Serialization-Induced Prediction Drift

Khudran M. Alzhrani
Computers Department, College of Engineering and Computing in Al-Qunfudhah, Umm Al-Qura University, Al-Qunfudhah, Mecca, Saudi Arabia
kmzhrani@uqu.edu.sa (corresponding author)

Received: 18 February 2026 | Revised: 6 March 2026, 22 March 2026, and 2 April 2026 | Accepted: 10 April 2026

 DOI: https://doi.org/10.48084/etasr.18259

ABSTRACT

Tabular Machine Learning (ML) workflows often export and reload numeric features through formats, such as CSV and Parquet, sometimes rounding values or casting between floating-point precisions (e.g., float64 to float32). Although commonly treated as engineering details, these steps can introduce systematic numerical perturbations that propagate into model predictions. This study presents a methodology to quantify how routine data-representation changes affect prediction drift and performance. Starting from a float64 Parquet baseline, CSV round-trip variants with 6, 3, and 1 decimal places and a float32 Parquet variant are generated. Fixed train-validation-test splits are reused across treatments, and two scenarios are evaluated: train-on-variant and evaluation-only (baseline-trained, perturbed-test). Value-level drift, prediction drift (score drift, rank correlation, and classification churn), and performance deltas are measured, with the results aggregated across three random seeds with bootstrap confidence intervals and Wilcoxon signed-rank tests. Experiments on the Breast Cancer Wisconsin (Diagnostic) classification dataset and the Diabetes and California Housing regression datasets, using multiple model families, show that mild perturbations (CSV 6/3 decimals and float32) generally yield negligible drift and no meaningful performance change, while rounding to 1-decimal place triggers a sharp instability onset, including threshold-crossing effects in classification and marked drift amplification in the most sensitive regression settings. Sensitivity varied by model family under aggressive rounding, and the added analysis of representative linear models showed that 1-decimal rounding perturbs the internal linear score and can also change the coefficient structure learned during retraining.

Keywords-prediction drift; data serialization; numerical precision; machine learning pipelines; tabular data

I. INTRODUCTION

Modern Machine Learning (ML) workflows often move numerical features through several storage and representation forms before a model makes a prediction. Values may be exported to text for data exchange. They may also be rounded during storage or reporting. In some cases, they are cast between floating-point precisions as part of normal data handling. These steps are often treated as minor implementation details rather than experimental variables. However, research in ML systems has shown that seemingly minor pipeline decisions can accumulate into technical debt and lead to unexpected behavior in deployed systems [1]. Dataset documentation work also stresses that datasets should record not only what data are used, but also how they are collected, processed, and transformed, because these choices can affect reproducibility and downstream modeling results [2].

Research suggests that even small numerical differences can affect output reproducibility. Work on large-model inference shows that runtime configuration and numerical effects can lead to nondeterministic outputs and unstable evaluation results under sensitive conditions [3]. Similar findings have appeared in other domains. Studies on the numerical stability of DeepGOPlus inference demonstrate that robustness depends on the perturbation mechanism and on where reduced precision is introduced [4]. More broadly, research on numerical uncertainty in scientific pipelines indicates that low-level floating-point perturbations can propagate through complex workflows and change the derived results in meaningful ways [5, 6]. Together, these studies show that numerical effects are a practical source of variability in computational pipelines. One underexplored source of this variability is the data serialization layer. At this stage, floating-point values are converted to decimal strings or lower-precision representations before model execution. In practical ML pipelines, this can happen when numerical tabular data are exported, shared, and reloaded across tools or storage formats. Research on floating-point printing algorithms shows that converting binary floating-point numbers to decimal strings involves important correctness and performance trade-offs [7, 8]. Empirical work also demonstrates that shortest-decimal conversion remains an active systems problem, with measurable differences across methods and implementations [9]. These findings suggest that serialization should be treated as a possible perturbation source rather than as a neutral storage step. Another related line of work studies how perturbed data affect predictive performance when the perturbations are introduced intentionally. Authors in [10] showed that statistical disclosure control techniques can affect the predictive performance of several ML algorithms. Reduced numerical precision for computational efficiency has also been studied. Studies on limited-precision deep learning, mixed-precision

training, BFLOAT16 training, and neural-network quantization show that carefully designed low-precision computation can preserve accuracy while still influencing optimization and inference behavior in some cases [11-15]. However, these studies mainly focus on intentional privacy-oriented perturbations or reduced precision inside the model, such as weights, activations, and arithmetic formats. They do not focus on routine representation changes applied to input data before model execution.

