Explainable Adversarial Drift Detection for MLOps Feature Monitoring

***Abstract*—Machine learning models in production suffer from distribution drift, where evolving input features degrade model performance. Existing unsupervised detectors signal that drift occurred but provide no insight into which features drifted or what corrective action is appropriate. This paper proposes Explainable Adversarial Drift Detection, a framework that extends adversarial validation with permutation testing for statistical rigor and feature attribution for root cause analysis. The framework trains a gradient boosting classifier to distinguish reference from current data, applies a permutation test to confirm statistical significance, and uses feature attribution to identify drifting features with automated remediation prescriptions. Evaluation on synthetic and 13 real-world datasets demonstrated detection of all four temporal drift types with zero missed detections, zero false alarms on stable data, and correct feature identification in all controlled scenarios, advancing drift detection from binary alarms to actionable diagnostics.**

***Index Terms*—concept drift, adversarial validation, feature attribution, distribution shift, MLOps, streaming data.**

Introduction and Motivation

Machine learning models deployed in production frequently encounter evolving data distributions that degrade performance [1], [2]. Detecting these changes early is critical, especially when ground-truth labels are delayed or unavailable. The Discriminative Drift Detector (D3) [3] achieves state-of-the-art accuracy using adversarial validation [4], but provides no insight into *which* features drifted—a critical gap for practical MLOps deployment.

This work proposes **Explainable Adversarial Drift Detection (EADD)**, which extends adversarial validation with: (1) permutation testing (B=50, α=0.01) for statistically rigorous drift confirmation, (2) SHAP-based feature attribution [5] identifying which features drive detected drift, and (3) automated prescriptions mapping drift diagnoses to MLOps remediation actions.

Methodology

EADD operates as a four-step pipeline on streaming data. **Step 1:** Reservoir sampling maintains a reference window (W_{ref}=500) capturing the global historical distribution, while a sliding current window (W_{cur}=200) captures recent observations. **Step 2:** A LightGBM classifier [6] is trained to distinguish W_{ref} from W_{cur}; high AUC-ROC indicates distributional divergence. **Step 3:** A permutation test shuffles source labels B=50 times to build a null distribution; drift is confirmed only if $p < 0.01$. **Step 4:** Upon confirmation, TreeSHAP extracts per-feature importance from the adversarial classifier, ranking features by drift contribution and classifying the drift pattern (univariate, subset, or multivariate) with corresponding prescriptions.

Key Results

Temporal Coverage: EADD detected all four drift types (abrupt, gradual, incremental, recurring) with 100% success rate across 5 runs each; D3 detected only 2/4, failing on gradual and incremental drift entirely.

Real-World Performance: Across 13 benchmark datasets [4], EADD achieved 0% missed detection rate with 3.7% lower mean time to detection than D3 (1,661 vs. 1,725 samples).

Explainability: In three controlled scenarios, SHAP correctly identified drifting features as the top contributors in all cases (AUC 0.767–0.801). Feature attribution accuracy was 100%.

Robustness: EADD produced zero false alarms across all stable stream types, while D3 triggered up to 87.4 false alarms on autocorrelated data (Mann–Whitney $p = 0.0101$).

Conclusion

EADD transforms drift detection from a binary alarm into a diagnostic tool with feature-level attribution and actionable prescriptions. It detects a broader range of drift types than D3, achieves zero false alarms via permutation testing, and correctly identifies drifting features in all tested scenarios. The framework is directly applicable to production MLOps pipelines where understanding *what* drifted is as important as knowing *that* drift occurred.

References

[1] J. Lu et al., "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, 2018.

[2] J. Gama et al., "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, 2014.

[3] O. Gözüaçık et al., "Unsupervised concept drift detection with a discriminative classifier," in *Proc. ACM CIKM*, 2019, pp. 2311–2314.

[4] B. Lukats et al., "Unsupervised concept drift detection from deep learning representations in real-time," in *Proc. CIKM*, 2025.

[5] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 4765–4774.

[6] G. Ke et al., "LightGBM: A highly efficient gradient boosting de-

cision tree," in *Proc. NeurIPS*, 2017, pp. 3149–3157.