Explanations for Machine Learning Pipelines under Data Drift

Jahid Hasan
Purdue University
West Lafayette, IN, USA
hasan89@purdue.edu

Romila Pradhan
Purdue University
West Lafayette, IN, USA
rpradhan@purdue.edu

Abstract

Ensuring the robustness of data preprocessing pipelines is essential for maintaining the reliability of machine learning model performance in the face of real-world data shifts. Traditional methods optimize preprocessing sequences for specific datasets but often overlook their vulnerability to future data variations. This research introduces a vulnerability score to quantify the susceptibility of preprocessing components to data shift. We propose a Linear Regression approach to establish a predictive relationship between the vulnerability of the pipeline components and changes in the model's performance. The generated relationships act as explanations for practitioners of the system and help them quantify the robustness of the pipeline to data shift. For a given pipeline, we generate an explanation that highlights a tolerable threshold beyond which a component is considered *shift-vulnerable* and is likely to contribute to performance degradation. For the shift-vulnerable scenarios, we further suggest a new pipeline for system maintainers that preserves the model performance without retraining. The proposed framework delivers a risk-aware assessment, empowering practitioners to anticipate potential performance changes and adapt their pipeline strategies accordingly. Experimental results on several real-world datasets generate valid explanations for pipeline robustness and demonstrate the opportunities in this field of research.

CCS Concepts

• **Information systems → Data mining**; • **Computing methodologies → Machine learning**; • **Human-centered computing**;

Keywords

Pipeline Robustness, Data Preparation, Explainable AI, Data Drift

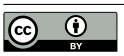
ACM Reference Format:
Jahid Hasan and Romila Pradhan. 2025. Explanations for Machine Learning Pipelines under Data Drift. In *Workshop on Human-In-the-Loop Data Analytics (HILDA' 25), June 22–27, 2025, Berlin, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3736733.3736744

1 Introduction

Machine learning (ML) pipelines have become a cornerstone in modern data-driven decision-making systems. These end-to-end pipelines comprise a sequence of operations (e.g., missing value imputation, outlier detection, encoding, etc) that systematically transform raw data into features for modeling [14, 17]. Optimizing such a pipeline is an essential practice in ML workflow. Conventional approaches aim to find the best sequence of preprocessing components that maximizes the model performance for a given training dataset [7]. Tools such as AutoML [15] have enabled automated selection and tuning of pipeline configuration, offering optimized results when both the training and deployment data are drawn from the same distribution. However, this process is inherently static, assuming that the data distribution remains unchanged post-deployment. In real-world, ML pipelines operate in dynamic scenarios where incoming data may deviate significantly from the learned distribution. This phenomenon is known as **data shift** [23], manifest primarily in two forms: covariate shift [11] and concept shift [8, 19]. Covariate shift refers to changes in the input feature distribution while the underlying correlation between feature and label remains stable. On the other hand, concept shift occurs when the relationship between inputs and outputs changes over time. Such shifts can potentially degrade the model's performance. In this paper, we focus on covariate shift, as it is a significant form of data shift [11]. The effect of covariate shift can be quantified by the shift in first and higher-order statistics [20, 25]. This paper addresses the critical research question: *How vulnerable is an optimized preprocessing pipeline when subjected to data shift?* In other words, we investigate whether a preprocessing pipeline optimized on the original data distribution remains optimal under distributional shift. Consequently, this study offers alternative pipeline suggestions aimed at preserving model performance without requiring an expensive retraining process. This paper introduces a novel perspective on quantifying the vulnerability of preprocessing components using a composite vulnerability score, denoted as VS. Instead of altering the model, we assess how shifts in the data affect each pipeline component and estimate their contributions to downstream performance changes. We synthetically generate several post-deployment shifted scenarios and analyze the corresponding vulnerabilities to generate a comprehensive explanation for the practitioner of the system. This user-facing report includes a robustness analysis of the existing model pipeline under various data shift conditions and proposes a tolerable threshold for VS, beyond which the pipeline is deemed vulnerable and significant performance degradation is likely to occur. In such vulnerable scenarios, we further recommend an alternative sequence of preprocessing steps designed to preserve model performance without the need for retraining. This study empowers users to anticipate potential vulnerabilities within the pipeline and adapt their strategies accordingly.

