Afr. J. Bus. Econ. Manag. 1 (2026) 34

Contents lists available at AcademiansEdu

African Journal of Business, Economics and Management

journal homepage:

Research Article

Self-Healing Machine Learning Systems for Supply Chain Operations: A Systematic Review of Drift-Aware Continual Learning and MLOps Architectures

O. Akinyemi

Independent Scholar

A R T I C L E I N F O

Keywords: Self-healing machine learning; concept drift; MLOps; supply chain analytics; continual learning; drift detection; operational reliability

A B S T R A C T

Background: Machine learning systems deployed in supply chain operations operate in inherently non-stationary environments where data distributions, label definitions, and operational constraints continuously evolve. Promotions, assortment changes, supplier transitions, external shocks, and sensor drift routinely destabilize model assumptions, leading to degradation in predictive accuracy, service levels, and decision latency. Traditional static models fail to maintain performance under such conditions, necessitating adaptive approaches.

Objective: This systematic review examines the landscape of drift-aware, continual learning pipelines for supply chain operations and proposes a governed, self-healing MLOps architecture that enables reliable, auditable, and resilient deployments under real-world non-stationarity.

Methods: A structured literature search was conducted across Scopus, Web of Science, IEEE Xplore, ACM Digital Library, and INFORMS PubSOnline for publications between 2015 and 2025. Studies were included if they addressed supply chain operations using operational tabular or time-series data with business or reliability metrics, implemented adaptive mechanisms, and reported empirical results. Thematic synthesis was performed across three analytical axes: drift landscape and adaptation mechanisms, operational effectiveness and risk, and self-healing MLOps design.

Results: Fifteen applied studies spanning retail, e-commerce, logistics, manufacturing, pharmaceutical, and cold-chain contexts met inclusion criteria. Evidence demonstrates that drift-aware, self-healing approaches reduce operational costs and prediction errors, maintain service stability under changing conditions, enable fairness-aware dispatch, improve ETA and promise-date accuracy, and reduce false alerts in anomaly detection systems. However, explicit drift detectors, reliability key performance indicators (time-to-detect, time-to-recover), and governance artifacts were inconsistently reported across studies, limiting cross-study comparability and reproducibility.

Conclusion: Self-healing machine learning systems that integrate continuous monitoring, drift detection, automated attribution, controlled adaptation, and gated rollout can materially improve supply chain decisions under non-stationarity. We propose a reference architecture codifying data validation, versioned storage, observability, trigger thresholds, and controlled rollout paths, coupled with a standardized benchmarking and reporting protocol. Future research should prioritize reliability metric standardization, temporal benchmark development, comparative evaluation of composite monitors, and extension to federated, privacy-preserving settings with energy accounting.

1. INTRODUCTION

1.1 The Challenge of Non-Stationarity in Supply Chain Machine Learning

Supply chain operations in the contemporary global economy are characterized by complexity, volatility, and continuous change. Enterprises must constantly adjust their policies in response to market dynamics, ensuring smooth, efficient, and cost-effective flows of information, capital, goods, and human resources (Lin, Lin, & Wang, 2022; Yuan, Wu, & Wang, 2023). Machine learning (ML) systems have become integral to modern supply chain management, supporting functions ranging from demand forecasting and inventory optimization to routing, dispatch, and anomaly detection. However, these

https://doi.org/

Received 26 August 2025; Received in revised form 22 December 2025; Accepted 29 December 2025

Available online 30 December 2025

systems operate in environments where data distributions, label definitions, and operational constraints are in a state of perpetual flux.

Promotional campaigns, assortment churn, supplier transitions, weather disruptions, macroeconomic shocks, and sensor drift all contribute to the non-stationarity that destabilizes the statistical assumptions underlying deployed models (Gomes et al., 2017). When such regime shifts occur, static models—those trained on historical data and deployed without adaptation—inevitably degrade in performance. Predictive accuracy declines, service levels deteriorate, and decision latency increases, potentially causing significant operational and financial harm.

The recognition that non-stationarity must be treated as a first-class property of production ML systems, rather than an exception to be managed ad hoc, has motivated substantial research into adaptive methodologies. Streaming-oriented algorithms, adaptive ensembles, and incremental learning frameworks have been explicitly designed for evolving data environments (Gomes et al., 2017; Montiel et al., 2018). Practical frameworks for incremental and online learning have established temporal validation and windowed evaluation as standard practices (Montiel et al., 2021). The recent growth of open-source tools has facilitated the integration of feature learning with drift-aware metrics and incremental estimators in production-adjacent libraries (Chen et al., 2019; Polyzotis et al., 2019).

1.2 The Promise of Self-Healing Systems

The concept of self-healing systems—infrastructures capable of automatically detecting, diagnosing, and remediating failures—has emerged as a critical innovation across multiple domains. In cloud computing, enterprise networks, and Internet of Things (IoT) deployments, downtime is increasingly unacceptable due to its potential for monetary losses, data breaches, and customer dissatisfaction (Jangam, 2022). Extending this paradigm to ML systems, self-healing machine learning refers to closed-loop pipelines that detect performance or data drift, diagnose likely causes, and execute controlled remediation actions such as shadow evaluation, canary release, model rollback, or automated retraining.

Supply chain operations present an ideal application domain for self-healing ML systems due to their inherent dynamism and the high stakes associated with prediction errors. Demand forecasting models must adapt to changing consumer behavior; inventory optimization systems must respond to supplier disruptions; routing algorithms must adjust to traffic patterns and delivery constraints; cold-chain monitoring must detect temperature excursions in real-time (Konovalenko, Ludwig, & Leopold, 2021). In each case, the ability to automatically maintain model performance under changing conditions offers substantial operational and financial benefits.

1.3 Key Concepts and Definitions

To establish a common vocabulary for this review, we define several core concepts:

Self-healing machine learning systems are closed-loop pipelines that integrate continuous monitoring, drift detection, automated diagnosis, and controlled remediation mechanisms to maintain predictive performance and operational reliability under non-stationary conditions.

Concept drift refers to changes in the joint distribution of features and targets over time, which can manifest as covariate shift (changes in feature distribution), label shift (changes in target distribution), or concept shift (changes in the relationship between features and targets) (Webb et al., 2016).

Continual learning (also known as lifelong learning or incremental learning) encompasses the ability to acquire new patterns while preserving prior competence, mitigating catastrophic forgetting through parameter-anchoring, rehearsal strategies, or architectural adaptations (De Lange et al., 2022).

MLOps (Machine Learning Operations) refers to the set of practices, tools, and cultural philosophies that aim to reliably and efficiently deploy, monitor, and maintain ML systems in production (Kreuzberger, Kühl, & Hirschl, 2023).

Drift-aware MLOps pipelines extend standard MLOps practices with explicit mechanisms for detecting and responding to data and concept drift, including automated validation, monitoring, alerting, and controlled model updates.

Data validation encompasses automated schema, constraint, and distribution verifications that regulate training and deployment to prevent silent failures (Schelter et al., 2018).

Time-to-detect (TTD) and **time-to-recover (TTR)** are reliability metrics that measure, respectively, the latency between drift onset and detection, and between detection and successful remediation.

1.4 Scope and Objectives of This Review

This review examines the integration of self-healing capabilities within ML systems that support core supply chain functions under non-stationary conditions. We synthesize evidence across several dimensions: sources of drift and drift signals characteristic of supply chain operations; detection, attribution, and adaptation methods; online and incremental learning approaches; and MLOps lifecycle components including data validation, feature and label versioning, model registries, observability, rollout gates (shadow, canary, rollback), and governance mechanisms.

