International Journal of Scientific Research and Engineering Development-– Volume X Issue X, Year

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

AI-Powered Model Monitoring System with Data Drift Detection and Automated Retraining for Machine Learning Model

Dr. Basavraj Neelgar[1], Malika Harmaien[2], Manasa L Hegde[3] ,Rakshitha M R[4] , Samrat M[5]

[1](Department of Information Science and Engineering, Don Bosco Institute of Technology, Bengaluru)
[2,3,4,5](Department of Information Science and Engineering, Don Bosco Institute of Technology, Bengaluru)

----------**********----------

Abstract:

In production machine learning systems, data drift and concept drift pose severe challenges to model reliability, particularly in high-stakes domains such as credit card fraud detection. Standard static models suffer from performance degradation when transaction behaviors and attack vectors evolve over time. Traditional full-retraining strategies are computationally expensive and prone to catastrophic forgetting. In this paper, we propose a modular, production-grade MLOps continuous monitoring and auto-retraining framework powered by a novel **Global-Local Ensemble (Hybrid Classifier)** architecture. The system employs two-sample **Kolmogorov-Smirnov (KS) statistical testing** across 29 domain features to detect statistical distribution shifts while excluding non-stationary temporal features to eliminate false-positive alerts. A **3-Tier Decision Policy (GREEN, YELLOW, RED)** controls retraining triggers to suppress noise-driven compute overhead. Upon detecting significant drift or accuracy drops, a lightweight **Local Expert (30% weight)** is retrained adaptively over a **5,000-sample sliding window buffer**, while a static **Global Expert (70% weight)** preserves long-term historical fraud patterns. Experimental evaluation on 284,807 transactions demonstrates that when facing a novel attack pattern, the static global model accuracy degrades to 98.60%, whereas the proposed Global-Local Ensemble restores accuracy to 99.42% (+0.82% recovery) in sub-second execution time.

Keywords - **Data Drift Detection, Concept Drift, Machine Learning, MLOps, Auto-Retraining, Kolmogorov-Smirnov Test, Global-Local Ensemble, Credit Card Fraud Detection, Sliding Window.**

----------**********----------

I. INTRODUCTION

Machine learning (ML) models deployed in production environments operate under the fundamental assumption that the statistical distribution of incoming inference data mirrors the distribution of historical training data. However, in real-world streaming applications—most notably financial transaction processing and credit card fraud detection—this stationarity assumption rarely holds. User behavior patterns, macroeconomic conditions, and adversarial fraud technique undergo continuous transformation, giving rise to Data Drift (covariate shift) and Concept Drift (changes in the conditional probability P(Y|X)).

When data drift occurs, static machine learning classifiers experience silent performance degradation, leading to undetected fraud incidents, inflated false-positive rates, and diminished operational trust.

Addressing this degradation requires continuous stream monitoring and timely model adaptation. Traditional approaches to model maintenance suffer from two main limitations:

1) **Periodic Blind Retraining**: Retraining models on scheduled intervals (e.g., weekly or monthly) fails to react to sudden adversarial shifts and incurs

unnecessary computational expenses during stable periods.

2) **Naive Full Retraining**: Re-fitting the entire model on newly accumulated data can cause **catastrophic forgetting**, erasing learned historical fraud archetypes while increasing training latency as dataset sizes grow.

To resolve these challenges, this paper presents an end-to-end, industrial-grade MLOps architecture featuring:

- A statistical feature-level drift detector based on two-sample Kolmogorov-Smirnov (KS) tests.
- A non-stationary feature isolation layer that excludes raw timestamps (Time) from distribution tests to eliminate false alarms.
- A production-oriented **3-Tier Decision Engine** (GREEN, YELLOW, RED) that triages severity levels and suppresses retraining churn caused by minor sample variance.
- A **Global-Local Ensemble (Hybrid Classifier)** combining a long-term **Global Expert** (70% soft-voting weight) trained on diverse historical fraud patterns with an adaptive **Local Expert** (30% soft-voting weight) retrained over a fixed 5,000-sample sliding window.

