INTERPRETABLE EARLY WARNING OF STATISTICS-INDUCED QUERY PLAN REGRESSION IN REAL-TIME ANALYTICAL WORKLOADS

Maithili S. Deshmukh (1,2), Sohel A. Bhura (1)*

(1) Department of Computer Science, Jhulelal Institute of Technology, Nagpur;
(2) Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur
India

* Corresponding Author, e-mail: msdesh22@gmail.com

Abstract: Analytical databases under continuous ingest accumulate a divergence between the data distribution assumed by the query optimiser and the data actually stored, and plans degrade silently when that divergence crosses a plan-selection boundary. This paper asks whether such regressions can be predicted before execution and attributed to a repairable cause. Three PostgreSQL databases receive identical drifting data under different statistics policies, so regression is measured as a latency ratio against a fresh-statistics reference rather than assumed. Across 1,620 query instances the optimiser's own cost estimate predicts nothing, whereas a two-predicate rule reading the catalogue and the unexecuted plan matches a gradient-boosted model, identifying all thirty held-out regressions without false positives. Repairing the attributed column resolves every regression, while repairing a second drifted column resolves none.

Key words: query optimisation, data drift, cardinality estimation, explainable artificial intelligence, PostgreSQL.

1. INTRODUCTION

Real-time analytical systems ingest data continuously while serving queries. The cost-based optimiser selecting plans for those queries does not consult the data; it consults a summary comprising most-common-value (MCV) lists, histograms and distinct-value counts, refreshed only when a statistics-collection process runs. Between refreshes the optimiser reasons about a table that no longer exists in the form it assumes.

This is ordinarily benign. Cost-based plans are frequently robust to estimation error, in that a sequential scan remains the preferred access path whether the optimiser anticipates one million rows or ten. Risk concentrates in the comparatively rare cases where an estimation error crosses a plan-selection boundary, leading the optimiser to prefer a nested loop over a hash join and thereby converting linear work into quadratic work. The distinction between error and consequence is central to this study: a pilot

reported in Section 3.2 produced estimation errors exceeding a factor of one thousand with no latency regression at all.

The purpose of this paper is to determine, first, whether statistics-induced plan regression can be predicted before a query executes using only information available in the catalogue and the unexecuted plan, and second, whether such a prediction can be attributed to a specific drifted column in a manner that is causally valid, meaning that repairing the attributed column demonstrably resolves the regression.

The contribution is threefold. A controlled three-arm testbed is constructed in which regression is measured against a fresh-statistics reference rather than defined by a heuristic threshold, every latency being a median over repeated executions. It is then shown that the optimiser's own cost estimate carries no signal about impending regression, that a single catalogue predicate attains full recall, and that a two-predicate rule obtained through attribution analysis matches an unrestricted gradient-boosted classifier. Finally, the attribution is validated by intervention: repairing the attributed column resolves every regression, whereas repairing a second column that also drifted resolves none.

2. RELATED WORK

Errors in cardinality estimation remain a principal cause of poor plan selection. The foundational measurements found that all evaluated estimators produce large errors routinely, with errors of a factor of one thousand or more on multi-join queries, and that no tested system detected join-crossing correlations [1]; bounding the impact of such errors on plan quality, rather than reducing the errors themselves, has also been studied [2]. The problem remains current: a recent analysis traces how estimation quality propagates into plan choice in a commercial engine [3], and learned estimation has since produced a substantial literature, including DeepDB [4] and comparative benchmarks of learned estimators [5], together with work directed at drift specifically, such as Warper on adapting learned estimators to data and workload change [6] and ALECE for dynamic workloads [7].

The present work is orthogonal to that line. It does not attempt to improve estimates but to predict the circumstances under which staleness in the classical estimator alters a plan for the worse. The distinction is material because the regressions observed here arise from a categorical failure, namely a value absent from the MCV list altogether, rather than from the graded inaccuracy that a better estimator would attenuate.

