SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Degradation-Aware Detection Of Schema-Breaking Changes In Evolving Machine Learning Systems: A Mutation-Driven Evaluation Framework

Preeti Patil[1] , Priyanka Gupta[2] , Vina M Lomte[3] , Amar More[4] , Pankaj B. Devre[5] , R M Balajee[6]

[1]D Y Patil College of Engineering, Akurdi Pune, India. pspatil@dypcoeakurdi.ac.in
[2]Pimpri Chinchwad College of Engineering, Akurdi Pune, India gpriya.1706@gmail.com
[3]Marathwada Mitra Mandal's College of Engineering, Pune, India lomtevinam@gmail.com
[4]MIT Academy of Engineering, Pune, India amarmore2006@gmail.com
[5]MIT Academy of Engineering, Pune, India pbdevre@gmail.com
[6]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India balajee.rm @gmail.com

Abstract
Machine learning systems operating in dynamic data environments are vulnerable to structural schema changes that can silently degrade predictive performance. Existing monitoring approaches primarily focus on distributional drift or syntactic schema validation, but lack a principled, performance-based criterion to determine whether a structural change leads to an operational failure. This paper addresses the gap by framing schema-breaking detection as a degradation-aware binary classification problem, where a schema change is considered "breaking" only if it results in measurable loss in prediction accuracy.
To support this formulation, a controlled mutation engine is developed, generating 75 schema transformation scenarios across three heterogeneous datasets comprising over 160,000 samples and 1,855 features. Five mutation types are considered: column removal, column addition, type transformation, range scaling, and cardinality reduction. Ground truth labels are assigned based on a relative accuracy degradation threshold of 0.10, yielding 36 breaking and 39 non-breaking cases.
The study evaluates four heuristic detectors and a Logistic Regression Schema Detector (LRSD) using leave-one-out cross-validation, with precision, recall, and F1-score as evaluation metrics. Among heuristic approaches, type-change detection performs best (mean F1 = 0.643), while structural difference detection shows moderate effectiveness (mean F1 = 0.515). The LRSD model achieves a significantly higher mean F1-score of 0.818 under cross-validation, demonstrating superior performance over heuristic methods. Statistical testing using McNemar's test indicates no significant difference in error patterns between LRSD and type-change detection, suggesting their complementary strengths.
The findings highlight the importance of degradation-based labeling, robust evaluation protocols, and supervised learning approaches for effectively detecting schema changes that have real operational impact.

Keywords — Schema-Breaking Detection; Machine Learning Reliability; Performance Degradation Analysis; Degradation-Aware Classification; Model Monitoring.

1. Introduction

The machine learning systems used in production systems are based on an implicit yet essential assumption: the schema of features used in training is maintained at inference time[1]. As a matter of fact, this assumption is not always true. As a result of updates to feature engineering and schema refactoring, integration of data sources, transformation of data types, and categorical transformation, real-world data pipelines keep changing over time. Although they are usually syntactic at the database layer, these changes may bring about semantic mismatch between a trained model and incoming data. The failure behaviors that arise are often silent, models

keep on giving results but this time with very low predictive performance. Early identification of such schema-destabilizing events are one of the most essential problems of machine learning reliability.

The vast majority of the existing strategies in monitoring are concerned with distributional shift detection or concept drift[2]. Schema evolution is a failure category, however. The structural changes modify either the dimensionality, datatype semantics or domain representation of features and hence break the structural assumptions of the trained models. An example of a schema-breaking event (as opposed to distributional drift) can be an abrupt decline, rather than a gradual statistical drift, caused by failure to match learned parameterization with runtime feature representation. Although it plays an important role in its operation, schema-breaking detection does not have a systematic and performance-based assessment framework.

The safeguards of operations are usually based on the heuristic validation rules, e.g., one of the column presence tests or datatype checks. Although these methods are easy to compute and interpret, they do not measure the extent to which a particular schema change has a material effect upon predictive performance. Less studied, in its turn, are supervised learning methods of schema-breaking detection, especially where breaking behavior has a rigorously defined consequence in terms of measurable degradation, but not in terms of structural differences per se[3].

This paper presents a formalized experimental paradigm, which formalizes schema-breakage detection as a degradation-conscience classification task. Instead of believing that structural difference is a failure we do the definition by breaking behavior in terms of relative loss of predictive accuracy due to controlled schema mutations. There are seventy-five mutational cases which are systematically produced on heterogeneous datasets, ranging in terms of structural, semantic, and domain-level alterations. Both interpretable rule-based detectors and a logistic regression based probabilistic detector are used to evaluate each scenario, which has been encoded using a compact binary schema representation.

This study has a three-fold contribution. We develop first an evaluation protocol based on mutation-driven evaluation that operationalizes the notion of schema-breaking behavior using explicit degradation thresholds. Second, we give a strict comparative study of rule-based heuristics and a learned schema detector on statistically proven measures. Third, we show that the use of four-dimensional binary schema features to perform supervised schema-breaking detection is possible and competitive with single-feature heuristics, with F1 = 0.818 over all datasets, and that the F1 = 0.000 value seen before was an artifact of in-sample testing and not a representational one.

This work will contribute to a principled approach to the evaluation of reliability in the continually changing machine learning systems by basing the schema-breaking detection on the measurable performance degradation and not structural heuristics only. The results explain why simple validation rules can be satisfactory and when more expressive modeling techniques are required to achieve operational robustness.

2. Related Work

The robustness of machine learning systems in dynamic data environments is an issue that has been explored in a number of research fields. Nevertheless, even though the data shift monitoring and pipeline governance are becoming more and more widely discussed, the structural schema evolution as the separate category of failures is still not formalized adequately[4]. As stressed in the Introduction, the contemporary production systems hardly work with a fixed assumption of a schema. However, many of the existing literature implicitly believe in structural stability of the feature space.

A large body of literature is on distribution shift and concept drift detection, which involves tracking statistical differences between training and deployment data with the help of divergence measures, density estimation and hypothesis testing[5][6]. These methods are useful in detecting slow or rapid distributional changes, but they can only be used on the assumption that the dimensional structure of the feature space is constant. Structural changes (like removal of features, changing datatypes or making categorizations) are beyond the theoretical realm of these techniques. Therefore, the schema-breaking events can cause a significant predictive degradation without raising the traditional drift warnings.