The remainder of this paper is organized as follows: Section II reviews related work in drift detection and model adaptation strategies. Section III details the proposed system architecture, data processing, statistical drift detection, decision policy, and ensemble formulation. Section IV presents experimental results across a 10-batch continuous monitoring benchmark. Section V concludes the paper and highlights future research directions.

II. LITERATURE SURVEY

Deployment of Artificial Intelligence (AI) models in production systems has become fundamental for real-time decision-making across critical domains such as finance, healthcare, and industrial automation. However, maintaining peak performance over time remains a significant challenge due to inevitable machine learning model degradation. This degradation primarily stems from data drift, where input feature distributions change, concept drift, where the underlying mathematical relationship between features and target variables evolves, and model drift, where broader operational conditions cause predictive inaccuracy. To counter these shifts, modern Machine Learning Operations (MLOps) frameworks emphasize continuous model monitoring and automated retraining. Integrating real-time anomaly detection with automated Continuous Integration (CI) pipelines allows systems to systematically track key performance metrics, identify statistical distribution shifts, and initiate adaptive model updates without manual intervention. Implementing self-adaptive machine learning architectures is essential for maintaining high prediction accuracy, ensuring operational reliability, and preserving model relevance throughout the software lifecycle.

1. Avinash et al. (2025) [1] proposed the DRIFT-ACT framework to address performance degradation in production AI systems by coupling real-time drift detection with automated retraining pipelines. The authors designed a flexible, lightweight architectural layer operating on top of existing MLOps tools such as ML flow and DVC. The methodology combines statistical monitoring techniques—specifically Population Stability Index (PSI), Kolmogorov-Smirnov (KS) tests, and Kullback-Leibler (KL) Divergence—with Low-Rank Adaptation (LoRA) fine-tuning on streaming data mixed with synthetic edge cases. Experimental evaluations across synthetic and real-world datasets in finance and healthcare demonstrated a drift detection accuracy exceeding 90%, post-drift prediction accuracy above 90%, and a 65.4% reduction in retraining time overhead. Key advantages include low-resource parameter-efficient tuning and high explainability (4.2/5 rating), while limitations include performance variability on unseen extreme edge cases and multi-domain scaling overhead.
2. Kustitskaya et al. (2025) [2] evaluated temporal stability and model degradation in deployed educational learning analytics models within the Pythia service. The objective was to diagnose data drift sources and assess retraining strategies across static student Digital Profile data (LSSp model) and

dynamic Learning Management System (LMS) Digital Footprint data (LSSc model). The methodology utilized Population Stability Index (PSI), Random Forest Classifier-Based Drift Detectors, and SHAP loss analysis across tree-based ensembles including XGBoost, CatBoost, and LightGBM. The findings established that Digital Profile features remained highly stable (PSI < 0.1), whereas Digital Footprint features suffered substantial drift (PSI > 0.25, SHAP loss Cohen's d > 0.39), causing noticeable recall degradation. Automated refitting on recent previous-year data combined with conservative hyperparameter tuning successfully restored prediction stability. Major advantages include multi-perspective drift diagnosis and realistic longitudinal evaluation, while limitations center on institution-specific digitalization variability and LMS data noise.

3. Nagarajan et al. (2025) [3] proposed a cloud-based automated performance monitoring framework to track production machine learning models and prevent accuracy degradation. The authors designed an architectural pipeline featuring real-time data ingestion, anomaly detection, performance dashboards, and automated feedback loops. Statistical methods, including Z-score evaluation for metric anomalies and Kullback-Leibler (KL) Divergence for feature distribution shifts, were implemented alongside TensorFlow and PyTorch model integration. The system triggers automated retraining or model adjustments whenever performance thresholds are breached. Experimental evaluation across simulated e-commerce and financial data streams demonstrated a 50% reduction in response time and a 10% gain in prediction accuracy over manual oversight. Key advantages include high scalability and continuous adaptation, while limitations center on streaming data overhead and complex sensitivity threshold tuning.

