

Original Article

Data Quality Engineering for AI Systems: A Survey of Validation, Drift Detection, and Monitoring Techniques

Raviteja Narra

Independent Researcher.

Received Date: 09 August 2025

Revised Date: 05 September 2025

Accepted Date: 30 September 2025

Abstract: *The increased dependence of industries on data-driven decision-making practices makes it obvious that quality data and governance systems are necessary. Contemporary organisations have to deal with extensive and heterogeneous data sets and are highly concerned about the quality of data and compliance. This survey overcomes such obstacles by suggesting a taxonomy of traditional data-quality dimensions, e.g., accuracy, completeness, and timeliness, and machine-learning-specific dimensions, e.g., training-serving skew, label noise, feature freshness, and feedback-loop contamination. It further offers a structured comparison, mapping validation, drift detection, and monitoring methods to the circumstances in which each method is best suited. The reviewed articles cover the period of 2022-2024 of algorithmic drift detection, serverless monitoring architecture, deep-learning-based validation, and industry-based reliability. With this synthesis, a number of relevant trends can be identified: model-based drift detection, cost-efficient data cleaning and adaptive monitoring. At the same time, there are still gaps, e.g., the lack of standardized benchmarks, the limited scalability of available solutions, and insufficient enterprise preparedness to implement them. This survey thus makes data-quality engineering an important foundation for trustworthy artificial intelligence, identifying its benefits and drawbacks. Sometimes, more specific taxonomies and practical frameworks can inform practitioners to build pipelines that sustain the results of the model and guarantee future confidence in real-world deployments.*

Keywords: *Data quality engineering, Artificial intelligence, Data validation, Concept drift, Data monitoring and drift detection.*

I. INTRODUCTION

In the modern data-driven ecosystem, effective artificial intelligence, quality decision-making, and enduring organisational success are determined not only by access to data but also by the quality of the data in its lifecycle. Data quality failures are not abstract: when quality degrades, the consequences are concrete, and models are silently undermined, their reliability eroding without warning. As an example, historical distributions were found to be unsuitable for models in the presence of changing environments: changes in user behaviour, seasonal, ageing sensors, or external shocks. These changes create latent inconsistencies and distributional drift, which cannot be addressed using the existing batch-based retraining or fixed validation pipelines.

A real-life example could be the financial services industry, where the model of fraud detection, being trained on the historical patterns of transactions, is initially successful; however, as the fraudsters change their methods to introduce new paths of transactions, new payment platforms, or imitate the behaviour of legitimate customers, the underlying data distribution shifts [1]. Lack of well-defined drift-detection and adaptive monitoring mechanisms decreases model accuracy, fraud is not identified, and financial losses are accrued, and thus undermines customer trust. It can also be seen in healthcare sensor networks, where the calibration drift of medical equipment can cause medical data to be inaccurate, and in customer churn prediction, where market dynamics change and cause the features of past data to become outdated [2].

These failure modes bring out the ineffectiveness of current measures, including periodical retraining, manual audits, or isolated detection of anomalies. They do not reflect recurrent patterns, trade-offs between stability and plasticity, or the use of past knowledge. As a result, artificial intelligence will bear an unjustifiable overhead, will slow down to evolve, and will silently deteriorate in organisations. This motivates the development of a Data Quality Engineering discipline for AI systems that integrates automated validation, drift detection, and lifecycle monitoring [3]. Structural correctness, completeness, and semantic consistency must be verified on a continuous basis, while distributional shifts and concept drift must be detected proactively to avert reliability collapse.

This survey systematically examines leading practices in data validation, drift detection, and monitoring, analyzing the strengths and limitations of each approach with respect to production AI deployments. It seeks to assist the researchers and practitioners in developing scalable, automated, and reliable data-quality pipelines to facilitate the use of AI in dynamic



operations by researching the most up-to-date frameworks and clarifying gaps, including the absence of cost-effective, consistent solutions.

The three major contributions of the survey are as follows. First, it synthesizes existing research on validation, drift detection, and monitoring techniques used in AI data pipelines. Second, it provides a comparative analysis of recent methods with regard to scalability, operational limitations and practical implementation factors. Third, it identifies key research gaps, including the lack of benchmark datasets, the inability to evaluate monitoring frameworks in production environments, and the problem of integrating governance mechanisms and AI monitoring systems.

A. Structure of Paper

The structure of the paper is as follows: Section II describes the principal data-quality dimensions, methods, and practices. Section III deals with data validation and monitoring procedures. Section IV addresses the concept drift, data drift, and detection algorithms. Section V provides a literature review of recent frameworks, and Section VI presents the conclusion of the paper, including the key findings, challenges, and future research directions.