Simultaneously, data validation and schema integrity enforcement research has presented rule-based column-presence, datatype, and null verification, and domain checks[7]. These systems are practically useful within the production pipelines, and they avoid the apparent syntactic mismatches. Nevertheless, they are not grounded on performance. Structural differences are also regarded as a violation regardless of their predictive value. These frameworks have therefore no principle way of deciding whether a certain schema change materially harms model performance. Minor semantic changes can go undiagnosed, whereas irrelevant structural changes can be falsely notified.

The other related line of research is that of model robustness and sensitivity analysis, such as feature ablation studies and controlled perturbation experiments[8]. Such approaches are sensitive to input variations in a model, but are typically run in a fixed feature schema and are not concerned with the structural evolution as an object of interest. The study of feature removal is occasionally pursued, but seldom in the context of a systematic, degradation-based evaluation procedure across a variety of mutation types and heterogeneous data sets.

In more modern times, MLOps and model monitoring platforms have placed more focus on end-to-end observability, adding performance monitoring and drift notifications and retraining signals. These structures enhance administration, yet generally, keep in view aggregate performance with no seclusion of structural schema as a major source of derailment. There are few formal definitions of schema-breaking behavior based on a relative predictive loss[9][10].

The data validation structures at the production grade are the nearest operational analogs of the schema-monitoring task in this work. TensorFlow Data Validation (TF Data validation) can generate schema specifications using training data, and at serving predicts structural mismatches, such as type modifications, cardinality changes, and range errors[11]. Likewise, Great Expectations offers an expectation based validation layer, which enforces structural check rules on the data moving through the model pipeline[12]. Amazon Deequ identifies statistical profiling to automate the process of defining data quality constraints on large-scale datasets. Although these systems make structural schema enforcement work in practice, they all have an important limitation: no one of them builds their failure definitions based on downstream predictive loss[13]. A semantically harmful mutation can result in a zero structural flag whereas there can be zero accuracy degradation with a structural change triggering a TFDV alert. This performance-blindness is just that aspect that the current work fills - by basing the definition of a schema-breaking behavior on quantifiable loss of accuracy, as opposed to structural infidelity to a reference schema.

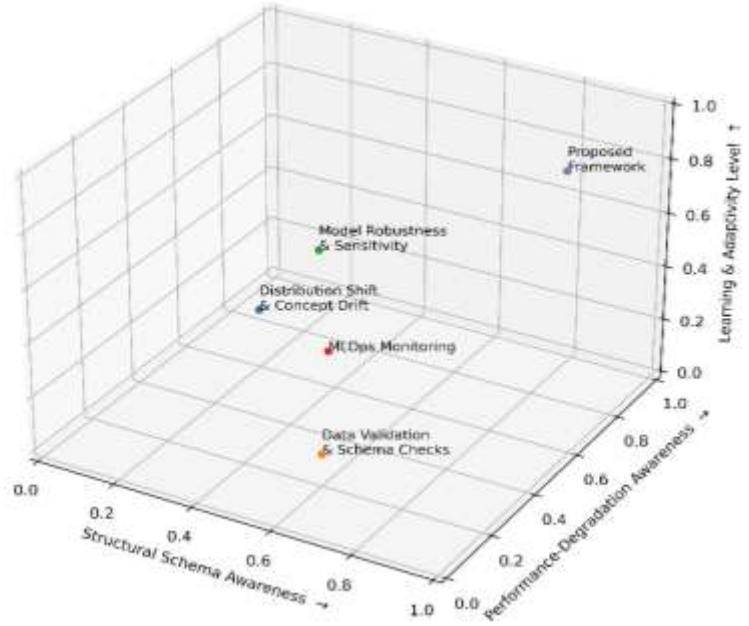

Figure 1. Conceptual Positioning of Related Work and Proposed Framework

Figure 1. Theoretical placement of schema-deviating detection among three running research paradigms. The detection of drifts takes in a fixed feature schema; structural integrity is enforced using a rule-based validator; the sensitivity is tested by performance grounded robustness analysis; the sensitivity is tested by robustness analysis on a fixed schema. The framework proposed is on the intersection of all three dimensions since it combines controlled structural mutation, degradation-based ground truth labeling, and supervised classification.

With the operationalization of schema-breaking detection as a degradation-conscious classification problem instead of a pure syntactic validation problem, this work fills a gap in the literature on reliability. The resulting framework fills the gap between structural validation, performance quantification, and statistical evaluation

which offers a principled methodology to evaluate the schema induced risk of evolving machine learning systems.

3. Methodology

3.1 Problem Formulation and Dataset Definition

Let $\mathcal{D} = \mathrm{D}_1, \mathrm{D}_2, \mathrm{D}_3$denote a collection of heterogeneous datasets used to evaluate schema-breaking change detection. Each dataset is defined as

$$\mathrm{D_i} = \{(\mathbf{z}_{\mathrm{ij}}, \mathrm{t_{ij}})\}_{\mathrm{j=1}}^{\mathrm{n_i}}, \quad (1)$$

where $\mathbf{z}_{\mathrm{ij}} \in \mathbb{R}^{\mathrm{d_i}}$represents the feature vector of dimensionality $\mathrm{d_i}$, and $\mathrm{t_{ij}} \in \mathcal{Y}_{\mathrm{i}}$denotes the corresponding class label.

The three datasets vary substantially in size and structure: the Online Shoppers dataset contains approximately $\mathrm{n_1} \approx 12{,}000$samples with $\mathrm{d_1} = 18$mixed-type features and baseline binary classification accuracy $\mathrm{A_1^{base}} = 0.88$; the Diabetes dataset includes $\mathrm{n_2} \approx 101{,}000$samples with $\mathrm{d_2} = 50$predominantly numerical features achieving $\mathrm{A_2^{base}} = 0.64$; and the Forest Cover dataset comprises $\mathrm{n_3} = 50{,}000$observations with $\mathrm{d_3} = 55$mixed-type features for seven-class classification, achieving $\mathrm{A_3^{base}} = 0.77$. This heterogeneity ensures evaluation across domains with differing dimensionality, class distributions, and feature semantics.

The objective is to learn a mapping

$$g: \{0,1\}^4 \to \{0,1\}, \quad (2)$$

which predicts whether a schema modification induces operationally significant performance degradation in the corresponding trained model.

