Assignment 01: Survey of Failure Modes in ML Systems

AI545 - W26

Ahsanul Kabir

February 25, 2026

1 Introduction

Machine Learning Operations (MLOps) is the discipline that unifies machine learning development and operations. It represents the lifecycle of machine learning systems, from data collection and model training to deployment, monitoring, and continuous improvement. It extends the principles of traditional DevOps into the Machine Learning (ML) domain, emphasizing automation, scalability, and governance.

Unlike regular software systems, where failures are deterministic and reproducible, ML systems can fail due to shifts in data distributions or environmental conditions in addition to bugs in code. These failure modes may quietly degrade performance, create bias, or produce misleading outputs. Understanding and mitigating such failure modes is therefore necessary to sustain the reliability and usability of an ML model. A good MLOps lifecycle should be able to address these challenges.

Failure modes are the particular ways in which a system can break down, meaning the recognizable patterns or mechanisms that lead it to behave incorrectly, or with reduced performance. In MLOps, it is crucial to recognize how an ML model or system might fail because understanding these issues enables practitioners to reduce risk and maintain system reliability.

2 Failure Modes

2.1 Label Noise

Label noise refers to inaccurate labels within a machine learning dataset. This occurs when the *ground truth* — the reference data the model uses to learn is incorrect.

2.1.1 Cause

- Human Error: Mistakes made by annotators or arising from subjective interpretations.
- Ambiguity: Data points that are inherently difficult to classify (e.g., visually similar images).
- Crowdsourcing: Low-cost labeling from large groups, which often introduces inconsistencies.
- Automated Labeling: Errors propagated from prior models or automated processes.

2.1.2 Example

If a model is trained to distinguish between healthy and diseased trees, but 10% of the images marked 'Healthy' actually exhibit signs of Oak Wilt, label noise is present.

Automated solutions that strictly scan data for patterns often fail due to a lack of contextual understanding. For example, a 64-bit number might be misidentified as a timestamp when it is, in fact, a user ID [1].

2.1.3 Mitigation

- Shifting responsibility *left*: Developers annotate data at the moment of creation within the code[1].
- Multi-signal classification: Utilization of multiple classification clues to verify labels[1].

2.2 Training-Serving Skew

Training-serving skew is a failure mode where a machine learning model performs poorly in production because the live data encountered differs significantly from the training data. It stems from the difference between how a model sees data during training and how it sees data when it is deployed to make real predictions in production.

2.2.1 Cause

- Data Source Discrepancies: Training data is often clean and complete, whereas serving data originates from real-time events that may be noisy or missing.
- Environmental Differences: Variations in hardware or software environments between the pre-deployment and deployment stages.
- Duration Consideration: Training on 10 days of data while serving utilizes a 1-month window.

2.2.2 Example

The performance of voice assistants like *Alexa* can experience skew because clean, annotated data is typically used for initial training, whereas the model encounters noisy, overlapping audio during actual production use [2][3].

2.2.3 Mitigation

- Data Simulation: Aligning training data with real-world usage. To address this, realistic noise and simulated room acoustics were introduced during the training of Alexa[3][2].

2.3 Distribution Shift

Distribution shift occurs when the statistical properties of data change between the training phase and deployment, causing the machine learning model's performance to degrade because the real-world environment no longer matches learned patterns.

2.3.1 Cause

- Temporal Change: A model trained on a past is deployed in a future environment with different characteristics.
- Population Shift: The model encounters a new demographic or population segment not represented in the training set.
- Random Outages: As noted in [4], specific events (such as technical outages) can cause shifts in data patterns.

2.3.2 Example

In [4], an observability system detected that an ML model consistently reduced traffic to a specific payment route every Tuesday. Investigation revealed that the route had experienced an outage on a previous Tuesday. The distribution of *successful transactions* for that specific day-of-week/route combination shifted significantly during the outage. The model encoded this temporary anomaly as a permanent rule, leading to subsequent performance degradation.

2.3.3 Mitigation

- Data Selection Strategy: [4] poses a question regarding whether outage data should be included in training. A primary mitigation strategy involves filtering out noisy or anomalous periods to prevent the model from learning false shifts.

2.4 Concept Drift

Concept drift occurs when the statistical properties of the target variable change over time. This renders the machine learning model's learned patterns and predictions inaccurate as the relationship between input data and what the model attempts to predict evolves.