Summary of contributions. Our main contributions can be summarized as follows:

- We formalize the notion of **pipeline vulnerability** to introduce a new perspective on understanding and addressing data drift in machine learning systems. (Section 3.1)



HILDA' 25, Berlin, Germany

ACM ISBN 979-8-4007-1959-2/2025/06
https://doi.org/10.1145/3736733.3736744

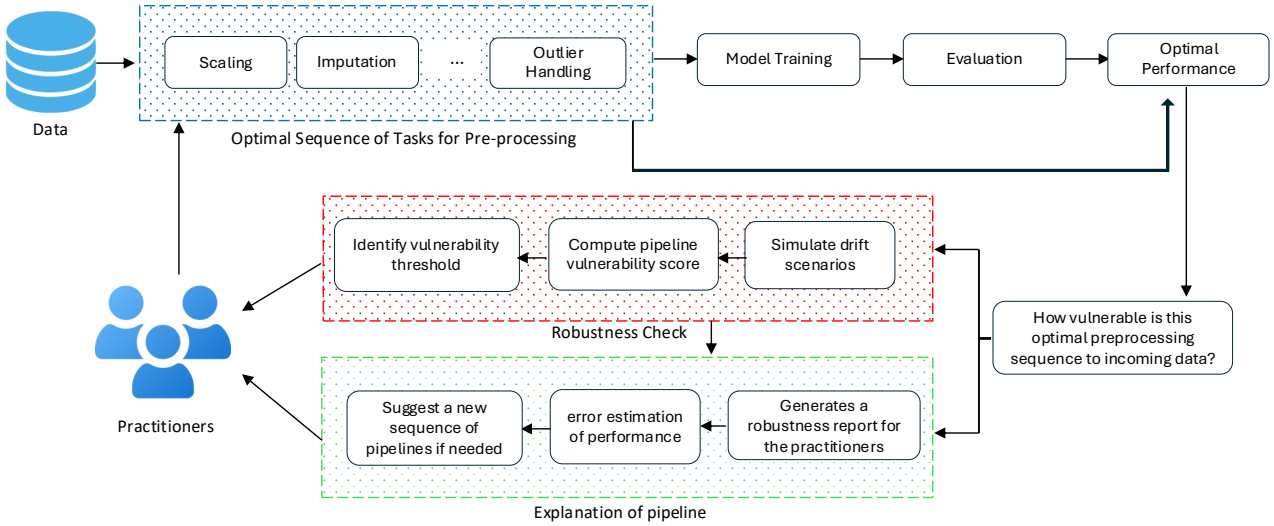


Figure 1: An overview of the proposed framework to generate explanations for practitioners about the robustness of ML pipelines under data drift. The blue-shaded section highlights the selection of the optimal sequence of tasks for preprocessing. The red-shaded section presents the robustness checking component, while the green-shaded section provides explanations.

- We introduce the vulnerability score **(VS)**, which quantifies the susceptibility of pipeline components under drifted data scenarios. (Section 3.1)
- We propose a linear regression approach to establish a relationship between the VS and performance changes. (Section 3.2)
- We generate user-friendly explanations of the preprocessing pipeline for a trained model and propose a tolerable threshold for VS. Beyond this threshold, the pipeline is considered vulnerable, and we suggest an alternative sequence of pipelines to preserve model performance without necessitating retraining. (Section 3.3)
- We conduct experiments on two real-world datasets to validate the research question and generate valid explanations. (Section 4)

We also highlight key research opportunities, outline existing challenges, and propose directions for future work (Section 5).

2 Preliminary

Supervised Learning. We consider a binary supervised learning task for a given dataset $\mathcal{D} = \{X, Y\}$ represent a dataset with $X \in \mathbb{R}^{n \times d}$ as the feature matrix of n samples and d features and $Y \in \mathbb{R}^n$ as the labels. Suppose there is a conditional distribution $p(y \mid \mathbf{x})$ defined over $\mathcal{D}$, where $\mathbf{x} \in X, y \in Y$. Given training dataset $\mathbf{D}_{train} = \{d_i\}_{i=1}^n = \{\mathbf{x}_i, y_i\}_{i=1}^n \in \mathcal{D}$, the learning task is to train a classifier $\mathcal{M}$ that represents a distribution g that captures the target distribution p as closely as possible. $\mathcal{M}$ learns a function $f : X \rightarrow \hat{Y}$ that associates each data point $\mathbf{x}$ with a prediction $\hat{y} = f(\mathbf{x}) \in \{0, 1\}$, and is evaluated on $\mathbf{D}_{test} \in \mathcal{D}$.

Data preprocessing pipeline. Input data $\mathcal{D}$ undergoes a series of transformations (e.g., missing value imputation, outlier handling, normalization) before training a model $\mathcal{M}$. Each such transformation, or preprocessing component, is denoted by P_i. A data preprocessing pipeline, denoted by $\mathcal{P}$, then is a sequence of m transformations that are applied to X before a model is trained:

$$\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$$

Given that dataset X undergoes the operations in pipeline $\mathcal{P}$, the transformed (or, preprocessed) dataset is denoted by $X' = \mathcal{P}(X)$. The preprocessing pipeline $\mathcal{P}$ is often optimized for model performance (e.g., accuracy, mean squared error) on a given dataset. Let $\mathcal{D}_{\text{orig}} = \{\mathbf{X}_{\text{orig}}, \mathbf{Y}_{\text{orig}}\} \in \mathcal{D}$ denote the original dataset used to optimize the pipeline $\mathcal{P}$.