B. Literature Search Methodology

The literature included in this survey was identified through systematic searches of IEEE Xplore, ACM Digital Library, Scopus, and Google Scholar. Relevant keywords were: data validation, data drift detection, machine learning monitoring, concept drift, data quality engineering, training-serving skew, feature freshness and ML pipeline reliability. The review has prioritized peer-reviewed articles published since 2020 with emphasis on practical methods of validation, drift detection and continuous monitoring of AI systems used in production. Survey papers, empirical studies, and framework proposals were all considered. Articles that only deal with database-layer quality management and not with direct applicability to ML were excluded from consideration.

II. DATA QUALITY IN THE AI ERA

With the rise of artificial intelligence and machine learning as the new paradigm of data-driven decision-making, the standards of data quality have been fundamentally transformed. Incomplete information does not just reduce efficiency; it introduces failure points that silently erode model reliability and compromise business outcomes. In one case, a failure of fraud-detection systems may occur where the attacker develops training-serving skew; that is, features that occur in the training phase but do not occur in the production phase, leading to failure of fraudulent activity detection and monetary loss. Similarly, leakage of noise or labels in churn-prediction pipelines can cause the model to learn spurious relationships and generate inaccurate retention policies. In sensor-based healthcare, feature freshness is crucial: delayed responses to vital signs can cause false alarms. Even when there is contamination of feedback loops, that is, when model predictions affect future data collection, there is a likelihood of bias and corruption of distributions.

These machine-learning-specific measures, including skew, noise, freshness and contamination, cannot be solved using the traditional data-quality models (accuracy, completeness, timeliness). Existing mechanisms, including periodical retraining or, at most, sustaining at rest, cannot reduce these problems, which result in silent degradation, enhanced bias and unprofitable business consequences. This necessitates the development of data-quality engineering for AI systems, which integrates both classical and machine-learning-specific dimensions into automated pipelines for validation, drift detection, and monitoring. Table I presents the key features of the data quality with the definitions, real-life illustrations, and measurement procedures.

Table 1: Core dimensions of data quality

Dimension	Definition	Practical Example	Measurement Approach
Accuracy	The extent to which data accurately depicts an actual thing or event[4].	The patient's documented blood pressure is consistent with the measurement.	Comparing with a reliable source or the real deal.
Completeness	The degree of presence of all necessary data[5].	The customer database has every client's contact information.	Proportion of missing fields; ratio of necessary to accessible values.
Consistency	Lack of inconsistencies across and within datasets.	The date of a patient's birth appears consistently in both insurance and electronic health records.	cross-database and cross-field validation tests.
Timeliness	The extent to which data is current and accessible when required.	Real-time updates on stock market pricing.	The delay between data creation and its use.
Validity	The extent to which data complies with specified formats, guidelines, or ranges.	The official national standard governs postal codes.	Range restrictions, format checks, and validation rules.

Relevance	Data suitability for the desired purpose.	Evaluation of a novel treatment incorporating data from clinical trials.	Expert judgement and conformity to analytical or decision-making requirements.
Uniqueness	The extent to which duplicate records are absent from the data.	A single, distinct medical record number is assigned to each patient.	Algorithms for record linkage and duplicate detection.

A. Data Quality Methods and Best Practices

The practices that need to be directly integrated in machine-learning pipelines, and not exist in abstract governance or customer-relationship-management operations, are necessary to guarantee the high-quality of data in AI systems [6]. Machine-learning reliability engineering is focused on automation, reproducibility and monitoring. The ML reliability engineering focuses on automation, reproducibility, and monitoring. Key practices include:

a) Feature Engineering and Validation

Feature engineering is the foundation of reliable ML pipelines, but it is also a common source of silent errors. To ensure that features are as expected, engineers would need to enforce schema validation, type checks, and range constraints at ingestion. Benchmarking training and serving features in shadow deployments, along with automated unit testing of feature transformations, can help identify training-serving skew before it negatively impacts model performance. This proactive validation would ensure that the features employed in production are exactly what the design intended.

b) Label Integrity

Supervised learning relies on labels, and the integrity of those labels directly influences the reliability of the model. Common failure modes include label noise (wrong labels), and label leakage (training on future information). To overcome these risks, engineers can cross-verify labels through cross-validation, use noise estimation algorithms, and deploy leakage detectors. Models can additionally be stress-tested through synthetic label injection to evaluate robustness under mislabeling conditions. These practices allow models to record massive patterns as opposed to chance correlations.

c) Pipeline Monitoring & Drift Detection

Machine-learning pipelines need to be monitored so that anomalies and model drift are identified after deployment. Statistical measures that are used to compute distribution changes include the population stability index (PSI), KL divergence, or Wasserstein distance. Monitoring concept drift, calibration and error rates is model-centric, and adaptive concept drift detectors such as ADWIN or DDM are real-time detectors of drift. A combination of these mechanisms can also be used to prevent silent degradation by warning engineers when data distributions or model behaviour deviate from expected norm.