The present study addresses this gap by evaluating serialization-induced prediction drift as a reliability problem in tabular ML pipelines. The study contributes a controlled experimental framework that isolates routine input-side representation changes and measures how they affect feature values, model predictions, and task performance. Starting from clean float64 baseline data, the study creates perturbed variants that reflect common handling operations, specifically decimal-limited CSV round-trips and float32 storage conversion. The design also separates training-time exposure from inference-time sensitivity through two complementary scenarios. In the train-on-variant setting, perturbed data are used for both training and evaluation. In the evaluation-only setting, the model is trained on the clean baseline and tested on perturbed inputs. By combining value-level drift, prediction-level drift, and standard performance measures across multiple model families and benchmark datasets, the study provides a practical framework for determining when routine representation changes are harmless and when they become decision-relevant.

II. METHODOLOGY

The proposed methodology quantifies how controlled data perturbations propagate through ML pipelines and affect model predictions. In this study, the term "controlled" refers to all treatments being derived from the same clean float64 baseline data; the same split indices are reused across treatments for each random seed, and model configurations and evaluation procedures are kept fixed across comparisons. The only factor intentionally varied is the representation of the input data. This design isolates the effect of routine serialization and storage operations from other sources of variation. The workflow consists of four main stages: generation of perturbed data variants, construction of fixed train-validation-test splits under multiple random seeds, model training and evaluation under two complementary scenarios, and computation of prediction-drift and performance-change metrics. Figure 1 summarizes this controlled experimental workflow and highlights the separation between the train-on-variant and evaluation-only settings.

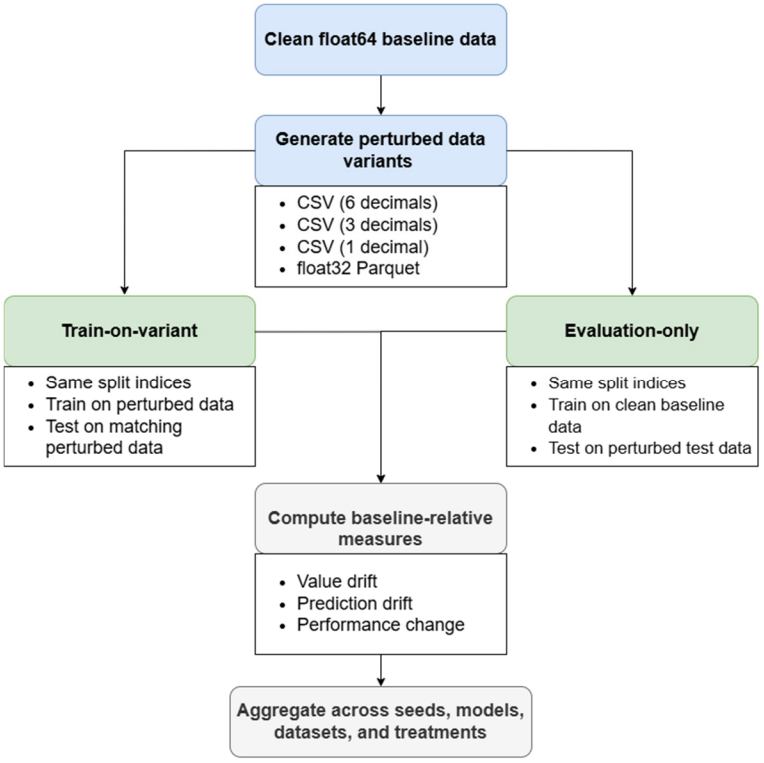

Fig. 1. Controlled experimental workflow used to quantify serialization-induced prediction drift.

A. Data Variants

All experiments start from a clean baseline dataset stored in double-precision floating-point format. From this baseline, several data variants are generated to emulate routine data handling operations encountered in real-world ML workflows. First, CSV rounding variants are created by exporting feature values to CSV files with restricted decimal precision (1, 3, or 6 decimal places) and subsequently reloading them. This process simulates precision loss arising from data exchange, logging, or intermediate storage in text-based formats. Second, a float32 variant is produced by casting all feature columns to 32-bit floating-point representation and saving the result in Parquet

format. This reflects the default numeric precision used by ML frameworks and data processing pipelines. For each data variant, value-level drift is quantified by comparing the perturbed feature values with their baseline counterparts. Three complementary metrics are computed: the mean absolute difference across all feature entries, the maximum absolute difference observed in any feature entry, and the percentage of feature entries whose values change by any non-zero amount. These statistics characterize both the magnitude and prevalence of numerical perturbations introduced by each data transformation.

