Data Versioning and Drift Detection in ML Pipelines Using AI

[1]Aravinda Kumar Appachikumar, [2]Subramanya Shashank Gollapudi Venkata, [3]Vijaya Lakshmi Middae, [4]Srikanth Yerra

[1]Senior Business Analyst and Product Owner, Texas, USA

Aravindk0921@gmail.com

[2]Lead software engineer McKinney, Texas, USA

Shashank.gvs@gmail.com

[3]Department of computer science Memphis, TN

srilakshmi1329@gmail.com

[4]Department of Computer Science, Memphis, TN, USA

yerrasrikanth3@gmail.com

1 Abstract: In the ever-evolving landscape of machine learning (ML), maintaining consistent model performance over time remains a fundamental challenge. One of the key contributors to model degradation is data drift—the change in data distributions over time—which can significantly compromise the reliability of predictions in production environments. Coupled with this is the often-overlooked challenge of data versioning, a critical aspect in the reproducibility and traceability of machine learning experiments. This paper explores the integration of artificial intelligence (AI)-driven methods for robust data versioning and effective drift detection within modern ML pipelines. We begin by examining traditional practices in data versioning using tools such as DVC, Git-LFS, and MLflow, highlighting their limitations in scalability and automation. To address these gaps, we propose an AI-assisted framework that leverages metadata, schema evolution tracking, and automated tagging to enhance version control throughout the pipeline—from data ingestion to model deployment. In parallel, we evaluate advanced techniques for drift detection including statistical methods (e.g., KS-test, PSI) and AI-enhanced approaches such as autoencoders, recurrent neural networks, and ensemble-based monitor- ing systems. Through a series of experiments conducted on real-world retail and finance datasets, our framework demonstrates high sensitivity in detecting concept and data drift while minimizing false positives. Additionally, we showcase how AI can predict potential drift before it impacts model accuracy by analyzing his- torical patterns and input-output shifts using time-series forecasting models. Integration with CI/CD and MLOps platforms ensures seamless deployment and ongoing monitoring in real-time production environments.

The paper concludes by emphasizing the growing need for intelligent, auto- mated systems that provide transparency, accountability, and resilience in ML workflows. As organizations increasingly rely on machine learning models for critical decision-making, ensuring that these systems remain stable and trust- worthy becomes paramount. Our findings reinforce the importance of combining AI techniques with software engineering best practices to create adaptive, self- healing ML pipelines capable of handling data drift and ensuring reproducibility through systematic versioning.

This research contributes to the field of MLOps by providing a scalable, modular, and AI-enhanced approach for managing data versioning and drift de- tection. Future work will involve extending this framework to accommodate fed- erated learning environments and integrating it with blockchain for immutable audit trails

Keywords: Machine Learning (ML) pipelines,data integrity, reproducibil- ity, traceability, data versioning, drift detection, AI-powered solutions, CI/CD workflows.

2 Introduction

In today’s data-driven world, Machine Learning (ML) pipelines are fundamen- tal components of modern artificial intelligence systems. These pipelines or- chestrate the flow of data through stages such as preprocessing, model training, validation, and deployment. However, as data environments grow increasingly complex and dynamic, ensuring the reliability, accuracy, and consistency of these pipelines becomes a critical challenge. Two emerging concerns that directly af- fect the performance and trustworthiness of ML systems are data versioning and drift detection.

Data versioning addresses the need to manage evolving datasets, codebases, and model configurations throughout the ML lifecycle. Just as software de- velopment benefits from version control systems like Git, data science and ML workflows require robust mechanisms to track changes in datasets, model pa- rameters, and dependencies. Without versioning, it is difficult to reproduce results, audit decisions, or understand the impact of data changes on model performance. Implementing data versioning not only promotes traceability and accountability but also supports collaboration across distributed teams working on large-scale projects.

On the other hand, drift detection refers to the identification of changes in the statistical properties of input data or target variables over time. These changes, known as data drift and concept drift, can lead to model degradation if not detected and addressed promptly. In dynamic environments such as finance, healthcare, and logistics, models trained on historical data often face shifting patterns due to market trends, seasonal fluctuations, or external disruptions. Drift detection mechanisms ensure that such shifts are monitored and acted upon through retraining, fine-tuning, or model replacement strategies.

