Counterfactual User Guidance for Improving Transparent Hyperparameter Tuning

Sabri Manai[1][0009−0009−2242−7391], Szymon Bobek[1][0000−0002−6350−8405], Mehrdad Asadi[2][0000−0003−3360−9579], Ann Nowé[2][0000−0001−6346−4564], and Grzegorz J. Nalepa[1][0000−0002−8182−4225]

[1] Department of Human-Centered Artificial Intelligence, Institute of Applied Computer Science, Jagiellonian University, Krakow, Poland
sabri.manai@doctoral.uj.edu.pl
szymon.bobek@uj.edu.pl, grzegorz.j.nalepa@uj.edu.pl
[2] AI Lab, Vrije Universiteit Brussel
mehrdad.asadi@vub.be, ann.nowe@vub.be

Abstract. Hyperparameter optimization (HPO) is usually framed as a fully automated search that returns a single "best" configuration, leaving human operators with little insight into alternatives or trade-offs. However, in many applied settings, practitioners arrive with concrete wishes such as higher precision, less labeling effort, or faster models that standard HPO tools cannot address directly. In this work, we propose a surrogate-based explainable AI (XAI) framework that turns informal requirements into counterfactual queries over the configuration space. A CatBoost model is trained on the resulting configuration-performance log and analysed with XAI tools, then queried by the DiCE and CFNOW explainers to generate counterfactual configurations under fixed soft constraints. Experiments on four standard binary classification benchmarks show that the framework can provide locally reliable guidance. In particular, DiCE often suggests alternative settings that result in consistent improvements when the main model is retrained.

Keywords: Hyperparameter optimization · Explainable AI · Counterfactual explanations

1 Introduction

Hyperparameter optimization (HPO) is a cornerstone of building high-performing machine learning models, yet it often remains a black-box process, leaving human operators with limited understanding of why certain configurations are selected. This lack of transparency is particularly critical in domains where human operators bear full responsibility for the final model, such as healthcare, finance, industry, or other safety-critical systems. In these contexts, understanding the rationale behind hyperparameter choices is essential for trust, accountability, and informed decision-making. With this insight, users can establish sufficient

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

trust to deploy optimized models [8]. In this work, we introduce a novel framework for explainable hyperparameter optimization that significantly enhances interpretability and actionability. Our approach brings two key contributions.

First, we integrate counterfactual explanations into HPO, allowing users to query the optimizer with actionable questions such as "What changes would increase model quality by 20%?". Unlike traditional feature-importance methods, counterfactuals provide direct guidance on how to achieve specific performance improvements, bridging the gap between model optimization and human decision-making.

Second, we introduce soft parameters, a new class of hyperparameters that are not intrinsic to the model but represent human-dependent actions, such as labeling more data, handling missing values more effectively, or modifying pre-processing strategies. By optimizing these parameters alongside traditional hyperparameters and providing counterfactual explanations, our framework enables users to understand both the importance of each hyperparameter and the concrete actions required to improve model performance.

Together, these innovations make hyperparameter optimization not only more transparent but also actionable, empowering practitioners to make informed decisions that combine algorithmic guidance with human expertise.

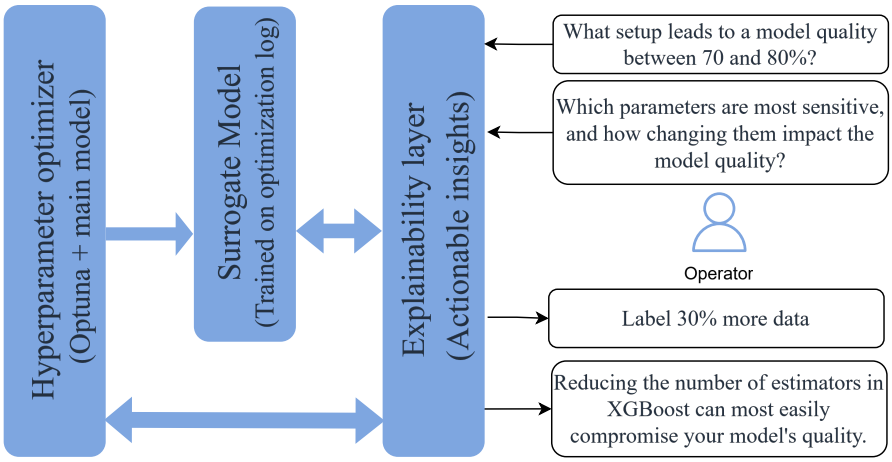

Fig. 1. Workflow of the explainable hyperparameter optimization approach

Figure 1 illustrates the overall workflow. The hyperparameter optimization procedure jointly considers hard parameters (model-specific) and soft parameters (human-dependent actions) to optimize performance. A surrogate model is trained on logged optimization data to provide detailed insight into the relationship between hyperparameters and model quality. Explanations are presented as

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

counterfactuals, which are immediately actionable: they specify concrete hyperparameter settings or human actions required to achieve targeted improvements.