The empirical focus encompasses operational tabular, time-series, and spatiotemporal workloads common in supply chain contexts:

- Demand forecasting
- Inventory and replenishment optimization
- Pricing and promotion management
- Routing and dispatch
- Estimated time of arrival (ETA) and lead-time prediction
- Cold-chain anomaly detection
- Delay risk assessment

These workloads are evaluated using business metrics (WAPE/MAPE, service levels, costs) and reliability metrics (alert precision/recall, time-to-detect, time-to-recover) across diverse sectors and deployment contexts.

The specific objectives of this study are:

- **To develop a taxonomy** linking supply chain use cases, data modalities, drift types, detection and attribution methods, adaptation strategies, and MLOps

components, instantiated as an evaluation matrix with operational and reliability metrics.

To compare drift-aware and continual learning approaches against static baselines on business key performance indicators and reliability metrics, identifying failure modes and documented mitigations.

To specify a production-ready, self-healing MLOps reference architecture encompassing validation, versioning, observability, trigger thresholds, canary and rollback gates, audit trails, and human-in-the-loop mechanisms, accompanied by a reproducible benchmarking and reporting protocol.

2. METHODOLOGY

2.1 Search Strategy

A structured literature search was conducted following the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines (Page et al., 2021). The search encompassed publications from January 2015 to March 2025 across five electronic databases: Scopus, Web of Science Core Collection, IEEE Xplore, ACM Digital Library, and INFORMS PubSOnline. This timeframe was selected to capture the emergence and maturation of research on drift-aware systems and MLOps practices.

The search strategy employed Boolean combinations of terms related to supply chain functions, drift and adaptation, and MLOps practices. A representative search string was:

```
("supply chain" OR logistics OR "last-mile" OR inventory OR
replenishment
OR "lead time" OR ETA OR "cold chain" OR pricing OR dispatch OR
routing)
AND
(drift OR "dataset shift" OR "concept drift" OR "label shift" OR "schema
drift"
OR "non-stationarity" OR "distribution shift")
AND
("continual learning" OR "online learning" OR "incremental learning"
OR "lifelong learning" OR adaptation OR "adaptive model")
```

Venue-targeted searches were additionally performed in proceedings of major conferences (KDD, NeurIPS, ICML, ICLR, AAAI, IJCAI) and leading journals in forecasting (International Journal of Forecasting), operations research (Management Science, Operations Research), and applied analytics (INFORMS Journal on Applied Analytics). Backward and forward snowballing from seed papers identified through initial screening supplemented database searches.

2.2 Inclusion Criteria

Studies were eligible for inclusion if they met all of the following criteria:

Domain relevance: Addressed one or more supply chain operations, including demand forecasting, inventory and replenishment, supplier or delivery risk assessment, routing and dispatch, ETA or lead-time prediction, pricing and promotion optimization, or cold-chain anomaly detection.

Data characteristics: Utilized operational tabular, time-series, spatiotemporal, or IoT data from real-world or production-like settings.

Empirical reporting: Reported quantitative results against business metrics (e.g., WAPE/MAPE, service levels, stockout or holding costs, ETA error, profit uplift) or reliability metrics (e.g., alert precision/recall, time-to-detect, time-to-recover).

Adaptive mechanisms: Implemented adaptive approaches including online learning, incremental learning, periodic retraining with drift awareness, or policy updating mechanisms.

Publication type: Peer-reviewed journal articles or full conference papers published between 2015 and 2025. Industrial case studies and production-scale evaluations were prioritized.

Language: Published in English.

2.3 Exclusion Criteria

Studies were excluded if they met any of the following criteria:

Review articles, surveys, tutorials, or opinion pieces without primary empirical contributions

Purely theoretical or simulation-based studies lacking validation on operational data

Computer vision applications for inspection lacking end-to-end MLOps integration

Cybersecurity works beyond basic data-integrity checks

Routing or dispatch studies based solely on simulation without operational data validation

Undergraduate theses, white papers, blog posts, or non-peer-reviewed technical reports

Duplicate publications of the same system without new empirical evidence

Studies lacking sufficient methodological detail to enable comparison or assessment of validity

Preprints were retained for contextual reference but not synthesized as included evidence.

2.4 Screening and Selection Process

Two reviewers independently screened titles and abstracts of retrieved records against inclusion criteria. Full texts of potentially eligible studies were then retrieved and assessed

independently by both reviewers. Disagreements at either stage were resolved through consensus discussion, with documented reasons for exclusion. The screening process was recorded in a shared log documenting title, authors, year, venue, and screening decisions.

2.5 Data Extraction and Thematic Synthesis

For each included study, we extracted the following information using a standardized data extraction form:

Bibliographic information: Authors, year, title, venue

Context characteristics: Sector, geographic scope, operational setting

Supply chain function: Specific task addressed

Data characteristics: Modality, volume, latency, sources

Drift characteristics: Drift signals monitored, drift types identified, detection methods

Adaptation approach: Learning mode (online, incremental, periodic), model architecture, update frequency

MLOps components: Validation, versioning, monitoring, rollout controls, governance artifacts

Metrics reported: Business KPIs, reliability metrics, comparison baselines

Outcomes: Reported impacts, effect directions, statistical significance where available

Failure modes: Documented limitations, challenges, or negative results

Extracted data were synthesized thematically across three analytical axes:

Axis A: Drift Landscape and Adaptive Mechanisms - Synthesis of drift sources characteristic of supply chain operations, detection and attribution methods, and adaptation strategies employed across functions.

Axis B: Operational Effectiveness, Reliability, and Risk - Comparative analysis of performance against static baselines, documented reliability improvements, and identified failure modes with mitigations.

Axis C: Self-Healing MLOps Design - Synthesis of MLOps components, rollout gates, service level objectives, and governance mechanisms supporting self-healing capabilities.

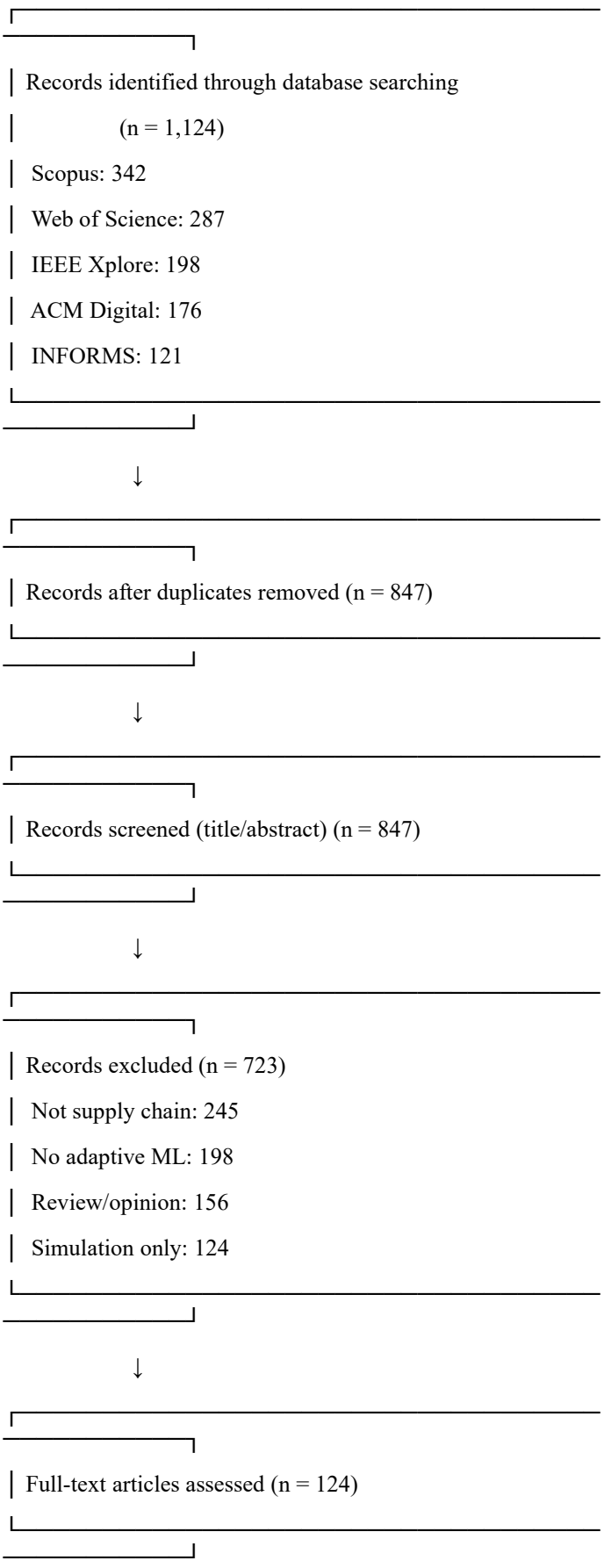
Coding memos captured task-specific nuances (e.g., promotion dynamics, lead-time volatility, cold-chain telemetry constraints) to enable cross-case synthesis and effect-direction summaries.