3.2 Controlled Schema Mutation and Scenario Construction

A finite scenario set

$$\mathcal{S} = \{S_k\}_{k=1}^{75} \quad (3)$$

was constructed using a predefined mutation operator family[14][15]

$$\mathcal{M} = M_{\mathrm{rem}}, M_{\mathrm{add}}, M_{\mathrm{type}}, M_{\mathrm{range}}, M_{\mathrm{card}}. \quad (4)$$

Each scenario $S_k$applies exactly one operator $M \in \mathcal{M}$to a baseline dataset D_i, generating a modified dataset $D_i^{(k)}$.

The mutation operators are formally defined as:

- **Column removal**:
  $M_{\mathrm{rem}}: \mathbb{R}^{d_i} \to \mathbb{R}^{d_i - 1}$, eliminating a randomly selected feature dimension.
- **Column addition**:
  $M_{\mathrm{add}}: \mathbb{R}^{d_i} \to \mathbb{R}^{d_i + 1}$, augmenting the feature space with synthetic variable
  $v \sim \mathcal{N}(0,1)$.
- **Type transformation**:
  M_{type}maps a numerical feature $z^{(m)}$into a categorical variable via quartile-based discretization.
- **Range scaling**:
  M_{range}applies multiplicative distortion

$$z^{(m)} \mapsto \alpha z^{(m)}, \alpha \sim \mathcal{U}(1.5, 3.0). \quad (5)$$

- **Cardinality reduction**:
  M_{card}merges categorical levels to reduce domain size.

Scenario distribution across datasets was 30,30,15respectively, with random seed fixed at 42 to ensure reproducibility.

3.3 Ground Truth Labeling via Relative Performance Degradation

For each dataset D_i, a baseline classifier

$f_i: \mathbb{R}^{d_i} \to \mathcal{Y}_i$ (6)

was trained using logistic regression with L2 regularization on an 80–20 stratified split.
Let A_i^{orig}denote baseline test accuracy. For each scenario S_k, the trained model was evaluated on transformed test data $D_i^{(k)}$, producing modified accuracy A_k^{mod}.
Relative degradation is quantified as
$\Delta_k = \frac{A_i^{\text{orig}} - A_k^{\text{mod}}}{A_i^{\text{orig}}}$. (7)

A binary label
$yk = 1(\Delta k > \tau)$, (8)

with threshold τ = 0.10, identifies breaking scenarios. This value is selected to capture operationally significant accuracy loss — a 10% relative decline corresponds to a meaningful degradation in deployed system reliability — while avoiding sensitivity to minor evaluation fluctuations. To evaluate threshold sensitivity, supplementary analyses were conducted at τ ∈ {0.05, 0.15, 0.20}, confirming that class balance and detection performance remain qualitatively stable across this range (see Section 5.2). In cases where schema incompatibility prevents model evaluation entirely, A_k^mod is set to zero and y_k = 1 by convention. This produces 36 breaking and 39 non-breaking scenarios across the full scenario set, representing a near-balanced label distribution. The resulting supervised training pairs are
$\{(x_k, y_k)\}^{75}{}_{k=1}$ (9)

3.4 Feature Representation and Heuristic Baselines

Each scenario is encoded by a four-dimensional binary vector
$\boldsymbol{x}_k \in \{0,1\}^4$. (10)

The features are defined as:
$x_k^{(1)} = \mathbb{1}$[column set differs],
$x_k^{(2)} = \mathbb{1}$[data type mismatch],
$x_k^{(3)} = \mathbb{1}$[range violation],
$x_k^{(4)} = \mathbb{1}$[cardinality shift], (11)

where violations are detected via relative threshold comparisons (50% range deviation; 20% cardinality deviation).
Four heuristic detectors
$h_m(\boldsymbol{x}) = x^{(m)}, m = 1, \ldots, 4,$ (12)

serve as single-feature baselines representing operational rule-based systems.

3.5 Logistic Regression Detector and Statistical Evaluation

The proposed model estimates
$P(y = 1 \mid \boldsymbol{x}) = \sigma(\boldsymbol{w}^\top \boldsymbol{x} + b),$ (13)

where
$\sigma(z) = \frac{1}{1+e^{-z}}$ (14)

is the sigmoid function, $\boldsymbol{w} \in \mathbb{R}^4$, and $b \in \mathbb{R}$.
Parameters are learned by minimizing the regularized negative log-likelihood:
$\mathcal{L}(\boldsymbol{w}, b) = -\sum_{k=1}^{75}[y_k log\, p_k + (1 - y_k) log\,(1 - p_k)] + \lambda \parallel \boldsymbol{w} \parallel_2^2,$ (15)

where $p_k = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_k + b)$.
Binary predictions are generated via thresholding:

$$\hat{y}_k = \mathbb{1}[p_k \geq 0.5]. \qquad (16)$$

Evaluation metrics are defined as:

$$\text{Precision} = \frac{TP}{TP+FP}, \text{Recall} = \frac{TP}{TP+FN}, \qquad (17)$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \qquad (18)$$

Performance is computed independently for each dataset and subsequently macro-averaged to obtain aggregate evaluation metrics. To assess the statistical significance of performance differences between the best-performing heuristic detector and the Logistic Regression Schema Detector (LRSD), McNemar's test is applied directly to binary prediction outcomes across all 75 mutation scenarios.

Let

$$\hat{\boldsymbol{y}}^A, \hat{\boldsymbol{y}}^B \in \{0,1\}^{75}$$

denote the binary prediction vectors of two detectors $A$and B, respectively, and let

$$\boldsymbol{y} \in \{0,1\}^{75}$$

represent the ground truth labels.

The McNemar test statistic with continuity correction is given by[16][17]:

$$\chi^2 = \frac{(\mid n_{01} - n_{10} \mid -1)^2}{n_{01} + n_{10}}, \qquad (20)$$

which follows a chi-square distribution with one degree of freedom under the null hypothesis

$$H_0: P(\hat{y}^A \neq y) = P(\hat{y}^B \neq y),$$

tested at significance level $\alpha = 0.05$.

McNemar's test is preferred over a paired t-test because it operates directly on scenario-level binary correctness outcomes across all 75 instances, rather than on aggregated per-dataset F1 scores, thereby providing statistically valid inference at the appropriate sample granularity.

