International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 08 August-2026 | Impact Factor: 3.5

From Drift Detection to Diagnosis: An Explainable Framework for Deployed Machine Learning

Tambe Ankit Sampat
Research Scholar and Assistant Professor
Modern Education Society's Wadia College of Engineering, Pune, Maharashtra, India
Email: ankittambe24@gmail.com

How to Cite this Article:

Sampat, T. A. (2026). From Drift Detection to Diagnosis: An Explainable Framework for Deployed Machine Learning. International Journal of Creative and Open Research in Engineering and Management, 02(8), 1-9. https://doi.org/10.55041/ijcope.v2i8.295

https://doi.org/10.55041/ijcope.v2i8.295

Abstract

Machine learning (ML) models are increasingly deployed in real-world environments where data characteristics, user behaviour, and operating conditions may change over time. Such changes can lead to data or concept drift and may gradually reduce model reliability. Existing monitoring approaches can identify distributional changes or performance degradation, but they often provide limited information about why the changes occur. Similarly, Explainable Artificial Intelligence (XAI) methods can explain model predictions but are frequently treated separately from continuous drift monitoring. This paper proposes a unified drift-aware framework for explainable diagnostics in deployed machine learning systems. The framework integrates data-drift detection, model-performance monitoring, explainability, and diagnostic decision-making into a continuous monitoring pipeline. Statistical drift detection methods are used to identify changes between reference and production data, while XAI techniques such as SHAP and LIME are used to identify important features associated with changes in model behaviour. The framework combines drift indicators and performance information to generate interpretable diagnostic reports and recommend suitable maintenance actions. The proposed approach aims to improve the transparency, reliability, and maintainability of deployed ML systems by connecting drift detection with actionable explanations. The framework can support proactive model monitoring and assist practitioners in deciding whether continued monitoring, investigation, recalibration, or model retraining is required.

Keywords: Explainable AI,Machine Learning Deployment, Model Monitoring,Drift Detection

1. Introduction

Machine learning has become an important technology in applications such as healthcare, finance, cybersecurity, recommendation systems, agriculture, and intelligent decision support. During development, ML models are generally trained and evaluated using historical datasets under the assumption that future data will follow a similar distribution. However, real-world environments are dynamic, and the characteristics of incoming data may change over time. Such changes can affect the relationship between input variables and target outcomes and are commonly associated with concept drift (Gama et al., 2014). Concept drift can occur because of changes in user behaviour, environmental conditions, business processes, or data-generation mechanisms. When a deployed model is not monitored continuously, these changes may result in reduced prediction accuracy and unreliable decisions. Research on concept drift has therefore emphasized the importance of detecting and adapting to changes in evolving data environments (Lu et al., 2018).Traditional drift monitoring approaches generally focus on identifying whether a statistical or performance change has occurred. However, detection alone does not necessarily explain the reason behind the

change. For example, a monitoring system may report that the production data distribution has changed but may not identify which features are responsible for the change or how these features influence the model.Explainable Artificial Intelligence provides methods for understanding machine-learning predictions. LIME provides local explanations of individual predictions, while SHAP provides feature contribution values based on Shapley-value concepts (Ribeiro et al., 2016; Lundberg & Lee, 2017). These methods can improve transparency, but their integration with continuous drift monitoring remains an important research challenge.Recent research specifically examining the intersection of concept drift and model explanations demonstrates that changes in model explanations can themselves provide useful information for detecting and understanding drift (Demšar & Bosnić, 2018).

Therefore, this study proposes a unified framework that combines **drift detection, performance monitoring, explainability, and diagnostic decision-making** in a single deployment pipeline.

2. Literature Review

2.1 Concept Drift in Machine Learning

Concept drift occurs when the underlying data-generating process changes over time. Gama et al. (2014) provided a comprehensive survey of concept drift adaptation and classified different types of drift and adaptation strategies. Their work established that machine-learning systems operating in dynamic environments require mechanisms for detecting and responding to changes. Lu et al. (2018) reviewed learning methods under concept drift and highlighted the limitations of static learning models when the data distribution changes continuously. Such findings indicate that deployed ML systems require continuous monitoring and adaptation.Webb et al. (2017) further emphasized that understanding drift requires more than simply detecting its occurrence. Quantitative descriptions of changes in data distributions can help practitioners understand the nature and impact of drift.