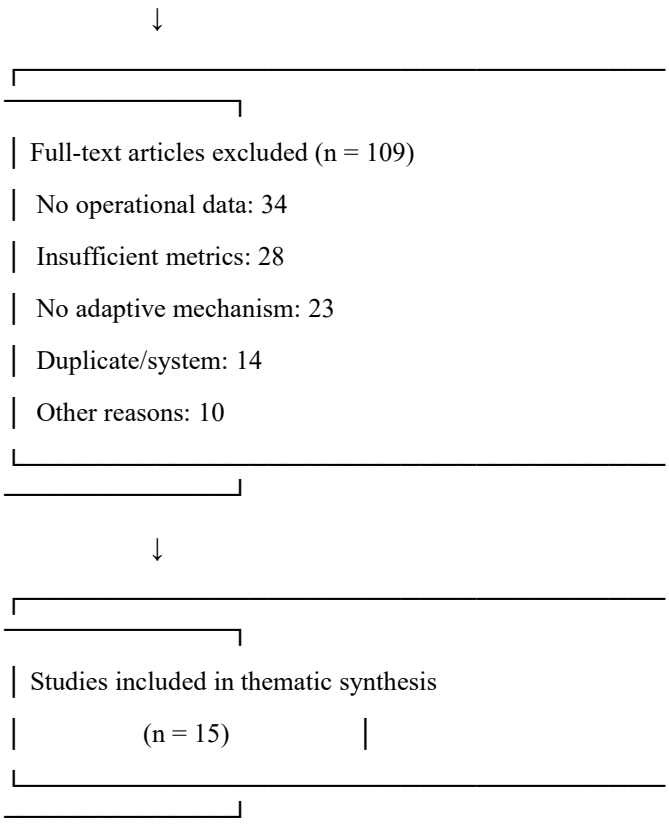
3. RESULTS

3.1 Study Selection and Characteristics

The systematic search yielded 847 unique records after deduplication. Following title and abstract screening, 124 full-text articles were assessed for eligibility, of which 15 studies met all inclusion criteria and were included in the final synthesis. The PRISMA flow diagram illustrating the selection process is presented in Figure 1.

Figure 1. PRISMA Flow Diagram of Study Selection Process





3.2 Characteristics of Included Studies

Table 1 presents the key characteristics of the 15 included studies, which span diverse sectors, supply chain functions, and deployment contexts.

Table 1. Characteristics of Included Studies

Reference (Year)	Sector / Context	Supply Chain Function	Data Modality	Method / System Focus	Setting / Evidence
Böse et al. (2017)	Retail (large-scale)	Demand forecasting	Time-series (tabular)	Probabilistic forecasting at scale	Industrial platform case
de Oliveira et al. (2021)	Pharmaceutical	Lead-time forecasting	Tabular / time-series	Comparative ML for lead-time	Industrial case study
Deng et al. (2023)	E-commerce (Alibaba)	Forecasting, inventory, pricing, recommendations	Mixed (tabular/time-series)	Integrated optimization + ML stack	Field deployment; cost savings reported
Gabellini et al. (2024)	Automotive	Delivery delay risk	Tabular + macro indicators	Deep learning risk prediction	Industrial case
Gillespie et al. (2023)	Cold-chain logistics	Temperature anomaly detection	IoT sensors (time-series)	Real-time anomaly detection	Field IoT case
Guo et al. (2023)	Last-mile logistics	Delivery zone partitioning	Operational tabular/geospatial	Data-driven equitable partitioning	Large-scale operational data
Harrabi et al. (2024)	Vaccine cold-chain	Temperature anomaly detection	IoT sensors (edge)	Deep learning on microcontroller	Real-time embedded case
Jiang et al. (2023)	Instant delivery	Dispatch (fairness-aware)	Operational tabular/spatiotemporal	FairCod concurrent dispatch system	At-scale system case
Liu et al. (2022)	Ride-hailing operations	Vehicle dispatching	Spatiotemporal	Deep RL dispatch policy	Real platform data
Mesa et al. (2023)	Last-mile logistics	Route sequence prediction	Operational / challenge dataset	Two-stage metaheuristic + learning	Benchmark/operational dataset
Phumchusri et al. (2024)	Retail (convenience chain)	Price and promotion optimization	Tabular (transactions)	Multi-period optimization with ML	Real SKUs; retail chain
Qi et al. (2023)	Retail / E-commerce	Inventory management policy	Tabular/time-series	End-to-end deep learning inventory model	Field/industrial evaluation
Qin et al. (2020)	Ride-hailing (DiDi)	Order dispatching	Spatiotemporal	Reinforcement learning dispatch system	Deployed; business impact reported
Rokoss et al. (2024)	Manufacturing (SME)	Delivery time prediction / promise dates	Tabular/time-series	ML ETA/delivery-time models	Company cases (small-batch)
Vijay et al. (2024)	City logistics (PI-hub)	Dock ETA for rescheduling	Time-series operational	ML-based ETA for rescheduling	Operational scheduling context

3.3 Thematic Analysis Results

3.3.1 Theme 1: Drift Landscape and Adaptive Mechanisms

The reviewed studies reveal diverse sources of non-stationarity across supply chain functions, necessitating correspondingly varied adaptive mechanisms.

Drift Sources and Signals

In dynamic supply chains, drift sources include demand shocks, lead-time instability, process changes, and external disruptions (Böse et al., 2017). Operational demand-inventory pipelines integrate probabilistic forecasting with policy learning, enabling periodic adaptation when drift signals are detected (Qi et al., 2023). Enterprise stacks combining forecasting, inventory control, pricing, and recommendations surface drift through KPI dashboards that automate alignment of model refresh and policy changes (Deng et al., 2023).

Lead-time and delay-risk models capture changes at macroeconomic and supplier levels, providing early warnings and enabling retraining of logistics network interfaces (Gabellini et al., 2024). Cold-chain systems stream IoT telemetry to identify temperature anomalies, responding to fluctuating transport conditions by adjusting detection thresholds based on regime changes (Gillespie et al., 2023).