B. Data Splitting and Reproducibility

To ensure fair and reproducible comparisons across all experimental conditions, fixed train–validation–test splits are generated once per random seed and reused for all data variants. The dataset is partitioned into 70% training, 15% validation, and 15% test subsets. For the classification task (breast cancer dataset), stratification is enforced such that both classes are present in the validation and test sets. To guarantee this condition, stratified splitting is repeated up to 100 times per seed until a valid split is obtained. All split indices are stored to disk and reused verbatim across treatments. Experiments are conducted using three distinct random seeds (42, 123, and 456) to assess the robustness of observed effects to stochastic variation in data partitioning and model initialization.

C. Model Training and Evaluation Scenarios

For each combination of dataset, model, data variant, and random seed, two complementary evaluation scenarios are considered. In the standard train-on-variant scenario, models are trained on the perturbed training data and evaluated on the corresponding perturbed test data. This setting reflects a fully affected pipeline in which both training and inference stages operate on altered data representations. In the evaluation-only scenario, a model is trained once on the clean baseline training data and subsequently evaluated on perturbed test data derived from the same test indices. This design isolates the effect of test-time input perturbations and enables separation of inference-time sensitivity from retraining effects. In both scenarios, a baseline reference model is trained on the clean training data and evaluated on the clean test data. All drift and performance change metrics are computed relative to this baseline reference.

D. Prediction Drift and Performance Metrics

Let $\hat{y}^b$ denote the baseline prediction scores produced by a model on the baseline test set, and let $\hat{y}^t$ denote the prediction scores obtained under a given data treatment, evaluated on the same test indices. Prediction drift is quantified using several complementary measures. The mean absolute score drift is defined as:

$$\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i^b - \hat{y}_i^t\right| \quad (1)$$

where n is the number of test instances, capturing the average magnitude of changes in model predictions induced by the data treatment. The maximum absolute score drift is defined as:

$$\max_i\left|\hat{y}_i^b - \hat{y}_i^t\right|, \quad (2)$$

which reflects the largest individual deviation between baseline and treated predictions. In addition, the Spearman rank correlation between $\hat{y}^b$ and $\hat{y}^t$ is computed to assess the extent to which the relative ordering of predictions is preserved under data perturbations.

For classification tasks, prediction churn is measured as the fraction of test instances whose predicted class label changes when applying a fixed decision threshold of 0.5. For regression tasks, normalized prediction drift is computed by dividing absolute score drift values by the standard deviation of baseline predictions, enabling comparisons across datasets with different target scales.

Changes in predictive performance are summarized using delta metrics defined as the difference between a treatment's evaluation metric and the corresponding baseline metric. Specifically, differences in accuracy, Area Under the Curve (AUC), and log loss are reported for classification, while differences in Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are reported for regression. All metrics are computed independently for each random seed. To further interpret the strongest 1-decimal effects, representative linear models are examined through fixed-model internal score shifts in the evaluation-only setting and coefficient changes after retraining on rounded data.

E. Statistical Summaries and Association Analysis

To obtain robust aggregate estimates, the results are summarized across random seeds, models, and datasets. For each treatment and metric, the median value is reported along with a 95% bootstrap confidence interval computed using 1000 bootstrap resamples and the percentile method. Statistical significance of treatment-induced changes is assessed using the Wilcoxon signed-rank test, testing whether the distribution of metric differences is centered at zero.

Finally, to examine the relationship between numerical perturbations at the data level and their downstream effects on model predictions, Spearman rank correlations are computed between value-drift metrics and prediction-drift metrics for each dataset–model pair. This analysis provides insights into whether larger input-level perturbations systematically correspond to increased prediction instability.

III. EXPERIMENTAL SETUP

The experimental evaluation is conducted based on the methodology described earlier using three benchmark datasets. The Breast Cancer Wisconsin (Diagnostic) dataset [16, 17] is used for binary classification. It contains 569 samples with 30 numerical features and a binary malignant/benign target. The diabetes dataset [18, 19] is utilized for regression. It consists of 442 samples with 10 numerical features and a continuous target measuring disease progression one year after baseline. To broaden coverage with a larger-scale regression benchmark, the California Housing dataset [20, 21] is also included. It contains 20,640 samples with 8 numerical features and a continuous target representing the median house value for California districts. For all datasets, the baseline condition preserves the original double-precision floating-point representation of feature values.

All data treatments evaluated in the experiments are listed in Table I. The baseline condition corresponds to clean data stored in Parquet format and serves as the reference for all comparisons. Perturbed variants include CSV serialization with restricted decimal precision (6, 3, or 1 decimal place) and conversion of feature values to 32-bit floating-point representation stored in Parquet format. For each perturbation, an evaluation-only counterpart is also considered, in which the transformation is applied exclusively at test time while training is performed on the clean baseline data, enabling isolation of inference-time sensitivity.