2.2 Drift Detection Techniques

Different statistical and machine-learning methods have been proposed for detecting drift. Distribution-based techniques compare reference and production data, while performance-based approaches monitor changes in prediction quality.Bifet and Gavalda (2007) proposed Adaptive Windowing (ADWIN), which dynamically adjusts the size of the observation window to detect changes in streaming data. Such techniques are useful for environments where data arrive continuously.Page (1954) introduced the CUSUM approach for detecting persistent changes in statistical processes. Similar statistical monitoring concepts can be adapted for ML model monitoring.However, conventional drift detectors generally answer whether a change has occurred rather than explaining the underlying reason for the change.

2.3 Explainable Artificial Intelligence

Explainable AI aims to make machine-learning decisions understandable to users and domain experts. Ribeiro et al. (2016) introduced LIME, which explains individual predictions by constructing an interpretable local approximation around a prediction.Lundberg and Lee (2017) introduced SHAP, a unified framework for assigning feature importance values to model predictions. SHAP can provide both local and global explanations and is therefore suitable for identifying changes in feature contributions.These techniques are particularly useful in diagnostic systems because users may need to understand not only the prediction but also the factors contributing to it.

2.4 Explainability and Concept Drift

An important development in the literature is the integration of explanation with drift detection. Demšar and Bosnić (2018) proposed a concept-drift detection approach based on changes in model explanations. Their work demonstrated that monitoring changes in feature contributions can provide information about concept drift and improve interpretability. More recent research has highlighted that drift explainability should move beyond simply detecting

distributional changes and should provide human-interpretable information about **why, where, and how** drift occurs. This supports the motivation for a unified framework combining conventional drift measures with feature-level explanations.

2.5 Machine Learning Deployment and Monitoring

Deploying an ML model into production introduces challenges that are not normally visible during offline experimentation. Sculley et al. (2015) discussed hidden technical debt in ML systems and emphasized issues associated with data dependencies, changing external environments, and system maintenance. Amershi et al. (2019) also demonstrated that real-world ML systems require continuous engineering and monitoring beyond model development. Similarly, Breck et al. (2017) emphasized testing and monitoring as important requirements for production-ready ML systems. These studies indicate that model deployment should be considered as a continuous lifecycle rather than a one-time activity. The increasing adoption of AI across organizations has also created a need for continuous adaptation and effective management of intelligent systems. Tambe (2025) highlighted the growing role of AI-driven automation, intelligent decision-making, and workforce adaptation in the global economy. This broader adoption of AI further emphasizes the importance of maintaining reliable and interpretable ML systems after deployment.

3. Research Gap

Many studies have focused on concept drift detection, machine learning model monitoring, and Explainable Artificial Intelligence (XAI). However, limited research specifically integrates these approaches for diagnosing problems in deployed machine learning systems. Existing studies mainly focus on detecting data drift or monitoring model performance, while XAI techniques are often used separately to explain individual predictions. There is insufficient research on connecting drift detection, performance degradation, feature-level explanations, and maintenance decisions within a single practical framework. Therefore, the present study attempts to bridge this gap by proposing a unified drift-aware framework that integrates **drift detection, performance analysis, explainability, diagnosis, and maintenance decisions** for reliable machine learning deployment.

4. Objectives of the Study

The major objectives of the proposed research are:

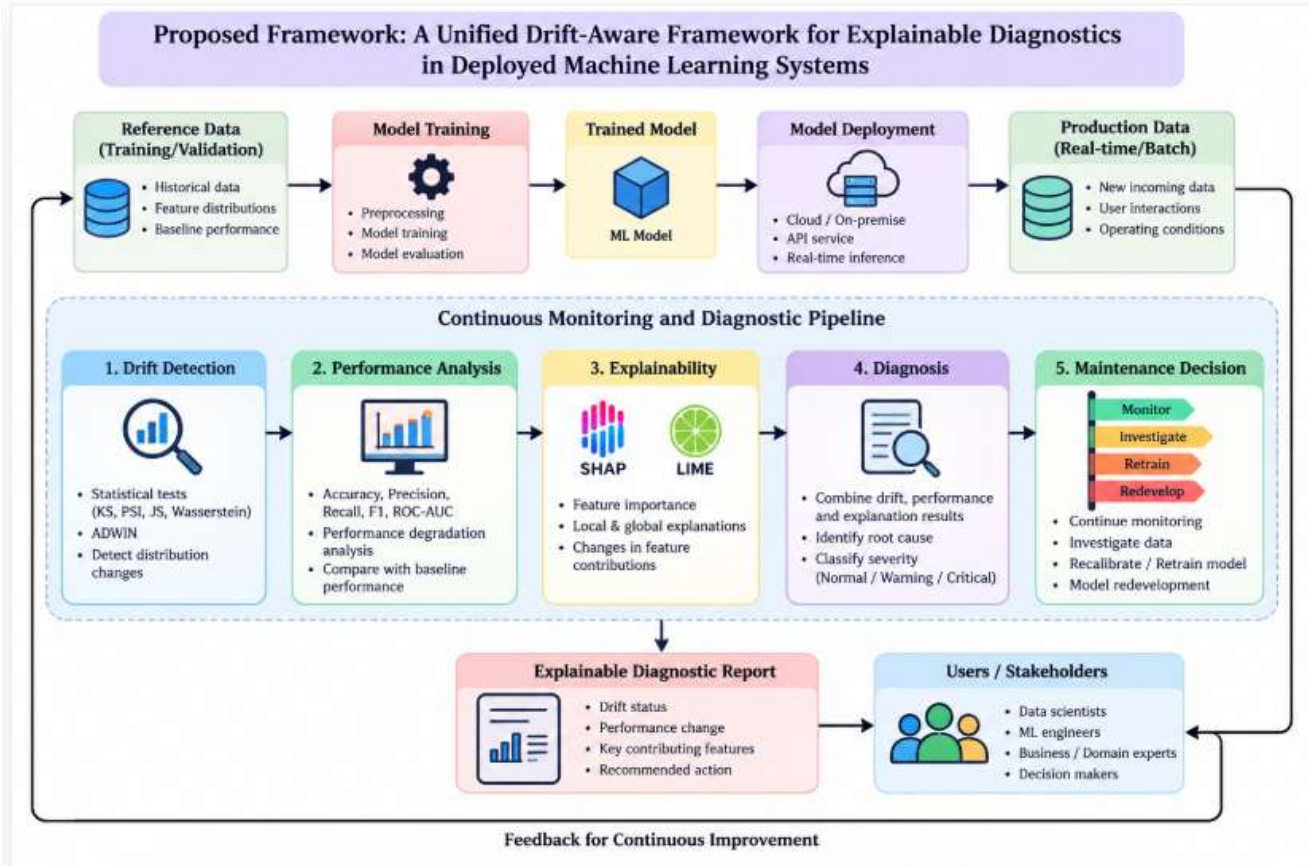
1. To develop a unified framework for monitoring deployed ML models.
2. To detect data and concept drift in production environments.
3. To monitor changes in model performance.
4. To use XAI techniques to identify important features associated with detected changes.
5. To generate an interpretable diagnostic report.
6. To provide appropriate maintenance recommendations based on drift and performance conditions.
7. To improve the reliability and transparency of deployed ML systems.

5. Proposed Methodology

The proposed framework continuously monitors deployed ML systems to identify changes in incoming data and model performance. **Statistical drift detection techniques** are used to compare current production data with reference data and identify significant distribution changes (Gama et al., 2014; Lu et al., 2018). When drift is detected, **SHAP or LIME** is applied to identify the features contributing to changes in model predictions (Ribeiro et al., 2016; Lundberg & Lee, 2017). The system then combines drift and performance information to generate an **explainable diagnostic report** and suggest suitable actions such as continued monitoring, investigation, recalibration, or model retraining. The

integration of drift detection and model explanation is supported by previous research demonstrating that changes in explanations can provide useful evidence of concept drift (Demšar & Bosnić, 2018).

6.Proposed Framework



6.1. Framework Components

Reference Data

Reference data represents the historical training or validation data used as the baseline. It contains the original feature distributions and model performance information. This baseline is later compared with incoming production data to identify changes.

Model Training

The reference data is preprocessed and used to train the machine learning model. The trained model is evaluated using suitable performance metrics such as accuracy, precision, recall, and F1-score. The best-performing model is then prepared for deployment.

Model Deployment

The trained model is deployed in a real-world environment through a cloud, server, or API-based application. It continuously receives new data and generates predictions. The deployed model becomes the main source for continuous monitoring.

Production Data

Production data consists of new, real-world data received after deployment. It may differ from the reference data because of changes in users, environment, or operating conditions. These changes are continuously monitored for possible drift.

Drift Detection

Drift detection compares production data with the reference data to identify significant distribution changes. Statistical techniques such as **KS Test, PSI, Jensen–Shannon Divergence, Wasserstein Distance, and ADWIN** can be used. A significant change generates a drift alert (Gama et al., 2014; Lu et al., 2018).