The remainder of the paper situates our approach within the landscape of existing explanation mechanisms for hyperparameter optimization and provides a detailed discussion of the proposed method along with its evaluation on benchmark datasets.

2 Related work

HPO is an essential but costly component of machine learning pipelines [1, 3]. To enhance the efficiency of this process, several Automated Machine Learning (AutoML) frameworks were developed, and a comparison of nine different optimizers showed that Optuna [2] is the best-performing AutoML model [11].

Recent research highlights the need for approaches that are not only efficient, but also interpretable and aligned with human operators, especially in high-stakes or industrial settings [21]. Table 1 summarizes selected approaches in explainable hyperparameter optimization and AutoML, indicating their optimization methods, explanation techniques, and purpose. While prior work emphasizes global insights or visual interfaces, our approach focuses on local, user-guided exploration through counterfactual queries.

Table 1. Overview of related work on explainable optimization and AutoML.

Work	**HPO Method**	**Explanations**	**Purpose**
[18] [14] [5] [6] [17]	Bayesian Optimization (BO)	TNTRules (TuneNOTune Rules), SHAP, PDPs, LIME, Systematic XAI	Explain BO-based HPO by visualising search behaviour, deriving tuning rules, and identifying influential hyperparameters.
[16]	ShrinkHPO	SHAP	Guide search space reduction and improve optimization efficiency.
[11]	AutoML (9 optimizers)	SHAP, LIME, PDPs, Feature importance	Analyse the contribution of hyperparameters across multiple AutoML systems.
[22] [10] [19]	AutoML Frameworks	Visual plots, Visual analytics, Web GUI	Provide explainable and interactive AutoML through visual interfaces.
[9] [20] [21]	AutoRL, AutoDO, SSHH	Visual analytics, Surrogate rule extraction, KL/HMM maps	Explain optimization behaviour in AutoRL and heuristic search using visual analytics and probabilistic diagnostics.
[13] [4]	R-XIMO, EXMOS	SHAP, Data-centric and Model-centric XAI	Support explainable multiobjective optimization and expert-guided parameter tuning.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

3 Counterfactual Explanations for Actionable Hyperparameter Optimization

Figure 2 sketches the full workflow. First, an Optuna study samples both model-specific hyperparameters and soft, human-controllable knobs to build a configuration-performance log. A surrogate model with XAI tools is then trained on this log. Counterfactual explainers such as DiCE [15], which generates diverse counterfactual examples, or CFNOW [7], a search-based counterfactual method, then query the surrogate under fixed soft constraints to propose alternative configurations, which are finally re-evaluated with the main XGBoost model on validation and test splits.

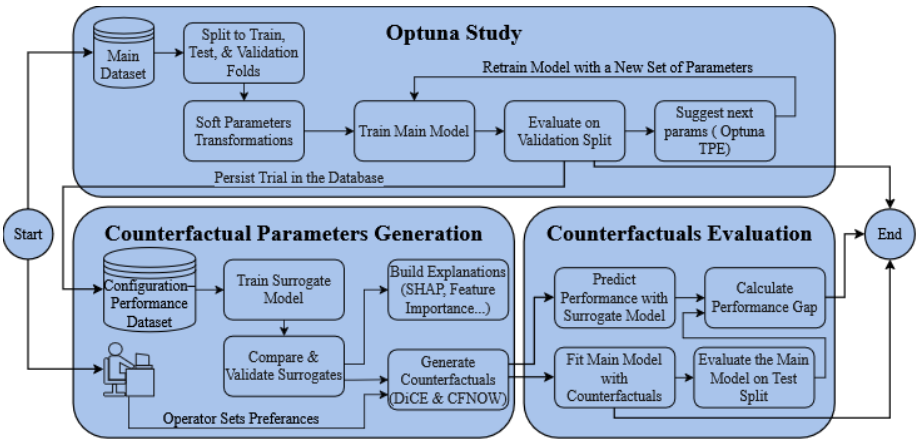

Fig. 2. End-to-end pipeline for counterfactual-based hyperparameter optimization

3.1 Data, Preprocessing, and Search Space

The methodological design assumes a general supervised learning setting in which datasets include an input feature space and an associated target variable, without restricting the task to any particular prediction type or number of classes. To structure model development, each dataset is partitioned into stratified or otherwise distribution-preserving splits for training, validation, and testing—commonly following a 60/20/20 ratio, though the procedure is agnostic to the specific proportions. During optimization, only the training split is subject to optional transformations governed by configurable parameters. These may include subsampling training instances through a row-sampling mechanism and controlled introduction of missing values regulated by a missing-rate parameter. After such transformations, categorical features may be one-hot encoded when applicable, while numerical features are passed through in their native form. The validation and test splits remain unchanged and serve solely for model selection and final performance assessment within the supervised learning pipeline.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

The optimization procedure treats both model-specific hyperparameters and broader, human-controllable soft knobs as part of a unified search space. In our demonstration, the predictive model is an XGBoost classifier, and the optimizer (e.g., Optuna) samples its standard hyperparameters jointly with the soft knobs to create a diverse configuration-performance log. This log is later used to train the surrogate model. In the subsequent counterfactual stage, however, the soft knobs are no longer optimized; instead, they define the operator's query constraints, while only the model-specific hyperparameters are varied to generate alternative configurations. However, the overall framework is model-agnostic in principle, and the same general procedure can be applied to other supervised learning models by redefining the model-specific parameter space and collecting the corresponding configuration-performance data.

