Counterfactual Guidance for Transparent Hyperparameter Tuning

Sabri Manai, Szymon Bobek, and Grzegorz J. Nalepa

AIRA – 08/01/2026

UNIVERSITY

GEIST Research Group

We are GEIST. We dream big and work hard.

Outlines

1. PEER Project
2. Proditec Case Progress
3. Counterfactual Guidance for Transparent Hyperparameter Tuning
4. Case Study: Human-Controlled Counterfactual Tuning

PEER is about Human-centric Artificial Intelligence in Practice, and Bidirectional Communication between AI Agent and User

Use cases:

- Amsterdam Smart City Navigation (AMS: amsterdam.nl/innovatie)
- Indoor Shopping (Sonae: https://sonae.pt/)
- Industrial Machines Configuration (Proditec: proditec.com)
- Optimization of warehouse operations (Datacation: datacation.nl)

Our focus is on XAI, Personalization & Preference Modeling and Industrial AI

Timeframe: from 2023/10 to 2027/09

This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101120406.

More on PEER

Partners Exploring Human-centric Artificial Intelligence

- VRIJE UNIVERSITEIT BRUSSEL
- ALPHA CONSULTANTS S.R.L.
- FUJITSU SERVICES GMBH
- CENTRE AQUITAIN DES TECHNOLOGIES DE L'INFORMATION ET ELECTRONIQUES
- INESC TEC - INSTITUTO DE ENGENHARIA DE SISTEMAS E COPT
- FUNDACIO EURECAT
- TECHNISCHE UNIVERSITEIT EINDHOVEN
- GEMEENTE AMSTERDAM
- SONAE
- PRODITEC
- DATACATION
- UNIVERZITA KARLOVA

More on PEER

Proditec

- PRODITEC is the European leading manufacturer of automated inspection machines for the pharmaceutical & minting industries.
- More than 500 visual inspection systems in operation in 60 countries.
- Easy-to-use and intuive equipment designed to simplify the life of operators, maintenance and quality managers.

Proditec – Detecting Faulty Coins

Build a human-in-the-loop AI system to guide the operators of coin sorting machines in the configuration of the ML algorithms for detecting faulty coins.

GOOD

BAD

Challenges in Visual Anomaly Detection at Proditec

- Few defective coins, evolving defect types and changing production conditions make it hard to build representative training data.
- Model configuration requires coordinated choices about data collection, labelling strategy, algorithm selection and hyperparameters.
- Non-expert users need guidance to understand false positives/false negatives and to know how to adjust the system when quality requirements or conditions change.

Counterfactual Guidance for Transparent Hyperparameter Tuning

Where Our PDT Research Fits

- Within VXAD, this work focuses on operator-centred configuration: exposing high-level “soft knobs” (risk preference, data usage, decision speed...) instead of raw hyperparameters.
- Surrogate models and counterfactual explanations are used later in the talk to propose a small set of alternative settings that respect these soft constraints.
- For PEER, the long-term goal is an interactive tool where operators query “what-if” configurations and safely adapt the Proditec anomaly-detection pipeline as products and conditions evolve.

Counterfactual Guidance for Transparent Hyperparameter Tuning

2. Clearer explanations of system decisions – Hyperparameter optimization process.

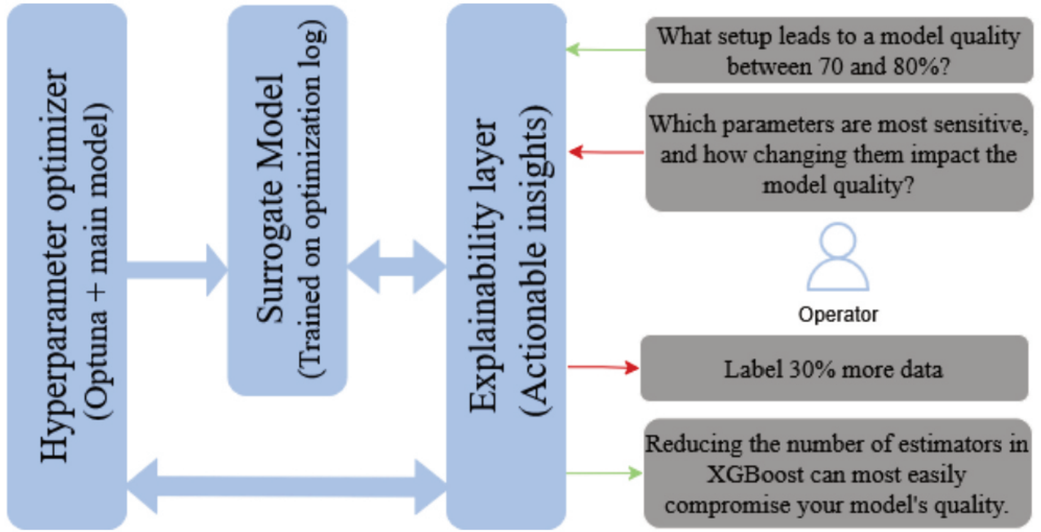

Fig 1: Workflow of the explainable hyperparameter optimization approach.