d) Feature Freshness and Latency Control

In production systems, stale features or delayed features may be just as bad as erroneous features. Feature-freshness checks are utilized in cases where there are time-sensitive features like sensor data or transaction logs that are required to be updated within acceptable latency constraints. Freshness dashboards and timestamp validation are tools used by engineers in order to detect delays in features. By implementing freshness service-level agreements (SLAs) teams can eliminate false alarms and prediction errors in safety-critical systems, including healthcare monitoring and fraud detection.

e) Feedback Loop Mitigation

The feedback loop occurs when the model results affect the subsequent data collection and hence pollute results and the cycle of bias [7]. As an example, it is possible that the recommender systems enhance popularity bias by repeatedly suggesting the same items. Engineers reduce this using counterfactual assessment, audit debiasing, and randomized exploration strategies. These practices encourage fairness and long-term reliability of system by making sure that training data are not corrupted by the feedback loops.

f) Reproducibility and Automation

Debugging, auditing and reproducibility are all necessary for scaling machine learning pipelines. However, engineers do so by versioning datasets, features, and models using tools such as MLflow or DVC. Automated CI/CD pipelines enforce validation checks and retraining workflows, while canary testing allows safe rollout of new models [8]. By making the pipeline reproducible and automatable, teams can ensure that experiments are reproducible, failures can be traced, and deployments do not vary across environments. Practically, these best practices are not theoretical principles of governance but tangible engineering work that will ensure the reliability of ML pipelines in production.

III. DATA VALIDATION AND MONITORING

Since ML systems operate on data, verifying the data consumed during training and prediction is a prerequisite for proper and healthy ML systems. ML systems can have relevant data validation dimensions that are broadly classified into

notions of data quality typically considered in classical database management systems (DBMS) and ML model-dependent dimensions [9]. Some of the dimensions are emphasized below, with examples for each category.

Combined data from different sources are presented in Fig. 1; features are isolated, and ML models translate these features into predictions, which are served to users, whose behavioral signals are provided as feedback to the system. It is only possible to monitor and verify these data streams at scale through automated tests. Nonetheless, in every phase, confirming the information requires various skills and, in many cases, domain knowledge. This poses new technical and conceptual challenges for engineering teams when automating ML production systems.

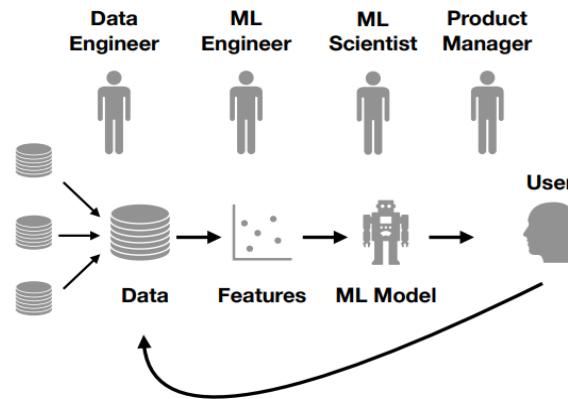


Figure 1: Responsibilities for ML production software system [10].

Effective validation requires both domain expertise and automated tooling; the appropriate method depends on data type, pipeline latency constraints, and the failure modes most likely to occur in a given production environment.

A. Data Monitoring Techniques

The various methods of monitoring (see Table II) are applied for different purposes within the ML pipeline [11]. It is up to the engineers to determine when to employ a specific type of failure mode and when not to, based on the nature of the data and the constraints on operations.

Table 2: Methods for Data Validation and Monitoring

Method	Detects	Best For	Downside	Reference
Rule Checks (Schema/Format Validation)	Missing fields, invalid ranges, type mismatches	Early-stage ingestion, enforcing business rules	Cannot detect subtle drift or semantic errors	[12]
Statistical Drift Tests (PSI, KS, KL, Wasserstein)	Distributional shifts in features	Large-scale tabular data, periodic monitoring	Sensitive to sample size; may flag benign changes	[13][14][15]
Adaptive Window Detectors (ADWIN, DDM)	Concept drift in streaming data	Real-time pipelines, non-stationary environments	May overreact to noise; tuning window size is critical	[16][17]
ML-based Detectors (Isolation Forest, Autoencoders, LSTM)	Complex anomalies, nonlinear drift	High-dimensional data, sensor networks, cybersecurity	Computationally expensive; risk of false positives	[18][19]
Model Performance Monitoring	Accuracy drop, calibration drift, confidence shifts	Production models with feedback loops	Requires labeled feedback; lag in detection	[20]

Note: Method selection should be guided by data type, pipeline latency requirements, and operational cost constraints. Hybrid combinations, pairing rule-based checks with statistical drift detection and model-centric monitoring provide the strongest defense against silent failures in production AI systems.