The user-controllable parameters influence data usage and learning behaviour, including the fraction of training rows, the injected missingness rate, the class-imbalance handling strategy (`resampling_strategy`), the feature fraction, and the proportion of labels retained for the semi-supervised SelfTraining scheme (`labeled_ratio`).

After the optimization run, we derive three additional feature-engineered soft parameters from each configuration. Let $N_{\text{samples}}^{\text{total}}$ be the size of the total training split and $\rho \in [0, 1]$ the used fraction for the actual configuration (`training_rows_fraction`); the effective number of training examples is given by Eq. (1).

$$N_{\text{samples}}^{\text{training}} = \text{round}\big(\rho\, N_{\text{samples}}^{\text{total}}\big) \tag{1}$$

To obtain a knob for decision-speed category, we define the complexity proxy in Eq. (2):

$$c = \texttt{max\_depth} \cdot \texttt{n\_estimators} \tag{2}$$

and compute its empirical tertiles $q_{0.33}, q_{0.66}$ per dataset; configurations with $c < q_{0.33}$, $q_{0.33} \leq c < q_{0.66}$, and $c \geq q_{0.66}$ are labelled `fast`, `balanced`, and `slow`, respectively. Finally, we derive the scalar risk score in Eq. (3):

$$s = \log(\texttt{scale\_pos\_weight}) + r \tag{3}$$

where $r = 1$ for `oversample`, 0.5 for `undersample`, and 0 otherwise. Trials with $|s| < 0.2$ are tagged as `balanced`, while $s > 0.2$ and $s < -0.2$ correspond to `recall_pref` and `precision_pref`, respectively.

These soft parameters were selected because, in our experience, they capture design choices that commonly arise in supervised learning workflows, such as data usage, missing data, class imbalance, feature selection, and the labeling effort needed from the user. They are either directly sampled or derived from configuration-level quantities, providing a compact and interpretable representation of variation that can be expected across a range of modelling setups.

Table 2 summarizes the configuration space explored in the study. For each parameter, we report its name, type, defined range or set of values, and the sampling scheme (strategy Optuna uses to draw candidate values). The table is split into two blocks, covering standard XGBoost hyperparameters and the soft, human-controllable knobs introduced in this work.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

Table 2. Search space specification for XGBoost hyperparameters and soft human-controllable knobs.

Parameter	Type	Range / Values	Sampling
Model hyperparameters (XGBoost)			
booster	categorical	{gbtree, dart}	categorical
n_estimators	integer	[200, 1000]	step=100
max_depth	integer	[3, 12]	uniform
learning_rate	float	[1e-4, 0.5]	log-uniform
min_child_weight	float	[0.1, 100.0]	log-uniform
gamma	float	[0.0, 10.0]	uniform
lambda (L2)	float	[1e-4, 100.0]	log-uniform
alpha (L1)	float	[1e-4, 100.0]	log-uniform
subsample	float	[0.2, 1.0]	uniform
colsample_bytree	float	[0.2, 1.0]	uniform
grow_policy	categorical	{depthwise, lossguide}	categorical
max_delta_step	integer	[0, 10]	uniform
rate_drop*	float	[0.0, 0.5]	uniform
skip_drop*	float	[0.0, 0.9]	uniform
sample_type*	categorical	{uniform, weighted}	categorical
normalize_type*	categorical	{tree, forest}	categorical
scale_pos_weight	float	[0.5, 5.0]	log-uniform
Soft, human-controllable knobs			
selftrain_threshold	float	[0.6, 1]	uniform
selftrain_max_iter	integer	[0, 12]	uniform
training_rows_fraction	float	[0.5, 1.0]	uniform
missing_rate	float	{0, 0.02, 0.05, 0.08, 0.1, 0.12, 0.14, 0.16, 0.18, 0.2, 0.22, 0.24, 0.26, 0.28, 0.3}	discretized grid
resampling_strategy	categorical	{none, undersample, oversample}	categorical
feature_fraction	float	[0.5, 1.0]	uniform
labeled_ratio	float	[0.05, 0.95]	uniform

* Only active when `booster = dart`.

3.2 Surrogate Models and Explanations

Once the Optuna study is complete, each trial yields a record linking a hyperparameter configuration to its resulting validation performance. Across the four benchmark datasets, filtering invalid or interrupted runs produces four surrogate datasets of comparable scale: 11,077 trials for Adult Income, 6,989 for Breast Cancer, 6,993 for Bank Marketing, and 4421 for Phishing Websites. Each dataset contains the sampled hard hyperparameters, the soft attributes derived from human-controllable knobs, and multiple evaluation metrics logged during optimization. These structured datasets serve as training material for surrogate modelling, whose goal is to approximate the relationship between hyperparame-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

ter configurations and resulting model performance, enabling fast counterfactual queries without repeated full model training.