Data drift. Let the original dataset follow the data distribution $\mathcal{D} \sim \mathbb{P}_D$, and let a new, unseen dataset with the same schema drawn from the same domain follow $\mathcal{D}_{\text{shift}} \sim \mathbb{P}_E$. We say $\mathcal{D}_{\text{shift}}$ exhibits data shift if the data distribution of $\mathcal{D}$ is not equal to that of $\mathcal{D}_{\text{shift}}$, i.e., $\mathbb{P}_D \neq \mathbb{P}_E$; in this paper, we consider covariate shift which is very common in real-world scenarios. Since it does not require labeled data from the target domain, it is practical for many applications [20, 23, 25].

Definition 2.1 (Optimal Pipeline). Given dataset $\mathcal{D}$ and a supervised learning algorithm, $\mathcal{P}^*$ denotes the optimal pipeline which is a set of task sequences for data pre-processing that result in the optimal performance(e.g., accuracy, F1 score, loss) for model $\mathcal{M}$ trained on dataset $\mathcal{P}^*(\mathcal{D})$.

Problem Definition. Given the optimal pipeline $\mathcal{P}^*$ on dataset $\mathcal{D}$, our objective is to quantify the vulnerability of the pre-processing pipeline $\mathcal{P}^*$ to data shift and generate explanations for practitioners under unseen data shift scenarios, i.e., when $\mathbb{P}_{\mathcal{D}} \neq \mathbb{P}_{\mathcal{D}_{\text{shift}}}$.

3 Methodology

In this section, we propose a novel vulnerability score to measure the susceptibility of each pre-processing component to covariate shift in data distribution and provide a performance error estimation framework of vulnerability caused by the vulnerability score.

We present our proposed framework in Figure 1 that illustrates an ML workflow followed by data pre-processing, training, and evaluation (upper half of figure). The lower half of the figure shows our research question, with our research efforts on robustness check (depicted in the red-shaded box) and pipeline explanation generation for the practitioner (depicted in the green-shaded box).

3.1 Preprocessing Vulnerability Score (VS)

Given pipeline $\mathcal{P}$ and shifted dataset $\mathbf{X}_{\text{shift}}$, we first quantify the vulnerability score of $\mathcal{P}$ in terms of the sensitivity of each preprocessing component $P_k \in \mathcal{P}$ to the shift $\mathbf{X}_{\text{shift}}$ from the original dataset $\mathbf{X}_{\text{orig}}$ and the resulting dissimilarity in the distribution of their corresponding post-processed datasets. We then collate the vulnerability scores under various data shift scenarios to generate a vulnerability score matrix for the pre-processing pipeline $\mathcal{P}$.

3.1.1 Sensitivity of preprocessing step P_k to $\mathbf{X}_{shift}$. The sensitivity S_{P_k} of a preprocessing component P_k to data drift is calculated as the average absolute difference between the original dataset and the drifted dataset, after component P_k is performed.

$$S_{P_k}(\mathcal{D}_{\text{orig}}, \mathcal{D}_{\text{shift}}) = \frac{1}{nd} \sum_{i=1}^{n} \sum_{j=1}^{d} \left| x_{ij}^{orig} - x_{ij}^{shift} \right| \tag{1}$$

where $x^{orig} \in P_k(\mathbf{X}_{\text{orig}})$ and $x^{\text{shift}} \in P_k(\mathbf{X}_{\text{shift}})$, n is the number of samples, and d is the number of features. Equation 1 captures the first-order shift statistics and calculates the sensitivity of each preprocessing component to the drift immediately after that particular transformation. For instance, in the case of normalization using standardization, we apply the transformation as follows:

$$P_k(\mathbf{X}_{\text{orig}}) = \frac{X - \mu_X}{\sigma_X}$$

$$P_k(\mathbf{X}_{\text{shift}}) = \frac{X' - (\mu_X + \Delta_X)}{\sigma_X + \Delta_\Sigma}$$

where, $\mathbf{X}_{\text{orig}} \sim (\mu_X, \sigma_X^2)$ belongs to the original distribution. We compute the sensitivity of a component P_k using Equation 1 which measures the distributional shift. Since evaluating all components in the entire pre-processing pipeline $\mathcal{P}$ can be computationally intensive, we adopt a sampling strategy to select a representative subset $\mathcal{P}' \subset \mathcal{P}$. To guide this selection, we leverage SHAP (SHapley Additive exPlanations) [9], a data valuation method that quantifies the contribution of each component, enabling us to identify and prioritize the most influential pre-processing steps.