Effective data validation and monitoring avoid error propagation, minimize model bias, increase system robustness, and control regulatory compliance [21]. In safety-critical healthcare, smart cities, cybersecurity, automated pipelines, and industrial automation, automated validation pipelines and drift-aware monitoring systems enable early detection of data and system drift and of malfunctions, ensuring sound decision-making and the long-term sustainability of operations.

IV. DATA DRIFT AND CONCEPT DRIFT

Drift in machine-learning systems may be classified into concept drift and data drift, and the difference between them is the most crucial thing for reliability engineering. Data drift occurs when the statistical distribution of input features shifts over time while the underlying relationship between inputs and outputs remains stable [22], [23]. As an illustration, sensor calibration errors can cause an offset in the temperature sensor's value distribution without the actual target changing. When this happens, the engineers are advised to use statistical drift-detection techniques (e.g., PSI, KS tests) and to recalibrate features or freshness checks to restore input integrity. The various notion drift kinds and the sorts of drifts in speed are likewise depicted in Figs. 2 and 3.

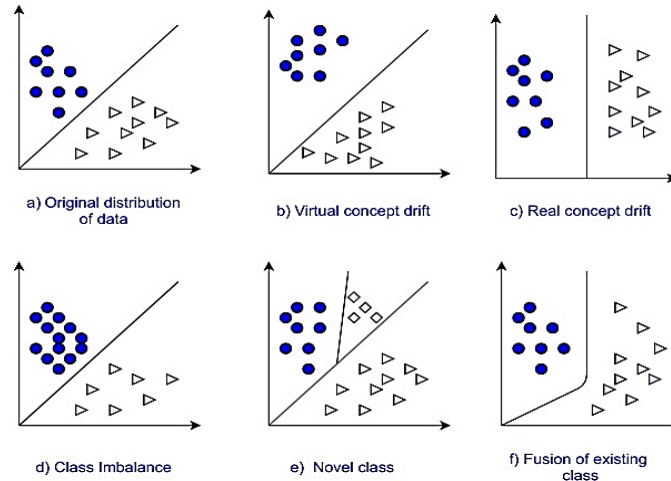


Figure 2: Representation of Different Types of Concept Drifts.

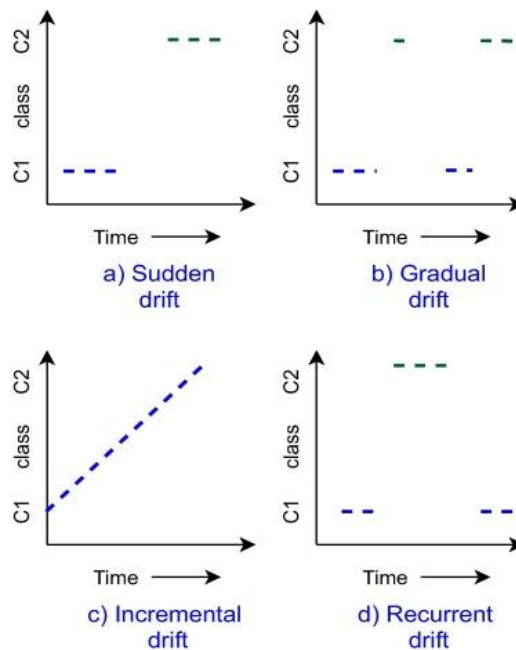


Figure 3: Types of Concept Drifts in Terms of Speed.

A. Real Concept Drift

It may be described as an effect in which the correlation between the goal variable and the input attributes differs. This can be due to the varying needs, fashion, or marketplace conditions. Example: This can be without working in a sentiment analysis model, which was trained on social media data, but the concept changes, such as through a change in slang or reference.

B. Virtual Concept Drift

Virtual concept drift occurs because $P(X|c)$ conditional probability, changes without modifying $P(C|X)$ (posterior probability). There is also preservation of the decision border. Data distribution within the same category is nonetheless affected. Virtual drift can exacerbate class imbalance. In a binary fraud detection setting, for instance, fraudulent transactions may constitute as little as 1% of all records, rendering the minority class progressively underrepresented over time.

C. Abrupt Drift

The previous concept is abruptly replaced by the new idea of the incoming data stream. Thus, abrupt drift refers to the moment when the old notion abruptly changes to the new concept. It detects abrupt declines in the classifier's accuracy. Following that, the model immediately picks up on the features of the new notion, and the modifications are implemented accordingly.

D. Gradual Drift

In contrast to abrupt drift, the idea change duration in gradual drift is comparatively longer. This kind of drift comes in two varieties: slow gradual drift and regular gradual drift. For example, when market dynamics change due to a recession or inflation, the steady drift is seen. This type of drift contains overlapping concepts that eventually stabilize.