Work on detecting plan regression is closer in spirit. PLANSIEVE detects suboptimal plans in real time by comparing the ordering of sub-plans under estimated and true cardinalities [8]. Eraser eliminates regression in learned query optimisers [9], DBG-PT diagnoses regressions after they occur using a large language model [10], and NeurBench benchmarks learned components under drift [11]. These approaches operate either on learned optimisers or retrospectively. The setting studied here differs on four counts taken together: the optimiser is the classical cost-based planner; the warning precedes execution; the ground truth is a measured counterfactual latency rather than an estimated notion of optimality; and the output is a column-level repair action validated by intervention.

Interpretability appears sparsely in this literature relative to its overall size. KnobTree applies explainable reinforcement learning to parameter configuration [12], EAST presents an interpretable knob estimation system [13], and related work addresses interpretable scheduling for data-processing clusters [14]. SafeLoad lists interpretability among its design goals for admission control and filters memory-safe queries through an interpretable discriminative rule before invoking its learned components [15], which is structurally close to the arrangement the present results support. This body of work generally renders a learned model more interpretable; the result reported here runs in the opposite direction, since attribution over a learned model yielded a rule that then replaced it.

Learned query optimisers form a further line of work in which plan selection itself is delegated to a model. Bao steers an existing optimiser through coarse hints and was designed for practical deployment [16], whereas Neo learns a complete optimiser end to end [17]. Both replace or override the cost model. The present study leaves the classical planner untouched and predicts only when its statistics have become unsafe to rely upon, which is a narrower objective but one that requires no change to the engine and no training pipeline in production.

A related concern runs through recent work on data-intensive systems, where the aim is to anticipate degradation rather than react to it. Analytical pipelines that load operational data into relational stores and expose it through business-intelligence tools have been described together with the costs that accumulate as they grow [18]; traffic spikes have been predicted using recurrent neural monitoring agents [19]; and programme-based monitoring of information processes has been examined for detecting load anomalies early [20]. These approaches observe the surrounding infrastructure: the pipeline, the network and the arriving workload. The present work applies the same anticipatory principle inside the database engine, where the degradation originates in the optimiser’s own statistics.

3. DESCRIPTION OF THE PROBLEM AND PROPOSED SOLUTION

3.1. Problem formulation

Let q denote a query submitted at time t against a table whose statistics were last refreshed at an earlier time. Let L_stale(q) be the latency of q under those statistics and L_fresh(q) its latency under statistics refreshed at t, both taken as medians over repeated executions for reasons established in Section 4.1. The regression ratio is defined as the quotient of the two, and q is labelled regressed when that ratio reaches or exceeds a threshold tau, set to 1.5 throughout, with sensitivity reported in Section 4.4.

The prediction task is to estimate this indicator using only features observable before q executes: the catalogue statistics, the plan produced by EXPLAIN without execution, and inexpensive sampled probes of the current data. The methodological commitment is that the ratio is measured rather than modelled, the label being obtained by executing the same query against two databases holding identical data under differing statistics policies.

3.2. Experimental design

Figure 1 summarises the pipeline. A single generator supplies byte-identical batches to three databases differing only in statistics-maintenance policy, listed in Table 1; the latency ratio between the frozen and reference arms supplies a measured label; features are drawn exclusively from the frozen arm before execution; and the attribution is tested by repair interventions applied and rolled back within a transaction.

Table 1. Statistics-maintenance policy of each experimental arm

Arm	Statistics policy	Role
dg_fresh	Explicit ANALYZE every epoch	Reference latency
dg_stale	Frozen at epoch 0; autovacuum disabled on the fact table	Drift accumulates
dg_auto	PostgreSQL default autovacuum and autoanalyze	Deployed baseline