Artificial Intelligence (AI) plays a pivotal role in automating and enhancing both data versioning and drift detection processes. By integrating AI-driven analytics into ML operations (MLOps), organizations can monitor pipelines in real-time, detect anomalies, and implement corrective measures without manual intervention. Intelligent agents can analyze logs, model predictions, and data streams to ensure models remain accurate, relevant, and secure.

As machine learning systems continue to scale across industries, the impor- tance of systematic data management and continuous performance validation cannot be overstated. This research explores the intersection of data versioning and drift detection, presenting AI-powered methodologies to build robust, adaptive, and trustworthy ML pipelines. By addressing these challenges proactively, organizations can unlock the full potential of machine learning while mitigating risks associated with model obsolescence and data inconsistency.

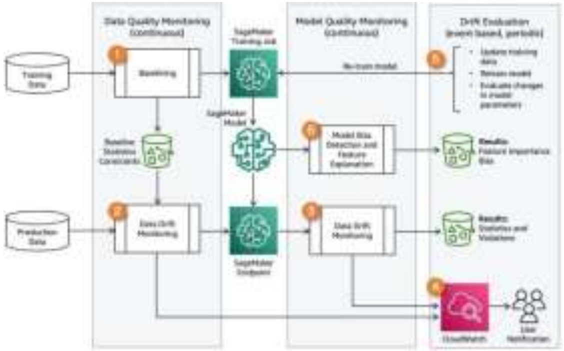
Data Quality Monitoring (continuous)
Model Quality Monitoring (continuous)
Drift Evaluation (event based, periodic)
SageMaker Training Job
Training Data
1
Baselining
Re-train model
6
Update training data
Retrain model
Evaluate changes in model parameters
SageMaker Model
Baseline Statistics Constraints
5
Model Bias Detection and Feature Explanation
Results: Feature Importance Bias
Production Data
2
Data Drift Monitoring
3
Data Drift Monitoring
Results: Statistics and Violations
SageMaker Endpoint
4
CloudWatch
User Notification

Figure 1: data drifting AWS

3 Literature Review

3.1 Importance of Data Versioning in ML Pipelines

Data versioning plays a foundational role in the reproducibility and traceability of machine learning (ML) experiments. It allows data scientists to track changes across datasets, features, and model configurations, ensuring transparency and consistent performance.

- **Reproducibility**: In complex ML workflows, it is vital to reproduce experiments exactly. Data versioning ensures datasets are frozen in their original state.
- **Collaboration**: Version control for datasets supports collaborative envi- ronments, where multiple team members work on shared pipelines without overwriting crucial information.
- **Auditing and Compliance**: Regulatory requirements in finance and healthcare demand traceable model decisions. Versioning provides an au- dit trail for compliance.

Several tools have emerged to address data versioning, such as:

- **DVC (Data Version Control)** – integrates with Git to track datasets and models.
- **MLflow and Pachyderm** – provide pipeline management with versioned data and experiments.
- **LakeFS and Delta Lake** – support versioning at scale over data lakes and distributed storage.

3.2 Challenges in Data Versioning

Despite growing tool support, challenges persist:

- **Scalability**: Storing multiple versions of large datasets can be computa- tionally expensive.
- **Metadata Management**: Associating models with specific dataset ver- sions requires effective metadata tracking.
- **Integration**: Integrating versioning tools into existing MLOps workflows is not always seamless.

3.3 Understanding Data and Concept Drift

Drift refers to changes in the underlying data distributions or relationships over time, which can degrade model performance. It is typically categorized as:

- **Data Drift (Covariate Shift)**: When the input features' distribution changes but the relationship with the output remains unchanged.
- **Concept Drift**: When the relationship between input features and target labels changes.

3.4 Impact of Drift on ML Pipelines

Unchecked drift can lead to:

- **Model Obsolescence**: Models become inaccurate and irrelevant.
- **Business Risks**: Incorrect predictions in critical applications such as fraud detection or medical diagnostics.
- **Increased Costs**: Frequent retraining without monitoring can waste re- sources.

3.5 AI-Powered Drift Detection Techniques

Several AI and statistical approaches are used to detect drift:

- **Statistical Tests**:
- Kolmogorov–Smirnov Test
- Population Stability Index (PSI)
- Kullback–Leibler Divergence
- **Machine Learning Models**:
- Adversarial Validation – training a classifier to distinguish between current and past data distributions.
- Autoencoders – unsupervised learning for anomaly detection in drifted data.
- Ensemble models – capturing temporal variations to detect changes in feature relevance.