Counterfactual Guidance for Transparent Hyperparameter Tuning

2. Clearer explanations of system decisions – Hyperparameter optimization process.

In this work, we introduce a novel framework for explainable hyperparameter optimization that significantly enhances interpretability and actionability

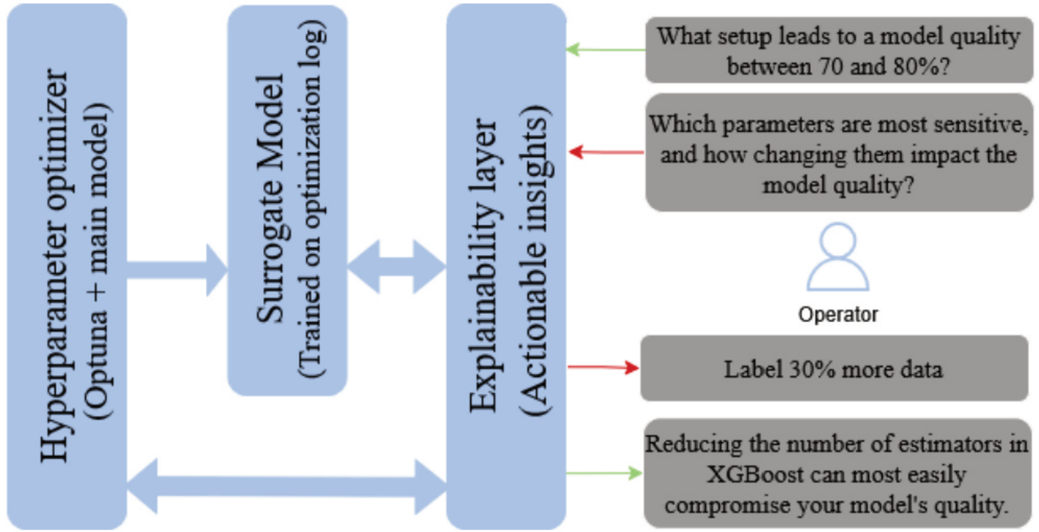

Fig 1: Workflow of the explainable hyperparameter optimization approach.