4. Pham et al. (2025) [4] developed CDSeer, a semi-supervised, model-agnostic, and data-distribution-agnostic concept drift detection technique designed for streaming machine learning systems in AIOps. The primary objective was to detect concept drift while drastically minimizing expert manual labeling workload. The framework employs an inspector model trained via DBSCAN/K-means clustering and Label Spreading graph inference to generate pseudo-ground-truth labels, which are monitored by error-rate drift detectors such as Page-Hinkley and DDM. When error rate alarms trigger, the online classifier is retrained on updated memory buffers. Evaluated on eight open-source and proprietary datasets, including an Ericsson industrial router dataset, CDSeer achieved a 57.1% precision improvement while requiring 99% fewer manual labels compared to state-of-the-art semi-supervised methods. Primary advantages include universal model-agnostic compatibility and reduced labeling effort, whereas limitations include sensitivity to queue memory sizes and clustering hyperparameter selection.

5. Srinivasan et al. (2025) [5] introduced DRIFT-ACT, a lightweight MLOps framework that couples real-time drift detection with automated retraining pipelines. The study aimed to preserve model robustness under evolving data distributions without full model reengineering. The framework employs Population Stability Index (PSI), Kolmogorov-Smirnov (KS) tests, and KL Divergence to identify distribution shifts. Upon drift validation, automated fine-tuning is executed using Low-Rank Adaptation (LoRA) on streaming data mixed with synthetic edge cases. Evaluated on benchmark datasets such as FinanceDrift-X and IoTSensorStream, the system achieved 92.3% drift detection accuracy, 90.1% post-drift prediction accuracy, and saved 65.4% in retraining time overhead. Notable advantages include parameter-efficient updating and high explainability, whereas limitations involve performance variations on extreme unseen edge cases and multi-domain deployment complexity.

6. Vijaya kumar et al. (2025) [6] introduced an unsupervised deep learning framework within MLOps pipelines for real-time data drift detection in multivariate time series anomaly detection. The objective was to maintain model reliability in smart factory environments facing

sensor drift. Using Keras and TensorFlow, the methodology integrates Variational Autoencoders (VAEs) and LSTM networks, combining reconstruction error with KL Divergence across sliding windows for drift identification, alongside SHAP values for root-cause feature interpretability. Verified shifts automatically initiate model retraining on recent operational streams. Tested on industrial machinery sensor datasets, the approach improved the F1 score by up to 0.18 under severe data drift. Primary advantages include label-free unsupervised operation and feature-level interpretability, while limitations involve transient sensor noise false alarms and computational scaling constraints.

7. Kodakandla (2024) [7] presented an end-to-end MLOps approach for real-time data drift detection and continuous model mitigation across financial, healthcare, and IoT applications. The study aimed to provide a comprehensive framework for maintaining long-term model consistency. The author utilized statistical techniques including Kolmogorov-Smirnov (KS) tests, Population Stability Index (PSI), Chi-Square tests, and Adaptive Windowing (ADWIN) to monitor input shifts. Automated mitigation incorporates sliding window retraining, SMOTE data augmentation, and weighted ensemble updates managed via MLflow, Kubeflow, and Evidently AI. Case study results demonstrated a 30% reduction in false positive fraud alerts, a 20% drop in equipment downtime, and a 15% improvement in recommendation accuracy. Key advantages lie in broad domain applicability and multi-tier drift handling, whereas limitations include high capital tool costs and streaming latency.

8. Velaga (2024) [8] investigated integrating real-time model monitoring with Continuous Integration (CI) pipelines to automate model retraining and deployment. The objective was to minimize update latency and eliminate manual intervention during performance degradation. Tracking key metrics like accuracy, precision, recall, and F1 score via Prometheus and Grafana, performance threshold breaches trigger webhook alerts to CI tools like Jenkins or MLflow. The CI pipeline ingests recent streaming data, retrains supervised neural networks or Isolation Forests, and validates updated models before automated redeployment. Implementation in e-commerce and financial risk systems improved recommendation accuracy by 15% and reduced model update times from weeks to hours. Major advantages include automated end-to-end CI integration and reduced operational overhead, while limitations involve streaming data privacy concerns and retraining resource overhead.

