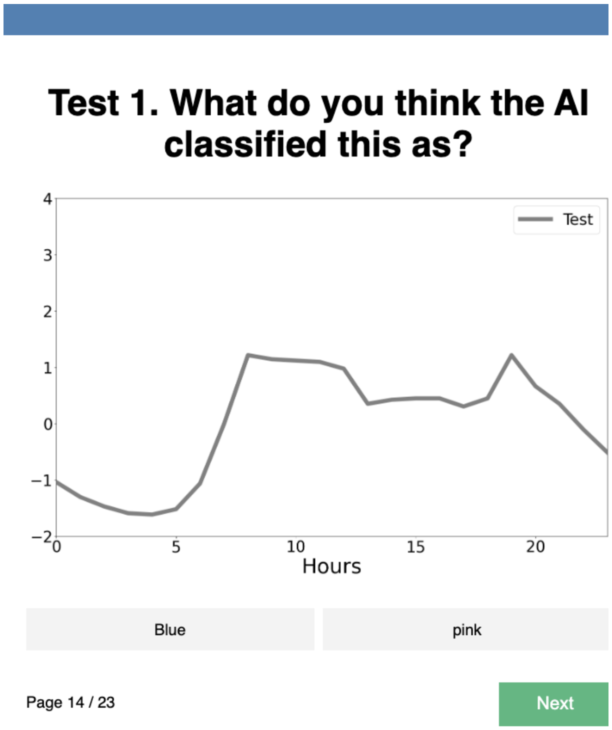
Table 1: Information of the datasets employed in the evaluation. We also include information about the proportion of data employed for training and testing the data, and the accuracy obtained by the learned model.

Dataset	Number of instances	Class distribution	Train/Test	Accuracy
ItalyPowerDemand	363	104/259	6%/94%	98%
ChinaTown	1096	547/549	5%/95%	96%
Car Charging	365	269/96	75%/25%	57%

Some details of the datasets employed in the experiments can be found in Table 1. We also include the accuracy obtained by the trained models. In the datasets we used the split between training and test defined and the repository except from Car charging where we employed a random selection of 75% train and 25% test.

As mentioned earlier, the three main components used in our interactive XAI system for time series are prototypes (PT), counterfactuals (CF), and model inspection (MI) with user-generated instances. Our survey groups will enter a 3-stage process, as follows:

- Intro: A short introduction is given to the relevant components, the dataset, training stage, and testing stage.
- Training: Group dependent.
- Testing: 10 randomly selected time series from the dataset are shown, and for each one the user must guess the AI model classification. See Figure below.



Test 1. What do you think the AI classified this as?

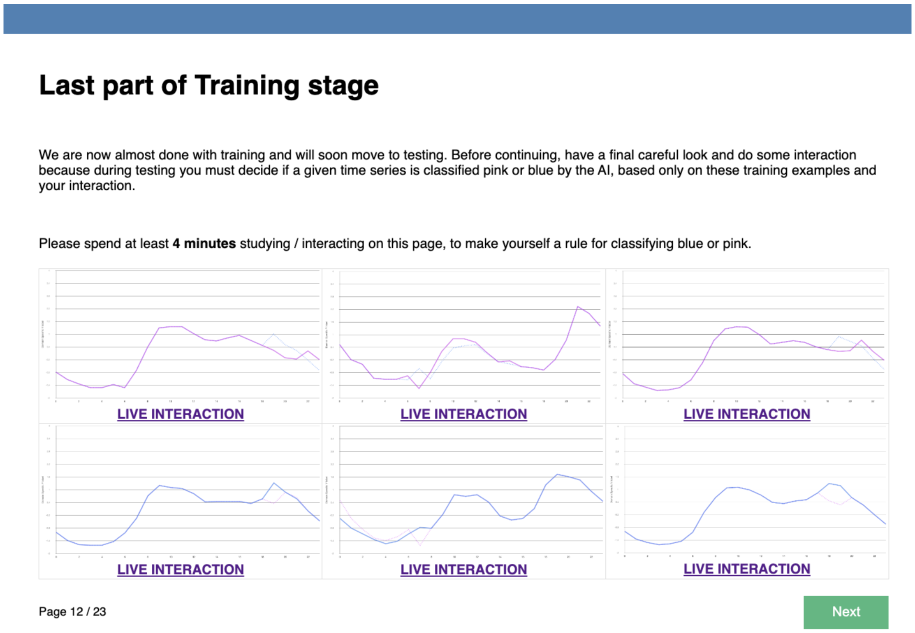
Fig. 6: Test case for ItalyPowerDemand.

To cover the distribution of time series within each class, we wanted to show users more than one prototype for each class. However, we did not want to overload the user with information, hence we opted to have 6 prototypes in total, three for each class. We will have three survey groups (PT, PT+CF, PT+CF+MI) depending on what is made available to users in the Training stage:

- PT: One at a time, the user is shown 3 prototypes from class A, then 3 from class B, and finally shown a screen with all 6 prototypes.
- PT+CF: One at a time, the user is shown 3 prototypes from class A together with a counterfactual from class B, then 3 converse pairs, and finally shown a screen with all 6 pairs.