3.1.2 Distribution dissimilarity between processed $\mathbf{X}_{orig}$ and processed $\mathbf{X}_{shift}$. Measuring the *local* distributional shift for each pipeline component is not sufficient to capture the *global* distortion due to the drift [4, 25]. We therefore measure the distributional dissimilarity between the processed original and the processed shifted datasets using KL-Divergence [18]. KL-Divergence measures the amount of information lost when using one probability distribution to approximate another distribution, and quantifies the higher-order shift between the two distributions. Formally, KL-divergence can be written as follows:

$$D_{\text{KL}}(\mathcal{P}(\mathbf{X}_{\text{orig}} \| \mathcal{P}(\mathbf{X}_{\text{shift}})) = \sum_{x} \mathcal{P}(\mathbf{X}_{\text{orig}}) \log \frac{\mathcal{P}(\mathbf{X}_{\text{orig}})}{\mathcal{P}(\mathbf{X}_{\text{shift}})} \tag{2}$$

3.1.3 Vulnerability of pipeline component P_k to $\mathbf{X}_{shift}$. Using the sensitivity measure and distribution dissimilarity as described above, we now compute the vulnerability score of component P_k to the shifted dataset $\mathbf{X}_{\text{shift}}$ as follows:

$$\text{VS}_{P_k} = \lambda_1 S_{P_k} + \lambda_2 D_{\text{KL}} \left(P_{\text{orig}}, P_{\text{drift}} \right) \tag{3}$$

where $\lambda_1, \lambda_2 \in (0, 1]$ are tunable parameters. A higher value of λ_1 places more emphasis on the contribution of pipeline sensitivity, while a higher value of λ_2 assigns greater importance to higher-order moments of shift to shape the vulnerability score.

3.1.4 Pipeline vulnerability to $\mathbf{X}_{shift}$. In the previous subsections, we quantified the vulnerability of each component of a pipeline to any shift in the original dataset. The vulnerability of pipeline $\mathcal{P}$ for a particular shift in $\mathbf{X}_{\text{shift}}$ can be denoted as $[\text{VS}_{P_1}, \ldots, \text{VS}_{P_m}]^T$ quantified as their aggregated sum:

$$\text{VS}(\mathcal{P}) = \sum_{k=1}^{m} \text{VS}_{P_k}$$

To compute the vulnerability of pipeline $\mathcal{P}$ to different shifts, we compute its vulnerability to several simulated shifts. We simulate n shifted datasets through a data drift function g by perturbing a controlled amount of noise into the original dataset where the perturb parameter ρ indicates the percentages of data points that are being perturbed. For the original dataset $\mathbf{X}_{\text{orig}}$, we generate n drifted datasets $\mathbf{X}_{\text{shift},j} = g_j \left(\mathbf{X}_{\text{orig}} \right)$ where $j \in [1, \ldots, n]$. Data set $\mathbf{X}_{\text{shift},j}$ represents a unique drift scenario. Computing the vulnerability of the pipeline to each of these drift scenarios results in the following pipeline vulnerability matrix:

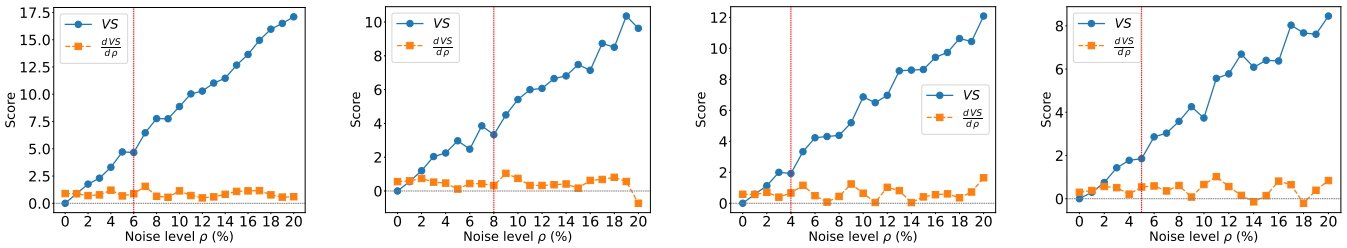
$$\text{VS} = \begin{bmatrix} \text{VS}_1^1 & \text{VS}_1^2 & \ldots & \text{VS}_1^n \\ \text{VS}_2^1 & \text{VS}_2^2 & \ldots & \text{VS}_2^n \\ \vdots & \vdots & \ddots & \vdots \\ \text{VS}_m^1 & \text{VS}_m^2 & \ldots & \text{VS}_m^n \end{bmatrix}$$

where VS_k^j denotes the vulnerability score of component P_k under the drifted dataset $\mathbf{X}_{\text{shift},j}$. The matrix VS across different datasets captures the susceptibility of each component to varying types of drift. A higher value of VS indicates relatively greater potential to vulnerability.

We aim to generate a report for the practitioner with a tolerable threshold τ beyond which a component is considered drift-vulnerable and might cause performance to decline. Identifying such a threshold for VS is challenging due to the absence of ground truth. Therefore, we empirically analyze the trend of VS, its corresponding gradient, and the associated performance degradation to define a tolerable threshold. We compute the gradient $\frac{\partial \text{VS}}{\partial \rho}$ at each VS point where ρ denotes the perturbation parameter. Future work will focus on developing a theoretical framework for determining this threshold more rigorously.