Several regression models are trained and compared as surrogates, including Random Forest, ExtraTrees, Gradient Boosting, XGBoost, LightGBM, and CatBoost. All models are trained to predict the validation F1 score recorded during optimization, and their performance is evaluated using a held-out split of the configuration–performance datasets. The best-performing surrogate is then selected for use in downstream counterfactual generation.

To analyze how the surrogate responds to different hyperparameters and soft knobs, multiple explanation techniques are applied. These include CatBoost's built-in global feature importance (PredictionValuesChange), permutation feature importance computed on the validation split, and SHAP-based explanations [12] derived using a TreeExplainer. The explanations are examined both at a global level (mean absolute SHAP values) and locally for selected configurations. Together, these views highlight parameters with strong influence on predicted performance, identify weak or redundant knobs, and guide the search toward actionable directions for counterfactual exploration.

3.3 Counterfactual Generation and Evaluation

To obtain counterfactual explanations while preserving the operator's constraints, instead of exposing the full hyperparameter space, the system relies on a small set of interpretable *soft controls* that an operator may specify for a given query. These include how much of the training data may be used (`sample_rows`), the amount of injected missingness (`missing_rate`), handling class imbalance (`resampling_strategy`), the allowed feature fraction, and settings related to semi-supervised behaviour (`labeled_ratio`, `selftrain_threshold`, and `selftrain_max_iter`). Each trial also includes two derived descriptors: a *risk preference label* (balanced, recall-preferring, or precision-preferring) and a *decision-speed category*, based on model complexity.

At query time, a configuration that satisfies the chosen soft controls is selected as a factual starting point. From this configuration, we generate alternative candidates using two counterfactual frameworks: DiCE and CFNOW. Both are applied under the same soft-control constraints, which remain fixed throughout the query, so that the counterfactuals represent realistic modifications of the model rather than changes to the underlying operational assumptions. Thus, DiCE and CFNOW are allowed to modify only the hard model hyperparameters within the permitted ranges. All proposed counterfactual configurations are evaluated in two steps.

First, a fast check uses the surrogate model to assess whether a candidate behaves consistently with the expected performance change. This stage filters out unstable or implausible counterfactuals.

Second, the remaining candidates are executed as full training runs, following the same soft-control settings as the factual configuration, and evaluated on the validation and held-out test splits. Each configuration is run across multiple seeds to assess robustness.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

4 Results

We evaluate the framework on four standard binary classification benchmarks: Adult Income (OpenML, version 2), Breast Cancer Wisconsin (Diagnostic, via `load_breast_cancer`), Bank Marketing (OpenML, version 1), and Phishing Websites (OpenML, version 1). For the Adult Income dataset, the target indicates whether income exceeds $50K; for Bank Marketing, we predict whether a client subscribes to a term deposit, and for Phishing Websites, whether a site is legitimate; in the Breast Cancer dataset, the target distinguishes malignant from benign tumours. For each dataset, we retain the original feature set and construct a stratified 60/20/20 split into train, validation, and test. During optimization, only the training split is modified: rows may be subsampled according to the `sample_rows` knob, and missing values are injected at a rate controlled by the `missing_rate` soft parameter (with categorical features one-hot encoded where applicable and numerical features passed through unchanged).

4.1 Surrogate Model Performance Across Datasets

Table 3 summarizes the performance of the top five surrogate models across the four configuration–performance datasets: Adult Income, Bank Marketing, Breast Cancer Wisconsin (Diagnostic), and Phishing Websites. Metrics are reported as RMSE, MAE, and R^2. Although the full study includes a broader set of regressors, including CatBoost, LightGBM, XGBoost, HistGradientBoosting, Random Forests, ExtraTrees, AdaBoost, Gradient Boosting, Decision Trees, SVR, KNN, Bayesian Ridge, ElasticNet, and Huber Regression, only the most competitive models are shown here for clarity. Across datasets, CatBoost generally provides the strongest predictive performance, while ExtraTrees attains the highest performance on the Breast Cancer dataset.

Table 3. Top five surrogate models across all four datasets (validation split). Metrics shown as RMSE / MAE / R^2. Best model per dataset in bold.

Model	Adult Income	Bank Marketing	Breast Cancer	Phishing Websites
	(RMSE ↓ / MAE ↓ / R^2 ↑)			
CatBoost	**0.06 / 0.04 / 0.92**	**0.10 / 0.04 / 0.70**	0.21 / 0.14 / 0.40	**0.17 / 0.11 / 0.43**
LightGBM	0.07 / 0.04 / 0.91	0.11 / 0.04 / 0.64	0.21 / 0.13 / 0.40	0.18 / 0.12 / 0.36
XGBoost	0.07 / 0.04 / 0.91	0.11 / 0.04 / 0.62	0.21 / 0.14 / 0.40	0.18 / 0.12 / 0.40
HistGBDT	0.07 / 0.04 / 0.91	0.10 / 0.04 / 0.67	0.22 / 0.14 / 0.37	0.18 / 0.11 / 0.39
ExtraTrees	0.07 / 0.04 / 0.90	0.11 / 0.04 / 0.63	**0.21 / 0.13 / 0.42**	0.18 / 0.11 / 0.39

4.2 Case Study: Human-Controlled Counterfactual Tuning

As an example, we apply the framework to the Adult Income dataset. The operator selects a set of soft constraints (Table 4) that reflects a recall-oriented and time-balanced setting, nearly all training data are used, no missingness is

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

injected, and no resampling is applied. This configuration serves as the factual seed for generating and evaluating counterfactual hyperparameter settings.