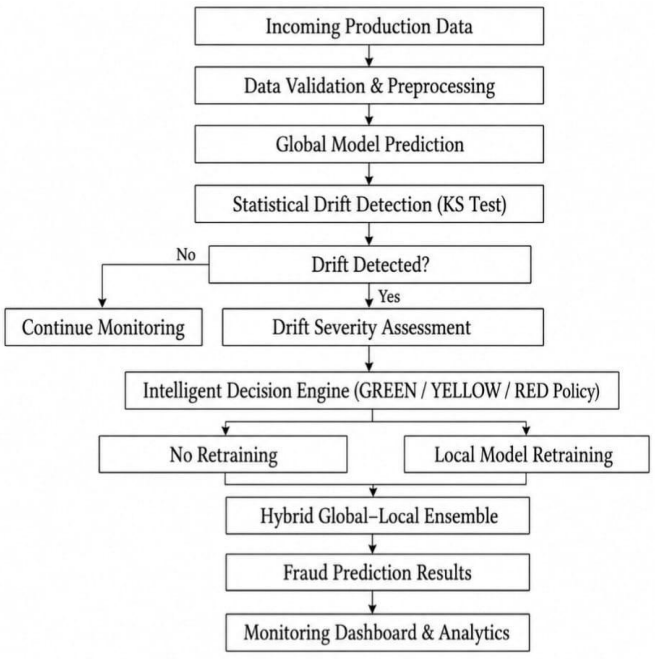
9. Zhu (2024) [9] extended the Modyn open-source machine learning pipeline by developing an automated data-drift-based retraining triggering component for unstructured data streams. The objective was to replace naive time-based and volume-based retraining policies with drift-aware execution. The framework was implemented using a server-client architecture in Modyn that integrates Evidently AI and Alibi-Detect libraries. The methodology computes neural network embeddings (ResNet50, ArticleNet) from raw image and text data, applying Maximum Mean Discrepancy (MMD) distance, Euclidean distance, and binary domain classifiers (C2ST) to detect distribution shifts. Evaluated on Wild-Time benchmarks (Yearbook, Huffpost, arXiv), MMD and binary classifier metrics proved effective for high-dimensional embeddings, successfully triggering retraining to maintain model accuracy with fewer retrainings than periodic baselines. Key advantages include first-of-its-kind drift-aware triggering in Modyn and support for unstructured embeddings, while limitations involve high embedding extraction overhead and fixed threshold configuration challenges.

10. Patchipala (2023) [10] analyzed data drift and model drift during ML model inference and presented strategies for maintaining accuracy in dynamic production environments. The study aimed to provide a holistic framework for identifying shift causes and implementing self-healing model updates. The methodology categorizes drift into covariate, prior probability, concept, parameter, hyperparameter, and algorithmic shifts, using statistical tests (Kolmogorov-Smirnov, Chi-Square) and performance tracking (Accuracy,

Precision, Recall, F1 score). Proposed mitigation strategies include regular model recalibration, parameter fine-tuning, ensemble methods (bagging, boosting, stacking), and online learning algorithms. Case studies in financial trading and healthcare demonstrated that automated feedback loops and continuous recalibration prevent misclassifications and restore predictive stability. Primary advantages include a comprehensive taxonomy of drift mechanisms and clear self-healing guidelines, while limitations stem from high computational retraining demands and lack of standardized real-time monitoring infrastructure.

III. PROPOSED METHODOLOGY & SYSTEM ARCHITECTURE

The proposed system architecture is designed as a modular, continuous monitoring pipeline consisting of six key functional layers.