TABLE I. TREATMENTS EVALUATED IN THE STUDY

Treatment	Data representation	Training data	Test data
baseline_parquet	Parquet (float64)	Baseline	Baseline
csv_dec6	CSV (6 decimal places)	Perturbed	Perturbed
csv_dec3	CSV (3 decimal places)	Perturbed	Perturbed
csv_dec1	CSV (1 decimal place)	Perturbed	Perturbed
float32_parquet	Parquet (float32)	Perturbed	Perturbed
eval_csv_dec6	CSV (6 decimal places)	Baseline	Perturbed
eval_csv_dec3	CSV (3 decimal places)	Baseline	Perturbed
eval_csv_dec1	CSV (1 decimal place)	Baseline	Perturbed
eval_float32_parquet	Parquet (float32)	Baseline	Perturbed

A set of models with diverse inductive biases is evaluated for each task. For classification, the experiments include Logistic Regression (LR) with L2 regularization, LR with L1 regularization, a Random Forest (RF) classifier with 300 trees, and a multi-layer perceptron with two hidden layers of 64 and 32 neurons. For regression, ridge regression, lasso regression, RF regression with 300 trees, and a multi-layer perceptron with the same architecture are employed. All models are trained using fixed random seeds and default hyperparameters unless otherwise specified. Linear models and neural networks use standard feature scaling, whereas tree-based models operate directly on unscaled features. Each combination of dataset, model, treatment, and random seed is executed independently, producing prediction outputs and evaluation metrics. Three random seeds (42, 123, and 456) are used to assess robustness. Fixed train, validation, and test splits are reused across all treatments for a given seed to ensure comparability. The results are aggregated across seeds, models, and datasets to produce the summary statistics and visualizations.

IV. RESULTS

A. Value-Level Drift Induced by Data Treatments

Table II summarizes value-level drift for the breast cancer dataset. Similar monotonic drift patterns were observed for the regression datasets (Diabetes and California Housing). As expected, CSV rounding introduces increasingly large numerical deviations as precision is reduced, while float32 conversion introduces very small absolute errors despite modifying a large fraction of entries.

Rounding to six decimal places results in negligible value drift, with a mean absolute difference on the order of 10^{-10} and fewer than 4% of feature values modified. In contrast, rounding to three decimal places affects more than 60% of entries and increases the mean absolute difference to approximately 1.6×10^{-4}. The most aggressive treatment, rounding to 1-decimal place, alters nearly 90% of feature values and introduces mean absolute deviations on the order of 10^{-2}. Although float32 conversion modifies over 95% of entries, the induced numerical deviations remain small, with mean absolute differences below 10^{-6}.

TABLE II. VALUE-LEVEL DRIFT INDUCED BY DATA TREATMENTS FOR THE BREAST CANCER DATASET

Treatment	Mean absolute value drift	Maximum absolute value drift	Percentage of entries changed (%)
baseline_parquet	0.0000	0.0000	0.0
csv_dec6	8.20×10^{-11}	3.00×10^{-7}	3.5
csv_dec3	1.61×10^{-4}	5.00×10^{-4}	61.1
csv_dec1	2.06×10^{-2}	5.00×10^{-2}	88.0
float32_parquet	8.30×10^{-7}	2.44×10^{-5}	95.5

B. Classification Stability and Threshold Effects

Tables III–V provide illustrative single-seed examples (seed = 42) for representative models, while Table VII reports bootstrap confidence intervals and Wilcoxon test results across seeds. Table III reports prediction drift and performance changes for LR on the breast cancer classification task. Mild perturbations, including six-decimal CSV rounding and float32 conversion, produce negligible prediction drift and do not affect accuracy or AUC. Rounding to three decimal places increases score drift but preserves prediction ranking and classification accuracy. In contrast, rounding to 1-decimal place results in a sharp degradation of stability. Mean absolute score drift increases to approximately 4.2×10^{-2}, the Spearman rank correlation drops below 0.99, and more than 3% of test predictions change class label. This abrupt increase in prediction churn is accompanied by a measurable drop in classification accuracy, indicating a threshold-crossing effect in which small additional perturbations push samples across the decision boundary.

TABLE III. PREDICTION DRIFT AND PERFORMANCE CHANGE FOR LR (CLASSIFICATION)

Treatment	Mean absolute prediction drift	Spearman's rank correlation	Prediction churn (%)	Accuracy change
csv_dec6	2.10×10^{-9}	1.000	0.0	0.000
csv_dec3	1.70×10^{-3}	0.9998	0.0	0.000
csv_dec1	4.20×10^{-2}	0.984	3.5	−0.035
float32_parquet	1.50×10^{-8}	1.000	0.0	0.000

C. Regression Sensitivity to Numerical Perturbations

Table IV presents corresponding results for ridge regression on the diabetes dataset. Unlike classification, regression models exhibit a continuous amplification of value-level perturbations. Even moderate rounding to three decimal places introduces measurable prediction drift and a small increase in RMSE. Rounding to 1-decimal place produces substantial instability, with mean absolute prediction drift exceeding 18 units and a pronounced increase in RMSE.