3.2 Performance Error Estimation

This work proposes a lightweight and efficient framework that leverages a vulnerability score to estimate performance degradation due to data shift. The key idea is to model the degradation as a



(a) AdultIncome(MVI) (b) AdultIncome(OI) (c) GermanCredit(MVI) (d) GermanCredit(OI)

Figure 2: The results demonstrate the vulnerability score (VS) and its gradient for mean value imputation under different types of noise—Missing Value Injection (MVI) and Outlier Injection (OI). As the severity of noise increases, VS exhibits a consistent trend, indicating its sensitivity to noise-induced perturbations.

function of a dataset-level VS using a simple regression model. This approach requires no retraining of the underlying model and allows for fast, real-time estimation of performance degradation in deployment scenarios. Let $A(\mathcal{D})$ and $A(\mathcal{D}_{\text{shift}})$ represent the model's accuracy on the clean and shifted datasets, respectively. The observed performance degradation is defined as:

$$\Delta = A(\mathcal{D}) - A(\mathcal{D}_{\text{shift}}).$$

Now, leveraging the vulnerability score $\text{VS} \in \mathbb{R}^{n \times m}$ that quantifies the degree of distributional shift between $\mathcal{D}$ and $\mathcal{D}_{\text{shift}}$, our goal is to predict Δ directly by utilizing VS. We construct a training dataset $\mathcal{T} = \{\text{VS}, \Delta\}$ by generating multiple $(\mathcal{D}, \mathcal{D}_{\text{shift, i}})_{i=1}^{n}$ pairs with varying levels and types of shifts. For each pair, we compute the vulnerability score VS_i^j and the corresponding performance degradation Δ_i. A regression model $\mathcal{R} : \text{VS} \rightarrow \Delta$ is then trained to learn the mapping from shift severity to performance degradation:

$$\hat{\Delta}(\text{VS}_{P_k}) = f(\text{VS}_{P_k})$$

To enhance interpretability and safety, we optionally estimate the standard deviation of the residuals from the regression model and construct a confidence-adjusted interval:

$$\Delta \in \left[\hat{\Delta}(\text{VS}_{P_k}) - z \cdot \hat{\sigma},\ \hat{\Delta}(\text{VS}_{P_k}) + z \cdot \hat{\sigma}\right],$$

or apply a fixed safety margin ϵ to account for worst-case scenarios:

$$\Delta = \hat{\Delta}(\text{VS}_{P_k}) + \epsilon.$$

Despite its simplicity, this approach provides a strong and interpretable baseline for predicting performance degradation using data-centric indicators, and can be extended with uncertainty-aware or meta-feature-based techniques in future work.

3.3 Generating Explanation

Given the VS matrix defined in Section 3.1.4, along with a tolerable threshold τ and a learned model $\mathcal{R}$, we generate user-facing explanations that characterize pipeline robustness under data drift. While the development of an automated method for recommending alternative pipeline sequences in drift-vulnerable scenarios is left for future work, this study demonstrates the potential of such recommendations. An illustrative example of the comprehensive explanation framework proposed in this study is presented below:

Quantification: This pipeline exhibits a **vulnerability score** of VS = 7 which means any shift in the training data will likely result in a decline of the model performance by 3-5%.

Explanation: The **most vulnerable component** is missing value imputation using mean.

Intervention: Change the missing value imputation from mean to median to preserve model performance with a drop of $< 1\%$.

4 Preliminary Experiments

4.1 Experimental setup

4.1.1 Datasets. We consider two real-world datasets popular in machine learning literature. **AdultIncome [6]** dataset is used to predict whether an individual's annual income exceeds \$50, 000 by analyzing a range of demographic and socio-economic factors. **German Credit** [16] dataset contains information on 1,000 individuals classified as having good or bad credit risk, based on 20 categorical and numerical attributes.

Table 1: Vulnerability Score for two datasets under different data shift scenarios. $\text{VS}_{\mathcal{P}_1}$ and $\text{VS}_{\mathcal{P}_2}$ represents VS for mean imputation and standardization respectively. MV represents Missing Value injection.