Table 4. Soft-control profile for the Adult Income case study.

Soft knob	Value	Soft knob	Value
Risk preference	`recall_pref`	Resampling strategy	`none`
Decision speed	`balanced`	Labeled ratio	1.00
Training rows fraction	0.976	Total train samples	28612
Feature fraction	0.933	Injected missing rate	0.00

Filtering the optimization log under these soft constraints yields 341 candidate configurations. Among them, we select a factual seed whose surrogate-predicted validation F1 lies in the range set by the operator $[0.70, 0.80]$; the chosen configuration has a surrogate F1 of approximately 0.716 and serves as the starting point for counterfactual exploration.

From this factual configuration, DiCE is applied in a human-controlled setting, where the soft knobs are treated as constraints for the current query (Table 4). What-if exploration is obtained by changing the soft-control profile and rerunning the query. For this query, DiCE varies only model-specific hyperparameters (e.g., depth, learning rate, number of trees) within their predefined ranges, and returns 16 counterfactual configurations whose surrogate-predicted scores remain in the target band.

Table 5. Example counterfactual configuration (DiCE).

Model hyperparameters		Soft parameters		Target (F1)
Name	Value	Name	Value	Value
alpha	0.00021	feature_fraction	0.933	0.717792
booster	gbtree	resampling_strategy	none	
colsample_bytree	0.317368	training_rows_fraction	0.976	
gamma	3.372142	labeled_ratio	1.0	
grow_policy	depthwise	injected_missing_rate	0.0	
lambda	0.055209	risk_preference	recall_pref	
learning_rate	0.222288	decision_speed	balanced	
max_delta_step	8	total_train_samples	28612	
max_depth	8			
min_child_weight	0.23602			
n_estimators	300			
decision_threshold	0.5			
selftrain_threshold	1.0			
selftrain_max_iter	0.0			
normalize_type	NA			
scale_pos_weight	1.2			
rate_drop	0.0			
sample_type	NA			
skip_drop	0.0			
subsample	0.790080			

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

Table 5 reports one representative counterfactual returned by DiCE. As required by the human-controlled setting, all soft parameters (i.e., risk preference, decision speed, resampling strategy, training-rows fraction, injected missing rate, feature fraction, and labeling ratio) are identical to the operator-defined profile in Table 4; only XGBoost hyperparameters are modified to explore alternative configurations. The F1 shown in the last column corresponds to the DiCE predicted score for this counterfactual, which satisfies the target band constraint used during generation.

Each counterfactual is then re-evaluated by retraining the XGBoost model $K = 5$ times with different random seeds, aggregating validation and test F1 scores. Figure 3 compares surrogate-predicted and realised test F1 for the obtained 16 DiCE counterfactuals and the selected factual seed. The points cluster near the diagonal and the association is moderate and positive (Pearson $r = 0.529$, $p = 0.024$ for all points including factual seed; $r = 0.514$, $p = 0.035$ for counterfactuals only), indicating that higher surrogate estimates tend to correspond to higher realised test performance in this local neighbourhood.