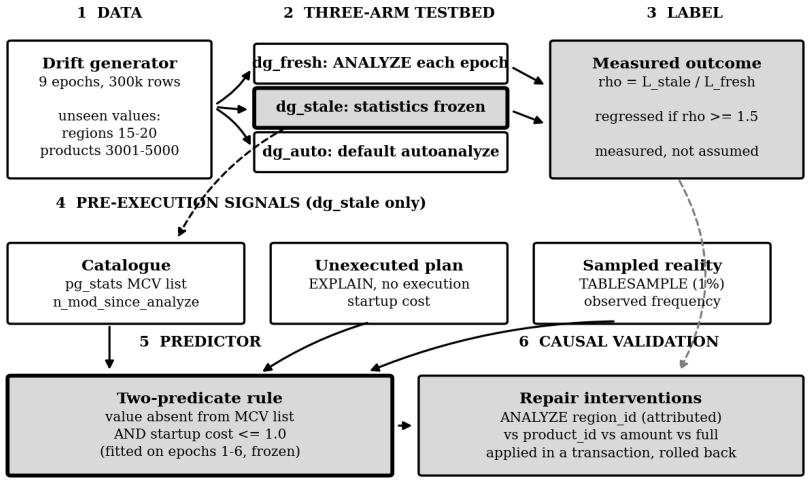


Figure 1. Overview of the experimental methodology. Stages 1 to 3 establish the measured label, stage 4 collects only signals available before execution, stage 5 applies the frozen rule, and stage 6 tests the attribution causally

The third arm was introduced in response to an experimental failure. An earlier version of this study produced no measurable effect at later epochs; investigation established that autovacuum had been re-analysing the nominally frozen arm. The default-policy configuration was therefore retained as a separate arm, and it yields one of the more consequential findings of the study, reported in Section 4.2.

The schema is a star configuration comprising a fact table with dimension tables for customers (50,000 rows), products (5,000) and regions (20). The fact table begins at 600,000 rows and grows by 300,000 per epoch across nine epochs, reaching 3,300,000 rows. Two independent drift channels operate, both of the unseen-value form. In the region channel, epoch 0 contains regions 1 to 14 only, and regions 15 to 20 appear from

epoch 1, rising to approximately 85 per cent of arrivals. In the product channel, epoch 0 contains products 1 to 3,000 only, with the remainder and their categories appearing subsequently. The second channel exists so that the attribution experiment of Section 4.5 has a competing drifted column to reject.

The unseen-value form was adopted after a pilot established that graded frequency drift is insufficient. In that pilot, hot-region mass migrated by approximately an order of magnitude, producing q-errors above one thousand with no latency regression, because the optimiser retained a sequential scan under both statistics regimes. This is consistent with earlier observations [1] and delimits the scope of the problem: only estimation errors that cross a plan-selection boundary have consequences.

Six analytical templates are evaluated with 30 parameter bindings each at every epoch on all three arms: a single-table filtered aggregate (T1), a two-way join with a category predicate (T2), a three-way join (T3), a top-N query over a range predicate (T4), a grouped aggregate with a HAVING clause (T5), and a distinct count across three tables (T6). Predicate values are drawn uniformly so that both drifted and undrifted regions are exercised. This yields 1,620 query instances, corresponding to 25,920 physical executions under the protocol below. Parallel execution was disabled and a statement timeout of 60 seconds applied. The full parameterised SQL, binding distributions, index definitions and planner settings accompany the replication package.

3.3. Measurement protocol

For every query instance, one execution on each of the two label-defining arms is discarded as warm-up. Five further executions are then taken on each arm, with the order of the arms alternated between successive repetitions so that any monotone drift in machine state, such as cache occupancy or checkpoint activity, cannot systematically favour one arm. The default-policy arm receives four executions, the first discarded. Reported latencies are medians; the interquartile range, coefficient of variation and all raw repetitions are retained so that dispersion and interval estimates can be computed.

3.4. Features and leakage control

Twenty-eight features are collected in three families, all observable before execution. The belief-versus-reality family comprises the optimiser's believed frequency for the queried value drawn from the catalogue, an MCV-membership indicator, a sampled frequency obtained through a one per cent table sample, and their ratio and absolute difference. The catalogue-staleness family comprises the live-tuple count, the modification count since the last analyse, and their ratio. The plan-structure family comprises estimated total and startup cost, estimated rows and width, plan depth, node count, and counts of join and scan operator types, all obtained from EXPLAIN without execution.