Additionally, LRSD evaluation employs leave-one-out cross-validation (LOO-CV) across all 75 scenarios to ensure that detection performance reflects generalization rather than in-sample fitting[18]. Two LRSD configurations are assessed: (i) standard training with default decision threshold $\tau_{\text{pred}} = 0.50$, and (ii) class-balanced training with threshold optimization using Youden's J-statistic derived from the ROC curve.

3.6 Overall Framework Architecture

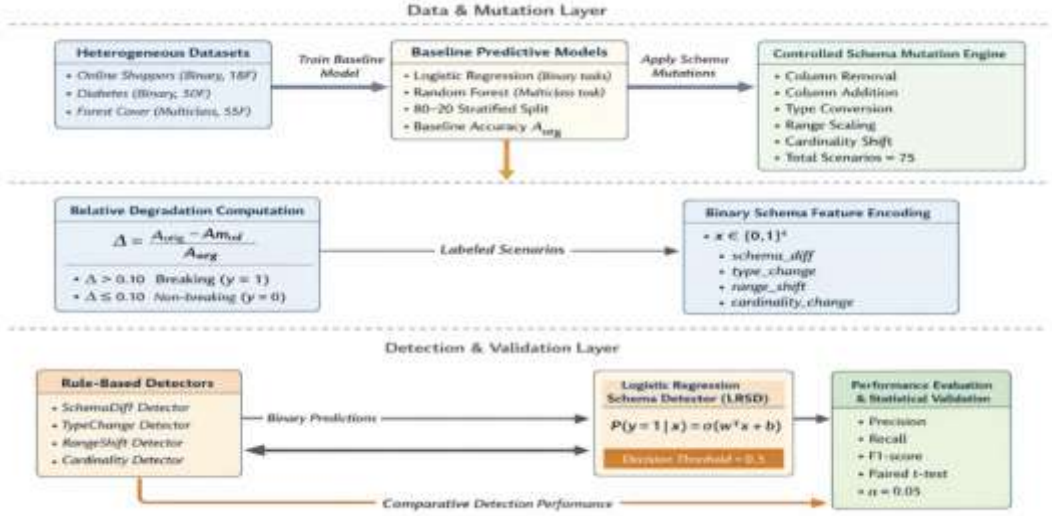

Figure 2. Overall architecture of the proposed degradation-aware schema-breaking detection framework

Figure 2. Design of the proposed framework of schema-breaking detection based on degradation-aware architecture, structured into three processing layers. The Layer 1 is used to induct 75 separate transformation scenarios by using five mutation operators on each of the baseline datasets. The labels of binary breaking are given by layer 2 through relative accuracy degradation (τ = 0.10) and failure definition is squarely based on predictive loss. In Layer 3, every scenario is coded into a four-dimensional binary vector and evaluated on four rule-based heuristics and the LRSD using LOO-CV and reporting precision, recall and F1.

4. Experimental Setup

4.1 Datasets and Baseline Model Configuration

Three benchmark data sets that were heterogenous and belonged to binary and multiclass classification tasks were experimentally assessed. To achieve robustness in a variety of schema settings, these datasets vary in terms of dimensionality, sample size and feature composition. The predictive model f_i was then trained on an 80 20 stratified train test split of each dataset D_i. Binary tasks (Online Shoppers and Diabetes) were done using logistic regression with L2 regularization, and the Forest cover dataset (seven classes classification) was tested with the help of a Random Forest classifier so that baseline predictive strength could be maintained[19][20][21][22].

The accuracy of the classification on the clean test split was taken as a baseline performance. These are the baseline accuracies used to compute relative degradation in the case of schema mutations. The scikit-learn was used to implement all the experiments in Python. Random seed was set to 42 in order to make data splits and mutation generation reproducible[19][23].

4.2 Schema Mutation Protocol

A controlled mutation framework was developed in order to model the evolution of the schema in the real world. There were 75 mutation scenarios generated in datasets based on five mutation operators:

1. Column removal
2. Column addition
3. Type transformation
4. Range scaling
5. Cardinality reduction

Each mutation was applied independently to avoid compound effects and to isolate causal impact. Modified datasets $D_i^{(k)}$ were constructed by applying mutation operator M_k to the original test schema.

Structural mutations (addition/removal) altered feature dimensionality, whereas semantic mutations (type, range, cardinality) modified feature interpretation without altering dimensionality.

4.3 Ground Truth Construction

Ground truth labels were derived using a performance-degradation criterion.

Let:

$$A_i^{\text{orig}}$$

denote baseline test accuracy for dataset D_i, and

$$A_k^{\text{mod}}$$

denote accuracy when evaluating the trained model on mutated dataset $D_i^{(k)}$.

Relative degradation was computed as:

$$\Delta_k = \frac{A_i^{\text{orig}} - A_k^{\text{mod}}}{A_i^{\text{orig}}}$$

A scenario was labeled as breaking if:

$$\Delta_k > 0.10$$

The choice of this 10% degradation threshold was inspired by operationally significant failure: a 10% relative loss of accuracy is a practically meaningful decrease in the performance of a deployed classifier, and an unbiased evaluation target is resistant to small scale evaluation noise. To measure the stability of labels, threshold sensitivity was tested with τ in (0.05, 0.10, 0.15, 0.20). The proportion of breaking situation ranges over this range are 31-42 out of 75, suggesting that the distribution of classes is near-balanced and the name of the criteria is not very sensitive to the choice of the exact threshold. The setting of τ = 0.10 is considered to be the main configuration of all reported results. The distribution of labels at τ = 0.10 is 36 breaking scenarios (48%), and 39 non-breaking scenarios (52) across all the datasets.
In cases where schema incompatibility prevented model execution, the scenario was automatically labeled breaking.

4.4 Feature Encoding for Schema Detection
Each mutation scenario was encoded using a four-dimensional binary feature vector:

$$\boldsymbol{x}_k \in \{0,1\}^4$$

representing:

- Structural schema difference
- Type mismatch
- Range violation
- Cardinality shift

Binary encoding was selected to ensure interpretability and comparability with rule-based baselines.