- PT+CF+MI: The user is shown same as PT+CF but model inspection is permitted, with prototypes being interactive to allow iterative changes of any chosen time points, and the ensuing classification and also new counterfactual continually updated.

In Figures 2-5 we see how a user in group PT+CF+MI is shown a pair consisting of a prototype and a counterfactual and is allowed to modify data points. A user in group PT+CF is shown a single such pair consisting of prototype and counterfactual but without the ability to make modifications, while a user in group PT is shown only the prototype. In the last part of the training stage the users are shown all 6 prototypes/counterfactuals on one screen, to easily make a comparison, as in Figure 7 for group PT+CF+MI.



Last part of Training stage

We are now almost done with training and will soon move to testing. Before continuing, have a final careful look and do some interaction because during testing you must decide if a given time series is classified pink or blue by the AI, based only on these training examples and your interaction.

Please spend at least **4 minutes** studying / interacting on this page, to make yourself a rule for classifying blue or pink.

Page 12 / 23

Fig. 7: Blue and pink prototypes with counterfactuals, for ItalyPowerDemand.

5.1 Evaluation results

Let us present the results of the user survey. In total, we had 65 voluntary participants, who were either students in a university-level informatics course, or researchers in informatics, thus with experience in using PCs, none of whom received compensation. The participants were presented with the survey and freely chose to participate. The participants were randomly divided into the 3 survey groups. After introduction and training they were asked to answer 10 test questions from each of up to 3 datasets. Those who spent less than a minimum amount of time (less than 1 minute for training and testing combined on a single dataset) were discarded, as we considered that it would be impossible to even read the provided information in that time.

In Table 2 we see the accuracy and number of tests (i.e. number of participants times 10) for each of the three survey groups, on each of the three datasets. Accuracy is the percentage of correct test answers. The rightmost column gives the aggregated information for each survey group, and the bottom row the aggregated information for each dataset. The value in the bottom right corner shows that the overall accuracy was 66.7%, thus clearly better than a random guess, and satisfactory given the complicated nature of these time series classifications.

Let us compare the performance between the three survey groups. Perhaps surprisingly, on aggregate we see that survey group PT that was only shown the prototypes, did slightly better than the other two, with accuracy of 70.2% versus 65.9% and 64%. This could indicate that non-expert users are not necessarily able to use the extra information provided by counterfactuals and model inspection in a meaningful way. It could also mean that a 'rule-of-thumb' that appears useful based only on prototypes (or prototypes + counterfactuals) actually works well on a high percentage, say 80%, of test instances. However, after close model inspection a user in group PT+CF+MI may discover that this rule is not precise (as it fails on several instances) leaving the user to discard this rule-of-thumb, and subsequently actually performing worse on the tests. A final possible explanation for why group PT+CF+MI did not achieve higher accuracy is the fact that the average time spent by users in total on all 3 datasets was about the same, around 15 minutes: group PT 954 seconds, group PT+CF 934 seconds, group PT+CF+MI 966 seconds. However, it seems natural that a user doing model inspection to learn a better rule should have spent more time than one who cannot do model inspection, making us question how careful the users in group PT+CF+MI were. On the other hand, there is not a clear trend when we compare accuracy versus time spent for users in this group.

Let us turn to comparing between the 3 datasets. Interestingly, all 3 survey groups had the highest accuracy on Chinatown and the lowest accuracy on CarCharging. We asked some users about rules they had been using for their own mental classification, and the most mentioned rule was for Chinatown, something like the following: 'if there is a sharp dip at the beginning of the series then it is Blue, and otherwise Pink'. Compare this rule to Figures 2-5 from the Chinatown dataset. See also Figures 7 and 6 which for ItalyPowerDemand gives all prototypes and counterfactuals, plus an example of a test, to get an impression of how difficult the classification indeed is for this dataset. For the test shown in Figure 6 the average accuracy was 59.3%. Note the users could not navigate back to see the prototypes when answering the tests.

Table 2: Overview of accuracy and number of tests by survey group and dataset.

Survey ID	Data	ItalyPowerDemand	Chinatown	CarCharging	All 3 datasets
PT	accuracy	65.9%± 10	79.4%± 13	65.3% ± 14	70,2%± 14
	tests	170	170	170	510
PT+CF	accuracy	61.9%± 17	81.3%± 13	54.4% ± 21	65.9%± 20
	tests	160	160	160	480
PT+CF+MI	accuracy	60.2%± 16	76.2%± 22	55.7%± 13	64.0%± 20
	tests	240	210	230	680
All groups	accuracy	62.7%± 15	79.0%± 17	58.4%± 17	66.7%± 19
	tests	570	540	560	1670

6 Conclusion

As companies and controllers must cope with the EU regulations in form of GDPR and the AI Act, explainability that allows model inspection by end-users may very well become the target for developers. In domains where we humans have poor intuition, such as time series, this may pose several challenges. In this work we have contributed a tool for XAI on time series classification that allows such model inspection. The evaluation results show that users can then successfully perform a difficult forward simulation test. However, to attain the full benefits from model inspection for such a complicated domain, it seems necessary to have highly motivated users that are willing to spend more time with such a tool, in order to form better mental models of the black-box classification. As future work, we propose to explore the use of simplified versions of time series prototypes generated by Machine Teaching techniques to explain time series classification models.