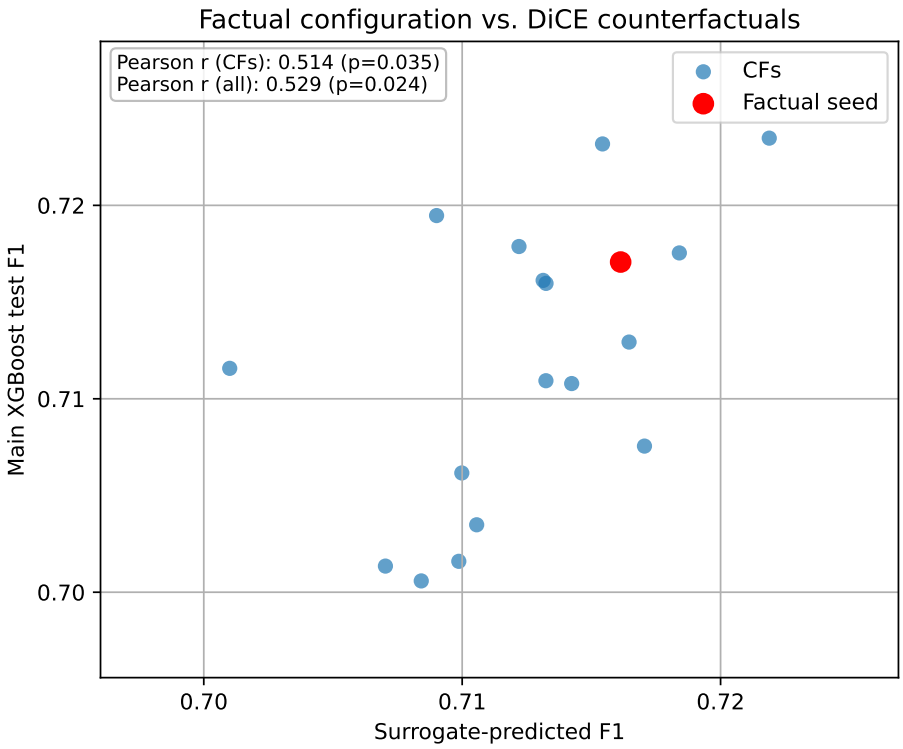

Fig. 3. Surrogate-predicted versus realised test F1 for the factual configuration and its counterfactual neighbours on Adult Income.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

Since F1 alone does not show how the precision/recall trade-off changes, we additionally report the main model's precision and recall on the Adult test split for the three highest-performing counterfactuals (ranked by test F1). This makes it possible to assess whether the generated configurations are consistent with the intended recall-oriented query, rather than relying on F1 as a single aggregate score. The results in Table 6 indicate that recall remains relatively high across the reported configurations, while precision varies as expected under a recall preference.

Table 6. Adult case study (query with recall preference): test precision, recall, and F1 for selected counterfactuals, with surrogate and DiCE F1.

Config	Precision	Recall	Main Model's F1 (test split)	Surrogate F1	DiCE F1
CF 11	0.660	0.793	0.720	0.709	0.705
CF 16	0.694	0.750	0.718	0.722	0.723
CF 8	0.704	0.750	0.723	0.715	0.720

4.3 Counterfactual Quality: Surrogate vs. Main Model

To assess the full pipeline, we compare three quantities per dataset and explainer: first, the surrogate's predicted F1, second, the test F1 obtained after refitting XGBoost with the counterfactual hyperparameters, and third, the mean F1 of the generated counterfactual configurations. For each setting, the reported values are averages over 10 counterfactual configurations, rounded to three decimal places. Each configuration is further evaluated by retraining XGBoost five times and averaging the resulting scores, which mitigates stochastic effects such as varying missing-value patterns when `missing_rate` is non-zero.

Figure 4 summarises these results for DiCE (left panels) and CFNOW (right panels). DiCE generally produces counterfactuals that stay close to the surrogate estimates and often yield small but consistent gains when the model is retrained. CFNOW generates stable but more conservative counterfactuals, which in some datasets do not translate into improvements for the main model.

5 Discussion

Some issues need to be considered when assessing the surrogate model and its stability. One of the main challenges is the granularity of the collected data. It was difficult to obtain configurations that cover the full range of performance, as Optuna's Tree-structured Parzen Estimator (TPE) sampler tends to focus on improving the model. Even when drawing random samples, many configurations converged either toward the maximum performance or toward zero. Only the Adult Income dataset produced a well-distributed set of hyperparameters, which likely explains the better surrogate results there. At the same time, the surrogate is not meant to remove the cost of the initial optimization run, but to reuse its