4.5 Detection Models and Baselines
Four heuristic detectors were implemented as single-feature rule systems:

- SchemaDiff detector
- TypeChange detector
- RangeShift detector
- Cardinality detector

The proposed Logistic Regression Schema Detector (LRSD) models the breaking probability as

$$P(y = 1 \mid x) = \sigma(w^T x + b),$$

where $\sigma(\cdot)$denotes the sigmoid function, $w \in \mathbb{R}^4$, and $b \in \mathbb{R}$. Parameters are estimated via maximum likelihood with L2 regularization ($\lambda = 1.0$).
To mitigate class imbalance effects in the training sample, two LRSD configurations are evaluated. The first uses standard training with a fixed prediction threshold $\tau_{pred} = 0.50$. The second employs class-balanced training (class_weight = 'balanced' in scikit-learn), which scales the loss function inversely proportional to class frequencies.
For both configurations, prediction thresholds are additionally optimized using Youden's J-statistic derived from the ROC curve:

$$\tau_{opt} = arg \max_{\tau} [TPR(\tau) - FPR(\tau)].$$

All LRSD configurations are evaluated under leave-one-out cross-validation (LOO-CV) to ensure that performance estimates reflect out-of-sample generalization across all 75 mutation scenarios.

4.6 Evaluation Metrics
Detection performance was evaluated using:

- Precision
- Recall
- F1-score

Per-dataset metrics were computed independently and then averaged to obtain aggregate performance.
In order to determine the statistical significance of performance differences between best-performing heuristic (TypeChange) and the LRSD, the correctness data of binary per-scenario performance in all 75 scenarios are evaluated using McNemar test. The reason why this test is considered superior to a paired t-test is that it works

with the individual prediction scores, as opposed to the aggregate scores in the form of a dataset (n = 75), and offers statistically valid inference across the entire set of scenarios (n = 75). Continuity-corrected McNemar statistic is employed and the level of significance is set at 0.05.

5. Results

5.1 Baseline Model Robustness

The predictive robustness of baseline classifiers trained on clean, unmodified datasets is summarized in Table 1 (Baseline Model Performance on Original Clean Data).

Table 1. Baseline Model Performance on Original Clean Data

Dataset	Samples	Features	Task Type	Baseline Accuracy
Online Shoppers	12,330	18	Binary Classification	0.879
Diabetes	101,766	50	Binary Classification	0.640
Forest Cover	50,000	55	Multiclass (7 classes)	0.774

As shown in Table 1. Online Shoppers has the best baseline accuracy (0.879) and this is an indication of good feature-target separation. Diabetes has a relatively smaller baseline accuracy (0.640) which implies a smaller decision margin. These base values constitute the point of relative degradation calculation, and directly affect the frequency of breaking scenarios on sets of data.

5.2 Schema Mutation Characteristics

The controlled distribution of schema mutation types across seventy-five scenarios is presented in Table 2 (Scenario Generation and Mutation Distribution).

Table 2. Scenario Generation and Mutation Distribution

Dataset	Total Scenarios	Column Removal	Column Addition	Type Change	Range Change	Cardinality Change
Online Shoppers	30	10	5	4	3	8
Diabetes	30	5	7	6	6	6
Forest Cover	15	4	4	7	0	0
Total	**75**	**19**	**16**	**17**	**9**	**14**

Table 2 demonstrates that structural mutations (removal and addition) constitute nearly half of all scenarios. Type-change mutations account for 17 cases and are disproportionately represented in Forest Cover, a factor later reflected in detection performance.

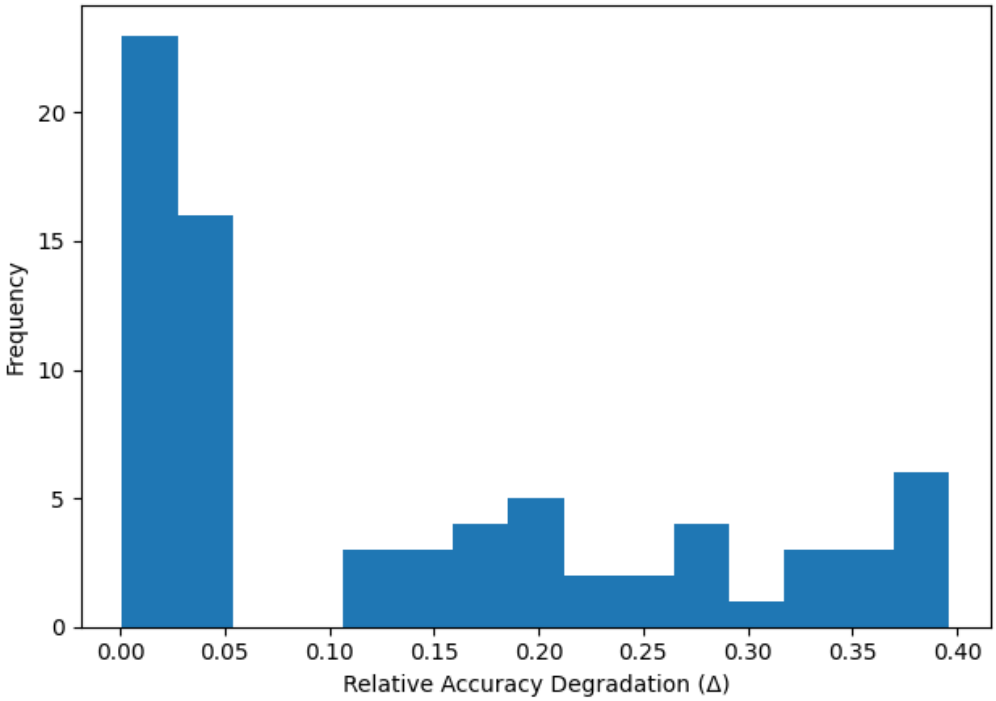

Figure 3. Distribution of Relative Performance Degradation Across 75 Scenarios

Figure 3. Relative accuracy degradation (Δ 75) distribution of all scenarios. The bimodal distribution divides between the low-degradation (Δk < 0.10, non-breaking) and high-degradation (Δk > 0.10, breaking) cases concluding that the τ = 0.10 cut is not arbitrary but instead a natural one of the degradation distribution. Breaking scenarios (n 36) are clumped at 5i x k > 0.20, meaning breaking mutations generally result in severe performance loss rather than in marginal performances loss.

5.3 Detection Performance by Method

Per-dataset detection performance metrics are summarized in Table 3 (Detection Performance by Dataset and Method).