One feature was removed after detection of leakage. An initial run included an indicator of plan change computed by comparing the stale plan against the fresh plan; this is unavailable at prediction time, since it requires the reference database. Features are namespaced at collection time and non-feature columns are excluded from modelling by construction.

3.5. Baselines, models and evaluation

Four baselines are considered. B1, the MCV-absence rule, flags a query when the queried value is absent from the MCV list. B2, a modification-ratio heuristic, flags a query when the modification count divided by the live-tuple count exceeds 0.1. It is described as a heuristic rather than as the engine policy, because the actual trigger combines a scale factor with an absolute threshold; the genuine default policy is represented by the third arm rather than by B2. B3 flags the top decile of estimated total cost, and B4 predicts regression for every query, bounding recall trivially.

A histogram gradient-boosting classifier is evaluated using all features (M1), with the MCV feature removed to test redundancy (M2), and as a cascade refining precision within the positives of the MCV-absence rule (M3). M3 serves a limited role as an exploratory attribution instrument rather than as a candidate detector: restricting the model to the rule-positive subset concentrates its decision surface on the discrimination of interest and makes the SHAP attributions easier to read. Those attributions exposed estimated startup cost as the dominant discriminant, and a two-predicate rule R2 was then fitted independently on the training epochs, its threshold chosen by maximising F1 and thereafter frozen.

The split is temporal, with epochs 1 to 6 for training (1,080 instances, 85 positives) and epochs 7 to 9 for testing (540 instances, 30 positives); a random split would leak later points of the same drift trajectory into training. Exact binomial intervals are reported for precision and recall, since bootstrap resampling degenerates to a point interval wherever a method separates the classes perfectly, which would misrepresent the evidence available from 30 positives.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Measurement noise and label stability

The coefficient of variation of stale-arm latency has a median of 0.069 and a ninetieth percentile of 0.285; the corresponding values for the reference arm are 0.050 and 0.310. Because the label is a threshold on a ratio of two noisy quantities, this dispersion propagates directly into the target variable. Forming all twenty-five pairings of one stale repetition with one reference repetition for each query, 44.0 per cent of queries do not receive a unanimous label, and the expected disagreement against the median-based label is 10.1 per cent. A single-measurement protocol would therefore have mislabelled roughly one query in ten, which bears on how the results of any single-execution study of plan regression should be read.

4.2. Regression prevalence and the effect of default autoanalyses

Under frozen statistics, 115 of 1,620 instances regressed, a rate of 7.1 per cent. The per-epoch rate was 5.6 per cent in every epoch except the second and third, which reached 11.1 and 13.9 per cent. Under the default policy the rate was 5.6 per cent in the first epoch and zero thereafter, as Figure 2 shows.

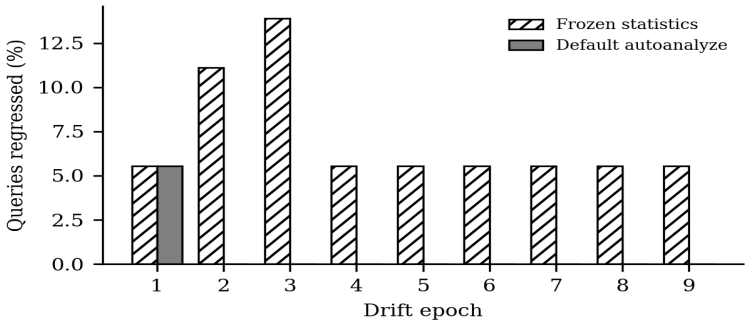


Figure 2. Proportion of queries regressing in each drift epoch under frozen statistics and under the default autoanalyses policy

The shipped statistics maintenance therefore prevents most of these regressions, which survive only within a transient window following a distribution shift and before the modification threshold is reached. The practical scope of this work is correspondingly restricted to settings in which that window is prolonged or the threshold is seldom reached: very large tables, for which the default scale factor represents a substantial absolute row count, and high-ingest systems in which autovacuum is throttled to protect foreground latency. Autovacuum was not detuned to enlarge the effect, since doing so would manufacture the result rather than measure it.