TABLE IV. PREDICTION DRIFT AND PERFORMANCE CHANGE FOR RIDGE REGRESSION (DIABETES DATASET)

Treatment	Mean absolute prediction drift	Spearman's rank correlation	RMSE change
csv_dec6	2.80×10^{-4}	1.000	−0.000
csv_dec3	2.70×10^{-1}	0.9996	+0.046
csv_dec1	18.40	0.889	+5.476
float32_parquet	6.70×10^{-5}	1.000	−0.000

Although float32 changes nearly all feature values, its effect on regression predictions is negligible. This highlights the difference between many small perturbations and fewer large deviations. To assess whether these instability patterns generalize beyond individual model choices, Figure 2 summarizes mean absolute prediction drift across all evaluated models, tasks, and datasets. Each curve corresponds to a dataset–model combination, with error bars indicating variability across random seeds. Across model architectures and inductive biases, mild perturbations (csv_dec6 and csv_dec3) induce limited drift, whereas aggressive rounding to 1-decimal place produces a clear escalation. In contrast, conversion to 32-bit floating-point representation consistently yields negligible drift, even for models that are otherwise sensitive to numerical perturbations. While low-bit quantization studies focus on internal model computation (weights and activations), the proposed experiments isolate the upstream data layer, showing that input representation changes alone can induce prediction drift even before any model-side quantization is applied.

Although drift magnitude varies by model family, the instability onset occurs consistently under the most aggressive rounding condition. At the output level, tree-based models and neural networks generally showed smoother degradation with increasing perturbation severity, whereas linear models showed more abrupt changes. These results suggest that the impact of data precision loss is not confined to a single algorithm or task, but appears consistently across the evaluated modeling approaches. To broaden dataset coverage and evaluate regression sensitivity on a larger scale, ridge regression is additionally assessed using the California Housing dataset (20,640 samples), with Table V showing negligible effects for csv_dec6/csv_dec3 and float32, but a noticeable increase in drift and RMSE under csv_dec1.

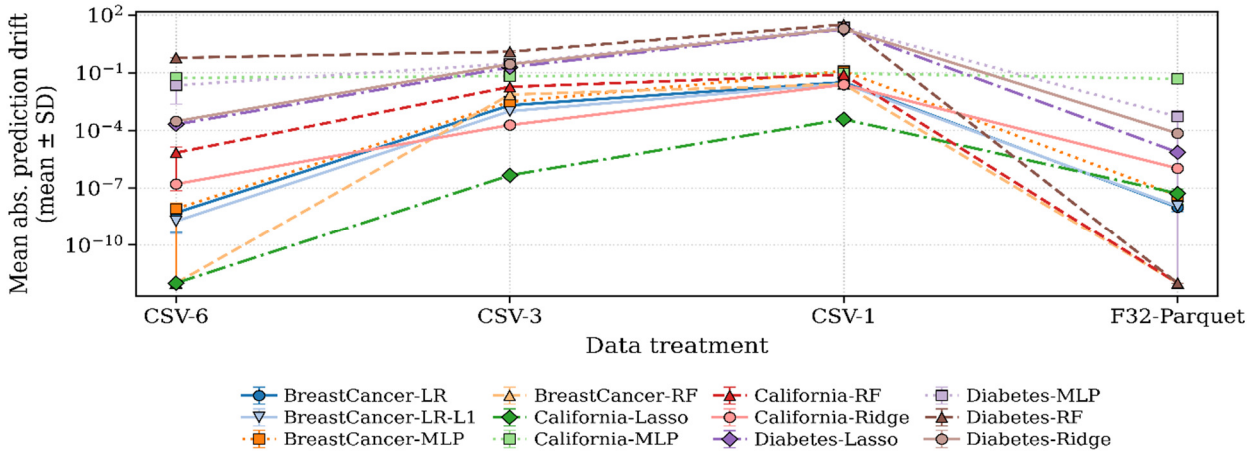

Fig. 2. Mean absolute prediction drift across all dataset–model combinations under increasing perturbation severity (mean ± standard deviation across seeds).