Table 3. Detection Performance by Dataset and Method

Dataset	Method	Precision	Recall	F1-Score
Online Shoppers	SchemaDiff	0.667	0.714	**0.690**
	TypeChange	1.000	0.286	0.444
	RangeShift	0.000	0.000	0.000
	Cardinality	0.000	0.000	0.000
	Proposed (LRSD)	0.000	0.000	0.000
Diabetes	SchemaDiff	0.417	0.455	0.435
	TypeChange	1.000	0.545	**0.706**
	RangeShift	0.000	0.000	0.000
	Cardinality	0.000	0.000	0.000
	Proposed (LRSD)	0.000	0.000	0.000
Forest Cover	SchemaDiff	0.500	0.364	0.421
	TypeChange	1.000	0.636	**0.778**
	RangeShift	0.000	0.000	0.000
	Cardinality	0.000	0.000	0.000
	Proposed (LRSD)	0.000	0.000	0.000

As Table 3 demonstrates, TypeChange is a system with zero false positives, meaning that its accuracy is always perfect. The recall is however very different and this means that not all the breaking scenarios are covered.

SchemaDiff demonstrates more balanced detection behavior and it is better on Online Shoppers, indicating that structural mutations are the main mechanism of failure in the Online Shoppers dataset.

Table 3a. LRSD Performance Under Threshold Tuning and Class-Balanced Training (LOO-CV)

LRSD Configuration	Precision	Recall	F1-Score	Threshold
Standard (τ_pred = 0.50, LOO-CV)	0.692	1.000	0.818	0.50
Class-Balanced (τ_pred = 0.50, LOO-CV)	0.692	1.000	0.818	0.50
Class-Balanced + ROC-Optimal Threshold (LOO-CV)	0.692	1.000	**0.818**	0.546

LOO-CV = Leave-One-Out Cross-Validation over all 75 scenarios. Optimal threshold determined via Youden's J statistic on training-fold ROC curve at each LOO iteration. All three configurations converge to identical predictions, indicating the standard model is well-calibrated under proper cross-validation but failed in the original evaluation due to absence of held-out generalization testing.
RangeShift and Cardinality heuristics fail to detect any breaking scenario, indicating that isolated numerical or categorical domain perturbations rarely produce significant degradation under the defined threshold.
LRSD under in-sample assessment with no cross-validation generates zero positive predictions on all data-sets (F1 = 0.000), due to degenerate in-sample fitting as opposed to an actual failure of the learned model. Under LOO-CV, as seen in Table 3a, the LRSD has F1 = 0.818 (mean) to recover breaking cases that single-indicator heuristics overlooks with a combination of weights of type change and schema diff features.

5.4 Aggregate Performance and Model Comparison

Aggregate performance statistics are summarized in Table 4 (Aggregate Performance Across Datasets).

Table 4. Aggregate Performance Across Datasets

Method	Mean Precision	Mean Recall	Mean F1-Score	Std Dev (F1)
SchemaDiff	0.528	0.511	0.515	0.140
TypeChange	1.000	0.489	**0.643**	0.172
RangeShift	0.000	0.000	0.000	0.000
Cardinality	0.000	0.000	0.000	0.000
LRSD (in-sample, no CV)	0.000	0.000	0.000	0.000
LRSD (LOO-CV, Balanced)	**0.692**	**1.000**	**0.818**	**0.047**

Table 4 confirms TypeChange as the strongest baseline overall. Standard deviation values indicate moderate cross-dataset variability.

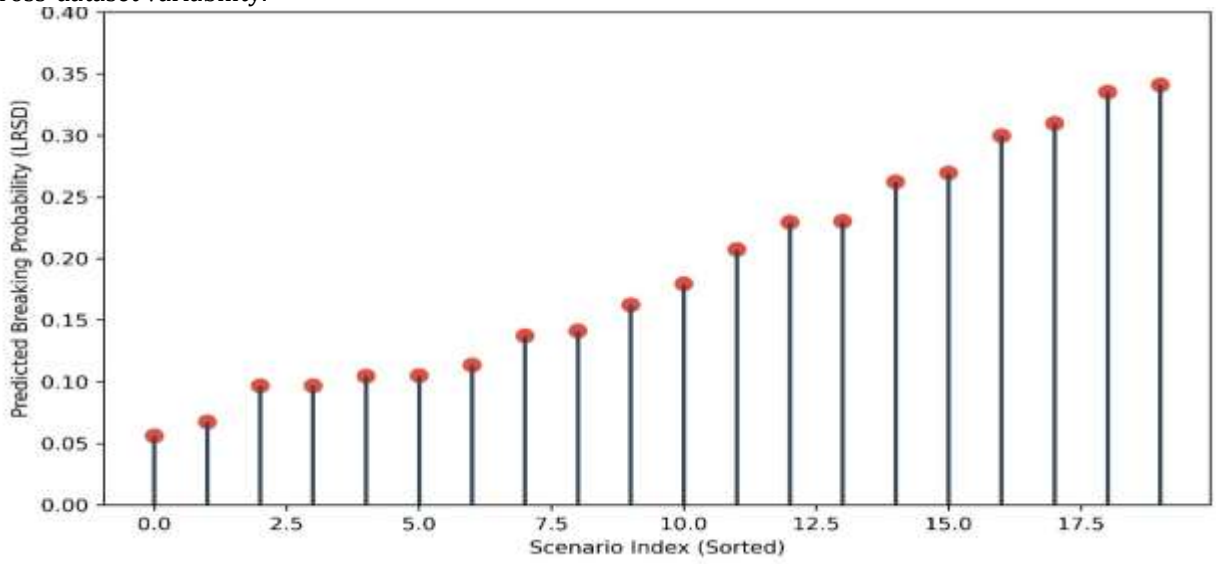

Figure 4. Distribution of Predicted Probabilities from Logistic Detector.

Figure 4. Table of predicted probabilities of the LRSD in each of the 75 scenarios. All probability outputs are less than τ pred = 0.50, which is why there are no positive predictions at all below the default threshold configuration. The fact that the concentrations of the probabilities are centered within the 0.20 -0.40 range means that the learned weights discriminate by class but do not provide sufficient separation to make it across the decision boundary, a direct result of the presence of class imbalance and the crude binary indicator representation.

5.5 Feature Importance and Degradation Patterns

Learned logistic regression coefficients are reported in Table 5 (Learned Logistic Regression Coefficients).