4.3. Detection performance

Table 2 reports performance on the held-out epochs. Of 381 queries flagged by high estimated cost, none regressed, and all 30 regressions fell outside the flagged set, so expensive queries and queries about to regress are close to disjoint populations in this workload. Triage by predicted cost therefore directs attention away from the queries at risk. In contrast, the MCV-absence predicate detected every observed regression at a precision of 0.189, using a catalogue lookup alone, with no execution, no sampling and no model.

Table 2. Detection performance on held-out epochs 7 to 9 (540 instances, 30 positives), with 95 per cent exact binomial intervals

Method	Precision [95%]	Recall [95%]	F1	Flagged
B1 MCV-absence rule	0.189 [0.131, 0.258]	1.000 [0.884, 1.000]	0.317	159
B2 modification-ratio	0.056 [0.034, 0.084]	0.667 [0.472, 0.827]	0.103	360
B3 optimiser cost	0.000 [0.000, 0.010]	0.000 [0.000, 0.116]	0.000	381
B4 always positive	0.056 [0.038, 0.078]	1.000 [0.884, 1.000]	0.105	540
M1 GBM, all features	1.000 [0.884, 1.000]	1.000 [0.884, 1.000]	1.000	30
M2 GBM, MCV removed	1.000 [0.884, 1.000]	1.000 [0.884, 1.000]	1.000	30
M3 cascade	0.600 [0.452, 0.736]	1.000 [0.884, 1.000]	0.750	50
R2 two-predicate rule	1.000 [0.884, 1.000]	1.000 [0.884, 1.000]	1.000	30

M1 and M2 perform identically, which is initially surprising given that MCV membership is the single most informative predicate available. The explanation is redundancy: the believed frequency drawn from the catalogue correlates with MCV membership at 0.999 and the count of index-scan nodes at 0.929, because both respond to the same underlying condition. Permutation importance over M2 assigns 0.920 to estimated startup cost and 0.726 to the index-scan count, with every remaining feature at exactly zero. The unrestricted model thus reconstructs a more elaborate proxy for information obtainable directly through a catalogue lookup.

4.4. A two-predicate rule matches the learned model

Startup cost acts as a proxy for plan shape, since a nested loop emits its first row immediately and therefore has near-zero startup cost, whereas a hash join must first build its hash table. Figure 3 shows the separation on the training epochs, where regressing queries exhibited startup costs between 0.7 and 1.0 and non-regressing queries between 8.4 and 18.0.

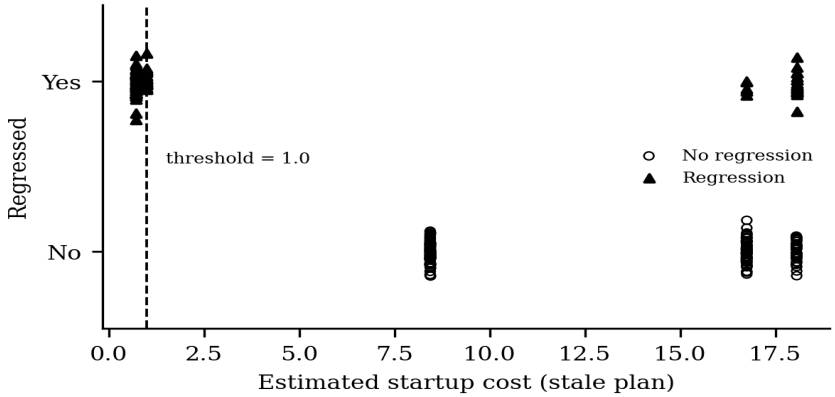


Figure 3. Estimated startup cost against regression outcome on the training epochs, restricted to queries whose predicate value is absent from the MCV list

The resulting rule flags a query when the queried value is absent from the MCV list and the plan's estimated startup cost does not exceed 1.0, the threshold having been fitted on the training epochs, where it attains an F1 of 0.828, and then frozen. On the held-out epochs the rule and the unrestricted models are indistinguishable, each flagging exactly the 30 regressing queries and nothing else. The cascade M3 is not equivalent to them, flagging 50 queries at a precision of 0.600; its role was to make the attributions legible rather than to serve as a detector. Table 3 reports sensitivity to the regression threshold.