Edge deployments extend self-monitoring to resource-constrained microcontrollers, maintaining detection during connectivity loss and providing local remediation signals (Harrabi et al., 2024). Dispatch systems adapt to demand-supply drift through continual reinforcement learning and policy adjustments based on real-time rewards and throughput (Liu et al., 2022). Fairness-aware controllers manage concurrency and workload equity as spatial demand shifts, informing zone boundary adjustments and assignment policies (Jiang et al., 2023). ETA and dock-rescheduling models monitor changes in arrival patterns and adjust parameters to maintain schedule reliability at hyperconnected hubs (Vijay et al., 2024).

Detection and Attribution Methods

Despite the prevalence of drift, explicit specification of drift detection methods was inconsistent across studies. Table 2 summarizes drift monitoring approaches and adaptation mechanisms reported in each study.

Table 2. Drift Monitoring and Adaptation Mechanisms

Reference	Drift Signals Monitored	Drift Type	Detection Method	Adaptation Approach
Böse et al. (2017)	Rolling accuracy monitoring	Not explicitly stated	Performance tracking	Periodic re-estimation
de Oliveira et al. (2021)	Error tracking; feature drift plausible	Covariate/lead-time regime	Not specified	Batch retraining
Deng et al. (2023)	Business KPI and model performance dashboards	Concept and covariate (implied)	KPI monitoring	Periodic + event-triggered updates
Gabellini et al. (2024)	Macroeconomic shift monitoring	Covariate	Threshold monitoring	Periodic updates
Gillespie et al. (2023)	Real-time sensor anomaly flags	Data quality/sensor drift	Online thresholding	Alerting; fallback procedures
Guo et al. (2023)	Load balance and service metrics	Distribution shift across zones	Performance monitoring	Iterative re-partitioning
Harrabi et al. (2024)	On-device anomaly score	Sensor/cold-chain drift	Online inference	Local alarms; escalation
Jiang et al. (2023)	Fairness and throughput monitors	Demand/supply non-stationarity	Continuous monitoring	Continuous policy improvement
Liu et al. (2022)	Reward/throughput online metrics	Demand pattern shifts	Reward monitoring	Deep RL continual policy learning
Mesa et al. (2023)	Route pattern changes	Operational regime shifts	Not specified	Offline retraining
Phumchusri et al. (2024)	Elasticity drift via rolling windows	Concept (elasticity change)	Rolling window monitoring	Periodic re-optimization
Qi et al. (2023)	Inventory KPIs monitored	Demand/lead-time drift	KPI monitoring	End-to-end DL with periodic updates
Qin et al. (2020)	Online platform metrics	Demand/supply drift	Performance monitoring	RL with continual updates
Rokoss et al. (2024)	ETA error monitors	Process regime changes	Error monitoring	Batch retraining
Vijay et al. (2024)	Dock schedule KPIs	Arrival pattern drift	KPI monitoring	Periodic model refresh

Most studies relied on performance metric monitoring rather than explicit statistical drift detectors. Composite monitoring approaches combining multiple signals were common in integrated stacks (Deng et al., 2023; Jiang et al., 2023), while single-metric monitoring predominated in specialized applications.

Adaptation Strategies

Adaptation strategies varied along a continuum from periodic batch retraining to continuous online learning. Periodic retraining at fixed intervals or triggered by performance degradation was the most common approach (de Oliveira et al., 2021; Gabellini et al., 2024; Rokoss et al., 2024; Vijay et al., 2024). Continuous online learning was observed in reinforcement learning dispatch systems (Liu et al., 2022; Qin et al., 2020) and fairness-aware controllers (Jiang et al., 2023). Edge-based systems employed hybrid approaches with local inference and periodic off-device updates (Harrabi et al., 2024).

3.3.2 Theme 2: Operational Effectiveness, Reliability, and Risk

Business Performance Improvements

Evidence from included studies demonstrates that self-healing, drift-aware approaches yield measurable operational improvements across multiple dimensions. Table 3 summarizes key findings and reported impacts.

Table 3. Summary of Key Findings and Reported Impacts

Reference	Business Metrics Reported	Reported Impact	Reliability Metrics
Böse et al. (2017)	WAPE/MAPE, coverage	Industrial-scale accuracy enabling downstream decisions	Not explicitly reported
de Oliveira et al. (2021)	MAE, MAPE, RMSE	Improved lead-time accuracy vs. baselines	Not explicitly reported
Deng et al. (2023)	Cost savings, service level, WAPE, stockouts	"Millions" in savings; service improvements	Not explicitly reported
Gabellini et al. (2024)	AUC, accuracy, delay-risk calibration	Improved delay-risk prediction	Not explicitly reported
Gillespie et al. (2023)	Precision, recall, latency	Reduced false alerts; timely detection	Alert precision/recall
Guo et al. (2023)	Equity metrics, service cost	Better workload equity without service loss	Not explicitly reported
Harrabi et al. (2024)	Accuracy, F1, resource use	High accuracy on constrained MCU	Detection latency
Jiang et al. (2023)	Fairness, completion rate, latency	Improved fairness + efficiency	Not explicitly reported
Liu et al. (2022)	GMV, ETA, acceptance, throughput	Gains vs. heuristic baselines	Not explicitly reported
Mesa et al. (2023)	Sequence accuracy, routing KPIs	Better predicted sequences	Not explicitly reported
Phumchusri et al. (2024)	Profit/revenue uplift, promo ROI	Improved profit under multi-period plan	Not explicitly reported
Qi et al. (2023)	Holding/stockout/total cost, service level	Cost reductions vs. static rules	Not explicitly reported
Qin et al. (2020)	Matching rate, ETA, GMV	Business impact reported	Not explicitly reported
Rokoss et al. (2024)	MAE, MAPE, R^2	Improved promise-date accuracy	Not explicitly reported
Vijay et al.	ETA error, rescheduling	Improved	Not explicitly

Reference	Business Reported	Metrics	Reported Impact	Reliability Metrics
(2024)	delay		rescheduling decisions	reported

Integrated stacks combining forecasting, inventory, and pricing modules demonstrated substantial cost savings and service level improvements when co-optimized and continuously updated (Deng et al., 2023). End-to-end deep inventory policies reduced holding and stockout costs compared to static rules, demonstrating adaptation to demand and lead-time fluctuations (Qi et al., 2023).

Reinforcement learning dispatch systems improved matching efficiency and timeliness under non-stationary demand-supply conditions, yielding platform-level reliability advantages (Liu et al., 2022; Qin et al., 2020). Fairness-aware controllers balanced workloads without compromising throughput, maintaining equity as spatial demand patterns shifted (Jiang et al., 2023).

Retail price promotion optimization increased profitability despite elasticity drift, with multi-period plans demonstrating resilience to concept drift (Phumchusri et al., 2024). Delivery-time and ETA models improved promise accuracy and responsiveness, enabling better resource allocation and customer communication (Rokoss et al., 2024; Vijay et al., 2024).

Cold-chain anomaly detection systems reduced false alerts while maintaining sensitivity, enabling timely response to temperature excursions (Gillespie et al., 2023). Edge-based inference maintained high detection accuracy during connectivity gaps, ensuring continuous monitoring for time-sensitive remediation (Harrabi et al., 2024). Delay-risk scoring provided advance warnings to logistics planners, enabling proactive mitigation (Gabellini et al., 2024).

Reliability Metrics and Reporting Gaps

Despite evidence of improved operational outcomes, reporting of reliability-specific metrics was inconsistent across studies. Time-to-detect (TTD) and time-to-recover (TTR)—fundamental reliability indicators for self-healing systems—were not explicitly reported in any included study. Proxy indicators such as alert precision/recall (Gillespie et al., 2023) and detection latency (Harrabi et al., 2024) were reported in a minority of cases. This gap limits cross-study comparability and assessment of self-healing effectiveness.