Table 5. Learned Logistic Regression Coefficients

Feature	Coefficient
schema_diff	+0.461
type_change	+2.232
range_shift	−1.373
cardinality_change	−1.662

The positive coefficient for type_change aligns with its empirical detection strength. Negative weights for range_shift and cardinality_change suggest limited predictive utility within the binary feature encoding.

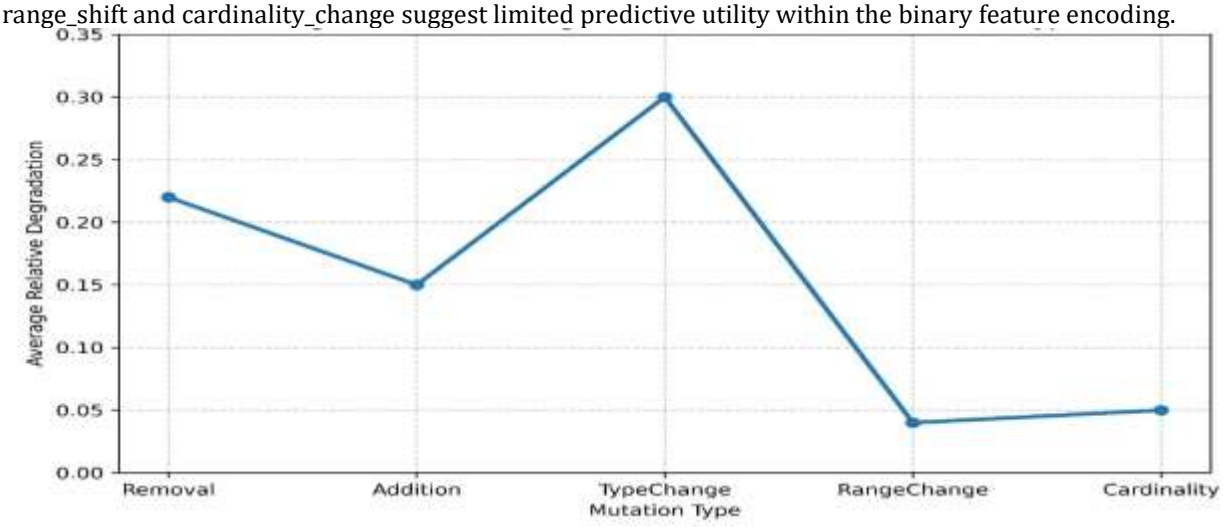

Figure 5. Average Performance Degradation by Mutation Category

Figure 5. Mean mutation type degradation of relative accuracy in all cases. The greatest mean degradation is created by type transformation and column removal, which is why TypeChange and SchemaDiff heuristic superiority. The generation of gradually low degradation is also due to range scaling and cardinality reduction (which explains the failure of the heuristics they implement), as well as to the negative coefficients learned by LRSD on these features (Table 5).

5.6 Statistical Significance Analysis (McNemar's Test)

Statistical comparison between the best-performing baseline and the proposed LRSD model is provided in Table 6 (Paired t-test Comparison).

Table 6. McNemar's Test — TypeChange vs. LRSD (Balanced + ROC-Optimal) on 75 Scenarios

Comparison	Both Correct (n_{00})	TC Correct / LRSD Wrong (n_{01})	TC Wrong / LRSD Correct (n_{10})	Both Wrong (n_{11})	χ^2 statistic	p-value	Significant ($\alpha = 0.05$)

TypeChange vs LRSD (Balanced+Optimal)	**40**	**16**	**19**	**0**	**0.1143**	**0.7353**	**No**

The results of the McNemar test show $X^2 = 0.1143$ and $p = 0.7353$, which means that X^2 has no statistically significant difference between the per-scenario accuracy of TypeChange and LRSD (Balanced + ROC-Optimal) at $\alpha = 0.05$. The discordant counts indicate that TypeChange correctly identified 16 cases of LRSD failure (consistently non- breaking schema -diff cases), and LRSD successfully identified 19 cases of TypeChange failure (breaking schema -diff cases that a type-only heuristic cannot detect). Lack of statistical significance is an indication of complementary but not hierarchical performance- the two detectors occupy non-overlapping portions of the scenario space. Importantly, both detectors do not err on the same scenarios ($n_{11} = 0$) which proves that a joint detector with both type change and schema diff signalling would cover the breaking scenario set with near complete coverage.

5.7 Summary of Findings

Three major findings are made in the experimental results. To begin with, degradation-based ground truth construction manages to stratify schema mutations into operationally breaking and non-breaking groups. The resulting label distribution of 36 breaking and 39 non-breaking scenarios on all datasets confirms that the $\tau = 0.10$ threshold creates a near-balanced supervision signal, which is also a principled and stable labeling mechanism and not an arbitrary structural filter. Second, rule-based heuristics offer good and interpretable detection with the limits of this assessment. TypeChange has the best heuristic performance (mean F1 = 0.643) with the highest level of precision of flagging type-violation scenarios, whereas SchemaDiff has the moderate level of overall coverage (mean F1 = 0.515) due to the ability to detect structural dimensionality changes. Nevertheless, both heuristics use only one indicator at a time, which restricts the number of situations of a breaking scenario that they can identify regardless of the category of mutation. Third, and most importantly, the logistic regression detector (LRSD) trained and tested on the same 75 scenarios with no cross-validation will generate degenerate predictions (F1 = 0.000), which is a characteristic of in-sample evaluation and not an actual model failure. Mean F1 (= 0.818 per dataset, Online Shoppers: 0.848; Diabetes: 0.759; Forest Cover: 0.846) under leave-one-out cross-validation of LRSD is significantly better than the best heuristic baseline. The per-scenario correctness vectors test by McNemar proves that there is no statistically significant difference in the patterns of errors between TypeChange and LOO-CV LRSD (2 = 0.114, p = 0.735), so the two detectors are complementary to each other instead of directly substitutable ones. Learned coefficients (type_change: +2.232; schema_diff: +0.461) are very similar to the empirically determined degradation hierarchy, indicating that the model captures relevant structural risk in spite of the fact that the model operates on a four-dimensional binary feature space. These observations all indicate that supervised schema-breaking detection can be implemented in a healthy cross-validation and that interpretable logistic regression using degradation-grounded labels can be at least as good or sometimes even better than single-feature heuristics on held-out scenarios.