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

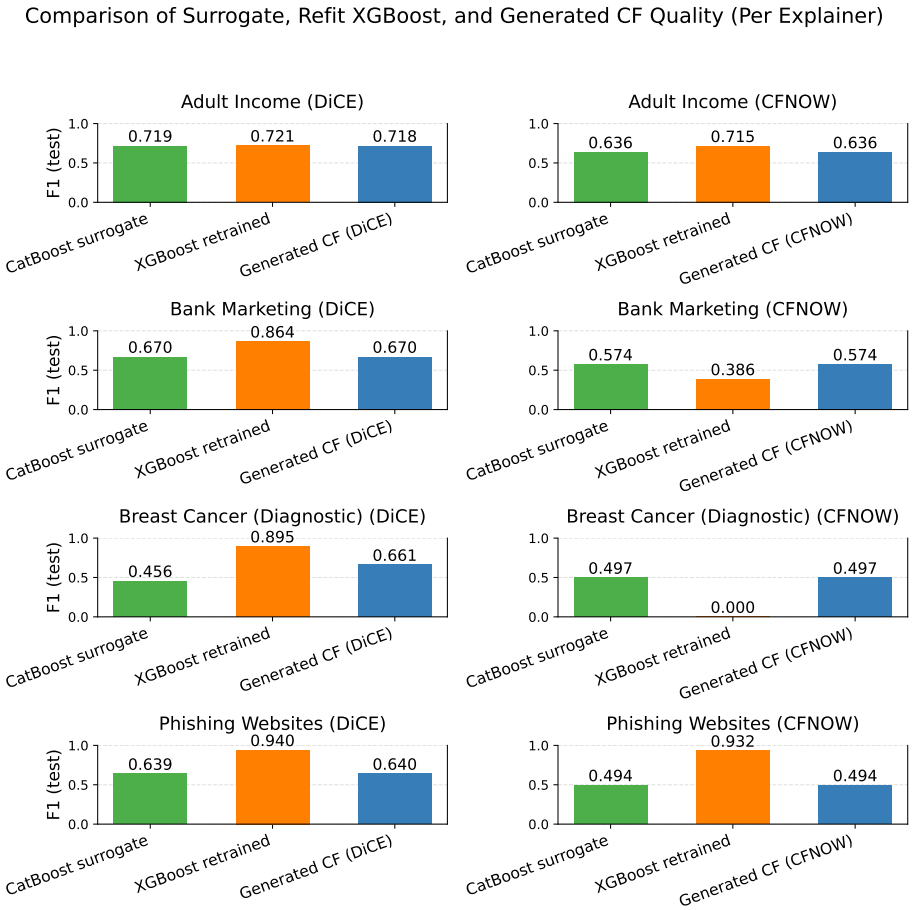

Fig. 4. Counterfactual quality across datasets for DiCE and CFNOW

results for fast counterfactual exploration. This is especially useful when several operator-guided alternatives must be examined instead of selecting only a single best configuration.

Another point is the amount of randomness introduced into the data. High missing-value rates or very low sampling fractions can affect both model performance and the surrogate's ability to approximate it. Moreover, while introducing soft hyperparameters can in principle generalise to different models and datasets, the current pipeline is still specific to XGBoost. Extending it would require collecting new configuration–performance datasets using the same procedure but adapted to the hyperparameters of the chosen main model. Furthermore, in our setting, the soft knobs define the counterfactual query. Within a given query, they are kept constant, while the search varies only model-specific hyperparam-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

eters around the factual seed. What-if exploration is obtained by adjusting the soft-knob constraints and re-running the search, which yields a different set of admissible configurations and counterfactual alternatives.

A limitation of the current approach is that the counterfactual generation process is restricted to producing a small number of solutions in the locality of the factual seed. While the design choice ensures a local counterfactual generator, it does not explicitly address the trade-off between counterfactual diversity and user cognitive load. In practical decision-support settings, presenting more than a few counterfactual examples (e.g., more than three) may overwhelm users and hinder effective reasoning.

6 Conclusions and future work

This work explored an explainable approach to hyperparameter optimization that combines surrogate modelling, soft human-controllable knobs, and counterfactual explanations. Instead of navigating the full configuration space, operators can fix a set of soft constraints and examine alternative hard hyperparameter settings proposed by DiCE and CFNOW. Across the four benchmark datasets, the CatBoost surrogate achieved reasonable predictive accuracy, with notably stronger performance on Adult Income, where the optimisation dataset was more granular.

Future work will focus on extending the method to additional model families and exploring XAI-guided sampling strategies that leverage surrogate explanations during search. An important direction for this is to move toward selectively searching and identifying a minimal set of meaningful and representative solutions. This could allow the users to benefit from richer exploratory coverage without unnecessary cognitive burden caused by a large number of trade-offs. Another direction is the use of large language models to convert sets of counterfactual configurations into concise natural-language summaries that may be easier for non-experts to interpret.

An important further direction involves applying this framework in an industrial setting as part of the PEER project (Horizon Europe, Grant No. 101120406), in which we participate. Specifically, the Proditec industrial use case in the project supports human-in-the-loop optimization of anomaly detection pipelines under high-level operational requirements.

To support reproducibility, the Python notebooks used to run the experiments are available at the project repository: GitHub repository.

7 Acknowledgments

This paper is part of a project that has received funding from the European Union's Horizon Europe Research and Innovation Programme, under Grant Agreement number 101120406. The paper reflects only the authors' views, and the EC is not responsible for any use that may be made of the information it contains.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

Disclosure of Interests. The authors declare that they have no competing interests to declare.

