Operationalising Active Learning for Power Grid Security Assessment: A Human-in-the-Loop Retraining Service for a Digital Twin Platform

Gašper Leskovec
leskovecg@gmail.com
Jožef Stefan Institute
Ljubljana, Slovenia

Costas Mylonas
kmylonas@ubitech.eu
UBITECH
Athens, Greece

Klemen Kenda
klemen.kenda@ijs.si
Jožef Stefan Institute
Ljubljana, Slovenia

Abstract

Active learning (AL) can reduce the number of expensive N-1 contingency simulations needed to train data-driven power grid security classifiers, but the reported gains come almost entirely from offline studies on a static pool of pre-computed operating points, with no human in the loop. This paper reports the integration of an AL-based security assessment service into an operational digital twin platform, describing the deployed architecture, the dashboard–backend interface contract, and the resulting workflow: day-ahead candidates are ranked by informativeness, the operator selects which of them justify simulation, and the labels are produced by the digital twin rather than by the human. The loop has so far been exercised manually — 31 retraining runs and 91 ranking calls over twelve distinct day-ahead sets, with no scheduled operation — so what we report is system behaviour and not label efficiency. Under a temporal split, aggregate discrimination is nearly unchanged while recall on the insecure class falls by ten points at the fixed decision threshold; the three uncertainty-based query strategies coincide in the binary setting, so an interface offering four options realises two behaviours; and the top of the ranking reorders under retraining that leaves the reported model quality unchanged.

Keywords

active learning, human-in-the-loop, digital twin, power grid security assessment, N-1 criterion, system integration

1 Introduction

Transmission grids are operated to withstand the loss of any single component [3]. Verifying this N-1 criterion is computationally demanding, since every candidate operating point requires a base-case power flow and one additional power-flow solution per credible contingency. Data-driven classifiers offer a fast alternative: trained on simulated operating points, they predict whether a state is secure at a fraction of the cost of simulating it [1]. The obstacle is the training data. Every label is the product of the same expensive simulations the classifier is meant to avoid.

Active learning (AL) reduces this cost by letting the model choose which points to label [6]. In the power system studies we are aware of, the reported gains are obtained offline, on a static pool, with a simulator queried automatically and no human between model and oracle. Day-ahead operation changes the problem: candidates arrive as a small daily set rather than a large pool, an operator decides which are worth simulating, and the model is retrained repeatedly over time.

This paper reports on that transition. We contribute a deployed architecture that turns an offline AL study into an operational component of a digital twin platform, comprising candidate ranking, simulator-based labelling and cumulative batch retraining; an interface contract for *operator-as-selector* human-in-the-loop AL, in which the human decides what is worth simulating while the digital twin provides the labels; and a report of what the recorded operation of that loop does and does not support. No label-efficiency experiment in deployment is reported: the loop has not yet accumulated the labels such an experiment requires.

2 Background and Case Study

2.1 Related work

In power systems AL has been applied to stability assessment [7] and, within a digital twin architecture, to day-ahead load forecasting with an operator-set query threshold [4]; we differ from the latter in that the task is security classification and the operator sets the simulation budget rather than a model parameter. Our previous work showed, on a digital twin of the Greek transmission network, that pool-based AL matches the classification quality of random sampling with roughly half the labelled operating points [2], and this paper carries that formulation into deployment.

2.2 Case study

The test system is a steady-state digital twin derived from the topology of the Greek transmission network (35 buses, 46 lines, 135 generators, 110 static generators, 20 loads), solved with AC load flow in `pandapower`. Each operating point is screened against 180 single-component contingencies: all line outages except one, and all generator outages. It is labelled *secure* if the base case and every contingency keep bus voltages within $[0.90, 1.10]$ p.u. and line loading below 100 %. A non-convergent power flow is also labelled insecure, and the label does not distinguish a violated limit from a non-convergent solution.

The reference table covers the 8 760 hours of 2023 and is approximately balanced (51.2 % secure). As delivered it held nine extra rows duplicating hours of 1 January, which we removed; seven were exact copies, but two carried conflicting labels for identical feature vectors. All figures below use the deduplicated table; full details of the offline study are in [2].