Failure Modes and Mitigations

Documented failure modes and mitigations were inconsistently reported. Several studies acknowledged limitations including:

Detection latency: Delays between drift onset and detection can permit performance degradation before remediation (Gillespie et al., 2023; Harrabi et al., 2024).

Remediation appropriateness: Automated retraining may not always yield improved performance if drift is transient or if new data is unrepresentative (Deng et al., 2023).

Catastrophic forgetting: Continual learning systems may lose performance on historical patterns when adapting to new distributions, though this was primarily discussed in reinforcement learning contexts (Liu et al., 2022; Qin et al., 2020).

Computational constraints: Edge deployments face trade-offs between model complexity, detection accuracy, and resource utilization (Harrabi et al., 2024).

Fairness degradation: Adaptation without fairness monitoring can exacerbate disparities as distributions shift (Jiang et al., 2023).

Mitigation strategies included conservative rollout with canary testing (Deng et al., 2023; Qin et al., 2020), human-in-the-loop escalation for critical decisions (Gillespie et al., 2023), fairness constraints in optimization (Jiang et al., 2023), and hybrid edge-cloud architectures balancing local responsiveness with global coordination (Harrabi et al., 2024).

3.3.3 Theme 3: Self-Healing MLOps Design and Governance

MLOps Component Adoption

The maturity of MLOps implementation varied considerably across studies. Table 4 summarizes MLOps components reported in each study.

Table 4. MLOps Components Reported

Reference	Data Validation	Versioning	Feature Store	Model Registry	Observability	Rollout Controls	Governance Artifacts
Böse et al. (2017)	Implied	Not specified	Not specified	Not specified	Yes	Not specified	Not specified
de Oliveira et al. (2021)	Not specified	Not specified	Not specified	Not specified	Limited	Not specified	Not specified
Deng et al. (2023)	Yes	Yes	Implied	Yes	Yes	Yes (rollback)	Not specified
Gabellini et al. (2024)	Not specified	Not specified	Not specified	Not specified	Limited	Not specified	Not specified
Gillespie et al. (2023)	Yes (schema)	Not specified	Not specified	Not specified	Yes	Limited (alerts)	Not specified
Guo et al. (2023)	Not specified	Not specified	Not specified	Not specified	Yes	Not specified	Not specified
Harrabi et al. (2024)	Not specified	Not specified	Not specified	Not specified	Yes (local)	Limited (escalation)	Not specified
Jiang et al. (2023)	Not specified	Not specified	Not specified	Not specified	Yes	Yes (concurrency)	Not specified
Liu et al. (2022)	Not specified	Not specified	Not specified	Not specified	Yes	Limited	Not specified
Mesa et al. (2023)	Not specified	Not specified	Not specified	Not specified	Limited	Not specified	Not specified
Phumchusri et al. (2024)	Not specified	Not specified	Not specified	Not specified	Limited	Yes (price guardrails)	Not specified
Qi et al. (2023)	Not specified	Not specified	Not specified	Not specified	Yes	Yes (business gates)	Not specified
Qin et al. (2020)	Not specified	Not specified	Not specified	Not specified	Yes	Yes (staged rollout)	Not specified

Reference	Data Validation	Versioning	Feature Store	Model Registry	Observability	Rollout Controls	Governance Artifacts
Rokoss et al. (2024)	Not specified	Not specified	Not specified	Not specified	Limited	Not specified	Not specified
Vijay et al. (2024)	Not specified	Not specified	Not specified	Not specified	Yes	Not specified	Not specified

Observability through KPI dashboards and performance monitoring was the most commonly implemented MLOps component, present in most studies. Rollout controls, including staged rollouts, canary testing, and rollback capabilities, were documented in integrated enterprise stacks (Deng et al., 2023) and reinforcement learning dispatch systems (Qin et al., 2020). Data validation was explicitly addressed in platform-scale forecasting systems (Böse et al., 2017) and IoT pipelines (Gillespie et al., 2023). However, versioning, feature stores, model registries, and governance artifacts such as model cards or datasheets were rarely documented.

Reference Architecture Elements

Synthesis of evidence across studies suggests key elements for self-healing MLOps architectures:

Continuous validation pipelines: Automated checks for data schema, statistical distributions, and data quality at ingestion points (Böse et al., 2017; Gillespie et al., 2023).

Versioned storage: Append-only storage with temporal integrity for features, labels, and model artifacts, enabling reproducible rollbacks and post-mortem analysis (Deng et al., 2023).

Multi-metric observability: Dashboards tracking business KPIs, model performance metrics, data drift indicators, and system health across the deployment lifecycle (Deng et al., 2023; Jiang et al., 2023).

Tiered trigger thresholds: Differentiated thresholds for alerting, shadow deployment evaluation, canary release, and automated rollback based on severity and confidence (Qi et al., 2023; Qin et al., 2020).

Controlled rollout gates: Mechanisms for shadow mode evaluation (parallel execution without affecting decisions), canary releases (limited traffic exposure), and automated rollback with fallback to previous versions (Deng et al., 2023; Qin et al., 2020).

Edge-cloud coordination: Lightweight detectors on edge tiers maintaining monitoring during network intermittency, with synchronization to cloud for model updates and telemetry aggregation (Harrabi et al., 2024; Gillespie et al., 2023).

Audit trails and governance: Comprehensive logging of data lineage, model versions, deployment decisions, and remediation actions supporting internal and regulatory accountability (implied but rarely documented).

4. DISCUSSION

4.1 Synthesis of Key Findings

This systematic review consolidates evidence on drift-aware, self-healing machine learning systems for supply chain operations across fifteen applied studies spanning diverse sectors and functions. The findings demonstrate that self-healing approaches—integrating continuous monitoring, drift detection, controlled adaptation, and gated rollout—can materially improve operational outcomes under non-stationary conditions.

Across functions, self-healing patterns produced reliable operational improvements. Integrated stacks co-optimizing forecasting, inventory, and pricing modules achieved substantial cost savings and service level improvements when continuously updated (Deng et al., 2023). End-to-end deep inventory policies reduced holding and stockout costs compared to static rules, demonstrating adaptation to demand and lead-time fluctuations (Qi et al., 2023). Reinforcement learning dispatch systems improved matching efficiency under volatile demand-supply conditions (Liu et al., 2022; Qin et al., 2020), while fairness-aware controllers balanced workloads without compromising throughput (Jiang et al., 2023).

Cold-chain and edge deployments demonstrated the feasibility of self-monitoring under resource and connectivity constraints. Anomaly detectors reduced false alerts while maintaining sensitivity (Gillespie et al., 2023), with edge-based inference preserving monitoring during connectivity gaps (Harrabi et al., 2024). These capabilities are essential for time-sensitive remediation in distributed supply chain environments.

Reliability improvements were evident in proxy metrics including alert precision/recall, detection latency, and service stability, though explicit TTD/TTR reporting was absent across all studies. This gap limits quantitative assessment of self-healing effectiveness and cross-study comparability.

4.2 Comparison with Existing Literature

Our findings align with and extend prior research on adaptive ML systems. The observed preference for periodic retraining and multi-metric monitoring over explicit statistical drift detectors reflects pragmatic engineering choices documented in production contexts (Sculley et al., 2015; Webb et al., 2016). While the academic literature thoroughly characterizes concept drift detection algorithms, field evidence suggests that composite monitoring with business-aware thresholds may be more practical and robust in operational settings (Rabanser, Günnemann, & Lipton, 2019).