References

1. A Ilemobayo, J., Durodola, O., Alade, O., J Awotunde, O., T Olanrewaju, A., Falana, O., Ogungbire, A., Osinuga, A., Ogunbiyi, D., Ifeanyi, A., E Odezuligbo, I., E Edu, O.: Hyperparameter Tuning in Machine Learning: A Comprehensive Review. Journal of Engineering Research and Reports **26**(6), 388–395 (Jun 2024). https://doi.org/10.9734/jerr/2024/v26i61188
2. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. p. 2623–2631. KDD '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3292500.3330701
3. Ali, Y.A., Awwad, E.M., Al-Razgan, M., Maarouf, A.: Hyperparameter search for machine learning algorithms for optimizing the computational complexity. Processes **11**(2) (2023). https://doi.org/10.3390/pr11020349
4. Bhattacharya, A., Stumpf, S., Gosak, L., Stiglic, G., Verbert, K.: EXMOS: Explanatory Model Steering through Multifaceted Explanations and Data Configurations. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 1–27. ACM, Honolulu HI USA (May 2024). https://doi.org/10.1145/3613904.3642106
5. Chakraborty, T., Wirth, C., Seifert, C.: Explainable bayesian optimization (2025), https://arxiv.org/abs/2401.13334
6. Chandramouli, S., Zhu, Y., Oulasvirta, A.: Interactive Personalization of Classifiers for Explainability using Multi-Objective Bayesian Optimization. In: Proceedings of the 31st ACM Conference on User Modeling, Adaptation and Personalization. pp. 34–45. ACM, Limassol Cyprus (Jun 2023). https://doi.org/10.1145/3565472.3592956
7. de Oliveira, R.M.B., Sörensen, K., Martens, D.: A model-agnostic and data-independent tabu search algorithm to generate counterfactuals for tabular, image, and text data. European Journal of Operational Research (2023). https://doi.org/10.1016/j.ejor.2023.08.031
8. Drozdal, J., Weisz, J., Wang, D., Dass, G., Yao, B., Zhao, C., Muller, M., Ju, L., Su, H.: Trust in automl: exploring information needs for establishing trust in automated machine learning systems. In: Proceedings of the 25th international conference on intelligent user interfaces. pp. 297–307 (2020)
9. Franke, L., Weidele, D.K.I., Dehmamy, N., Ning, L., Haehn, D.: AutoRL X: Automated Reinforcement Learning on the Web. ACM Transactions on Interactive Intelligent Systems **14**(4), 1–30 (Dec 2024). https://doi.org/10.1145/3670692
10. Garouani, M., Bouneffa, M.: Unlocking the black box: Towards interactive explainable automated machine learning. In: Quaresma, P., Camacho, D., Yin, H., Gonçalves, T., Julian, V., Tallón-Ballesteros, A.J. (eds.) Intelligent Data Engineering and Automated Learning – IDEAL 2023. pp. 458–469. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-48232-8_42
11. Khan, M.S., Peng, T., Akhlaq, H., Adeel Khan, M.: Comparative analysis of automated machine learning for hyperparameter optimization and explainable artificial intelligence models. IEEE Access **13**, 84966–84991 (2025). https://doi.org/10.1109/ACCESS.2025.3566427

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19

12. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4768–4777. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017). https://doi.org/10.48550/arXiv.1705.07874
13. Misitano, G., Afsar, B., Lárraga, G., Miettinen, K.: Towards explainable interactive multiobjective optimization: R-XIMO. Autonomous Agents and Multi-Agent Systems **36**(2), 43 (Oct 2022). https://doi.org/10.1007/s10458-022-09577-3
14. Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., Bischl, B.: Explaining hyperparameter optimization via partial dependence plots. In: Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J.W. (eds.) Advances in Neural Information Processing Systems. vol. 34, pp. 2280–2291. Curran Associates, Inc. (2021), https://proceedings.neurips.cc/paper_files/paper/2021/file/12ced2db6f0193dda91ba86224ea1cd8-Paper.pdf
15. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. p. 607–617. FAT* '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3372850
16. Mu, T., Wang, H., Tang, H., Shao, X.: ShrinkHPO: Towards Explainable Parallel Hyperparameter Optimization. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE). pp. 4897–4910. IEEE, Utrecht, Netherlands (May 2024). https://doi.org/10.1109/ICDE60146.2024.00371
17. Van Stein, N., Vermetten, D., V. Kononova, A., Bäck, T.: Explainable Benchmarking for Iterative Optimization Heuristics. ACM Transactions on Evolutionary Learning and Optimization **5**(2), 1–30 (Jun 2025). https://doi.org/10.1145/3716638
18. Vardhan, N.J., Chandana, D., Dheepak Raaj, R., Shanmukhi, S., Radhakrishnan, A.: A Comparative Study of Hyperparameter Tuning in Deep Learning Models using Bayesian Optimization and XAI. In: 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). pp. 1–6. IEEE, Kamand, India (Jun 2024). https://doi.org/10.1109/ICCCNT61001.2024.10725868
19. Wang, D., Andres, J., Weisz, J.D., Oduor, E., Dugan, C.: AutoDS: Towards Human-Centered Automation of Data Science. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. pp. 1–12. ACM, Yokohama Japan (May 2021). https://doi.org/10.1145/3411764.3445526
20. Weidele, D.K.I., Afzal, S., Valente, A.N., Makuch, C., Cornec, O., Vu, L., Subramanian, D., Geyer, W., Nair, R., Vejsbjerg, I., Marinescu, R., Palmes, P., Daly, E.M., Franke, L., Haehn, D.: AutoDOViz: Human-Centered Automation for Decision Optimization. In: Proceedings of the 28th International Conference on Intelligent User Interfaces. pp. 664–680. ACM, Sydney NSW Australia (Mar 2023). https://doi.org/10.1145/3581641.3584094
21. Yates, W.B., Keedwell, E.C., Kheiri, A.: Explainable Optimisation through Online and Offline Hyper-heuristics. ACM Transactions on Evolutionary Learning and Optimization **5**(2), 1–29 (Jun 2025). https://doi.org/10.1145/3701236
22. Zöller, M.A., Titov, W., Schlegel, T., Huber, M.F.: XAutoML: A Visual Analytics Tool for Understanding and Validating Automated Machine Learning. ACM Transactions on Interactive Intelligent Systems **13**(4), 1–39 (Dec 2023). https://doi.org/10.1145/3625240

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29918-5_19