The sensitivity analysis qualifies the headline result. At a threshold of 1.25 the rule retains perfect precision but recall falls to 0.370, indicating that it is selective for more severe regressions rather than being a general detector of mild degradation. No query in the held-out epochs regressed beyond a factor of three. Regressions are also concentrated in the join templates: of the 30 held-out regressions, 21 arise in T2 and 9 in T3, while

T1, T4, T5 and T6 produced none. The effective diversity of the regressing workload is therefore two templates rather than six.

Table 3. Sensitivity of the proposed rule to the regression threshold, with the startup-cost threshold frozen at its training value

Threshold tau	Positives	Precision	Recall	F1
1.25	81	1.000	0.370	0.541
1.50	30	1.000	1.000	1.000
2.00	30	1.000	1.000	1.000
3.00	0	n/a	n/a	n/a

4.5. Causal validation across both drift channels

Before intervening, it is necessary to establish which channel is responsible. All 115 measured regressions involve a region value absent from the MCV list, and none involves a region value present in it. Of 216 queries that target a newly introduced product category while querying an MCV-present region, none regressed. The product channel therefore produced no regressions in this workload despite drifting as designed, and the causal claim concerns the region channel alone. At the final epoch the frozen rule flagged 10 queries, all of which regressed, with a median ratio of 2.509. Table 4 reports the interventions.

Table 4. Effect of each repair intervention on the ten flagged regressions

Intervention	Resolved	Median ratio after
None (stale)	0 / 10	2.509
ANALYZE on region_id, the attributed column	10 / 10	0.987
ANALYZE on product_id, the second drifted channel	0 / 10	2.565
ANALYZE on amount, an undrifted decoy	0 / 10	2.591
ANALYZE on the full table	10 / 10	0.988

Repairing the attributed column resolved every regression and matched full-table repair, while repairing the product column, which had drifted substantially, left the regressions untouched, as did the undrifted decoy. Within this intervention setting the attribution therefore behaves causally rather than merely correlationally, and it discriminates among drifted columns rather than only between drifted and undrifted ones. Targeted repair is also cheaper, though modestly: on 3.3 million rows it required a median of 112.8 ms against 155.2 ms for the full table, a factor of 1.38.

4.6. Discussion

Under the evaluated workload the unrestricted gradient-boosted classifiers provide no measurable predictive advantage over the two-predicate rule obtained through attribution analysis. This does not generalise to cardinality estimation, join ordering or knob tuning, where learned methods have demonstrated substantial gains, nor would it

necessarily survive a workload with greater plan diversity. The narrower claim is that this task did not require learning, and that the attribution analysis is what revealed as much. The resulting artefact is auditable by an administrator and carries no training pipeline, no retraining schedule and no distribution-shift failure mode of its own. The pilot of Section 3.2 adds a related caution: q-errors above one thousand produced no regression when the plan did not change, so the actionable condition is not that an estimate is wrong but that it is wrong in a way that has altered the shape of the plan.

4.7. Threats to validity

Hardware is the most serious limitation. All measurements were taken on a single-core virtual machine with 3.9 GB of memory and parallel execution disabled, so regressions arising from parallelism, from input-output once data exceeds memory, or from multi-core contention lie outside the reach of this study. The fact table fits within page cache, so all latencies are CPU-bound and the index-versus-scan regression driven by random input-output cannot appear.

The workload is synthetic and narrow, and only two of the six templates regress at all, so the observed separation may partly reflect limited plan diversity; external validation is required before the fitted threshold can be regarded as portable. The perfect scores in Table 2 warrant caution rather than confidence: they rest on 30 held-out positives concentrated in two templates, and the recall of 0.370 at the lower threshold is the more informative figure. The causal claim is bounded by what was tested, since all measured regressions arose from the region channel. A single engine and a single drift form were examined, and the reference latency is an idealisation rather than an optimal plan.