The integration of forecasting, inventory, and pricing modules with continuous adaptation aligns with architectural guidance for multi-horizon control and interpretability (Lim et al., 2021). However, our finding of inconsistent reliability metric reporting echoes concerns raised about evaluation rigor in time-series forecasting research (Benidis et al., 2022). Competition-grade benchmarking practices (Makridakis, Spiliotis, & Assimakopoulos, 2022) and shift-aware benchmarks (Dragoi et al., 2022) offer templates for improving comparability.

The sparse documentation of governance artifacts such as model cards (Mitchell et al., 2019) and datasheets (Gebru et al., 2021) represents a significant gap. These practices help communicate assumptions, risks, and intended use boundaries in changing environments, and their absence limits

accountability and reproducibility. Pipeline practices showing highest alignment with TFX principles of validation, versioning, and orchestration (Baylor et al., 2017) corresponded with the most successful deployments, supporting the value of disciplined MLOps engineering.

Feature store architectures, which improve model retraining and reduce rollback frequency through point-in-time correct feature retrieval, were implied in some enterprise stacks (Deng et al., 2023) but rarely explicitly documented. Research on feature store optimization (Liu et al., 2023) and production implementations (de la Rúa Martínez et al., 2024) suggests these components are critical for maintaining temporal accuracy and preventing leakage.

4.3 Implications for Practice

For engineering and operations teams, operationalizing self-healing ML requires embedding software engineering discipline, auditability, and data governance within MLOps frameworks. Role definitions, test suites, and deployment gates must encompass data validation, versioning, CI/CD, observability, and rollback capabilities (Kreuzberger, Kühl, & Hirschl, 2023). Engineering playbooks should codify requirements capture, design reviews, experiment tracking, and post-mortem processes, ensuring that failure signals trigger active remediation rather than improvisation (Amershi et al., 2019).

For organizational leaders, successful implementation requires anticipating deployment bottlenecks and budgeting for process change, tooling investment, and skill development beyond modeling alone (Paleyes, Urma, & Lawrence, 2022). Cross-functional collaboration between data science, engineering, operations, and compliance teams is essential for building trusted, governable systems.

For governance and compliance, periodic algorithmic audits should assess resilience to drift, documentation completeness, and rollback readiness (Raji et al., 2020). Supplier-style declarations, such as service fact sheets, can standardize information reported about models, data, and operating conditions (Arnold et al., 2019). Fairness must be operationalized through practitioner guidance and checklists that surface trade-offs during monitoring, adaptation, and release decisions (Holstein et al., 2019; Madaio et al., 2020).

For data management, investment in data stewardship, lineage tracking, and contract enforcement is essential to prevent cascading failures (Sambasivan et al., 2021). Data validation and quality monitoring should be treated as first-class components of ML pipelines, not afterthoughts.

4.4 Limitations of This Review

Despite systematic methodology, several limitations should be acknowledged:

Heterogeneity in tasks, datasets, and reporting prevented meta-analysis and necessitated narrative synthesis. Variability in baseline strength and evaluation rigor across studies limits direct comparison.

Publication and selection biases may affect findings, as negative results and implementation failures are less likely to be published. English-only language requirements may exclude relevant research from non-English sources.

Inconsistent reporting of reliability indicators (TTD/TTR), drift detectors, and governance controls limits cross-study comparability and assessment of self-healing effectiveness.

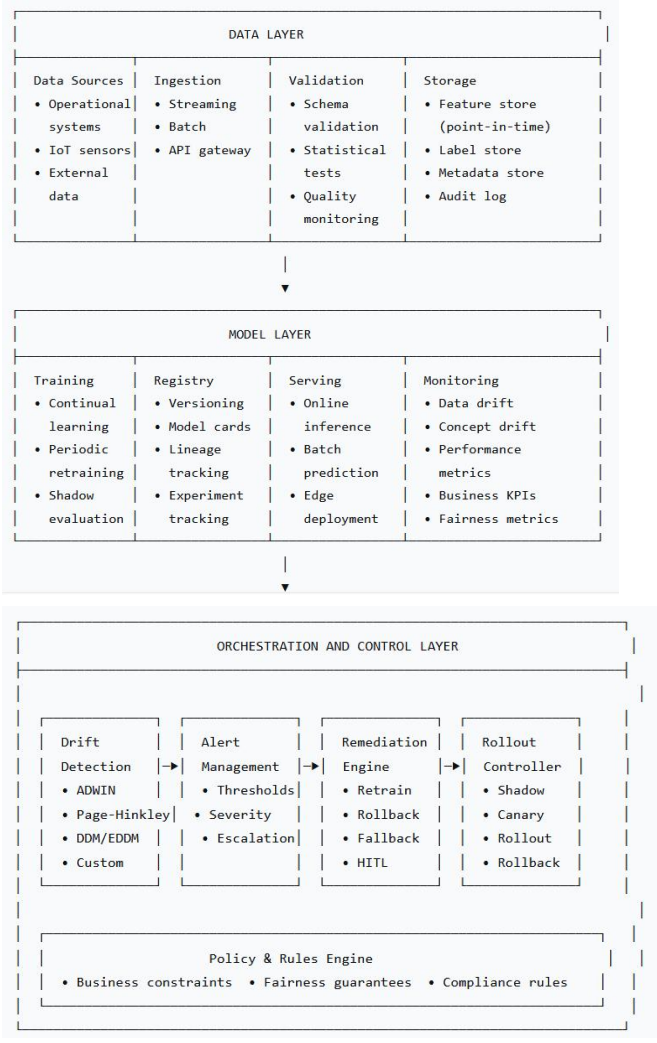
Proprietary platforms restricted reproducibility and detail on rollout gates and internal validation practices in several industrial case studies.

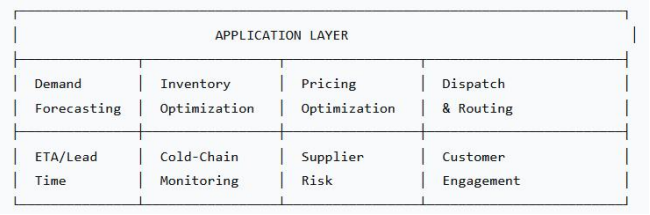
Excluded domains including vision-only inspection and cybersecurity studies may omit relevant edge cases where pipeline integration or data-integrity failures indirectly affect operations.

5. REFERENCE ARCHITECTURE FOR SELF-HEALING MLOPS

Based on synthesis of evidence from included studies and alignment with established MLOps principles, we propose a reference architecture for self-healing, drift-aware ML systems in supply chain operations. Figure 2 presents a high-level overview of the architecture.

Figure 2. Self-Healing MLOps Reference Architecture





5.1 Architecture Components

5.1.1 Data Layer

The data layer encompasses all components for acquiring, validating, and storing operational data with temporal integrity:

Data sources: Integration with operational systems (ERP, WMS, TMS), IoT sensor networks, and external data providers (weather, economic indicators, traffic).

Ingestion pipelines: Support for both streaming (real-time sensor data, transaction streams) and batch (daily sales, weekly inventory) modalities with appropriate latency guarantees.

Validation services: Automated schema validation, statistical distribution tests, and data quality monitoring at ingestion points, with alerting on anomalies (Schelter et al., 2018).

Feature store: Point-in-time correct feature retrieval with versioning, enabling reproducible training and inference while preventing leakage (de la Rúa Martínez et al., 2024; Liu et al., 2023).

Label store: Versioned storage of ground truth labels with clear lineage to upstream data sources.

Metadata store and audit log: Comprehensive tracking of data lineage, transformations, and access patterns supporting reproducibility and governance.