Counterfactual Explanations for Actionable Hyperparameter Optimization

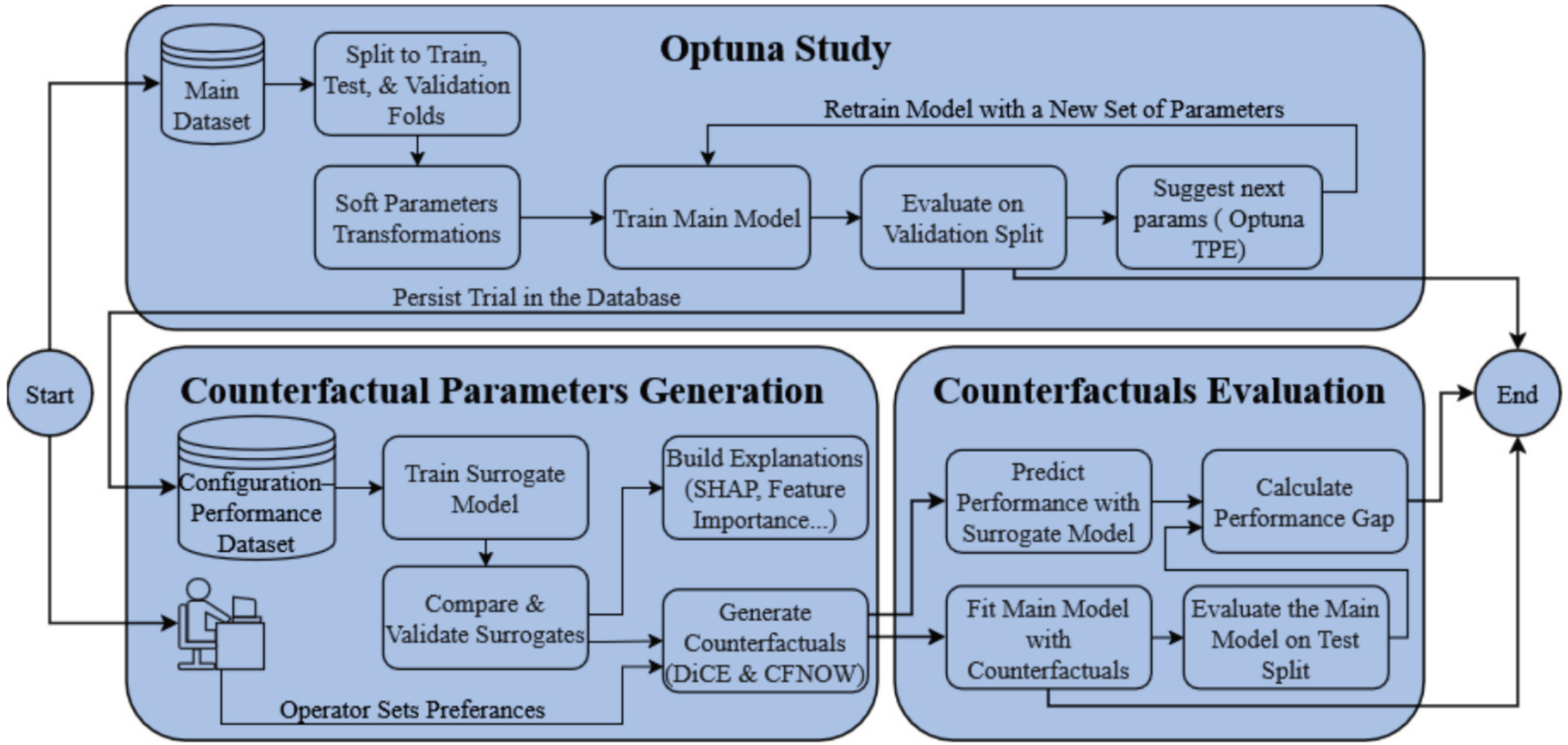

Fig 2: End-to-end pipeline for counterfactual-based hyperparameter optimization.

Search Space for Soft and Model's Parameters

Two different sets of parameters are optimized:

1. Model Hyperparameters: These are model specific parameters; in this example the parameters; in this example, the optimized model is XGBoost.
2. A set of Soft Parameters that could be generalized to any machine learning problem.

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, 0.95]	uniform
selftrain_max_iter	integer	[5, 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`.

Table 1: Search space specification for XGBoost hyperparameters and soft human-controllable knobs.

Derived Soft Parameter 1: Number of Samples

GOAL: Capture the *effort* level in terms of how much data is actually used.

START POINT: $N_{(total\ samples)}$ the size of the total training split.
$\rho \in (0, 1]$ the used fraction for the actual

DEFINITION: $$N_{\text{samples}}^{\text{training}} = \text{round}\left(\rho\, N_{\text{samples}}^{\text{total}}\right)$$

INTERPRETATION: Low ρ → fewer rows → lower preprocessing and compute efforts

High ρ → more rows → higher effort but potentially higher quality

Exposed to users as: “How many samples did we effectively train on?”

Derived Soft Parameter 2: Decision-Speed Category

GOAL: summarise model complexity
as a proxy for decision speed.

COMPLEXITY PROXY: $c = \texttt{max\_depth} \cdot \texttt{n\_estimators}$

On the optimization log (per dataset),
we compute empirical tertiles: q0.33, q0.66 for c

LABELING RULE: c < q0.33 → fast

q0.33 ≤ c < q0.66 → balanced

c ≥ q0.66 → slow

INTERPRETATION: Hardware-agnostic notion of speed:
“Is this configuration light, medium, or heavy to run?”

Derived Soft Parameter 3: Risk-Preference Label

GOAL: Encode the **precision vs. recall** attitude implied by imbalance handling.

WE COMBINE: scale_pos_weight

Resampling strategy r ∈ {none = 0, undersample = 0.5, oversample = 1}

RISK SCORE: $s = \log(\texttt{scale\_pos\_weight}) + r$

LABELING RULE: (margin τ=0.2)

$|s| < 0.2 \rightarrow$ balanced

$s > 0.2 \rightarrow$ recall_pref

$s < -0.2 \rightarrow$ precision_pref

INTERPRETATION: Single, human-readable tag summarising how aggressively positives are treated(recall-oriented, precision-oriented, or neutral).