The classifier sees only the operating point: the active power set-point of every load, conventional generator and static generator, 265 variables, the same set used in [2]. Under a fixed topology and control configuration an operating point is specified by its injections and demands, so a classifier given those variables addresses the same problem the simulator does, from the same starting information; everything recorded during or after the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
Information Society 2026, Ljubljana, Slovenia
© 2026 Copyright held by the owner/author(s).
https://doi.org/10.70314/is.2026.sikdd.108

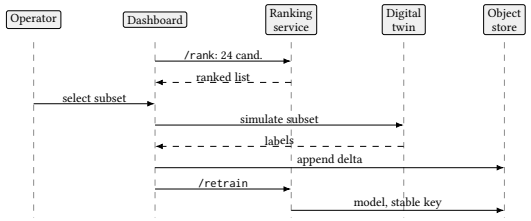


Figure 1: Deployed human-in-the-loop active learning workflow. The operator decides which candidates are simulated; labels come from the digital twin. The ranking service ranks from its own local copy of the current model, so no model is fetched from the store during a call.

simulation is withheld. The two conflicting rows show that the qualification is needed: identical vectors received different verdicts, implying a non-zero error floor. Aggregate outcomes such as worst line loading, against which the label is defined, are withheld; an early version trained on them reached perfect held-out accuracy, a symptom rather than a result.

3 Deployed System Architecture

3.1 Components

The system spans two organisations. The operator dashboard, the orchestration backend and the digital twin simulator are run by the industrial partner; a shared object store is the exchange layer; the ranking and retraining service runs at JSI as a containerised application (Figure 1). Neither side holds a copy of the other's system: they are coupled only through a small HTTP contract and agreed object keys, since the two codebases are released independently.

3.2 Service interface

The service exposes two operational endpoints behind a shared API key held as a deployment secret.

Candidate ranking. `POST /rank` receives the day-ahead candidates, one per forecast hour, each with a timestamp and the full 265-dimensional feature vector, together with the operator's chosen strategy; the dashboard supplies the vectors, and the service does not observe whether they are forecasts or replayed profiles. It ranks them against its local copy of the current model, returning each candidate with an informativeness score and a rank. Four strategies are exposed: least-confident, margin and entropy sampling, and a fourth, named `random` in the interface, which is in fact a deterministic passthrough assigning a constant score and so preserving submission order. Since candidates are submitted in clock order, taking the top k under that option returns the first k hours of the day, which is not a neutral sample. The name is a legacy of the offline study; it must not be read as the randomised baseline of [2], and renaming it is pending.

Retraining. `POST /retrain` retrieves the newly labelled rows from the object store, merges them with the base table held locally, refits a random forest of 400 trees with a fixed seed, and publishes the model with its metrics.

3.3 Engineering decisions

Two decisions were forced by deployment rather than design, and are the parts of this work most likely to transfer.

Delta-based exchange. Transferring the full training table every cycle proved unreliable: the shared object-store gateway timed out on larger payloads. Only new labelled rows are transferred; the base table is held locally and merged at training time, and the published model is likewise cached on local disk and served from there. The service is therefore not stateless, a trade-off between reproducibility and transfer reliability, and nothing pins the contents of the base table; a checksum would close that gap cheaply.

Logging, and what the operator controls. Every call is appended to a structured log with its configuration and outcome — for ranking, candidate timestamps and scores but not feature vectors; log writes are non-fatal. This record is the evidence base for Section 5. What the log does not contain is any operator-set model parameter: the operator controls how many candidates are simulated and nothing else, while thresholds, tree counts and seeds remain configuration owned by the ML engineer.

4 Operator-in-the-Loop Workflow

The role of the human here differs from the standard AL formulation. The operator does not assign labels but acts as a *selector* exercising operational judgement: the operator inspects the ranked day-ahead candidates and decides which justify the cost of a full N-1 analysis, after which the digital twin produces the reference label that is appended to the training data, so the labelling oracle remains a physics-based simulation. Because the human is not the oracle, a poor selection does not inject an incorrect label; it merely spends the budget on a less informative point, so the failure mode is inefficiency rather than contamination, which justifies giving the operator latitude. The selection is meant to sit alongside established day-ahead security assessment practice [3], the score being weighed against expected loading, planned outages and corridor flows; none of that context reaches the service today.