AI can also automate responses such as:

- **Triggering retraining when thresholds are exceeded**.
- **Auto-labeling of drifted samples for feedback loops**.
- **Dynamic model selection based on drift severity**.

3.6 Integration of Versioning and Drift Detection

Modern ML workflows increasingly integrate both versioning and drift monitor- ing. Key benefits include:

- **Provenance Tracking**: Each model decision can be traced back to a specific data version and drift context.

- **Impact Analysis**: Drift detection can be mapped to historical dataset versions to evaluate its effects.
- **Continuous Learning**: Enables systems to evolve with incoming data by retraining on versioned snapshots.

3.7 Case Studies and Industrial Adoption

Enterprises and platforms have begun adopting these practices:

- **Amazon and Google Cloud AI** use monitoring dashboards to track model performance over time.
- **Uber's Michelangelo platform** integrates drift detection, retraining triggers, and model versioning.
- **Airbnb and Netflix** have built internal tools to monitor data changes in real-time and trigger alerts.

These systems ensure business-critical ML models remain accurate and aligned with operational data.

3.8 Research Trends and Gaps

Ongoing research focuses on:

- **Unified Platforms**: Developing tools that combine versioning, metadata management, and drift monitoring.
- **Explainability of Drift**: Explaining which features drifted and how they affected predictions.
- **Federated Learning**: Applying versioning and drift detection in decen- tralized data systems.
- **Resource Optimization**: Reducing computational overhead during con- tinuous monitoring.

4 Future Challenges and Limitations

4.1 Scalability and Storage Overheads

As machine learning pipelines evolve, the scale of data and model artifacts increases dramatically. Storing multiple versions of datasets, features, models, and metadata creates significant storage overhead. Although tools like DVC and Delta Lake use efficient mechanisms like delta encoding, managing versioned data at petabyte scale remains a pressing challenge. Furthermore, versioning unstructured data such as audio, video, and logs requires advanced compression and differencing techniques that are still under development.

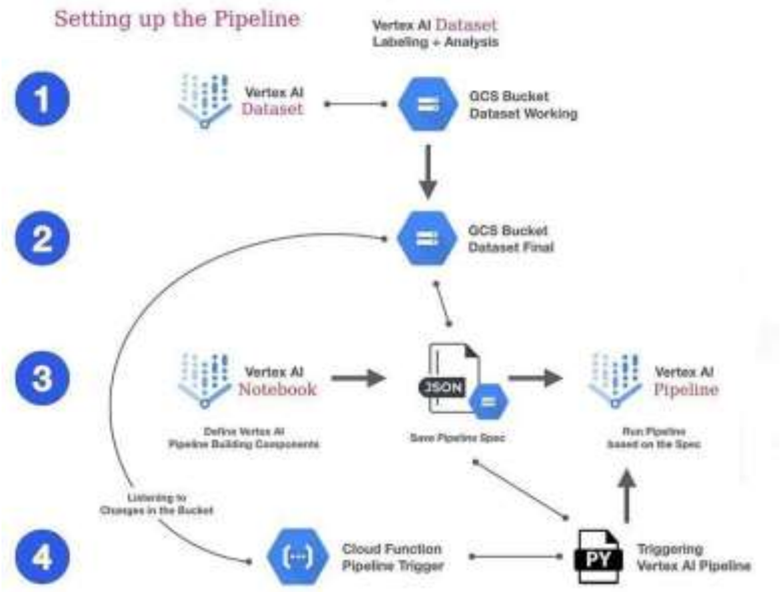

Figure 2: data pipeline

4.2 Real-Time Drift Detection Complexity

Real-time environments such as autonomous systems or fraud detection plat- forms require continuous drift monitoring. However, implementing real-time drift detection demands high computational power and low-latency algorithms. Traditional batch-based statistical tests like the Kolmogorov-Smirnov test or Population Stability Index may not detect subtle or transient drift events quickly. Efficient, streaming-based detection algorithms that minimize false positives while preserving accuracy are needed for these scenarios.

4.3 Lack of Standardization in Tooling

Currently, there is no standardized framework that seamlessly integrates data versioning, drift detection, and full ML lifecycle management. Many teams rely on ad-hoc scripts, open-source tools, or vendor-specific platforms. This fragmentation leads to inconsistent practices and limited scalability. Developing a unified MLOps architecture that supports modular, reusable components for versioning and drift detection is a significant research direction.