TABLE V. PREDICTION DRIFT AND PERFORMANCE CHANGE FOR RIDGE REGRESSION ON CALIFORNIA DATASET

Treatment	Mean absolute prediction drift	Spearman's rank correlation	RMSE change
csv_dec6	1.87×10^{-7}	1.000	$+1.45\times10^{-9}$
csv_dec3	2.13×10^{-4}	0.999	-5.52×10^{-6}
csv_dec1	0.0258	0.9991	+0.000981
float32_parquet	9.19×10^{-7}	0.999	-4.48×10^{-8}
eval_csv_dec6	1.87×10^{-7}	1.000	-1.84×10^{-9}
eval_csv_dec3	2.13×10^{-4}	0.999	-4.32×10^{-6}
eval_csv_dec1	0.0260	0.9991	+0.00156
eval_float32_parquet	1.72×10^{-6}	0.999	-1.29×10^{-8}

D. Sensitivity of Dense and Sparse Linear Models Under Aggressive Rounding

Table VI summarizes the output-level effect of aggressive 1-decimal rounding across the evaluated model families. In classification, both logistic models show larger prediction drift and accuracy loss than the tree-based and neural baselines, and the L1 variant is more affected than the L2 variant. In regression, the linear models are also sensitive to coarse rounding, although the relative ordering among all model families is not uniform across datasets. Overall, Table VI shows that aggressive rounding does not affect all models equally and that the greatest changes appear in representative linear models. Because their internal prediction score is directly interpretable, these models provide a useful basis for examining how coarse serialization propagates through a fixed learned mapping and how retraining on rounded data changes the fitted coefficients. To provide statistical transparency, Table VII reports median effects with 95% bootstrap confidence intervals and Wilcoxon signed-rank tests for the 1-decimal rounding condition (and its evaluation-only counterpart). These results confirm that the instability-onset effects observed under csv_dec1 are statistically significant across seeds.

TABLE VI. PREDICTION DRIFT AND PERFORMANCE DEGRADATION UNDER CSV_DEC1

Task	Model	Mean absolute prediction drift	Performance change
Classification	LR (L2)	4.2×10^{-2}	−3.5% accuracy
Classification	LR (L1)	5.8×10^{-2}	−5.1% accuracy
Classification	RF	1.9×10^{-2}	−1.2% accuracy
Classification	MLP	2.6×10^{-2}	−2.0% accuracy
Regression	Ridge	18.4	+5.48 RMSE
Regression	Lasso	26.7	+7.92 RMSE
Regression	RF	7.3	+2.11 RMSE
Regression	MLP	11.6	+3.84 RMSE

E. Interpreting the 1-Decimal Effect in Representative Linear Models

To better understand the stronger instability observed under 1-decimal rounding, this study next examines representative linear models, for which the internal linear score (logit) has a direct analytical form. This focus is motivated by the preceding results: Table VI shows that the sharpest output-level changes under csv_dec1 occur in the linear models, and the evaluation-only setting isolates the effect of rounded inputs on a fixed trained model. Table VIII demonstrates that, under eval_csv_dec1, 1-decimal rounding produces a substantial shift in the internal linear score for both breast cancer classifiers. The resulting prediction flips are concentrated near the decision boundary. For the L1 model, all flipped cases occur when the clean absolute logit is at most 1, and for the L2 model, the majority of flips also arise in that boundary region. This indicates that the observed churn is not a diffuse numerical effect across the test set, but a boundary-sensitive response in which coarse rounding perturbs the learned dot-product computation enough to move borderline cases across the classification threshold. The train-on-variant results suggest that aggressive rounding affects not only the test-time inputs but also the fitted linear mapping obtained during retraining. Table IX shows that retraining on csv_dec1 changes the learned coefficient vectors, with the largest reorganization appearing in the breast cancer logistic models. In the L1 classifier, the reduced active-feature overlap indicates that coarse rounding can also change which features remain selected by the sparse model, not merely the magnitude of their coefficients. The diabetes regression models also change under retraining, although the effect is more moderate than in the breast cancer classifiers. Taken together, Tables VIII and IX clarify the mechanism behind the stronger 1-decimal effect in representative linear models: coarse serialization perturbs the internal linear score directly at evaluation time and can also alter the coefficient structure learned during retraining.

TABLE VII. BOOTSTRAP 95% CI AND WILCOXON TESTS FOR 1-DECIMAL ROUNDING

Dataset	Task	Treatment	Statistic (median [95% CI])	Wilcoxon p-value
Breast Cancer	Classification	csv_dec1	Churn: 0.0174 [0.0116, 0.0349]	4.88×10^{-4}
Breast Cancer	Classification	eval_csv_dec1	Churn: 0.0349 [0.0174, 0.0465]	4.88×10^{-4}
Diabetes	Regression	csv_dec1	ΔRMSE: 8.353 [6.063, 9.739]	4.88×10^{-4}
Diabetes	Regression	eval_csv_dec1	ΔRMSE: 7.734 [5.62, 9.432]	4.88×10^{-4}
California Housing	Regression	csv_dec1	ΔRMSE: 0.0006 [0.0002, 0.0018]	0.0039
California Housing	Regression	eval_csv_dec1	ΔRMSE: 0.0006 [0.0002, 0.0017]	0.0039