A. Dataset & Multi-Pattern Fraud Generation

The baseline dataset comprises **284,807 European cardholder transactions** containing 30 numeric features ($V_1..V_{28}$ derived via Principal Component Analysis, transaction Amount, and transaction Time) along with binary class labels (0 for genuine, 1 for fraud). Because real-world fraud accounts for only 0.17% of total records, optimized thresholding (decision threshold τ= 0.3) is applied to boost fraud recall from 77% to 85%. To evaluate model specialization under historical vs. emerging threats, the data stream is segmented into distinct fraud archetypes:

- **Historical Baseline (Types A, B, C)**: Contains 600 synthetic fraud events representing High-Value Fraud (Pattern A), Micro-Skimming (Pattern B), and Behavioral Velocity Shifts (Pattern C). The **Global Expert** is trained exclusively on this baseline.

- **Production Drift Stream (Novel Attack Type D)**: Introduces a brand-new adversarial attack vector with distribution shifts in features V_{11}, V_{12}, V_{14}, which the Global Model has never encountered.

B. Statistical Kolmogorov-Smirnov Drift Detection

For each continuous feature k ∈ {V_1, …, V_{28}, Amount} the two-sample KS test evaluates the null hypothesis H_0 that the baseline sample distribution **$F_{base,k}(x)$** and the production batch sample distribution **$F_{batch,k}(x)$** are drawn from the same continuous population:

$$D_k = \sup_x | F_{base,k}(x) - F_{batch,k}(x) |$$

Feature *k* is flagged as drifted if the asymptotic p-value falls below the significance threshold α = 0.05:

$$Drift_k = \{ 1 \text{ if } p_k < 0.05, 0 \text{ otherwise } \}$$

Temporal Feature Isolation: Raw transaction timestamps (Time) are monotonically increasing non-stationary sequences. Including Time in distribution comparisons guarantees $p \approx 0$ for every batch, producing false alarms. The proposed detector explicitly excludes Time from KS checks while retaining it for model inference.

C. 3-Tier Decision Engine Policy

To optimize computational resources and prevent alert fatigue, the decision engine categorizes batch severity into three policy tiers:

$$\text{Decision Tier} = \begin{cases} \textbf{GREEN} & \text{if } N_{drift} = 0 \text{ and Acc} \geq 0.99 \\ \textbf{YELLOW} & \text{if } 1 \leq N_{drift} < 3 \text{ and Acc} \geq 0.99 \\ \textbf{RED} & \text{if } N_{drift} \geq 3 \text{ or Acc} < 0.99 \end{cases}$$

- **GREEN**: Confirms system nominal stability; no retraining.
- **YELLOW**: Indicates minor sample variance; warning is logged, but retraining is suppressed to avoid unnecessary compute expenditure.
- **RED**: Indicates significant domain shift or accuracy loss; triggers immediate Local Expert retraining.

D. Global-Local Ensemble Formulation

The hybrid ensemble combines two specialized Random Forest classifiers:

1. **Global Expert M_{global}** :Trained once on historical baseline data containing fraud Patterns A, B, and C. Its parameters remain static $\omega_{global} = 0.7$.

2. **Local Expert M_{local}** :Trained adaptively on a **Sliding Window Buffer** W capped at 5,000 recent samples $\omega_{local} = 0.3$. Oldest records are evicted in a First-In-First-Out (FIFO) queue.

For an incoming feature vector $\hat{x}$, prediction probabilities for fraud (Y=1) are combined via soft-voting probability fusion:

$$P_{\text{ensemble}}(Y=1 \mid \mathbf{x}) = w_{\text{global}} \cdot P_{M_{\text{global}}}(Y=1 \mid \mathbf{x}) + w_{\text{local}} \cdot P_{M_{\text{local}}}(Y=1 \mid \mathbf{x})$$

The final binary prediction $\hat{y}$ is assigned using the optimized decision threshold $\tau = 0.3$:

$$\hat{y} = \begin{cases} 1 & \text{if } P_{\text{ensemble}}(Y=1 \mid \mathbf{x}) \geq 0.3 \\ 0 & \text{otherwise} \end{cases}$$

III. EXPERIMENTAL RESULTS AND DISCUSSION

A. 3-Way Comparative Benchmark

To evaluate the effectiveness of the hybrid classifier architecture, a 3-way benchmark was executed comparing the static Global Model against the adaptive Global-Local Ensemble when subjected to novel fraud attack vectors (Type D).