E. Class Prior

This form of drift arises when the prior probability distribution over target classes shifts over time, altering the baseline frequency of each class in the incoming data stream. Example: The rise in fraudulent transactions around the holidays is one event that causes prior probability drift in a credit card fraud detection system.

F. Recurrent Drift

In recurrent drift, previously observed concepts re-emerge in the data stream after a significant interval. This pattern may be cyclic, as with seasonal fluctuations, or acyclic, when a concept recurs without a fixed period. When seasonal fluctuations occur, the cyclic phenomena are employed. For instance, the summer season increases sales of chilled goods. The acyclic impact is seen when the price of fuel rises and, in a typical situation, the price of electricity falls back to its initial level.

a) Drift Detection Algorithms

A number of drift detection techniques evaluate the classifier's accuracy [24]. because variations in the data distribution cause the classifier's accuracy to deteriorate over time. The computational procedures for addressing drift are given in Table III, and the mapping of drift types to engineering response is shown in Table IV. It still needs to keep an eye on things. Concept drift may be monitored using a number of techniques, some of which are below:

Table 3: Algorithmic steps to handle concept drift

Steps	Name of Steps	Methodology
Step1	Data processing methods	<ul style="list-style-type: none"> • Statistical-based methods • Similarity and dissimilarity-based methods • Significance analysis-based methods • Data distribution-based methods • Decision boundary-based methods • Model-dependent methods • Window-based methods: Fixed and Adaptive windowing methods • Sequential analysis-based methods
Step2	Data learning process	<ul style="list-style-type: none"> • Single learner • Ensemble learner
Step3	Concept drift monitoring methods	<ul style="list-style-type: none"> • Supervised methods • Unsupervised methods • Semi-supervised methods
Step4	Concept drift handling	<ul style="list-style-type: none"> • Informed methods • Blind methods
Step5	Performance measures	<ul style="list-style-type: none"> • Recall, Precision, F-measure, AUC, ROC • Time and space complexity • Change in Accuracy • Parameter settings • Reliability

Table 4: Recommended Engineering Responses by Drift Type

Drift Type	Recommended Response
Abrupt Drift	Immediate full retraining on recent data window
Gradual Drift	Sliding window retraining with increasing weight on recent samples

Recurrent Drift	Seasonal or ensemble models that retain historical concept memory
Virtual Drift	Feature recalibration and class rebalancing; no model retraining required
Class Prior Drift	Decision threshold recalibration based on updated class frequencies

V. LITERATURE REVIEW

This section offers a concise synopsis of the body of research on data quality tool surveys. Table V highlights the Analysis of Related Works on Data Quality Engineering for AI System in Below:

A. Drift Detection Frameworks

Yu et al. (2024) address concept drift in dynamic and heterogeneous networks supporting H2M applications by proposing a novel framework, the concept drift detection and adaptation (CDDA) framework, in order to quickly react and adjust to the concept drift. CDDA learns the traffic characteristics of H2M applications and combines offline and online learning processes to enhance H2M traffic prediction and improve bandwidth prediction performance. Results from an investigation using experimental traffic from H2M applications over a 10Gb/s passive optical network simulator show that CDDA can more rapidly respond to concept drift and better predict the bandwidth of changing H2M applications and network load [25]. Wang et al. (2023) provide a model-centric method for identifying concept drift that emphasizes model change above model output, employing deep neural networks to identify concept drift. Transfer learning has also been created to minimize computational complexity by freezing portions of the network and speeding up the drift detection procedure. Suggest using both large and small time periods to distinguish between the actual and possible detected drifts in order to further reduce false detected drifts. The efficacy of the suggested framework has been demonstrated through experiments using both simulated and real-world datasets [26]. Zhang, Tino and Yao (2023) propose the observation of both class-conditional and marginal distributions in a lower-dimensional space that is specifically selected in relation to the given classification task. Comparative tests have demonstrated that HRDD outperforms its rivals in detection recall, precision, and F-measure across a range of idea drift types, including subtle drifts, and achieves high-quality performance on high-dimensional data streams [27]. Sisniega et al. (2024) promote the use of these methods in ML deployment pipelines and emphasize the need to include drift detection as a basic prerequisite for creating reliable and trustworthy AI systems. In this manner, businesses may take advantage of serverless computing's advantages while proactively identifying and reducing the negative consequences of covariate drift [28].