Obtained F1 Distribution Training XGB on 4 Datasets

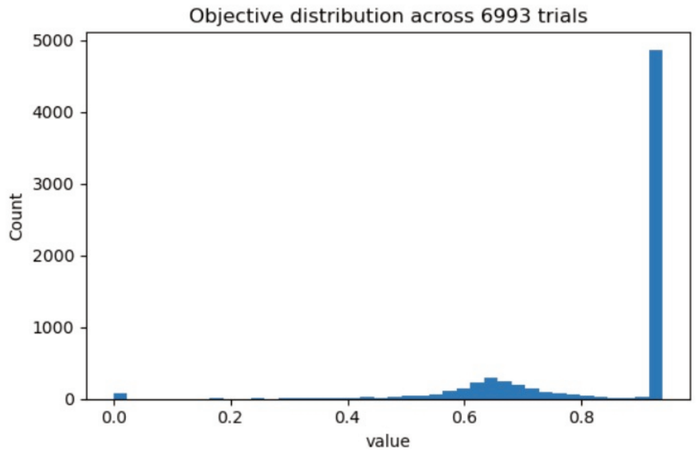

Fig 3: Distribution Training on Adult Dataset

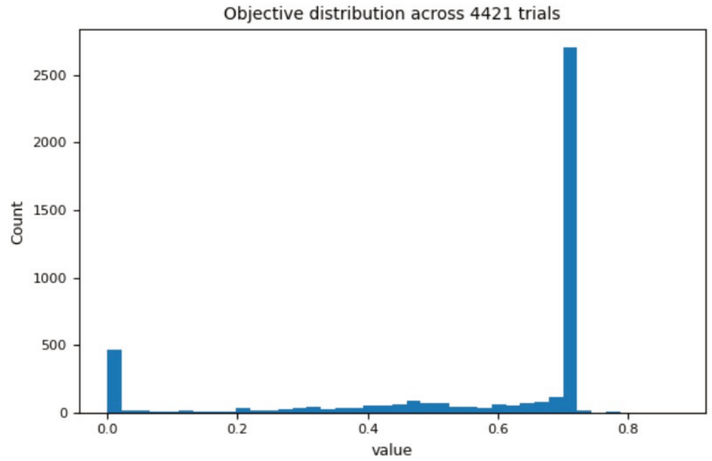

Fig 4: Distribution Training on Bank Marketing Dataset

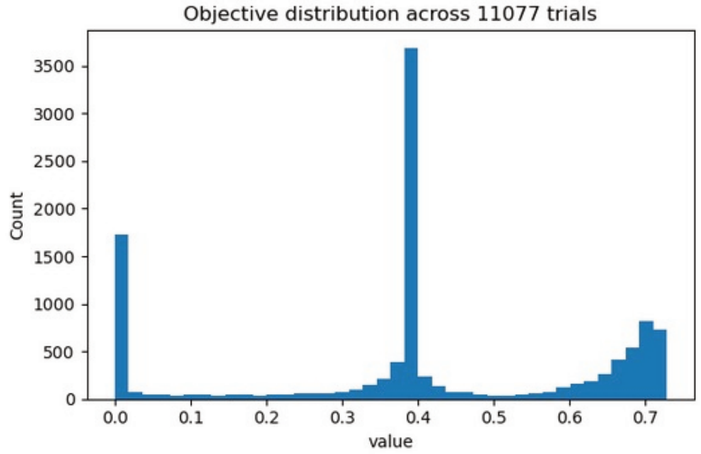

Fig 5: Distribution Training on Breast Cancer Dataset

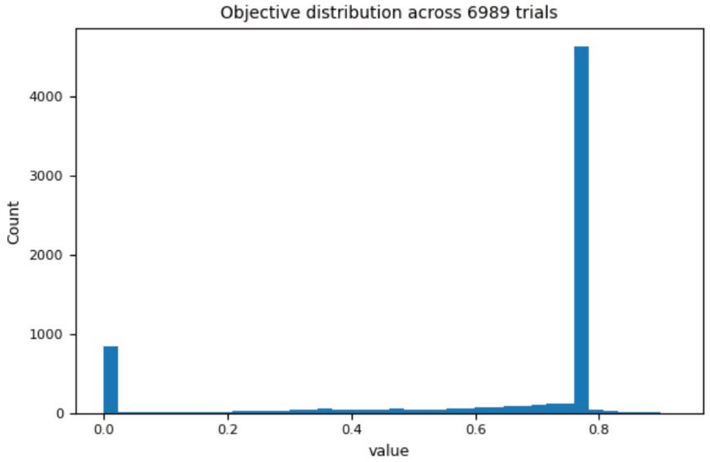

Fig 6: Distribution Training on Phishing Websites Dataset

Surrogate Models' Performance

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

Table 2: Top five surrogate models across all four datasets (validation split).

Across datasets, CatBoost generally provides the strongest predictive performance, while ExtraTrees attains the highest performance on the Breast Cancer dataset.

Catboost Predictions vs True Values

20% of the obtained configuration/performance dataset was used for validation

Results proved to be consistent, however, data imbalance towards high-performing configuration could be noisy prediction.