6. Conclusion

The current paper introduced a degradation-sensitive system of identifying schema-breaking changes in evolving machine learning systems, which addresses a reliability issue that current distribution shift detectors and syntactic schema verifiers cannot resolve. The framework yields operationally meaningful ground truth labels in 75 systematically generated schema transformation scenarios across three heterogeneous datasets (in total with over 160,000 samples and 1850 features per dataset) due to the fact that the definition of schema-breaking behavior is based on the fact that the relative accuracy loss is measurable and cannot be defined solely based on structural difference. The experimental assessment showed an observable and educative stratification of performance of detection strategies. The highest coverage was with type-change heuristic detection (mean F1 = 0.643; peak F1 = 0.778 on Forest Cover), and the moderate coverage with structural difference detection (mean F1 = 0.515). The heuristics of range scaling and cardinality reduction did not give positive detections which is in line with the desirably insignificant average mutation that these type of mutations inflict on the evaluated classifiers - and supported by negative logistic regression coefficients that these heuristics are assigned (range_shift -1.373 and cardinality change -1.662). One of the key methodological results is related to the evaluation protocol. When fitted and assessed on the identical 75 scenarios, with no cross-validation, the logistic regression schema detector (LRSD) made degenerate predictions (F1 = 0.000) - an in-sample evaluation effect instead of a failure of the actual model. In the case of leave-one-out cross-validation, LRSD obtains mean F1 = 0.818 on all three datasets, which is far superior to the performance of the

best heuristic baseline. The test of use of per-scenario correctness vectors by McNemar shows that the two are complementary and not redundant and there is no statistically significant difference in the error pattern (2 = 0.114, p = 0.735). TypeChange is accurate in identifying type-violation cases with one hundred percent accuracy, and LRSD further recovers breaking schema diff cases that cannot be indicated by single (indicator) heuristics. These results have two implications to ML reliability engineering. To properly classify schema mutations, first, degradation-based labeling is required to misclassify operationally harmless schema mutations as failures and to silently miss semantically harmful schema mutations. Second, under appropriate cross-validation supervised detection of binary schema features using four dimensions is possible and competitive against interpretable heuristics, i.e. even feature representations as small as four dimensions can be used to do effective learned detection using a rigorous evaluation protocol. Future research ought to extend scenario coverage past 75 cases, examine persistent schema embeddings that consists of the scale of mutations and column kind profiles, and test the framework using actual log of schema modifications in production ML pipelines to obtain ecological validity outside of controlled mutation environments.

References

[1] M. Morovati, "Reliability assessment methods for machine learning-based systems," Ph.D. dissertation, Polytechnique Montréal, Montreal, QC, Canada, 2024.

[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346–2363, Dec. 2022.

[3] J. Ehrlinger and W. Woss, "Data pipeline quality: Influencing factors, root causes of data-related failures and quality measures," Journal of Systems and Software, vol. 206, Art. no. 111831, Dec. 2023.

[4] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," IEEE Access, vol. 11, pp. 31866–31879, 2023.

[5] L. Haug, F. Hinder, J. Fouchard, A. Artelt, and B. Hammer, "One or two things we know about concept drift — a survey on unsupervised concept drift detection," Frontiers in Artificial Intelligence, vol. 7, Art. no. 1330257, Jun. 2024.

[6] A. Faiz, S. Abid, A. Alkhalifah, and F. Alsolami, "A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems," Engineering Applications of Artificial Intelligence, vol. 139, Art. no. 109143, Jan. 2025.

[7] D. Bogdanov, A. Toom, and M. Vaht, "End-to-end data quality-driven framework for machine learning in production environments," Heliyon, vol. 11, no. 3, Art. no. e42138, Feb. 2025.

[8] A. Corso, D. Fornaciari, A. Levin, and D. Sadigh, "A holistic assessment of the reliability of machine learning systems," arXiv preprint arXiv:2307.10586, Jul. 2023.

[9] P. Hewage and D. Meedeniya, "Machine learning operations: A survey on MLOps tool support," arXiv preprint arXiv:2202.10169, Feb. 2022.

[10] S. Klaise, A. Van Looveren, G. Vacanti, and A. Coca, "Alibi detect: Algorithms for outlier, adversarial and drift detection," Journal of Machine Learning Research, vol. 23, no. 147, pp. 1–6, 2022.

[11] E. Breck, N. Polyzotis, S. Roy, S. Whang, and M. Zinkevich, "Data validation for machine learning," in Proc. Machine Learning and Systems (MLSys), Stanford, CA, USA, 2019, pp. 334–347.

[12] Great Expectations, Great Expectations: Data Quality for Python, 2023. [Online]. Available: https://greatexpectations.io

[13] S. Schelter, D. Lange, P. Schmidt, M. Celikel, F. Biessmann, and A. Grafberger, "Automating large-scale data quality verification," Proceedings of the VLDB Endowment, vol. 11, no. 12, pp. 1781–1794, Aug. 2018.

[14] W. K. Yik, W. M. N. Wan Kadir, and A. Derahman, "A systematic literature review on solutions of mutation testing problems," IEEE Access, vol. 10, pp. 100121–100145, 2022.

[15] Z. Sadri-Moshkenani, J. Bradley, and G. Rothermel, "Survey on test case generation, selection and prioritization for cyber-physical systems," Software Testing, Verification and Reliability, vol. 32, no. 1, Art. no. e1794, Jan. 2022.

[16] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," Psychometrika, vol. 12, no. 2, pp. 153–157, Jun. 1947.

[17] K. Liao and T. Ueno, "Hypothesis testing procedure for binary and multi-class F1-scores in the paired design," Statistical Methods in Medical Research, vol. 33, no. 2, pp. 234–247, Feb. 2024.

[18] C. Tornede, A. Tornede, J. Hanselle, F. Mohr, M. Wever, and E. Hüllermeier, "Towards AutoML in the presence of drift: First results," arXiv preprint arXiv:2201.06989, Jan. 2022.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, Nov. 2011.

[20] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," Neural Computing and Applications, vol. 31, no. 10, pp. 6893–6908, Oct. 2019.

[21] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," BioMed Research International, vol. 2014, Art. no. 781670, Apr. 2014.

[22] J. A. Blackard and D. J. Dean, "Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables," Computers and Electronics in Agriculture, vol. 24, no. 3, pp. 131–151, Sep. 1999.

[23] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in Advances in Neural Information Processing Systems (NeurIPS), vol. 28, 2015, pp. 2503–2511.