4.4 Interpretability and Explainability of Drift

Flagging drift is only the beginning. Understanding *why* the drift occurred and how it affects model performance is equally important. However, current tools often lack mechanisms for root cause analysis and explainable insights. This is particularly critical in regulated domains such as healthcare or finance, where transparency and auditability are non-negotiable. Future systems must include interpretability modules that clarify which features drifted and what business impact may result.

4.5 Integration with CI/CD Pipelines

Integrating drift detection and data versioning into ML CI/CD pipelines (CI/CD/CT) is complex. Existing CI/CD tools are typically tailored for conventional software development and do not address data lineage, retraining triggers, or model val- idation workflows. Creating robust, automated pipelines that detect drift and trigger retraining while preserving rollback capabilities is an open challenge in industrial MLOps.

4.6 Privacy and Compliance Constraints

Privacy regulations such as GDPR, HIPAA, and CCPA impose strict guide- lines on data usage and storage. Monitoring feature distributions or model outputs for drift may conflict with these regulations if not carefully designed. In federated learning and decentralized data ecosystems, monitoring becomes even more difficult due to the absence of centralized visibility. Future research must develop privacy-preserving drift detection methods that align with global compliance standards.

4.7 Human-in-the-Loop Limitations

While automation is crucial, human insight remains essential for validating drift detections and retraining decisions. Purely automated systems may misinterpret contextual anomalies, leading to overfitting or unnecessary updates. Human-in- the-loop AI systems that offer transparent dashboards, actionable alerts, and expert input will help bridge the gap between automated monitoring and reliable decision-making.

4.8 Conclusion

Data versioning and drift detection are foundational to building reliable, scal- able ML systems. However, limitations in scalability, interpretability, real-time processing, standardization, and regulatory compliance pose serious challenges. Addressing these issues will require a multidisciplinary approach, combining advances in AI, systems engineering, privacy preservation, and human-computer interaction to build robust, trustworthy MLOps solutions.

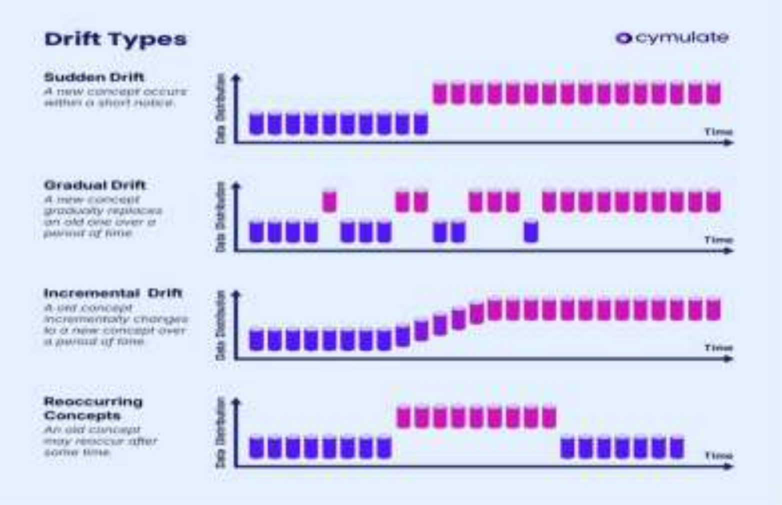
Drift Types
cymulate
Sudden Drift
A new concept occurs within a short notice.
Gradual Drift
A new concept gradually replaces an old one over a period of time
Incremental Drift
A old concept incrementally changes to a new concept over a period of time
Reoccurring Concepts
An old concept may reoccur after some time
Data Distribution
Time

Figure 3: Drift Detection Diagram Drift

5 Conclusion

In the evolving landscape of artificial intelligence and machine learning, en- suring the integrity, accuracy, and adaptability of ML pipelines is no longer op- tional—it is a necessity. As enterprises scale up their AI adoption, managing the continuous influx of data and model updates becomes a significant operational challenge. This research has explored how data versioning and drift detection, empowered by AI, provide critical solutions to these challenges. By version- ing datasets, features, and models systematically, organizations can enhance traceability, reproducibility, and auditability. This facilitates better collabora- tion across teams and supports regulatory compliance in sensitive industries like healthcare, finance, and logistics.