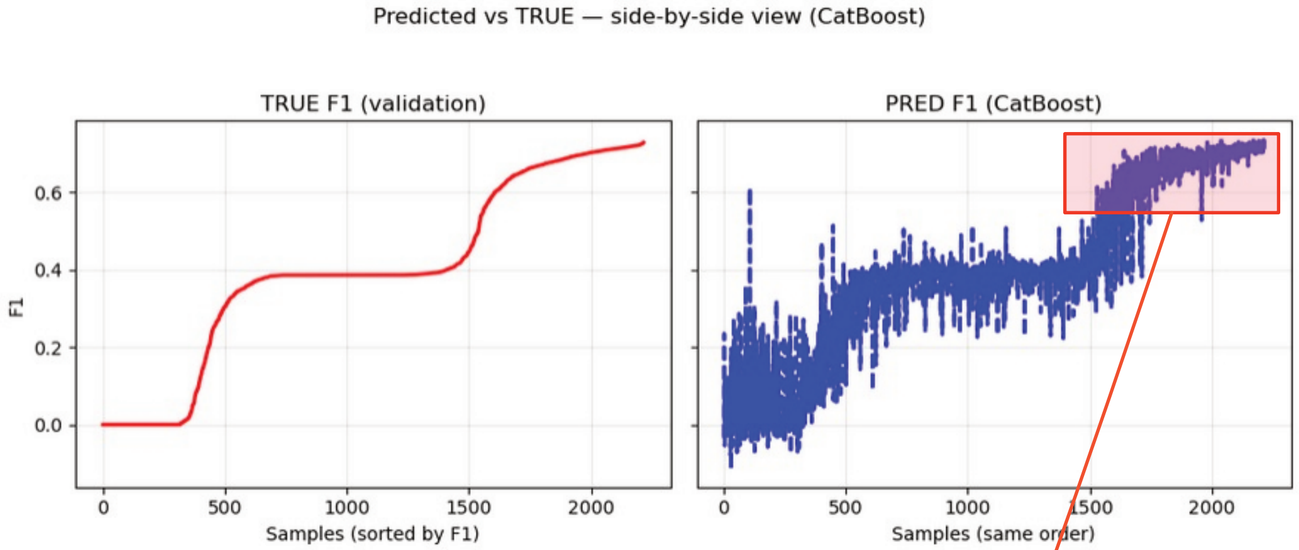

Fig 7: True and predicted performance comparison

We can notice that the noise decreases for samples with F1> 0.6

SHAP Beeswarm for the CatBoost surrogate on the Bank Marketing

Resampling strategy (soft) has the largest global impact on F1.

Among hard hyperparameters, the strongest effects come from *alpha, gamma, subsample, lambda, and n_estimators.*

Semi-supervised controls *(labeled_ratio and selftrain_threshold)* also appear high in the ranking.

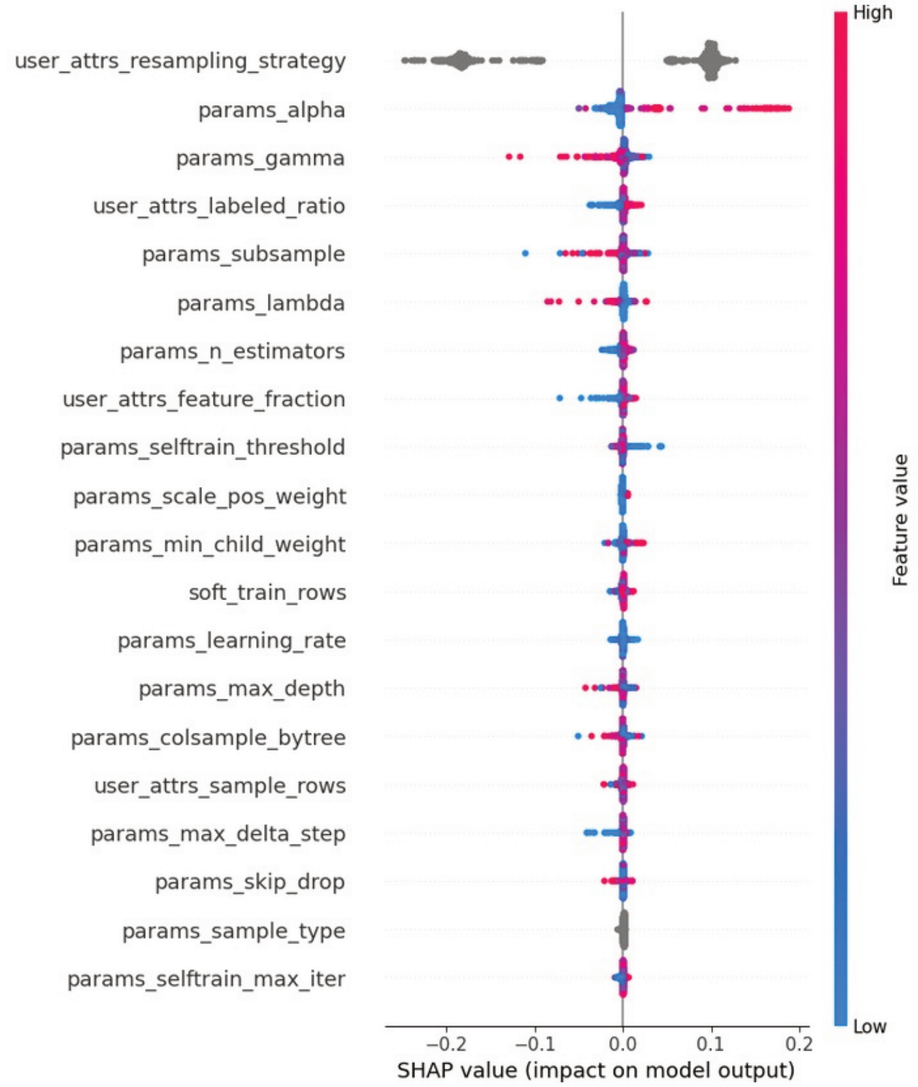

Fig 8: SHAP Beeswarm plot for surrogate on the Bank Marketing

CatBoost's global importance confirms the SHAP analysis

Resampling_strategy is by far the dominant knob for Bank Marketing, followed by *alpha, gamma, and subsample.*

The remaining model's and soft parameters have much smaller contributions, which is consistent with their narrow SHAP ranges.