Performance Analysis

The system compares the current model performance with its baseline performance. Metrics such as accuracy, precision, recall, F1-score, and ROC-AUC can be used for classification models. This helps determine whether detected drift is affecting model performance.

Explainability

When drift or performance degradation is detected, **SHAP and LIME** are used to understand the model's behaviour. They identify important features and their contribution to predictions. This helps explain **which features are responsible for changes in model behaviour** (Ribeiro et al., 2016; Lundberg & Lee, 2017).

Diagnosis

The diagnosis component combines drift information, performance changes, and XAI results. It identifies the possible reasons for model degradation and classifies the situation as **Normal, Warning, or Critical**. This converts technical monitoring information into understandable diagnostic information.

Maintenance Decision

Based on the diagnostic results, the system recommends an appropriate action. These actions may include **continue monitoring, investigate data, recalibrate/retrain the model, or redevelop the model**. This supports proactive maintenance of deployed ML systems.

Explainable Diagnostic Report

The framework generates a report containing the **drift status, performance change, major contributing features, severity level, and recommended action**. This makes the monitoring results easier for ML engineers and domain experts to understand and use.

Feedback for Continuous Improvement

The diagnostic results and subsequent maintenance actions are fed back into the monitoring process. This helps update the reference data, model, and monitoring thresholds when necessary. Thus, the framework supports **continuous improvement throughout the ML deployment lifecycle**.

7. Proposed Algorithm

Algorithm: Drift-Aware Explainable Diagnostic Framework

Input: Reference dataset DrD_r, production dataset DpD_p, trained model MM

Output: Diagnostic report RR

1. Load reference dataset DrD_r.
2. Load production data DpD_p.
3. Perform data preprocessing.
4. Calculate feature-wise drift scores.
5. Compare drift scores with predefined thresholds.
6. Monitor model performance.
7. If significant drift is detected, identify affected features.
8. Apply SHAP/LIME to generate model explanations.
9. Compare current feature contributions with baseline contributions.
10. Determine the severity of model degradation.
11. Generate an explainable diagnostic report.
12. Recommend appropriate maintenance action.
13. Store monitoring information for future comparison.

8. Experimental Design

The proposed framework can be evaluated using a publicly available classification dataset. The experiment follows the sequence **Dataset → Model Training → Baseline Evaluation → Controlled Drift → Drift Detection → Performance Evaluation → SHAP Analysis → Diagnostic Decision**. Controlled changes can be introduced in selected features to simulate different drift conditions, as commonly considered in concept-drift research (Gama et al., 2014; Lu et al., 2018). Five scenarios can be evaluated: **no drift, single-feature drift, multiple-feature drift, drift with performance degradation, and severe drift**. ADWIN and statistical methods can be used for drift detection (Bifet & Gavalda, 2007).

9. Evaluation Metrics

The framework can be evaluated using **drift detection, model performance, explainability, and system-level metrics**. Drift detection can be measured using **detection rate, false-positive rate, detection delay, precision, and recall** (Gama et al., 2014). Model performance can be evaluated using **accuracy, precision, recall, F1-score, and ROC-AUC**. For explainability, **top-k important features, feature contribution consistency, and explanation stability** can be considered using SHAP-based analysis (Lundberg & Lee, 2017). Finally, **detection time, processing overhead, and diagnostic generation time** can be used to evaluate the practical efficiency of the proposed framework.

10. Results

Since the proposed framework requires experimental implementation, actual numerical values should be generated after conducting the experiments. The expected results can be presented using the following structure.

Table 1. Experimental Results Format

Experimental Scenario	Drift Detected	Performance Change	Major Contributing Features	Diagnostic Decision
No Drift	No	Stable	No significant change	Continue Monitoring
Single Feature Drift	Yes	Minor/Stable	Feature X	Investigate Data
Multiple Feature Drift	Yes	Moderate	Feature X, Y	Detailed Investigation
Drift + Performance Loss	Yes	Significant	Feature X, Y	Retraining Recommended
Severe Drift	Yes	Severe	Multiple Features	Model Redevelopment

11. Discussion

The proposed framework addresses the limitation of treating drift detection and explainability as independent activities. Conventional monitoring can indicate that a deployed model is experiencing a change, but the information may not be sufficient for understanding the cause of that change.By integrating XAI, the proposed framework can provide feature-level diagnostic information. For example, if a model experiences performance degradation after deployment, the system can identify whether the change is associated with one particular feature or with multiple features.This approach is particularly useful because not every detected drift requires immediate retraining. A distribution change without significant performance degradation may simply require continued monitoring. In contrast, significant drift accompanied by performance degradation and major changes in feature contributions may indicate the need for retraining.The framework therefore supports a transition from **simple monitoring to explainable diagnosis**.