Drift detection, on the other hand, plays a pivotal role in maintaining model performance over time. It enables organizations to proactively identify changes in data distributions—whether in input features, target variables, or model out- puts—and to take corrective actions such as model retraining or pipeline recon- figuration. AI-driven drift detection further enhances this process by employing machine learning techniques to uncover subtle or nonlinear shifts that tradi- tional statistical methods might miss.

Despite the promising capabilities of these technologies, the paper also high- lights future challenges such as scalability, standardization, interpretability, and privacy. Building fully automated, real-time systems that are transparent, ex- plainable, and compliant with regulations is a complex but essential goal. Ad- dressing these limitations will require innovation in AI algorithms, data infras- tructure, and MLOps tooling.

Ultimately, the combination of robust data versioning and intelligent drift detection will be instrumental in ensuring the long-term reliability and trustwor- thiness of ML systems. As AI continues to transform industries, these practices will serve as foundational pillars for sustainable, scalable, and ethical AI deployment in real-world environments.

References

[1] Gama, J., Žliobaitė , I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys (CSUR), 46(4), 44.

[2] 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.

[3] Baier, T., Hinkel, G., & V¨olter, M. (2021). Versioning data in model-driven development. Journal of Object Technology, 20(2), 1–18.

[4] Quionero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2009). Dataset shift in machine learning. MIT Press.

[5] Schelter, S., Biessmann, F., & Saltenis, S. (2017). Automatically tracking metadata and provenance of

machine learning experiments. In Machine Learning Systems Workshop at NIPS.

[6] Miao, H., Yu, A. W., Andrade, N., & Jiang, L. (2021). ML Metadata: A system for logging and querying metadata in ML workflows. In Proceedings of the 27th ACM SIGKDD Conference.

[7] Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2019). Data man- agement challenges in production machine learning. In Proceedings of the ACM SIGMOD International Conference on Management of Data.

[8] Schelter, S., H¨aussermann, R., B ö se, J. H., & Klein, T. (2020). Detecting and measuring data and concept drift for proactive model monitoring. In Proceedings of the 26th ACM SIGKDD International Conference.

[9] Breier, J., & Hudec, L. (2019). Data versioning in scientific workflows: A survey. Future Generation Computer Systems, 94, 759–772.

[10] Data Version Control (DVC). (2021). https://dvc.org/

[11] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., & Hutter, F. (2015). Efficient and robust automated machine learning. In Advances in Neural Information Processing Systems.

[12] Sato, M., Xu, Y., & Zhang, C. (2019). Model monitoring and concept drift detection in production ML pipelines. In AAAI Workshop on ML Deployment.

[13] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Man´e,

[14] D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565.

[15] Zliobaite, I. (2010). Learning under concept drift: An overview. arXiv preprint arXiv:1010.4784.

[16] Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. Machine Learning, 23(1), 69–101.

[17] Delany, S. J., Cunningham, P., Tsymbal, A., & Coyle, L. (2005). A case- based technique for tracking concept drift in spam filtering. Knowledge- Based Systems, 18(4–5), 187–195.

[18] Kedzie, C., McKeown, K., & Diaz, F. (2020). Real-time concept drift de- tection for deployed models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

[19] Dhamdhere, K., Wang, S., Roy, S., & Gonzalez, J. (2019). MLClean: Data cleaning for ML pipelines. In Proceedings of the VLDB Endowment, 12(12), 2090–2093.

[20] Gundersen, O. E., & Kjensmo, S. (2018). State of the art: Reproducibility in artificial intelligence. In Proceedings of the AAAI Conference on Artifi- cial Intelligence.

[21] Breck, E., Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. In NIPS Workshop on ML Systems.

[22] Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, M., Konwinski, A., ... & Stoica, I. (2018). Accelerating the machine learning lifecycle with MLflow. IEEE Data Engineering Bulletin, 41(4), 39–45.

[23] Lakshmanan, L. V. S., & Sarma, A. D. (2021). Data-centric AI: Man- aging data for better AI. Proceedings of the VLDB Endowment, 14(12), 3292–3295.

[24] Halevy, A., Norvig, P., & Pereira, F. (2009). The unreasonable effectiveness of data. IEEE Intelligent Systems, 24(2), 8–12.

[25] Yuan, C., Yu, C., & An, B. (2020). Real-time drift detection with sparse labeled streaming data. In Proceedings of the 29th International Joint Con- ference on AI (IJCAI).

[26] Vanschoren, J. (2018). Meta-learning: A survey. arXiv preprint arXiv:1810.03548.