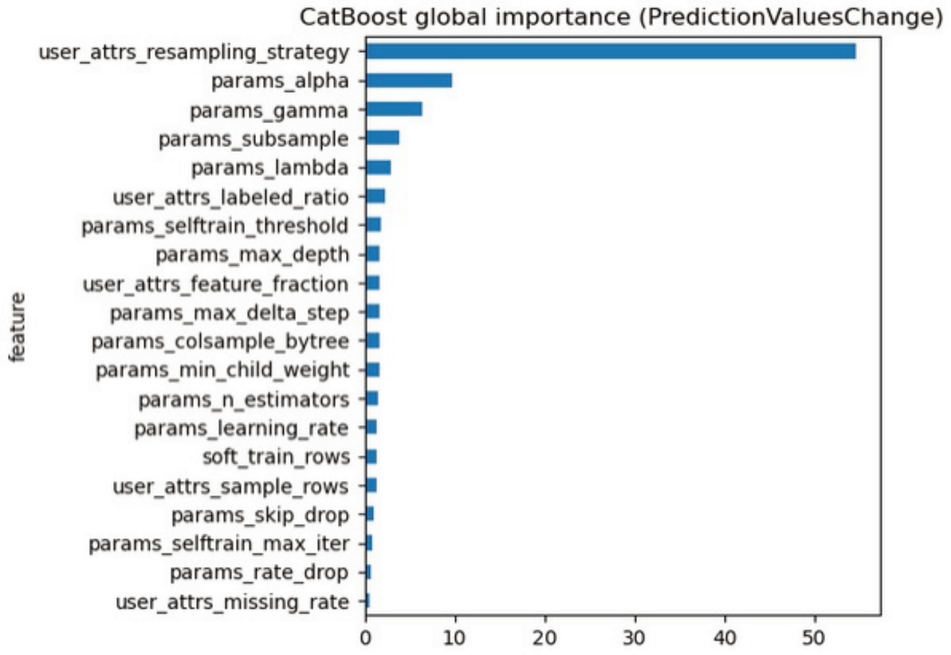

Fig 9: CatBoost global importance on the Bank Marketing

Counterfactual Generation

INPUTS: User preferences (risk, speed, sample_rows...).
Performance target range.
Trained surrogate model (CatBoost regressor).

STEP 1: Select factual configuration

STEP 2: Generate counterfactuals (DICE, CFNOW)

STEP 3: Validate counterfactuals (2 steps validation)

Counterfactual Quality: Surrogate vs. Main Model

To assess the full pipeline, we compare three quantities per dataset and explainer:

1. First, the surrogate's predicted F1
2. Second, the test F1 obtained after refitting XGBoost with the counterfactual hyperparameters
3. Third, the mean F1 of the generated counterfactual configurations.

For each setting, the reported values are averages over 10 counterfactual configurations