5. CONCLUSION

This paper set out to determine whether statistics-induced plan regression can be predicted before execution and explained in an actionable form. In the held-out epochs the MCV-absence catalogue predicate detected every observed regression, and adding a single plan field produced a rule indistinguishable from unrestricted gradient-boosted classifiers, each identifying the 30 regressions without false positives. Within the tested intervention setting the attribution behaves causally: repairing the attributed column resolved every regression, while repairing a second genuinely drifted column resolved none. The practical advantage is a warning that requires no model, no training pipeline and no execution of the query, together with a column-level repair action that is cheaper than full-table analysis.

Two findings temper this. The default autoanalyze policy already prevents most such regressions, confining the problem to a transient window following a distribution shift, and the recall of the rule falls to 0.370 once the regression threshold is lowered to 1.25, so it addresses severe degradation rather than mild degradation. A third observation concerns method rather than result: with 44 per cent of queries receiving non-unanimous labels across single-run pairings, repeated measurement is a precondition for studying this phenomenon at all.

Future work should validate the approach on multiple cores with parallelism enabled and data exceeding memory, which is the limitation most likely to alter these conclusions, and then on an external benchmark with genuine plan diversity, on additional engines, and under drift configurations in which several columns are simultaneously causal.

REFERENCES

[1] Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., Neumann, T. How good are query optimizers, really? *Proceedings of the VLDB Endowment,* vol.9, no.3, 2015, pp. 204-215. DOI: 10.14778/2850583.2850594.

[2] Moerkotte, G., Neumann, T., Steidl, G. Preventing bad plans by bounding the impact of cardinality estimation errors. *Proceedings of the VLDB Endowment*, vol.2, no.1, 2009, pp. 982-993. DOI: 10.14778/1687627.1687738.

[3] Lee, K., Dutt, A., Narasayya, V., Chaudhuri, S. Analyzing the impact of cardinality estimation on execution plans in Microsoft SQL Server. *Proceedings of the VLDB Endowment*, vol.16, no.11, 2023, pp. 2871-2883. DOI: 10.14778/3611479.3611494.

[4] Hilprecht, B., Schmidt, A., Kulessa, M., Molina, A., Kersting, K., Binnig, C. DeepDB: learn from data, not from queries! *Proceedings of the VLDB Endowment*, vol.13, no.7, 2020, pp. 992-1005. DOI: 10.14778/3384345.3384349.

[5] Wang, X., Qu, C., Wu, W., Wang, J., Zhou, Q. Are we ready for learned cardinality estimation? *Proceedings of the VLDB Endowment*, vol.14, no.9, 2021, pp. 1640-1654. DOI: 10.14778/3461535.3461552.

[6] Li, B., Lu, Y., Kandula, S. Warper: efficiently adapting learned cardinality estimators to data and workload drifts. *Proceedings of the 2022 International Conference on Management of Data (SIGMOD),* Philadelphia, USA, June 2022, pp. 1920-1933. DOI: 10.1145/3514221.3526179.

[7] Li, P., Wei, W., Zhu, R., Ding, B., Zhou, J., Lu, H. ALECE: an attention-based learned cardinality estimator for SPJ queries on dynamic workloads. *Proceedings of the VLDB Endowment*, vol.17, no.2, 2023, pp. 197-210. DOI: 10.14778/3626292.3626302.

[8] Datta, A., Izenov, Y., Tsan, B., Amanbayev, A., Rusu, F. PLANSIEVE: real-time suboptimal query plan detection through incremental refinements. arXiv preprint arXiv:2501.16544, 2025. URL: https://arxiv.org/abs/2501.16544 (Visited on 06.09.2026).

[9] Weng, L., Zhu, R., Wu, D., Ding, B., Zheng, B., Zhou, J. Eraser: eliminating performance regression on learned query optimizer. *Proceedings of the VLDB Endowment*, vol.17, no.5, 2024, pp. 926-938. DOI: 10.14778/3641204.3641205.