		AdultIncome		GermanCredit	
Injection Type (g)	ρ	$\text{VS}_{\mathcal{P}_1}$	$\text{VS}_{\mathcal{P}_2}$	$\text{VS}_{\mathcal{P}_1}$	$\text{VS}_{\mathcal{P}_2}$
MV Injection	0.1	7.93	0.14	6.70	0.014
	0.2	18.07	0.28	11.89	0.024
	0.3	27.23	0.39	16.96	0.030
	0.4	37.94	0.58	22.87	0.040
	0.5	46.92	0.77	29.46	0.060
Outlier Injection	0.1	5.48	0.06	3.15	0.01
	0.2	10.14	0.10	7.63	0.02
	0.3	14.84	0.15	9.10	0.05
	0.4	18.74	0.19	12.80	0.06
	0.5	22.79	0.24	15.57	0.09

4.1.2 Settings. We consider optimizing the pipeline for the entire dataset $\mathcal{D}$, which implies that the model $\mathcal{M}$ is trained and evaluated using the same dataset. Regarding the preprocessing component, we consider the following methods: Missing value imputation methods ∈ {Mean, Median, Mode}, scaling methods ∈ {Standardization, Min-Max}, and Outlier Handling techniques ∈ {IQR, Z-score}. For each dataset, we first optimize the pipeline, resulting in a return set $\mathcal{P}$ for the pre-processing component. Next, we evaluate the model's performance under various scenarios by introducing controlled noise, specifically injecting outliers and missing values. The parameters λ_1 and λ_2 are set arbitrarily to 0.2 and 1, respectively. We use logistic regression for the classification tasks.

Source code. The source code for the study is available at this link:Code

4.2 Primary results

4.2.1 Reporting VS. In this series of experiments, we report the vulnerability of mean value imputation and standardization scaling used to transform the original data for model training for both datasets. To analyze the vulnerability score (VS), we introduce two different types of noise: missing values and outliers, varying from 10% to 50% of the total data size. Each dataset is then preprocessed using the previously mentioned components. Subsequently, we calculate the VS using the equation 3 for each modified dataset. The resulting vulnerability score matrix is presented in table 1. VS_{P_1} and VS_{P_2} represent the vulnerabilities associated with missing values imputation and scaling, respectively. The results indicate that, among the two datasets, the pre-processing components are more susceptible to data shifts caused by missing value injection than to outlier injection. The inclusion of additional missing values, which undergo mean value imputation during the preprocessing stage, shifts the distribution significantly. In contrast, most injected extreme values were effectively clipped by the IQR outlier handling method, making the components less vulnerable to shifts caused by outliers. As discussed in Section 3.1.1, the sensitivity of each preprocessing component is measured immediately after it is applied. Consequently, standardization (VS_{P_2}) tends to exhibit relatively smaller vulnerability scores compared to mean imputation, due to its placement later in the pipeline. To ensure a fair and comparable evaluation across components, we additionally compute the corresponding gradient magnitude as a normalized sensitivity metric.

4.2.2 Pipeline Explanation. In this section, we provide an intuitive user-facing explanation to help end-users understand how susceptible the optimized preprocessing components are to distributional shifts in real-world scenarios. We simulate n perturbed data sets by injecting controlled levels of noise ($\rho \in [0, 20]\%$ of the data) and compute the corresponding vulnerability scores (VS) for each preprocessing component. Through this empirical evaluation, define a tolerable noise threshold (τ described in Section 3.1.4) beyond which the pre-processing step becomes vulnerable, indicated by a significant degradation in model performance and vice-versa. Assuming a user-defined tolerable performance degradation threshold δ, we define the vulnerability threshold τ as the point at which the model's performance drops by more than δ relative to the baseline model trained on the original dataset. In our experiments, we set $\delta = 0.03$, indicating that the user is only willing to tolerate up to a 3% decline in performance compared to the baseline. Figure 2 illustrates experiments conducted on two distinct datasets, evaluating the VS of a representative preprocessing method—mean value imputation—under two types of perturbations: missing value injection and outlier injection. The vertical red line highlights the identified threshold, implying that beyond this point, performance declines by 3%. We observe a notable pattern at the identified threshold points: the model performance exhibits a sharp decline, and correspondingly, the vulnerability score (VS) of the current pipeline experiences a sudden increase—reflected by a steep gradient trend. This relationship underscores the correlation between the vulnerability score (VS) and performance degradation under distributional shifts. To operationalize this insight, we train a regression model $\mathcal{R}$ as described in Section 3.2 that maps the VS values computed across various drifted scenarios to the corresponding performance changes. This allows us to quantify and anticipate the degree of degradation solely based on the vulnerability profile of the preprocessing pipeline. However, for instance, our proposed user-facing report of the pipeline robustness from the experiment is as follows:

> *Quantification:* Given a shifted version of the Adult Income dataset, a given pipeline exhibiting a vulnerability score of VS = 6.2 > τ = 4.9, the performance of model $\mathcal{M}$ with the current pre-processing pipeline is likely to decline by around 4%.
>
> *Explanation:* The **most vulnerable component** is missing value imputation using mean.

This positive correlation between VS and Δ motivates the assumption that, to generalize a pre-trained model $\mathcal{M}$ under distributional shift, one can strategically alter the sequence of preprocessing components—prioritizing those with lower vulnerability scores (VS ↓) to mitigate performance degradation without retraining the model.