5.1.2 Model Layer

The model layer manages model development, versioning, serving, and monitoring:

Training pipelines: Support for continual learning, periodic retraining, and shadow evaluation of candidate models against current deployments.

Model registry: Versioned storage of model artifacts with associated metadata including training data lineage, hyperparameters, evaluation metrics, and model cards documenting intended use, limitations, and risks (Mitchell et al., 2019).

Serving infrastructure: Low-latency online inference for real-time decisions (dispatch, anomaly detection), batch prediction for planning applications (demand forecasting, inventory optimization), and edge deployment for resource-constrained environments.

Monitoring services: Continuous tracking of data drift (feature distributions), concept drift (feature-target relationships), model performance metrics, business KPIs, and fairness indicators (Jiang et al., 2023).

5.1.3 Orchestration and Control Layer

The orchestration and control layer implements self-healing capabilities through coordinated detection, alerting, remediation, and rollout:

Drift detection: Ensemble of statistical detectors (ADWIN, Page-Hinkley, DDM) calibrated for each data modality and business context, with configurable sensitivity (Gomes et al., 2017).

Alert management: Tiered alerting based on drift severity, business impact, and confidence, with defined escalation paths for critical conditions.

Remediation engine: Automated responses including model retraining (triggered by sustained drift), rollback to previous versions (triggered by performance degradation), fallback to simpler rules or heuristics (during uncertainty), and human-in-the-loop escalation for high-stakes decisions.

Rollout controller: Staged deployment mechanisms including shadow mode (parallel evaluation without decision impact), canary releases (limited traffic exposure with close monitoring), gradual rollout (incremental traffic increase), and automated rollback with predefined criteria.

Policy and rules engine: Enforcement of business constraints (service level agreements, inventory targets), fairness guarantees (Jiang et al., 2023), and compliance requirements during adaptation and rollout.

5.1.4 Application Layer

The application layer encompasses the supply chain functions supported by self-healing ML systems, as documented in the reviewed literature: demand forecasting, inventory optimization, pricing, dispatch and routing, ETA prediction, cold-chain monitoring, supplier risk assessment, and customer engagement.

5.2 Operationalization Guidelines

Successful implementation of the reference architecture requires attention to:

Service level objectives (SLOs) defining acceptable bounds for prediction accuracy, decision latency, and reliability metrics (TTD, TTR) across operational contexts.

Trigger threshold calibration balancing detection sensitivity against false alarms, informed by business impact analysis and historical drift patterns.

Remediation playbooks documenting automated responses for each drift type and severity level, with clear escalation criteria for human intervention.

Governance reviews at regular intervals assessing drift patterns, remediation effectiveness, and documentation completeness (Raji et al., 2020).

Post-mortem processes analyzing incidents where self-healing mechanisms failed or produced suboptimal outcomes, feeding back into system improvement.

6. RECOMMENDATIONS FOR FUTURE RESEARCH

Based on gaps identified in the reviewed literature and challenges in operationalizing self-healing ML systems, we propose the following priorities for future research:

6.1 Standardization of Reliability Reporting

The absence of consistent reliability metrics across studies limits cross-study comparability and assessment of self-healing effectiveness. Future research should adopt standardized reporting of:

- **Time-to-detect (TTD)**: Latency between drift onset and detection
- **Time-to-recover (TTR)**: Latency between detection and successful remediation
- **Alert precision and recall**: Accuracy of drift detection mechanisms
- **Remediation success rate**: Proportion of drift events successfully addressed
- **False positive rate**: Rate of unnecessary remediations

These metrics should be reported alongside traditional business KPIs to enable holistic assessment of self-healing system performance.

6.2 Development of Temporal Benchmarks

Public benchmarks coupling forecasting and policy tasks with realistic drift injections and label latency are needed to enable controlled evaluation of self-healing approaches. Benchmarks should:

- Include multiple drift types (covariate, label, concept) with varying severity and duration
- Incorporate realistic label availability delays characteristic of operational settings
- Provide clear evaluation protocols separating detection, adaptation, and decision components
- Enable comparison of composite monitoring strategies against single detectors
- Support ablation studies isolating the contribution of each self-healing component

Competition-grade forecasting benchmarks (Makridakis et al., 2022) and shift-aware anomaly detection benchmarks (Dragoi et al., 2022) offer templates for such efforts.

6.3 Comparative Evaluation of Detection and Remediation Strategies

Research should systematically compare:

- **Composite monitors vs. single detectors**: Do ensembles of drift detection methods outperform individual approaches in operational settings?
- **Statistical detectors vs. performance monitoring**: Under what conditions does explicit drift detection add value beyond monitoring business KPIs?
- **Online learning vs. periodic retraining**: What are the cost-latency-performance trade-offs between continuous and batched adaptation?
- **Rollback vs. fallback vs. retain**: What criteria should determine whether to roll back to previous models, fall back to simpler heuristics, or continue with degraded performance?
- **Cost-aware triggering**: How should trigger thresholds be calibrated considering the costs of false alarms (unnecessary remediation) vs. missed detections (performance degradation)?

6.4 Stress Testing of Rollout Policies

The safety and effectiveness of automated rollout mechanisms under realistic conditions require systematic evaluation. Research should examine:

- Canary release policies: What traffic fractions and monitoring durations are appropriate given detection latency and business impact?
- Rollback criteria: What performance degradation thresholds should trigger automated rollback?
- Shadow deployment evaluation: How should candidate models be evaluated against production traffic without affecting decisions?
- Coordinated rollouts across interdependent models: How should updates be coordinated when multiple models (e.g., forecasting, inventory, pricing) interact?

6.5 Continual Learning with Constraints

As self-healing systems evolve continuously, maintaining alignment with business constraints and fairness guarantees becomes challenging. Research should address:

- **Constraint-preserving adaptation**: How can continual learning respect inventory service level agreements, pricing guardrails, and fairness guarantees?
- **Fairness under drift**: How do fairness properties of models evolve as distributions shift, and how can adaptation maintain equity?
- **Policy compliance**: How can automated remediation ensure continued compliance with regulatory requirements and data use agreements?
- **Stability-plasticity trade-offs**: What mechanisms best balance adaptation to new patterns against retention of historical knowledge?

6.6 Edge-Cloud Architectures

The proliferation of edge deployments in supply chain contexts (cold-chain monitoring, warehouse automation, delivery vehicles) creates opportunities and challenges for self-healing architectures. Research should examine:

- **Split inference**: How should detection and remediation responsibilities be partitioned between edge devices and cloud platforms?
- **Synchronization under intermittency**: How can models be updated and telemetry aggregated when connectivity is unreliable?
- **Resource-constrained learning**: What lightweight adaptation mechanisms are feasible on microcontrollers and edge gateways?
- **Federated drift detection**: How can drift patterns be detected across distributed edge nodes while preserving data privacy?

6.7 Privacy-Preserving Observability

As supply chains involve multiple partners with competing interests and data privacy obligations, observability

mechanisms must respect confidentiality. Research should explore:

Federated monitoring: How can drift be detected across organizational boundaries without sharing raw data?

Differential privacy for telemetry: What privacy guarantees can be provided for aggregated monitoring data?

Auditable lineage without data sharing: How can compliance be demonstrated when data cannot be shared across partners?

6.8 Energy and Carbon Accounting

The environmental impact of continuous learning and automated remediation deserves attention. Research should examine:

Energy costs of adaptation: What is the carbon footprint of frequent retraining, and how does it compare to the operational benefits?

Efficiency-aware triggering: Should energy costs be factored into decisions about when and how to adapt?