4.1 What an explanation can and cannot show

What the interface can usefully display is constrained by the model. We computed TreeSHAP attributions for the deployed forest over the 2 208-point temporal test set of Section 5. In aggregate, generator injections account for 61 % of the total mean absolute attribution, static generators for 33 % and loads for 6 %. Read per feature rather than per group, these shares mostly track group size: the three groups differ by less than a factor of two per variable. The model does not rest on a small set of dominant injections, and a global importance ranking offers the operator little that knowledge of the network does not supply.

The per-candidate picture is less conclusive than it first appears. The test point closest to the boundary has a predicted insecure probability of 0.500, which for a nearly balanced training set is about where the model's expected output lies, so attributions there sum to almost nothing by construction and their small, mixed-sign magnitudes follow from that choice of point rather than being evidence about the boundary; all it shows is that no single element was worth displaying, consistent with the aggregate result. Attribution is on this evidence better spent on framing: where a candidate sits relative to the rest of the day, and which candidates are tied with it.

Table 1: Randomised and temporal split of the deduplicated reference table ($n = 8\,760$), same seed and code path in a single execution. Brackets give 95 % Wilson intervals; ROC AUC has none, not being a proportion. The temporal split trains on January–September ($n = 6\,552$) and tests on October–December ($n = 2\,208$); the randomised split trains on 7 008 rows and holds out 1 752.

Metric	Randomised split	Temporal split
Accuracy	0.941 [0.929, 0.951]	0.910 [0.897, 0.921]
Recall (insecure)	0.939 [0.921, 0.953]	0.841 [0.818, 0.861]
Precision (insecure)	0.939 [0.921, 0.953]	0.981 [0.971, 0.988]
ROC AUC	0.985	0.978

5 Evaluation and Deployment Observations

5.1 Temporal generalisation

This subsection is an offline analysis of the 2023 reference table, not a measurement of the deployed loop. The offline study of [2] used a time-aware split, but the service reports metrics on a randomised hold-out, which is the usual convention and also the optimistic one: it interleaves training and test across the year, so the model is tested on hours whose neighbours it has seen, whereas a temporal split tests only on the period after a cut date, which is what deployment looks like. We evaluated the same model family under both regimes in one run (Table 1).

The accuracy loss is modest, but the composition of the errors changes: under the temporal split the model misses 181 of 1 135 insecure operating points while raising only 18 false alarms on 1 073 secure ones, whereas under the randomised split the two error types are exactly balanced (52 and 52). Recall falls by about ten percentage points, precision rises by four, and ROC AUC is nearly unchanged. We compare the regimes descriptively, through the intervals in Table 1, rather than with a two-sample test: the randomised hold-out is drawn from the whole year, so it overlaps the October–December test period, and the two proportions come from models fitted on different data — and on different amounts of it, 6 552 rows against 7 008.

The unchanged AUC is consistent with the posterior ordering largely surviving; what degrades is performance at the fixed threshold, where the model becomes systematically reluctant to flag the insecure class and so errs in the operationally unfavourable direction. Two mechanisms can produce this and the data do not separate them: drift in the relation between injections and security, and a shift in the class prior, the training period being 47.9 % insecure against 51.4 % in the test period. Both metrics are reported at the default threshold of 0.5, which the service uses and which we did not tune. Under this single cut the loss is attributable to threshold placement rather than to a degraded ranking, so the cheaper remedy is recalibration on a recent labelled window rather than retraining; that window is not free of labels, but it is far smaller than a retraining set.

The deployed model is fitted on 2023, while the candidates it ranks are 2026 hours without digital twin labels, so its accuracy there cannot be measured. Its own confidence gives a label-free indication: over the 288 candidates of the twelve 2026 sets, the mean posterior of the predicted class is 0.78 and half fall below 0.8, against an accuracy of 0.94 on held-out 2023 hours. A calibrated model with that confidence would be about 0.78 accurate; forests are not calibrated by construction, and the service cannot tell forecasts from replayed profiles, so this indicates rather than measures a gap.

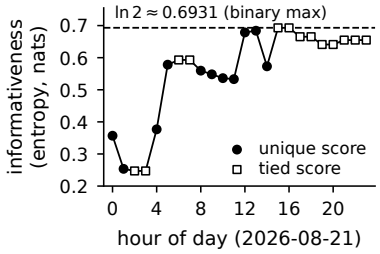


Figure 2: Informativeness scores returned by `/rank` for the twenty-four candidate hours of 2026-08-21, in clock order rather than rank order. Open markers denote tied candidates; the dashed line is the maximum $\ln 2$ for a binary classifier.