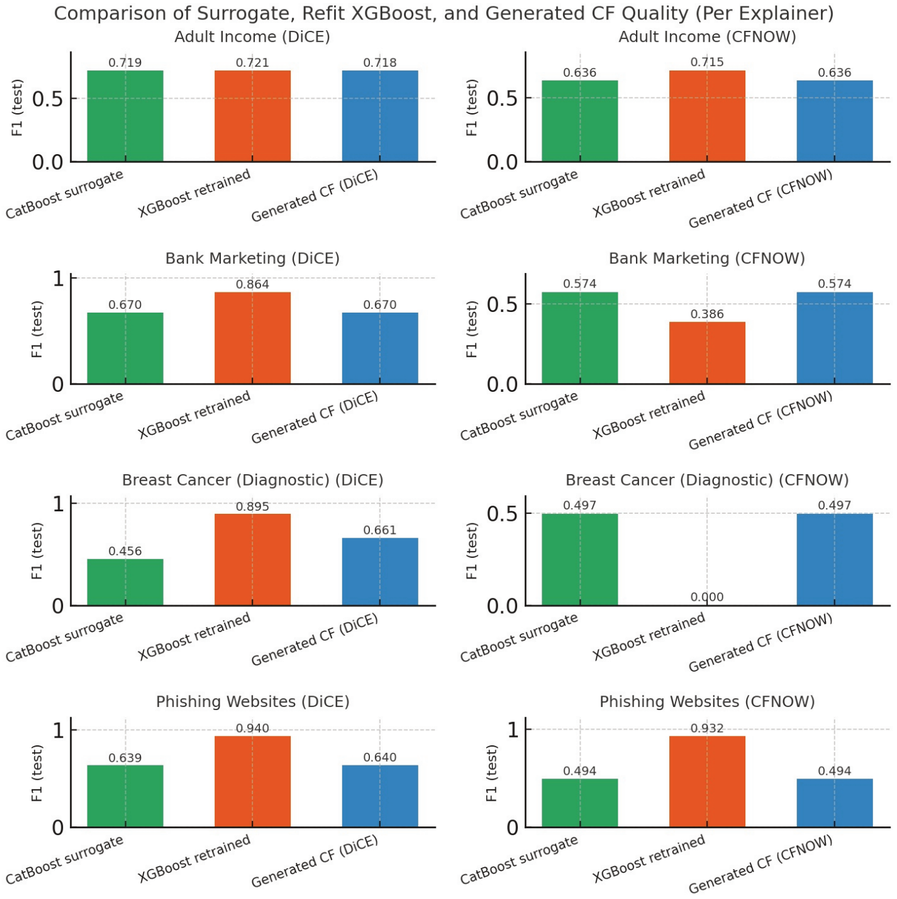

Fig 10: Counterfactual quality across datasets for DiCE and CFNOW.

Case Study: Human-Controlled Counterfactual Tuning

Operator's preferences:

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

Table 3: Soft-control profile for the Adult Income case study.

Risk & speed: recall-oriented, medium-complex model.

Data effort: almost all rows and labels used (high-effort setting).

Step 1 – Select factual configuration

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 0.72 and serves as the starting point for counterfactual exploration.

Step 2 – Generated Counterfactual Instance

Under the operator's constraints, DiCE returns 16 counterfactual configurations.

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			

Table 4: Example counterfactual configuration (DiCE).

Step 3 – Validate counterfactuals

Re-train using 16 DiCE counterfactuals as parameters 5 times, and aggregating test F1.

Points cluster near the y = x diagonal
→ surrogate is locally faithful around the seed.

Moderate positive association:
Pearson r ≈ 0.53 including factual seed
and 0.51 for only CFs.

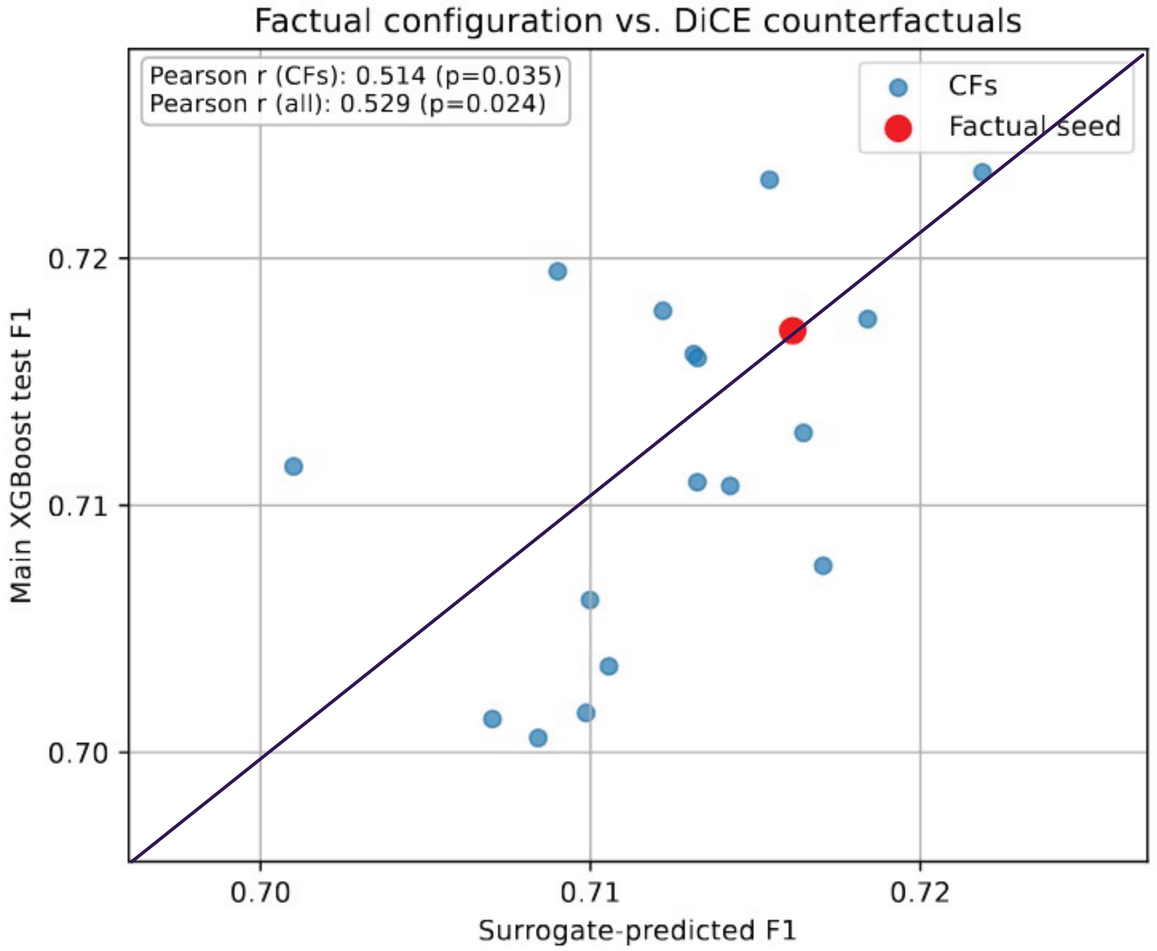

Fig 11: Surrogate-predicted versus realised test F1 for the factual configuration and its counterfactual neighbours on Adult Income.