This idea is consistent with research showing that model explanations can themselves be monitored to identify concept drift (Demšar & Bosnić, 2018). It also addresses the broader production challenges identified in ML engineering research, where monitoring and maintenance are essential components of reliable ML systems (Sculley et al., 2015; Breck et al., 2017). Another advantage is improved communication between technical and domain users. Instead of presenting only a statistical drift score, the system can provide an interpretable message such as:

“Warning: Feature X shows significant distribution change and has become a major contributor to model predictions. Model performance has decreased; retraining should be considered.”Such explanations can make model monitoring more useful for decision-makers.

12. Practical Significance

The proposed framework can be applied in several domains.

Domain	Possible Application
Healthcare	Patient-risk prediction
Finance	Credit-risk prediction
Agriculture	Crop disease prediction

Domain	Possible Application
Education	Student performance prediction
Cybersecurity	Intrusion detection
E-commerce	Recommendation systems
Transportation	Traffic or demand prediction

13. Limitations

The proposed framework has some limitations.

First, reliable performance monitoring requires actual labels, which may not always be immediately available in real-world applications. Second, different drift detection techniques may produce different results depending on the dataset and threshold selection. Third, XAI explanations are themselves approximations and should not always be interpreted as causal relationships. Finally, the framework may introduce additional computational overhead when explanations are generated continuously.

Future implementations should therefore evaluate the trade-off between monitoring frequency, computational cost, detection accuracy, and explanation quality.

14. Conclusion

Machine learning models deployed in real-world environments are exposed to changing data distributions and operating conditions. Such changes can cause concept drift and gradually reduce model performance. Conventional monitoring techniques can detect distribution changes or performance degradation, but they often provide limited information about the underlying causes.This paper proposed a **Unified Drift-Aware Explainable Diagnostic Framework** that integrates drift detection, model-performance monitoring, XAI, and diagnostic decision-making into a single deployment pipeline.

The key contribution of the framework is its ability to connect three important questions:

What changed? → How did the model respond? → Why did the change occur?

Statistical drift detection methods identify changes in production data, while SHAP or LIME can identify the features associated with changes in model predictions. The framework then combines these findings with model-performance information to generate an interpretable diagnostic report.The proposed approach can help transform conventional ML monitoring from a simple alert-generation mechanism into an **explainable diagnostic system**. Future work will focus on implementing the framework using real-world datasets, comparing multiple drift detection algorithms, evaluating explanation stability, and developing a real-time monitoring dashboard.

References

1.Amershi, S., Begel, A., Bird, C., DeLine, R., Gall, H., Kamar, E., Nagappan, N., Nushi, B., & Zimmermann, T. (2019). Software engineering for machine learning: A case study. *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*, 291–300. https://doi.org/10.1109/ICSE-SEIP.2019.00042

2.Bifet, A., & Gavalda, R. (2007). Learning from time-changing data with adaptive windowing. *Proceedings of the 2007 SIAM International Conference on Data Mining*, 443–448.

3.Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. *2017 IEEE International Conference on Big Data*, 1123–1131. https://doi.org/10.1109/BigData.2017.8258038

4.Demšar, J., & Bosnić, Z. (2018). Detecting concept drift in data streams using model explanation. *Expert Systems with Applications, 92*, 546–559. https://doi.org/10.1016/j.eswa.2017.10.003

5.Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*.

6.Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys, 46*(4), Article 44, 1–37. https://doi.org/10.1145/2523813

7.Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems, 30*, 4765–4774.

8.Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering, 31*(12), 2346–2363.

9.Molnar, C. (2022). *Interpretable machine learning*. https://christophm.github.io/interpretable-ml-book/

10.Page, E. S. (1954). Continuous inspection schemes. *Biometrika, 41*(1/2), 100–115. https://doi.org/10.1093/biomet/41.1-2.100

11.Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

12.Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. *Advances in Neural Information Processing Systems, 28*, 2503–2511.

13.Webb, G. I., Lee, L. K., Petitjean, F., & Goethals, B. (2017). Understanding concept drift. *arXiv preprint arXiv:1704.00362*.

14. Minakshi Vasant Tambe. Reshaping The Labor Landscape: Artificial Intelligence and The Future of Workforce Transformation in The Global Economy. International Journal of Contemporary Research in Multidisciplinary. 2025: 4(3):38-43