5.2 Ranking behaviour on full-day candidate sets

The uncertainty strategies coincide. Least-confident, margin and entropy sampling are distinct in the multi-class case and are offered as distinct choices in the interface, inherited from the option set of the offline study [2]; for binary classification they are not, a reduction already noted in the AL literature [5, 6]. Writing $q = \max(p, 1-p) \in [0.5, 1]$ for the posterior of the predicted class, and negating the margin so all three are maximised,

$$s_{\mathrm{LC}} = 1 - q, \qquad s_{\mathrm{M}} = -(2q - 1), \qquad s_{\mathrm{H}} = H(q), \tag{1}$$

with H the binary entropy in nats. Each is a strictly decreasing function of q alone, so the three induce exactly the same ordering of any candidate set, including the same ties. The offline study observed learning curves lying almost on top of each other, with a small residual gap [2]; it follows from (1) that any such gap originates outside the query policy — in tie resolution, in rounding of the score transform, or in non-determinism in the learner. What deployment adds is the operational consequence: four options realise two behaviours, a single uncertainty ordering and the passthrough. Widening the strategy space requires leaving the uncertainty family altogether.

Scores are well spread but coarsely quantised. One might expect a model fitted on several thousand operating points to assign nearly identical posteriors to all twenty-four hours of a day, leaving nothing to order. It does not. Across the twelve full-day sets the within-day range of scores lies between 0.331 and 0.637 nats, and on seven of them the top candidate comes within 0.001 of $\ln 2 \approx 0.6931$. The scores are, however, coarsely quantised: a day of 24 candidates resolves into only 16 to 21 distinct values, in two to six groups of exact ties, which reach the top three on three of the twelve days. With trees grown to purity the forest averages leaf proportions of 0 or 1 and so resolves posteriors on a grid of $1/400$, and 178 of the 182 distinct entropy values returned lie exactly on it.

Top candidates cluster, but not in a fixed part of the day. Figure 2 plots one call in clock order. Across the twelve full-day sets, the five top-ranked hours lie on average 4.4 h apart in clock distance, against 6.3 h for five hours drawn at random (permutation test, $p < 0.001$). Which hours are favoured, however, varies: pairwise Spearman correlations between the hourly orderings of different days range from −0.84 to +0.87, with a median of +0.27, so the demand-peak shape of the day in Figure 2 does not recur reliably. Informativeness is assessed per candidate within a single day,

with no term rewarding diversity, so a budget spent on top-ranked candidates would draw labels repeatedly from a narrow band of adjacent hours.

The ranking is not stable under retraining. One 24-hour candidate set was ranked repeatedly over a morning with eight retraining runs interleaved, during which the training table grew by 13 rows out of about 8 800. The five top hours at the first and the last call have none in common (Spearman +0.05), while the accuracy and ROC AUC of the successive models varied within one percentage point. With one small unexplained exception, calls with no data-changing retraining between them returned identical scores, so the change comes from the refitted model, although with input vectors unlogged an input change is not strictly excluded. The operator can read all 24 candidates; what the ranking offers is guidance on spending a limited simulation budget, and its top is fragile enough that near-tied candidates are better shown as a group and the ranking regenerated after every refit.

5.3 Scale of the deployed loop

In operation since late June 2026, the service has recorded 31 retraining runs and 91 ranking calls, 89 of them on a full twenty-four-hour candidate set, covering twelve distinct sets from January, August and September 2026. All were issued manually during integration testing: the dashboard exposes the ranking step but does not yet call it on a day-ahead schedule, and the 36 rows appended over the whole deployment were added while exercising the interface rather than through operator-driven simulation. This bounds every claim made above: each step behaves as specified under real inputs, but a few dozen rows on a base of several thousand cannot move the model's aggregate quality, and run-to-run variation in the reported metrics reflects the resampled hold-out rather than anything learned. The frequency of operator interaction, not the ranking policy, is the binding constraint.