2.4.1 Cause

- Changing Population Behavior: Strategies employed by fraudsters or the nature of spam content change.
- External Environment Shifts: Economic factors, such as a recession, may alter behaviors (e.g., loan repayment) compared to the pre-recession data used for training.
- Periodic Effects: Gradual changes over time, such as seasonality affecting energy demand relative to temperature.

2.4.2 Example

As stated in [5], concept drift occurs when the underlying statistical properties of a data stream change over time, such as a shift in species from red fox to gray fox in zoo surveillance imagery.

2.4.3 Mitigation

- Retraining: Periodically retraining the model with recent data to capture current trends.
- Detection: Continuously monitoring model performance metrics (e.g., accuracy, error rate) to identify degradation.

2.5 Data Leakage

Data leakage in machine learning occurs when information from outside the training dataset is included in the model's training process. Training models based on future or target information results in artificially high performance during training and validation but leads to poor, inaccurate predictions in production.

2.5.1 Cause

- Target Leakage: Features include data that will not be available at the time of prediction. For example, including a loan status information when trying to predict if a person will default or not.
- Train-Test Contamination: Data from the test set is unintentionally used to train the model, such as performing preprocessing on the entire dataset prior to splitting.
- Data Splitting Errors: Splitting time-series data non-chronologically, allowing future data points to leak into past training sets.

2.5.2 Example

In the 2019 Kaggle Santander Customer Transaction Prediction competition, data leakage became a central issue. The leakage was a statistical artifact introduced during dataset generation. Competitors identified a *magic feature* that enabled near-perfect prediction performance. The competition organizers later confirmed that this behavior resulted from the data synthesis process, effectively causing unintended data leakage [6] [7].

2.5.3 Mitigation

- Strict Temporal Partitioning: When data possesses a time component, chronological splitting must be used rather than random shuffling.
- Feature-Target Audit: Examine the relationship between input features and the target variable to identify predictors that are unrealistically correlated.

3 Discussion

Analyzing these MLOps failure modes identifies key actions necessary for a machine learning system to function correctly.

- **Monitoring**: To ensure system integrity, it is essential to implement robust performance metrics, data quality assessments, and automated anomaly detection. Given the unpredictable nature of MLOps failures, a comprehensive monitoring framework is required to keep humans informed while minimizing the need for manual intervention.
- **Iterations**: The MLOps lifecycle is inherently iterative, as offline training metrics can not perfectly predict real-world performance. Moreover, external factors can significantly impact model behavior. So establishing a continuous feedback loop is essential. This allows for the systematic integration of live performance data back into the development cycle to maintain model accuracy and relevance.
- **Context**: The utility of a dataset is dictated by its application context. For example, a computer vision dataset of apples may be used for general class detection or granular breed classification. Defining the specific objective is critical to mitigating risks such as temporal drift or population shift.

References

[1] Vasileios Lakafosis et al. *How Meta understands data at scale*. Engineering at Meta Blog. Accessed: 2026-01-30. Apr. 2025. URL: https://engineering.fb.com/2025/04/28/security/how-meta-understands-data-at-scale/.

[2] Minhua Wu. *Machine-labeled data + artificial noise = better speech recognition*. Amazon Science Blog. Accessed: 2026-01-30. Mar. 2019. URL: https://www.amazon.science/blog/machine-labeled-data-artificial-noise-better-speech-recognition.

[3] Ladislav Mošner et al. *Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning*. 2019. arXiv: 1901.02348 [eess.AS]. URL: https://arxiv.org/abs/1901.02348.

[4] Tanya Tang and Andrew Mehrmann. *ML Observability: Bringing Transparency to Payments and Beyond*. Accessed: 2026-01-30. Netflix Technology Blog. Aug. 2025. URL: https://netflixtechblog.com/ml-observability-bring-transparency-to-payments-and-beyond-33073e260a38.

[5] Fabian Hinder et al. "Model-based explanations of concept drift". In: *Neurocomputing* 555 (2023), p. 126640. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2023.126640. URL: https://www.sciencedirect.com/science/article/pii/S0925231223007634.

[6] Marios Michailidis. *KKM 5th Place Solution: Santander Customer Transaction Prediction*. Kaggle Competition Writeups. Accessed: 2026-01-30. Apr. 2019. URL: https://www.kaggle.com/competitions/santander-customer-transaction-prediction/writeups/kkm-5th-place-solution.

[7] Andrej Tschalzev et al. *A Data-Centric Perspective on Evaluating Machine Learning Models for Tabular Data*. 2024. arXiv: 2407.02112 [cs.LG]. URL: https://arxiv.org/abs/2407.02112.