B. Industrial Reliability and Predictive Maintenance

Nahvi et al. (2022) developed a technique that uses data from non-faulty cooling towers to determine the extent of the issue. A multilayer perceptron model's model estimate error is tracked using a concept drift detection method. An increase in model estimating error suggests a shift in system behaviour and a higher chance of failure [29]. Xia et al. (2022) suggested the COPPM algorithm finds the most economical maintenance cycle, work sequence, and machine choice for every task. To validate the cost-effectiveness of COPPM, computational studies involving the wind turbine blade process are shown. Under different issue sizes, the COPPM reduces overall costs by 13.68% compared to the conventional production and maintenance scheduling system [30]. Zürcher et al. (2022) suggest using the current process monitoring system to infer information about equipment status in three steps. The difficulties associated with using process monitoring data for a secondary use are addressed by the established technique. The implementation of the technique using real-world industry data is demonstrated through two case studies. The results of the case study were confirmed by analyzing data from the facility's historical maintenance records. The application's success demonstrates that maintenance should be a component of overall system design and marks a significant step towards data-driven equipment reliability evaluation and maintenance planning in pharmaceutical manufacturing [31].

Table 4: Comparative Analysis of Related Works on Data Quality Engineering for AI system

Author	Framework	Research Focus / Objective	Limitation	Contribution
Yu et al. (2024)	Concept Drift Detection and Adaptation Framework (CDDA)	To detect and adapt rapidly to concept drift in heterogeneous human-to-machine (H2M) network traffic.	Requires continuous retraining and substantial computational resources; effectiveness depends on the realism of traffic simulations.	Combines offline and online learning for fast drift adaptation and improves bandwidth prediction accuracy under dynamic network conditions.
Wang et al.	Transfer Learning-Based	Analysing internal model behaviour	May fail to detect very subtle drifts; the	Reduces computational cost, minimizes false drift alarms via

(2023)	Model-Centric Deep Drift Detection	instead of output statistics in order to identify idea drift.	effectiveness of transfer learning depends on the similarity between the source and target domains.	long-short time-window analysis, and accelerates drift detection.
Zhang, Tino & Yao (2023)	HRDD (High-Resolution Drift Detection)	To monitor marginal and class-conditional distributions in reduced-dimensional feature space for high-dimensional data streams.	Dimensionality reduction may lose critical features; performance is sensitive to parameter tuning.	Demonstrates high recall, precision, and robustness across multiple drift types, including gradual and subtle drifts.
Céspedes Sisniega et al. (2024)	Drift Detection for Serverless ML Pipelines	To integrate drift detection as a core component of trustworthy AI deployment pipelines.	Lacks empirical validation on large-scale real-world production systems; mainly conceptual and architectural.	Establishes drift monitoring as a mandatory requirement for reliable ML deployment and promotes proactive mitigation strategies in serverless environments.
Nahvi et al. (2022)	MLP Error-Based Drift Detection	To detect early defects by monitoring estimation error changes in industrial systems.	Limited to single-system environments and supervised model dependency; not suitable for complex multi-variable systems.	Enables predictive maintenance by linking increasing model error with system behavior change and failure risk.
Xia et al. (2022)	COPPM (Cost-Optimized Production & Preventive Maintenance)	Determines cost-effective maintenance cycle, job order, and machine selection	Focused on wind turbine blades; domain-specific	Demonstrates 13.68% average cost reduction compared to conventional production and maintenance scheduling systems, validated through computational studies on wind turbine blade manufacturing processes.
Zürcher et al. (2022)	Three-Step Process Monitoring Methodology	Uses existing monitoring infrastructure to infer equipment condition	Limited to case studies; secondary application challenges	Provides lessons for integrating maintenance into holistic system design; validated with industrial data

C. Research Gaps and Open Problems

- **Absence of Standardized Benchmarks:** No common evaluation datasets or unified metrics currently exist to enable direct, reproducible comparison of drift detection methods across application domains.
- **Scalability Limitations:** The majority of reviewed methods are validated on small-scale or simulated datasets. Rigorous evaluation under production-scale conditions remains an unaddressed gap.
- **Enterprise Readiness:** Organizational, tooling, and integration barriers to deploying hybrid pipelines are consistently acknowledged across reviewed works, but have not been systematically studied or resolved.

Table VI presents the data validation and monitoring methods.

Table 5: Practical Comparison of Data Validation and Monitoring Methods

Method	Best Use	Strengths	Limitations	Practical Takeaway
Rule-Based Validation	Schema enforcement, ingestion checks	Simple, fast, interpretable	Cannot detect subtle drift or semantic errors	Always use at ingestion to prevent deterministic failures
Statistical Drift Detection (PSI, KS, KL, Wasserstein)	Monitoring tabular feature distributions	Quantitative thresholds, scalable	Sensitive to sample size; may flag benign changes	Best for batch/tabular monitoring; combine with model monitoring

Adaptive Detectors (DDM)	Window (ADWIN)	Streaming/non-stationary environments	Dynamic adjustment, real-time detection	Mays are hyper-reactive to noise; turnitiveness.	Ideal for real-time pipelines where concepts evolve
ML-Based Detectors (Isolation Forest, Autoencoders, LSTM)		High-dimensional, nonlinear, sensor/cybersecurity data	Capture complex anomalies	Complex to compute; probability of false positive results.	Use for complex domains; balance cost vs accuracy
Model-Centric Monitoring		Production models with feedback loops	Directly tracks accuracy, calibration, and confidence drift	Requires labeled feedback; detection lag	Essential for production reliability; complements all other methods