5 Research Challenges and Future Works

The proposed user-facing pipeline for explanation development is currently in its early stages. The experimental results are not yet fully intuitive, and there is still a bunch of research to be conducted to address various challenges. We will now discuss some of these research challenges and outline further work that needs to be done.

Our proposed vulnerability score for preprocessing components serves as a proxy metric, lacking direct ground truth. This presents a key challenge in assessing the accuracy and reliability of the vulnerability quantification. To address this issue, we aim to investigate a potential causal relationship between the computed vulnerability score and the observed performance changes. While our experimental results in Figure 2 exhibit a consistent upward trend in vulnerability with increasing noise severity, this correlation alone does not establish causality. Further validation is required across diverse datasets and drift types. Alternatively, expert judgment from domain practitioners could serve as a form of ground truth calibration. In the experiments, we focused on injecting only one type of noise into the dataset at a time. However, in the real world, multiple forms of noise can coexist in a dataset. Our future work will involve

Table 2: A user-facing explanation for a new drifted dataset $\mathcal{D}_{\text{shift}}$, assessing the pipeline vulnerability, consequences, and suggesting the probable remedy with respect to a model $\mathcal{M}$ trained on the original dataset $\mathcal{D}$.

Description	Explanation (for a particular scenario)
Vulnerability Score (VS)	4.5
Vulnerability threshold, τ	3
Existing Pipeline	$\mathcal{P}$ = (Imputation: Mean → Outlier: IQR → · · · → Dimentionality Reduction: PCA)
Most Vulnerable Component	Mean Value Imputation
Performance Degradation (Δ)	3.5%
Suggested Pipeline	$\mathcal{P}$ = (Imputation: Median → Outlier: IQR → · · · → Dimentionality Reduction: PCA)
Retraining needed?	No

considering various types of noise simultaneously. We aim to extend the experiments on non-structured data and different learning settings as well. In addition, rather than training a regression model $\mathcal{R}$ to predict performance degradation Δ, we aim to develop a shift profiling approach that leverages meta-features of each pipeline run to estimate Δ. This strategy significantly enhances efficiency by avoiding repeated model retraining.

A significant body of work has addressed data shift from a model- and data-centric perspective, focusing on building generalized models that adapt to new data distributions by updating learned parameters or transforming data [23]. The current work explores a complementary direction: addressing data shift from the preprocessing pipeline perspective. In Section 4.2.2, we observe an interesting correlation between the vulnerability score (VS) and performance (Δ), suggesting that VS can act as a surrogate indicator of downstream performance changes. This raises an intriguing possibility—can we mitigate performance degradation by optimizing the preprocessing pipeline alone for unseen data in the deployed phase, without modifying the trained model? If VS can be minimized under data drift, it may be possible to identify new, more robust pipeline configurations, thereby avoiding expensive retraining procedures. Future work must investigate both theoretical guarantees and empirical trade-offs between pipeline optimization and full model retraining to understand when one is sufficient or preferable over the other. We further aim to offer comprehensive, user-facing explanations that encompass the vulnerability assessment, potential consequences, and actionable remedies, as illustrated in Table 2. These explanations are designed to assist end-users in understanding the implications of pipeline vulnerabilities and in making informed decisions. Data scientists may consider including such interpretive reports alongside deployed models, outlining how the pipeline behaves under different scenarios. This approach can help users anticipate potential risks and implement appropriate mitigation strategies proactively.

6 Related Work

The study in this paper is related to the following research areas: data shift, model-centric vs data-centric solutions, and robustness and explanation of pipeline configurations. While these areas have been studied extensively, our approach of providing a pipeline explanation to the end-user along with a vulnerability report and suggesting a new sequence for the pipeline is novel. Traditionally, most solutions to deal with data shift have been model-centric, focusing on making models more robust through fine-tuning, regularization, or ensembling [12, 13, 21]. However, the emerging data-centric AI paradigm emphasizes improving data quality and pipeline configurations, identifying root causes rather than altering model parameters [5, 27]. This transition underscores the need to explore robustness beyond model weights—particularly in how data is preprocessed before modeling. Pre-processing pipelines typically comprise steps such as imputation, outlier detection, etc. While pipeline optimization methods (e.g., AutoML, Reinforcement Learning, Meta Learning-based, Human-centric) aim to maximize performance on a given dataset [1, 2, 7, 24], these static sequences are rarely evaluated under data shift. Several research studies quantify the impact of data pre-processing on datasets [21, 28]. Some recent works have addressed pipeline what-if analysis [10], fairness, and interpretability [3, 26]. Most of the Explainable AI research primarily focuses on describing the model's behavior [22]. To our knowledge, this is the first study that quantifies the vulnerability of individual pipeline components under drift and explains the overall behavior of the pipeline. Understanding this behavior is crucial, as even optimal models can fail if upstream transformations introduce bias or noise when facing shifted data.