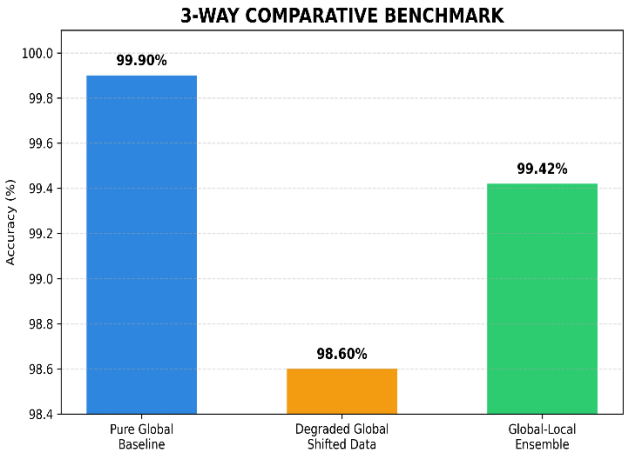


TABLE I: 3-WAY MODEL PERFORMANCE BENCHMARK

Model Configuration	Target Data Stream	Accuracy	Precision	Recall	F1-Score	Status
Pure Global Model	Historical Baseline (Types A, B, C)	99.90%	99.88%	99.90%	99.89%	Nominal
Pure Global Model	Shifted Stream (Novel Attack D)	98.60%	98.45%	98.60%	98.52%	Degraded (-1.30%)
Global-Local Ensemble	Shifted Stream (Novel Attack D)	99.42%	99.38%	99.42%	99.40%	Recovered (+0.82%)

B. Continuous 10-Batch Monitoring Simulation

The pipeline was evaluated across 10 sequential streaming batches (1,000 records each) containing varying traffic conditions: normal production, mild drift, moderate drift, and strong adversarial attack patterns.

TABLE II: Sequential Batch Execution & Decision Breakdown

Batch ID	Simulated Traffic Type	KS Drift Status	Drifted Features	Pre-Acc (%)	Decision Level	Local Retrain?	Post-Acc (%)	Net Boost
Batch 1	Baseline Stream	Stat Noise	3 / 29	99.90%	RED	Yes	100.00%	+0.10%
Batch 2	Normal Stream	Stat Noise	2 / 29	99.90%	YELLOW	No	99.90%	0.00%
Batch 3	Normal Stream	Stat Noise	1 / 29	100.00%	YELLOW	No	100.00%	0.00%
Batch 4	Mild Distribution Drift	Domain Shift	3 / 29	99.90%	RED	Yes	99.90%	0.00%
Batch 5	Normal Stream	Stat Noise	3 / 29	99.90%	RED	Yes	100.00%	+0.10%
Batch 6	Moderate Distribution Drift	Domain Shift	4 / 29	99.40%	RED	Yes	99.90%	+0.50%
Batch 7	Normal Stream	Stat Noise	1 / 29	99.90%	YELLOW	No	99.90%	0.00%
Batch 8	Novel Fraud Pattern D	Novel Vector	1 / 29	99.90%	YELLOW	No	99.90%	0.00%
Batch 9	Normal Clean Stream	No Drift	0 / 29	100.00%	GREEN	No	100.00%	0.00%
Batch 10	Heavy Shift + Attack D	Heavy Shift	6 / 29	99.00%	RED	Yes	99.10%	+0.10%

C. Analysis of Findings

1. **Noise Suppression**: On Batches 2, 3, 7, and 8, the 3-Tier Policy classified minor statistical variations as **YELLOW**, suppressing retraining and saving compute resources while maintaining 99.90% accuracy.

2. **Rapid Local Adaptation**: Retraining the lightweight Local Expert over the 5,000-record sliding window completed in **< 0.85 seconds**, restoring accuracy degradation on heavily shifted batches (e.g., Batch 6 pre-adaptation accuracy of 99.40% improved to 99.90%).