VI. CONCLUSION AND FUTURE WORK

With the acceleration of digital transformation, data quality has emerged as a foundational enabler of trusted analytics and credible artificial intelligence systems. Because organisations are increasingly dependent on automated decision-making using data, the quality and consistency of data are no longer a one-off issue of timeliness and validity, but a permanent operational process. This survey makes two primary contributions to the literature on data quality engineering for AI systems. First, it presents a taxonomy that brings together classical dimensions of data quality, including accuracy, completeness, and timeliness, as well as ML-specific ones, such as training-serving skew, label noise or leakage, feature freshness, and feedback loop contamination. Second, it provides an effective roadmap for engineers, mapping validation, drift detection, and monitoring techniques to real-world applications and explaining when each technique is most useful. The future of the research should follow the following directions:

- **Benchmark Datasets:** Publicly available datasets of drift-detection evaluation are needed to make the comparison of methods across domains and scales of deployment reproducible.
- **Scalable Real-Time Monitoring:** The existing approaches are not yet validated on a production scale; the studies should be conducted on monitoring overhead, the latency bottlenecks, and distributed implementation designs.
- **Automated Remediation Pipelines:** Beyond detection, future systems should automate the remediation response, including selective retraining, threshold recalibration, and feature refresh with minimal human intervention.
- **Governance and Monitoring Integration:** Enterprise AI systems require more integration between data governance programs and operational monitoring equipment, especially in regulatory compliance contexts.

VII. REFERENCES

- [1] X. Zheng, P. Li, X. Hu, and K. Yu, "Semi-supervised classification on data streams with recurring concept drift and concept evolution," *Knowledge-Based Syst.*, vol. 215, p. 106749, Mar. 2021, doi: 10.1016/j.knsys.2021.106749.
- [2] F. Bachinger, L. Ehrlinger, G. Kronberger, and W. Wössl, "Data Validation Utilizing Expert Knowledge and Shape Constraints," *J. Data Inf. Qual.*, vol. 16, no. 2, pp. 1-27, Jun. 2024, doi: 10.1145/3661826.
- [3] T. Reynolds, I. Painter, and L. Streichert, "Data Quality: A Systematic Review of the Biosurveillance Literature," *Online J. Public Health Inform.*, vol. 5, no. 1, Mar. 2013, doi: 10.5210/objphi.v5i1.4376.
- [4] C. Bidlack and M. P. Wellman, "Exceptional Data Quality Using Intelligent Matching and Retrieval," *AI Mag.*, vol. 31, no. 1, pp. 65-73, Mar. 2010, doi: 10.1609/aimag.v31i1.2280.
- [5] L. Ehrlinger and W. Wössl, "A Survey of Data Quality Measurement and Monitoring Tools," *Front. Big Data*, vol. 5, pp. 1-30, Mar. 2022, doi: 10.3389/fdata.2022.850611.
- [6] D. Sculley et al., "Hidden Technical Debt in Machine Learning Systems," *NIPS'15 Proc. 29th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, pp. 2503-2511, 2015.
- [7] S. Amrale, "A Novel Generative AI-Based Approach for Robust Anomaly Identification in High-Dimensional Dataset," *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 4, no. 2, pp. 709-721, Oct. 2024, doi: 10.48175/IJARST-19900D.
- [8] D. Kreuzberger, N. Kühn, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, vol. 11, pp. 31866-31879, 2023, doi: 10.1109/ACCESS.2023.3262138.
- [9] R. Dattangire, R. Vaidya, D. Biradar, and A. Joon, "Exploring the Tangible Impact of Artificial Intelligence and Machine Learning: Bridging the Gap between Hype and Reality," in *2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET)*, IEEE, Aug. 2024, pp. 1-6. doi: 10.1109/ACET61898.2024.10730334.
- [10] F. Biessmann, J. Golebiowski, T. Rukat, D. Lange, and P. Schmidt, "Automated Data Validation in Machine Learning Systems," *IEEE Comput. Soc.*, pp. 1-14, 2021.
- [11] S. Garg, "AI/ML Driven Proactive Performance Monitoring, Resource Allocation and Effective Cost Management in SAAS Operations," *Int. J. Core Eng. Manag.*, vol. 6, no. 6, pp. 263-273, 2019.
- [12] H. Li, X. Liu, H.-C. Chiu, D. Li, N. Zhang, and C. Xiao, "Drift: Dynamic rule-based defense with injection isolation for securing LLM agents," *arXiv p*, 2025.
- [13] R. Taplin and C. Hunt, "The Population Accuracy Index: A New Measure of Population Stability for Model Monitoring," *Risks*, vol. 7, no. 2, p. 53, May 2019, doi: 10.3390/risks7020053.