TABLE VIII. INTERNAL LOGIT PERTURBATION FOR REPRESENTATIVE LOGISTIC MODELS UNDER EVAL_CSV_DEC1 ON THE BREAST CANCER DATASET

Model	Mean absolute logit shift	Prediction churn (%)	Flipped cases with |clean logit| ≤ 1 (%)
LR (L2)	1.267 [1.262, 1.362]	4.65 [3.49, 4.65]	75.0 [50.0, 100.0]
LR (L1)	1.190 [1.090, 1.333]	2.33 [2.33, 4.65]	100.0 [100.0, 100.0]

Values are median [min, max] across 3 seeds

TABLE IX. COEFFICIENT CHANGES AFTER RETRAINING ON CSV_DEC1

Dataset	Model	Coefficient cosine similarity	Sign changes	Active-feature Jaccard
Breast Cancer	LR (L2)	0.037 [0.006, 0.071]	7 [6, 7]	—
Breast Cancer	LR (L1)	0.281 [0.273, 0.421]	6 [5, 11]	0.625 [0.421, 0.688]
Diabetes	Ridge	0.706 [0.693, 0.707]	2 [2, 3]	—
Diabetes	Lasso	0.961 [0.939, 0.973]	3 [1, 3]	0.700 [0.667, 0.900]

Values are median [min, max] across three seeds; active-feature Jaccard is shown only for sparse models.

V. CONCLUSION

Routine data serialization and precision changes are often treated as harmless implementation details in Machine Learning (ML) workflows, yet the results of this study show that their effect depends strongly on perturbation severity and model type. Using a controlled experimental methodology, the study examined how common representation changes introduced before model execution, specifically CSV round-trips with restricted decimal precision and float32 storage conversion, propagate into value-level drift, prediction drift, and performance change. Across the evaluated classification and regression settings, float32 conversion and CSV rounding to 6 or 3 decimal places were generally benign, producing minimal prediction drift and stable performance across models. In contrast, rounding to 1-decimal place produced a clear instability onset: prediction scores shifted substantially, classification churn emerged through threshold-crossing effects, and regression errors increased markedly. The results also showed model-dependent differences under aggressive rounding, and the added analysis of representative linear models indicated that 1-decimal rounding perturbs the internal linear score and can also alter the coefficient structure learned

during retraining, while the internal propagation mechanisms of tree-based and neural models were not directly analyzed in this study.

These findings place data representation choices within the broader context of ML reliability by showing that upstream numeric transformations can influence downstream predictive behavior even when the learning algorithm itself is unchanged. The main contribution of this work is therefore a controlled and reproducible framework for quantifying serialization-induced prediction drift and for separating the effects of train-time exposure from evaluation-only sensitivity. In this sense, the study complements broader recent work on preserving model reliability under changing data conditions [22], while focusing specifically on routine serialization and precision changes applied to input data before model execution in static tabular pipelines. From a practical perspective, the results suggest that coarse CSV rounding should be avoided in workflows where prediction stability, reproducibility, or decision consistency matters. Future work can extend this analysis to additional datasets, storage formats, and model classes, and can further investigate how serialization-induced drift interacts with preprocessing pipelines, feature engineering, and deployment environments.

DECLARATION OF COMPETING INTERESTS

The author declares no known competing financial interests or personal relationships that could have influenced the work reported in this paper.

ACKNOWLEDGMENT

The author extends appreciation to Umm Al-Qura University, Saudi Arabia for funding this research work through grant number: 26UQU4340018GSSR01.

DATA AVAILABILITY

The data used in this study were collected from the Breast Cancer Wisconsin (Diagnostic) dataset [16, 17], Diabetes dataset [18, 19], and California Housing dataset [20, 21].

FUNDING STATEMENT

This research work was funded by Umm Al-Qura University, Saudi Arabia under grant number: 26UQU4340018GSSR01.

REFERENCES

[1] D. Sculley *et al.*, "Hidden Technical Debt in Machine Learning Systems," in *NIPS'15: Proceedings of the 29th International Conference on Neural Information Processing Systems*, Montreal, Canada, Dec. 2015, vol. 2, pp. 2503–2511.

[2] T. Gebru *et al.*, "Datasheets for Datasets," *Communications of the ACM*, vol. 64, no. 12, pp. 86–92, Dec. 2021, https://doi.org/10.1145/3458723.

[3] J. Yuan *et al.*, "Understanding and Mitigating Numerical Sources of Nondeterminism in LLM Inference." arXiv, Oct. 24, 2025, https://doi.org/10.48550/arXiv.2506.09501.