7 Conclusion

This work initiates a novel direction in assessing machine learning pipeline vulnerability and providing an explanation to the practitioner. This study also offers a new pipeline sequence by optimizing the vulnerability to preserve the performance, avoiding retraining. Through a vulnerability score framework, we present a lightweight, interpretable method to anticipate performance degradation under data drift without retraining. While promising, the current results are preliminary, limited by the absence of ground truth and the complexity of real-world scenarios. Moving forward, establishing causality between vulnerability and model degradation, supporting multi-drift analysis, and offering actionable, user-centered explanations remain essential to prioritizing robustness in ML deployment.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful feedback. Funding for this research was provided by the National Science Foundation (NSF) under Grant #2237149 and the UL Research Institutes through the Center for Advancing Safety of Machine Intelligence.

References

[1] Laure Berti-Equille. 2019. Learn2clean: Optimizing the sequence of tasks for web data preparation. In *The world wide web conference*. 2580–2586.

[2] Besim Bilalli, Alberto Abelló, Tomàs Aluja-Banet, and Robert Wrembel. 2016. Automated data pre-processing via meta-learning. In *International Conference on Model and Data Engineering*. Springer, 194–208.

[3] Sumon Biswas and Hridesh Rajan. 2021. Fair preprocessing: towards understanding compositional fairness of data transformers in machine learning pipeline. In *Proceedings of the 29th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*. 981–993.

[4] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. 2020. Homm: Higher-order moment matching for unsupervised domain adaptation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3422–3429.

[5] Sijie Dong, Qitong Wang, Soror Sahri, Themis Palpanas, and Divesh Srivastava. 2024. Efficiently mitigating the impact of data drift on machine learning pipelines. *Proceedings of the VLDB Endowment* 17, 11 (2024), 3072–3081.

[6] Dheeru Dua, Casey Graff, et al. 2017. UCI machine learning repository. (2017).

[7] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. *Advances in neural information processing systems* 28 (2015).

[8] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM computing surveys (CSUR)* 46, 4 (2014), 1–37.

[9] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*. PMLR, 2242–2251.

[10] Stefan Grafberger, Paul Groth, and Sebastian Schelter. 2023. Automating and optimizing data-centric what-if analyses on native machine learning pipelines. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–26.

[11] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. 2009. Covariate shift by kernel mean matching. *Dataset shift in machine learning* 3, 4 (2009), 5.

[12] Ishaan Gulrajani and David Lopez-Paz. 2020. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434* (2020).

[13] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*. PMLR, 1321–1330.

[14] Jiawei Han, Jian Pei, and Hanghang Tong. 2022. *Data mining: concepts and techniques*. Morgan kaufmann.

[15] Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. AutoML: A survey of the state-of-the-art. *Knowledge-based systems* 212 (2021), 106622.

[16] Hans Hofmann. 1994. Statlog (German Credit Data). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5NC77.

[17] Sotiris B Kotsiantis, Dimitris Kanellopoulos, and Panagiotis E Pintelas. 2006. Data preprocessing for supervised leaning. *International journal of computer science* 1, 2 (2006), 111–117.

[18] Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 22, 1 (1951), 79–86. doi:10.1214/aoms/1177729694

[19] Katsiaryna Mirylenka, George Giannakopoulos, Le Minh Do, and Themis Palpanas. 2017. On classifier behavior in the presence of mislabeling noise. *Data mining and knowledge discovery* 31 (2017), 661–701.

[20] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. 2012. A unifying view on dataset shift in classification. *Pattern recognition* 45, 1 (2012), 521–530.

[21] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems* 32 (2019).

[22] Romila Pradhan, Aditya Lahiri, Sainyam Galhotra, and Babak Salimi. 2022. Explainable ai: Foundations, applications, opportunities for data management research. In *Proceedings of the 2022 International Conference on Management of Data*. 2452–2457.

[23] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2022. *Dataset shift in machine learning*. Mit Press.

[24] El Kindi Rezig, Mourad Ouzzani, Ahmed K Elmagarmid, Walid G Aref, and Michael Stonebraker. 2019. Towards an end-to-end human-centric data cleaning framework. In *Proceedings of the workshop on human-in-the-loop data analytics*. 1–7.

[25] Masashi Sugiyama and Motoaki Kawanabe. 2012. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press.

[26] Saeid Tizpaz-Niari, Ashish Kumar, Gang Tan, and Ashutosh Trivedi. 2022. Fairness-aware configuration of machine learning libraries. In *Proceedings of the 44th International Conference on Software Engineering*. 909–920.

[27] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip S Yu. 2022. Generalizing to unseen domains: A survey on domain generalization. *IEEE transactions on knowledge and data engineering* 35, 8 (2022), 8052–8072.

[28] Carlos Vladimiro González Zelaya. 2019. Towards explaining the effects of data preprocessing on machine learning. In *2019 IEEE 35th international conference on data engineering (ICDE)*. IEEE, 2086–2090.