6 Limitations and Future Work

What we report is system behaviour under real inputs, so no claim about label efficiency in deployment follows from it. The temporal comparison is a single train/test cut of one year and does not separate drift from the seasonal prior shift; a rolling-origin evaluation would. The 2023 to 2026 gap is bounded in Section 5.1 through the model's confidence but not measured, which would require labels for 2026 hours. The label records only the verdict, not whether an insecure outcome followed from a violated limit or a non-convergent solution, so the classifier is in part fitted to solver behaviour. The explanation results of Section 4 rest on one boundary candidate and on aggregate attributions, not on a study with operators. The forest uses library defaults apart from 400 trees and a fixed seed; the grid model and the reference table are partner-owned and not public.

Within the uncertainty family there is nothing left to compare in the binary case, so committee-based or diversity-aware batch selection is the only direction that changes the ordering, and the clustering of top candidates makes it the more urgent. Exact ties at the top are resolved by submission order; a principled tie-break, or presenting ties as an unordered group, would avoid conveying a precision the score lacks. Threshold recalibration on a recent window and risk-aware selection are the remaining directions, the latter requiring per-sample grid metadata from the twin. Finally, the case study is a single 35-bus system with single-component contingencies; larger topologies, N-k sets and dynamic assessment all raise the cost of a label and strengthen the case for selective simulation.

7 Conclusion

We integrated pool-based active learning for N-1 security assessment into an operational digital twin platform, with the operator selecting the day-ahead points to simulate and the twin producing the labels. The loop has so far been driven manually, so what we report is its behaviour rather than its label efficiency. Under a temporal split, aggregate discrimination is nearly unchanged while recall on the insecure class falls at the fixed threshold, so the randomised hold-out the service reports on is the optimistic case and the threshold is what most needs periodic review. The three uncertainty strategies induce the same ordering in the binary case, so an interface offering four of them offers two behaviours, one of which simply returns the first hours of the day. And the top of the ranking clusters in adjacent hours and reorders under retraining that leaves aggregate quality unchanged, so a fixed budget needs a diversity term and the ranking should be regenerated after every refit. The next steps are threshold recalibration, diversity-aware selection, a principled treatment of ties, and scheduled day-ahead operation, so that label efficiency can be evaluated under operational conditions.

Acknowledgments

This work was supported by the EU-funded HumAIne project [grant number 101120218]. The authors acknowledge the use of LLMs for language editing; all conceptual frameworks, analyses and interpretations remain the sole responsibility of the authors.

References

[1] José-María Hidalgo Arteaga, Fiodar Hancharou, Florian Thams, and Spyros Chatzivasileiadis. 2019. Deep Learning for Power System Security Assessment. In *2019 IEEE Milan PowerTech*. IEEE, 1–6. doi:10.1109/PTC.2019.8810906

[2] Gašper Leskovec, Costas Mylonas, and Klemen Kenda. 2025. Active Learning for Power Grid Security Assessment: Reducing Simulation Cost with Informative Sampling. In *Proceedings of the 28th International Multiconference Information Society – SiKDD 2025*. Ljubljana, Slovenia. doi:10.70314/is.2025.sikdd.11

[3] Kip Morison, Lei Wang, and Prabha Kundur. 2004. Power System Security Assessment. *IEEE Power and Energy Magazine* 2, 5 (2004), 30–39. doi:10.1109/MPAE.2004.1338124

[4] Costas Mylonas, Titos Georgoulakis, and Magda Foti. 2024. Facilitating AI and System Operator Synergy: Active Learning-Enhanced Digital Twin Architecture for Day-Ahead Load Forecasting. In *2024 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, 1–6. doi:10.1109/SEST61601.2024.10693980

[5] Vu-Linh Nguyen, Mohammad Hossein Shaker, and Eyke Hüllermeier. 2022. How to Measure Uncertainty in Uncertainty Sampling for Active Learning. *Machine Learning* 111, 1 (2022), 89–122. doi:10.1007/s10994-021-06003-9

[6] Burr Settles. 2009. *Active Learning Literature Survey*. Technical Report 1648. University of Wisconsin–Madison.

[7] Zhongtuo Shi, Wei Yao, Yong Tang, Xiaomeng Ai, Jinyu Wen, and Shijie Cheng. 2024. Intelligent Power System Stability Assessment and Dominant Instability Mode Identification Using Integrated Active Deep Learning. *IEEE Transactions on Neural Networks and Learning Systems* 35, 7 (2024), 9970–9984. doi:10.1109/TNNLS.2023.3238168