Challenges and limitations

1. **Log granularity**

   TPE concentrates near extremes; only Adult yielded a well-distributed configuration–performance log, improving surrogate quality.

2. **Stochasticity from soft knobs**

   High missing-rate and low sampling fraction add noise, degrading both main-model scores and surrogate fit.

3. **Scope and CF trade-off**

   Pipeline is XGBoost-specific and generalizing this to include several models requires high computing resources.

Challenges and limitations

1. **Log granularity**

   TPE concentrates near extremes; only Adult yielded a well-distributed configuration–performance log, improving surrogate quality.

2. **Stochasticity from soft knobs**

   High missing-rate and low sampling fraction add noise, degrading both main-model scores and surrogate fit.

3. **Scope and CF trade-off**

   Pipeline is XGBoost-specific and generalizing this to include several models requires high computing resources.

Challenges and limitations

1. **Log granularity**

   TPE concentrates near extremes; only Adult yielded a well-distributed configuration–performance log, improving surrogate quality.

2. **Stochasticity from soft knobs**

   High missing-rate and low sampling fraction add noise, degrading both main-model scores and surrogate fit.

3. **Scope and CF trade-off**

   Pipeline is XGBoost-specific and generalizing this to include several models requires high computing resources.

Challenges and limitations

1. **Log granularity**

   TPE concentrates near extremes; only Adult yielded a well-distributed configuration–performance log, improving surrogate quality.

2. **Stochasticity from soft knobs**

   High missing-rate and low sampling fraction add noise, degrading both main-model scores and surrogate fit.

3. **Scope and CF trade-off**

   Pipeline is XGBoost-specific and generalizing this to include several models requires high computing resources.

Future work

Extend beyond XGBoost

Adapt the pipeline to additional model families by collecting new configuration–performance logs with the same procedure.

XAI-guided sampling

Leverage surrogate explanations during search.

LLM summarisation of CFs

Convert counterfactual configurations into concise natural-language summaries to support non-expert interpretation.

Future work

Extend beyond XGBoost

Adapt the pipeline to additional model families by collecting new configuration–performance logs with the same procedure.

XAI-guided sampling

Leverage surrogate explanations during search.

LLM summarisation of CFs

Convert counterfactual configurations into concise natural-language summaries to support non-expert interpretation.

Future work

Extend beyond XGBoost

Adapt the pipeline to additional model families by collecting new configuration–performance logs with the same procedure.

XAI-guided sampling

Leverage surrogate explanations during search.

LLM summarisation of CFs

Convert counterfactual configurations into concise natural-language summaries to support non-expert interpretation.

Future work

Extend beyond XGBoost

Adapt the pipeline to additional model families by collecting new configuration–performance logs with the same procedure.

XAI-guided sampling

Leverage surrogate explanations during search.

LLM summarisation of CFs

Convert counterfactual configurations into concise natural-language summaries to support non-expert interpretation.

Key takeaways

Extend beyond traditional model's hyperparameter optimization

Operators set soft preferences, the system proposes hard alternatives (actionable tuning without exposing full search complexity).

Transparent guidance

Surrogate + explanations provide a fast “what-if” layer over the optimisation log to support decision-making.

Practical validation loop

Counterfactuals are verified by re-training, providing evidence for local trust in the recommendations.

Key takeaways

Extend beyond traditional model's hyperparameter optimization

Operators set soft preferences, the system proposes hard alternatives (actionable tuning without exposing full search complexity).

Transparent guidance

Surrogate + explanations provide a fast “what-if” layer over the optimisation log to support decision-making.

Practical validation loop

Counterfactuals are verified by re-training, providing evidence for local trust in the recommendations.

Key takeaways

Extend beyond traditional model's hyperparameter optimization

Operators set soft preferences, the system proposes hard alternatives (actionable tuning without exposing full search complexity).

Transparent guidance

Surrogate + explanations provide a fast “what-if” layer over the optimisation log to support decision-making.

Practical validation loop

Counterfactuals are verified by re-training, providing evidence for local trust in the recommendations.

Key takeaways

Extend beyond traditional model's hyperparameter optimization

Operators set soft preferences, the system proposes hard alternatives (actionable tuning without exposing full search complexity).

Transparent guidance

Surrogate + explanations provide a fast “what-if” layer over the optimisation log to support decision-making.

Practical validation loop

Counterfactuals are verified by re-training, providing evidence for local trust in the recommendations.

Thank you!

Email: sabri.manai@uj.edu.pl

JAGIELLON IN KRAKÓW UNIVERSITY

peer

GEIST Research Group

We are GEIST. We dream big and work hard.