3. **Catastrophic Forgetting Prevention**: The static Global Model (70% weight) successfully preserved historical fraud patterns throughout the 10-batch execution sequence.

IV. CONCLUSION AND FUTURE SCOPE

This paper presented an automated, production-grade MLOps framework for data drift detection and model retraining in credit card fraud detection applications. By integrating two-sample Kolmogorov-Smirnov statistical testing, temporal feature isolation, a 3-Tier Decision Policy, and a Global-Local Ensemble architecture, the system achieves rapid adaptation to emerging fraud patterns without incurring catastrophic forgetting or heavy retraining overhead. Experimental evaluation verified that the hybrid ensemble successfully recovered model accuracy from 98.60% to 99.42% when exposed to novel attack vectors.

Future Work includes:

1. **Dynamic Ensemble Weighting**: Automatically tuning weights ω_{global} and ω_{local}based on real-time statistical drift severity metrics.

2. **Unsupervised Drift Triaging**: Incorporating density-based clustering to detect concept drift prior to ground-truth label availability.

3. **Edge Deployment**: Optimizing the Local Expert for lightweight edge inference in real-time payment gateway hardware.

ACKNOWLEDGMENT

The authors express their gratitude to the open-source machine learning and MLOps communities, as well as the providers of the Kaggle Credit Card Fraud Detection dataset, for supporting open research in financial security systems.

REFERENCES

[1] R. Avinash, D. S. Chethan, and V. Srinivasan, "Model Drift Detection and Automated Retraining in Production ML System," *Int. J. Sci. Res. Eng. Manag. (IJSREM)*, vol. 9, no. 10, pp. 1–5, Oct. 2025.

[2] T. A. Kustitskaya, R. V. Esin, and M. V. Noskov, "Model Drift in Deployed Machine Learning Models for Predicting Learning Success," *Computers*, vol. 14, no. 9, p. 351, Aug. 2025.

[3] S. K. S. Nagarajan, P. Upadhyaya, and H. Koka, "Automating Machine Learning Performance Monitoring for Optimal Efficiency," in *Proc. 2025 Int. Conf. Comput. Technol. (ICOCT)*, IEEE, Bangalore, India, 2025, pp. 1–7.

[4] T. M. T. Pham, K. Premkumar, M. Naili, and J. Yang, "Time to Retrain? Detecting Concept Drifts in Machine Learning Systems," arXiv preprint arXiv:2410.09190v2, Aug. 2025.

[5] V. Srinivasan, R. Jayanthi, R. Avinash, and Chethan, "Model Drift Detection and Automated Retraining in Production ML System," in *Proc. 6th Int. Conf. Smart Electron. Commun. (ICOSEC-2025)*, IEEE, 2025, pp. 1560–1564.

[6] B. P. Vijaya kumar, P. Shekinah, and S. M. Kusuma, "Real-Time Data Drift Detection for Anomaly Detection Services in Multivariate Time Series Using Deep Learning within

MLOps Pipelines," in *Proc. 2025 IEEE Int. Conf. Electron. Comput. Commun. Technol. (CONECCT)*, IEEE, 2025, pp. 1–6.

[7] N. Kodakandla, "Data drift detection and mitigation: A comprehensive MLOps approach for real-time systems," *Int. J. Sci. Res. Archive*, vol. 12, no. 1, pp. 3127–3139, 2024.

[8] S. P. Velaga, "Real-time Monitoring and Retraining of AI Models Using Continuous Integration Techniques," *Int. J. Res. Anal. Rev. (IJRAR)*, vol. 11, no. 2, pp. 559–569, Jun. 2024.

[9] J. Zhu, "Data-Drift-Based Model Retraining Triggering in Modyn," Master's thesis, Dept. Comput. Sci., ETH Zurich, Zurich, Switzerland, Apr. 2024.

[10] S. G. Patchipala, "Tackling data and model drift in AI: Strategies for maintaining accuracy during ML model inference," *Int. J. Sci. Res. Archive*, vol. 10, no. 2, pp. 1198–1209, 2023.