[4] I. Gonzalez Pepe, Y. Chatelain, G. Kiar, and T. Glatard, "Numerical Stability of DeepGOPlus Inference," *PLOS ONE*, vol. 19, no. 1, Jan. 2024, Art. no. e0296725, https://doi.org/10.1371/journal.pone.0296725.

[5] G. Kiar *et al.*, "Numerical Uncertainty in Analytical Pipelines Lead to Impactful Variability in Brain Networks," *PLOS ONE*, vol. 16, no. 11, Nov. 2021, Art. no. e0250755, https://doi.org/10.1371/journal.pone.0250755.

[6] G. Kiar *et al.*, "Comparing Perturbation Models for Evaluating Stability of Neuroimaging Pipelines," *The International Journal of High Performance Computing Applications*, vol. 34, no. 5, pp. 491–501, Sep. 2020, https://doi.org/10.1177/1094342020926237.

[7] M. Andrysco, R. Jhala, and S. Lerner, "Printing Floating-Point Numbers: A Faster, Always Correct Method," in *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, St. Petersburg, FL, USA, Jan. 2016, pp. 555–567, https://doi.org/10.1145/2837614.2837654.

[8] U. Adams, "Ryū: Fast Float-to-String Conversion," in *Proceedings of the 39th ACM SIGPLAN Conference on Programming Language Design and Implementation*, Philadelphia, PA, USA, Jun. 2018, pp. 270–282, https://doi.org/10.1145/3192366.3192369.

[9] J. Champagne Gareau and D. Lemire, "Converting Binary Floating-Point Numbers to Shortest Decimal Strings: An Experimental Review," *Software: Practice and Experience*, vol. 56, no. 4, pp. 462–478, Apr. 2026, https://doi.org/10.1002/spe.70056.

[10] T. Johnson III and S. A. Mostafa, "Impact of Data Perturbation for Statistical Disclosure Control on the Predictive Performance of Machine Learning Techniques," *Journal of Data Science*, vol. 23, no. 2, pp. 312–331, Jan. 2025, https://doi.org/10.6339/25-JDS1186.

[11] S. Gupta, A. Agrawal, K. Gopalakrishnan, and P. Narayanan, "Deep Learning with Limited Numerical Precision," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, Jul. 2015, vol. 37, pp. 1737–1746.

[12] P. Micikevicius *et al.*, "Mixed Precision Training." arXiv, 2017, https://doi.org/10.48550/ARXIV.1710.03740.

[13] D. Kalamkar *et al.*, "A Study of BFLOAT16 for Deep Learning Training." arXiv, 2019, https://doi.org/10.48550/ARXIV.1905.12322.

[14] B. Jacob *et al.*, "Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, Jun. 2018, pp. 2704–2713, https://doi.org/10.1109/CVPR.2018.00286.

[15] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A Survey of Quantization Methods for Efficient Neural Network Inference," in *Low-Power Computer Vision*, 1st ed., Boca Raton: Chapman and Hall/CRC, 2022, pp. 291–326.

[16] W. Wolberg, O. Mangasarian, N. Street, W. Street, "Breast Cancer Wisconsin (Diagnostic)." UCI Machine Learning Repository, 1993, https://doi.org/10.24432/C5DW2B.

[17] W. N. Street, W. H. Wolberg, and O. L. Mangasarian, "Nuclear Feature Extraction for Breast Tumor Diagnosis," presented at the IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology, San Jose, CA, USA, Jul. 1993, pp. 861–870, https://doi.org/10.1117/12.148698.

[18] T. Hastie, B. Efron, I. Johnstone, and R. Tibshirani, "Diabetes Data." North Carolina State University, 2004, [Online]. Available: https://www4.stat.ncsu.edu/~boos/var.select/diabetes.html.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *The Annals of Statistics*, vol. 32, no. 2, Apr. 2004, https://doi.org/10.1214/009053604000000067.

[20] L. Torgo, "California Housing Prices." Kaggle, Apr. 2025, [Online]. Available: https://www.kaggle.com/datasets/camnugent/california-housing-prices/data.

[21] R. K. Pace and R. Barry, "Sparse Spatial Autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, May 1997, https://doi.org/10.1016/S0167-7152(96)00140-X.

[22] S. Sudianto, A. Sa'adah, and B. F. Arkana, "Utilization of Adaptive Machine Learning for Streaming Sentiment Analysis: The Effects of Batch and Drift Types," *Engineering, Technology & Applied Science Research*, vol. 16, no. 1, pp. 32384–32390, Feb. 2026, https://doi.org/10.48084/etasr.16379.