Acknowledgement. This paper is part of NRF project 329745 Machine Teaching For XAI. We thank the anonymous reviewers for their comments and the volunteers who participated in the experiments. This work was funded by ValGrai, CIPROM/2022/6 (FASSLOW) and IDIFEDER/2021/05 (CLUSTERIA) funded by Generalitat Valenciana, the EC H2020-EU grant agreement No. 952215 (TAILOR), US DARPA HR00112120007 (RECoG-AI) and Spanish grant PID2021-122830OB-C42 (SFERA) funded by MCIN/AEI/10.13039/501100011033 and “ERDF A way of making Europe".

Disclosure of interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abdul, A.M., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.S.: Trends and trajectories for explainable, accountable and intelligible systems: An HCI research

agenda. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, CHI 2018. p. 582. ACM (2018)

2. Beretta, I., Cappuccio, E., Manerba, M.M.: User-driven counterfactual generator: A human centered exploration. In: Conf. on eXplainable Artificial Intelligence (xAI-2023). pp. 83–88. CEUR Workshop Proceedings (2023)
3. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML: The UCR time series classification archive (2018)
4. Delaney, E., Greene, D., Keane, M.T.: Instance-based counterfactual explanations for time series classification. In: Case-Based Reasoning Research and Development - Int. Conference, ICCBR 2021. pp. 32–47. Springer (2021)
5. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. CoRR **abs/1702.08608** (2017)
6. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
7. Ford, C., Keane, M.T.: Explaining classifications to non-experts: an XAI user study of post-hoc explanations for a classifier when people lack expertise. In: International Conference on Pattern Recognition. pp. 246–260. Springer (2022)
8. Guillemé, M., Masson, V., Rozé, L., Termier, A.: Agnostic local explanation for time series classification. In: 31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019. pp. 432–439. IEEE (2019). `https://doi.org/10.1109/ICTAI.2019.00067`, `https://doi.org/10.1109/ICTAI.2019.00067`
9. Höllig, J., Kulbach, C., Thoma, S.: Tsinterpret: A python package for the interpretability of time series classification. J. Open Source Softw. **8**(87), 5220 (2023)
10. Ismail, A.A., Gunady, M.K., Bravo, H.C., Feizi, S.: Benchmarking deep learning interpretability in time series predictions. In: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual (2020)
11. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A.: Deep learning for time series classification: a review. Data mining and knowledge discovery **33**(4), 917–963 (2019)
12. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley (1990). `https://doi.org/10.1002/9780470316801`, `https://doi.org/10.1002/9780470316801`
13. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
14. Poché, A., Hervier, L., Bakkay, M.C.: Natural example-based explainability: A survey. In: Explainable Artificial Intelligence - 2023. vol. 1902, pp. 24–47. Springer (2023)
15. Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Hardt, M., Liu, P.J., Liu, X., Marcus, J., Sun, M., et al.: Scalable and accurate deep learning with electronic health records. NPJ digital medicine **1**(1), 1–10 (2018)
16. Schlegel, U., Arnout, H., El-Assady, M., Oelke, D., Keim, D.A.: Towards a rigorous evaluation of xai methods on time series. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 4197–4201. IEEE (2019)
17. Shneiderman, B.: Human-centered artificial intelligence: Reliable, safe & trustworthy. International Journal of Human–Computer Interaction **36**(6), 495–504 (2020)
18. Siddiqui, S.A., Mercier, D., Munir, M., Dengel, A., Ahmed, S.: Tsviz: Demystification of deep learning models for time-series analysis. IEEE Access **7**, 67027–67040

(2019). https://doi.org/10.1109/ACCESS.2019.2912823, https://doi.org/10.1109/ACCESS.2019.2912823

19. Su, L., Zhang, S., McGaughey, A.J., Reeja-Jayan, B., Manthiram, A.: Battery charge curve prediction via feature extraction and supervised machine learning. Advanced Science **10**(26), 2301737 (2023)
20. Susto, G.A., Cenedese, A., Terzi, M.: Time-series classification methods: Review and applications to power systems data. Big data application in power systems pp. 179–220 (2018)
21. Theissler, A., Spinnato, F., Schlegel, U., Guidotti, R.: Explainable AI for time series classification: A review, taxonomy and research directions. IEEE Access **10**, 100700–100724 (2022)
22. Wang, Z.J., Vaughan, J.W., Caruana, R., Chau, D.H.: GAM coach: Towards interactive and user-centered algorithmic recourse. In: Proc. Conf. on Human Factors in Computing Systems,. pp. 835:1–835:20. ACM (2023)
23. Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 2921–2929. IEEE Computer Society (2016)