- [14] Y. Qu, J. Blanchet, and P. Glynn, "Computable bounds on convergence of Markov chains in Wasserstein distance via contractive drift," *Ann. Appl. Probab.*, vol. 35, no. 4, pp. 2678–2715, 2025.
- [15] T. Nitta, Y. Shi, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Detecting data drift with ks test using attention map," in *Asian Conference on Pattern Recognition*, 2023, pp. 68–80.
- [16] A. Bifet and R. Gavaldà, "Learning from Time-Changing Data with Adaptive Windowing," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, Philadelphia, PA: Society for Industrial and Applied Mathematics, Apr. 2007, pp. 443–448. doi: 10.1137/1.9781611972771.42.
- [17] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*, 2004, pp. 286–295.
- [18] B. X. Yong, Y. Fathy, and A. Brintrup, "Bayesian Autoencoders for Drift Detection in Industrial Environments," in *2020 IEEE International Workshop on Metrology for Industry 4.0 & IoT*, IEEE, Jun. 2020, pp. 627–631. doi: 10.1109/MetroInd4.0IoT48571.2020.9138306.
- [19] Y. Tan, C. Hu, K. Zhang, K. Zheng, E. A. Davis, and J. S. Park, "LSTM-based anomaly detection for non-linear dynamical system," *arXiv Prepr. arXiv2006.03193*, 2020.
- [20] M. Yusuff, "Model Drift Monitoring: Continuously Tracking Model Performance Metrics to Detect Accuracy Degradation." December, 2024.
- [21] F. Hinder, V. Vaquet, and B. Hammer, "One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift," *Front. Artif. Intell.*, vol. 7, Jun. 2024, doi: 10.3389/frai.2024.1330257.
- [22] T. R. Hoens, R. Polikar, and N. V. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Prog. Artif. Intell.*, vol. 1, no. 1, pp. 89–101, Apr. 2012, doi: 10.1007/s13748-011-0008-0.
- [23] A. Mallick, K. Hsieh, B. Arzani, and G. Joshi, "Matchmaker: Data Drift Mitigation in Machine Learning for Large-Scale Systems," in *Proceedings of Machine Learning and Systems*, D. Marculescu, Y. Chi, and C. Wu, Eds., 2022, pp. 77–94.
- [24] R. S. M. Barros and S. G. T. C. Santos, "A large-scale comparison of concept drift detectors," *Inf. Sci. (Ny)*, vol. 451–452, pp. 348–370, Jul. 2018, doi: 10.1016/j.ins.2018.04.014.
- [25] X. Yu, L. Ruan, J. S. Evans, and E. Wong, "Novel Concept Drift Detection and Adaptation (CDDA) Framework for Human-to-Machine (H2M) Applications over Future Communication Networks," in *ICC 2024 - IEEE International Conference on Communications*, IEEE, Jun. 2024, pp. 5509–5514. doi: 10.1109/ICC51166.2024.10622259.
- [26] P. Wang, N. Jin, D. Davies, and W. L. Woo, "Model-centric transfer learning framework for concept drift detection," *Knowledge-Based Syst.*, vol. 275, p. 110705, Sep. 2023, doi: 10.1016/j.knosys.2023.110705.
- [27] S. Zhang, P. Tino, and X. Yao, "Hierarchical Reduced-space Drift Detection Framework for Multivariate Supervised Data Streams," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2628–2640, 2023, doi: 10.1109/TKDE.2021.3111756.
- [28] J. C. Sisniega, V. Rodríguez, G. Moltó, and Á. L. García, "Efficient and scalable covariate drift detection in machine learning systems with serverless computing," *Futur. Gener. Comput. Syst.*, vol. 161, pp. 174–188, Dec. 2024, doi: 10.1016/j.future.2024.07.010.
- [29] S. Nahvi et al., "Enhancing cooling tower performance with condition monitoring and machine learning based drift detection," *Procedia CIRP*, vol. 112, pp. 146–150, 2022, doi: 10.1016/j.procir.2022.09.063.
- [30] T. Xia et al., "Collaborative production and predictive maintenance scheduling for flexible flow shop with stochastic interruptions and monitoring data," *J. Manuf. Syst.*, vol. 65, pp. 640–652, Oct. 2022, doi: 10.1016/j.jmsy.2022.10.016.
- [31] P. Zürcher, S. Badr, S. Knüppel, and H. Sugiyama, "Data-driven equipment condition monitoring and reliability assessment for sterile drug product manufacturing: Method and application for an operating facility," *Chem. Eng. Res. Des.*, vol. 188, pp. 301–314, Dec. 2022, doi: 10.1016/j.cherd.2022.09.005.