[10] Fent, P., van Renen, A., Kipf, A., Leis, V., Neumann, T., Kemper, A. DBG-PT: a large language model assisted query performance regression debugger. *Proceedings of the VLDB Endowment*, vol.17, no.12, 2024, pp. 4337-4340. DOI: 10.14778/3685800.3685869.

[11] NeurBench: benchmarking learned database components with data and workload drift modelling. *Proceedings of the ACM on Management of Data*, 2026. DOI: 10.1145/3802091.

[12] Chen, J., Qi, S., Li, Y., Dong, Z., Ding, M., Wu, Y., Wang, X. KnobTree: intelligent database parameter configuration via explainable reinforcement learning. *arXiv preprint* arXiv:2406.15073, 2024. URL: https://arxiv.org/abs/2406.15073 (Visited on 06.09.2026).

[13] EAST: An interpretable knob estimation system for cloud database. *Proceedings of the 41st IEEE International Conference on Data Engineering (ICDE)*, 2025, pp. 4588-4591. DOI: 10.1109/ICDE65448.2025.00359.

[14] Hu, Z., Wang, C., Paik, H., Shu, Y., Zhu, L. Learning interpretable scheduling algorithms for data processing clusters. *arXiv preprint* arXiv:2405.19131, 2024. URL: https://arxiv.org/abs/2405.19131 (Visited on 06.09.2026).

[15] Wu, Y., Li, Y., Wang, Z., Xie, Z., Yang, D., Chen, K., Shou, L., Tang, B., Lin, L., Li, H., Chen, G. SafeLoad: efficient admission control framework for identifying memory-overloading queries in cloud data warehouses. *Proceedings of the VLDB Endowment*, vol.19, no.4, 2026, pp. 715-727. DOI: 10.14778/3785297.3785311.

[16] Marcus, R., Negi, P., Mao, H., Tatbul, N., Alizadeh, M., Kraska, T. Bao: making learned query optimization practical. *Proceedings of the 2021 International Conference on Management of Data (SIGMOD)*, Xian, China, June 2021, pp. 1275-1288. DOI: 10.1145/3448016.3452838.

[17] Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., Papaemmanouil, O., Tatbul, N. Neo: a learned query optimizer. *Proceedings of the VLDB Endowment*, vol.12, no.11, 2019, pp. 1705-1718. DOI: 10.14778/3342263.3342644.

[18] Sreedhar, T.S., Islam, S., Atomsa, M., Yazdan Doust, E., Elnaim, M.S., Mishra, S., Vemparala, V.N., Bajpai, R. Applications of BIG DATA in renewable energy systems based on cloud computing. *International Journal on Information Technologies and Security*, vol.16, no.3, 2024, pp. 121-128.

[19] Kravets, O.Ja., Aksenov, I.A., Redkin, Yu.V., Rahman, P.A., Kochegarov, M.V., Gorshkov, A.V., Sorokin, S.A. Modeling of neural network monitoring agent to predict traffic spikes and agent training. *International Journal on Information Technologies and Security*, vol.16, no.3, 2024, pp. 49-56.

[20] Romansky, R. Preliminary organization of virtual network based on program monitoring. *International Journal on Information Technologies and Security*, vol.16, no.3, 2024, pp. 111-120.

Information about the authors:

Maithili Sanjay Deshmukh is a Ph.D. researcher in Computer Science at the Jhulelal Institute of Technology (JIT), Nagpur, India. Her research interests include SQL query optimization, real-time data analytics, database systems, parallel query processing, and workload optimization. She also has professional experience in data engineering and cloud data platforms.

Dr. Sohel A. Bhura is associated with the Department of Computer Science, Jhulelal Institute of Technology (JIT), Nagpur, India. He holds a Ph.D. in Computer Science and Engg. His academic and research interests include Artificial Intelligence, Data Analytics, Machine Learning and emerging intelligent computer applications. He supervises doctoral research in data management.

Manuscript received on 11 September 2026