Green MLOps: What architectural patterns minimize the environmental impact of self-healing systems?

7. CONCLUSION

This systematic review has examined the landscape of self-healing, drift-aware machine learning systems for supply chain operations, synthesizing evidence from fifteen applied studies across diverse sectors and functions. The findings demonstrate that self-healing approaches—integrating continuous monitoring, drift detection, controlled adaptation, and gated rollout—can materially improve operational outcomes under non-stationary conditions.

Across demand forecasting, inventory optimization, pricing, dispatch, ETA prediction, and cold-chain monitoring, evidence indicates that adaptive systems reduce errors and costs while maintaining service stability. Fairness-aware dispatch and edge-based detectors enhance robustness under connectivity and workload constraints. Integrated stacks co-optimizing multiple functions achieve substantial business impact when continuously updated.

However, the synthesis also reveals significant gaps in reliability metric reporting (TTD, TTR), explicit drift detector specification, governance artifact documentation, and reproducibility. These gaps limit cross-study comparability and assessment of self-healing effectiveness, while constraining the transferability of research findings to practice.

In response, we have proposed a reference architecture codifying data validation, versioned storage, observability, trigger thresholds, controlled rollout paths, and governance mechanisms essential for self-healing MLOps. The architecture provides a blueprint for organizations seeking to operationalize continual learning research within audited, cost-aware operations that remain reliable under non-stationarity.

For future research, we recommend prioritizing: (1) standardization of reliability reporting metrics, (2) development of temporal benchmarks with realistic drift injections, (3) comparative evaluation of detection and remediation strategies, (4) stress testing of rollout policies, (5) continual learning with fairness and policy constraints, (6) edge-cloud architectural patterns, (7) privacy-preserving observability, and (8) energy and carbon accounting.

As supply chains continue to digitize and face increasing volatility, the ability to maintain ML system performance under changing conditions will become ever more critical. Self-healing, drift-aware systems offer a path toward resilient, reliable, and responsible AI in supply chain operations—but realizing this potential requires continued research, disciplined engineering, and commitment to transparency and accountability. This review provides both an evidence base and a roadmap for that journey.

REFERENCES

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. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)* (pp. 291-300). IEEE. https://doi.org/10.1109/ICSE-SEIP.2019.00042

Arnold, M., Bellamy, R. K., Hind, M., Houde, S., Mehta, S., Mojsilovic, A., Nair, R. G., Ramamurthy, K. N., et al. (2019). FactSheets: Increasing trust in AI services through supplier's declarations of conformity. *IBM Journal of Research and Development*, 63(4/5), 6:1-6:13. https://doi.org/10.1147/JRD.2019.2942288

Baylor, D., Breck, E., Cheng, H. T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., et al. (2017). TFX: A TensorFlow-based production-scale machine learning platform. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1387-1395). ACM. https://doi.org/10.1145/3097983.3098021

Benidis, K., Rangapuram, S. S., Flunkert, V., Wang, Y., Maddix, D., Turkmen, C., Gasthaus, J., Bohlke-Schneider, M., et al. (2022). Deep learning for time series forecasting: Tutorial and literature survey. *ACM Computing Surveys*, 55(6), 1-36. https://doi.org/10.1145/3533382

Böse, J. H., Flunkert, V., Gasthaus, J., Januschowski, T., Lange, D., Salinas, D., Schelter, S., Seeger, M., & Wang, Y. (2017). Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12), 1694-1705. https://doi.org/10.14778/3137765.3137775

Chen, Y., Qian, Y., Yao, Y., Wu, Z., Li, R., Zhou, Y., ... Xu, Y. (2019). Can sophisticated dispatching strategy acquired by reinforcement learning? A case study in dynamic courier dispatching system. *arXiv preprint arXiv:1903.02716*.

de la Rúa Martínez, J., Buso, F., Kouzoupis, A., Ormenisan, A. A., Niazi, S., Bzhalava, D., Mak, K., Jouffrey, V., Ronström, M., Cunningham, R., Zangis, R., Mukhedkar, D., Khazanchi, A., Vlassov, V., & Dowling, J. (2024). The Hopsworks feature store for machine learning. In *Companion of the 2024 International Conference on Management of Data* (pp. 135-147). ACM. https://doi.org/10.1145/3626246.3653389

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., & Tuytelaars, T. (2022). A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), 3366-3385. https://doi.org/10.1109/TPAMI.2021.3057446

de Oliveira, M. B., Zucchi, G., Lippi, M., Cordeiro, D. F., da Silva, N. R., & Iori, M. (2021). Lead time forecasting with machine learning techniques for a pharmaceutical supply chain. In *Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS)* (Vol. 1, pp. 634-641). https://doi.org/10.5220/0010434406340641

Deng, Y., Zhang, X., Wang, T., Wang, L., Zhang, Y., Wang, X., Zhao, S., Qi, Y., Yang, G., & Peng, X. (2023). Alibaba realizes millions in cost savings through integrated demand forecasting, inventory management, price optimization, and product recommendations. *INFORMS Journal on Applied Analytics*, 53(1), 32-46. https://doi.org/10.1287/inte.2022.1145

Dragoi, M., Burceanu, E., Haller, E., Manolache, A., & Brad, F. (2022). AnoShift: A distribution shift benchmark for unsupervised anomaly detection. *Advances in Neural Information Processing Systems*, 35, 32854-32867.

Gabellini, M., Civolani, L., Calabrese, F., & Bortolini, M. (2024). A deep learning approach to predict supply chain delivery delay risk based on

macroeconomic indicators: A case study in the automotive sector. *Applied Sciences*, 14(11), 4688. https://doi.org/10.3390/app14114688

Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J. W., Wallach, H., Daumé III, H., & Crawford, K. (2021). Datasheets for datasets. *Communications of the ACM*, 64(12), 86-92. https://doi.org/10.1145/3458723

Gillespie, J., da Costa, T. P., Cama-Moncunill, X., Cadden, T., Condell, J., Cowderoy, T., Ramsey, E., Murphy, F., Kull, M., Gallagher, R., & Ramanathan, R. (2023). Real-time anomaly detection in cold chain transportation using IoT technology. *Sustainability*, 15(3), 2255. https://doi.org/10.3390/su15032255

Gomes, H. M., Bifet, A., Read, J., Barddal, J. P., Enembreck, F., Pfahringer, B., Holmes, G., & Abdesselam, T. (2017). Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9), 1469-1495. https://doi.org/10.1007/s10994-017-5642-8

Guo, B., Wang, S., Wang, H., Liu, Y., Kong, F., Zhang, D., & He, T. (2023). Towards equitable assignment: Data-driven delivery zone partition at last-mile logistics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM. https://doi.org/10.1145/3580305.3599915

Harrabi, M., Hamdi, A., Ouni, B., & Bel Hadj Tahar, J. (2024). Real-time temperature anomaly detection in vaccine refrigeration systems using deep learning on a resource-constrained microcontroller. *Frontiers in Artificial Intelligence*. https://doi.org/10.3389/frai.2024.1429602

Holstein, K., Wortman Vaughan, J., Daumé III, H., Dudik, M., & Wallach, H. (2019). Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1-16). ACM. https://doi.org/10.1145/3290605.3300830

Jangam, S. K. (2022). Role of AI and ML in enhancing self-healing capabilities, including predictive analysis and automated recovery. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, 3(4), 47-56.

Jiang, L., Wang, S., Guo, B., Wang, H., Zhang, D., & Wang, G. (2023). FairCod: A fairness-aware concurrent dispatch system for large-scale instant delivery services. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 4229-4238). ACM.

Konovalenko, I., Ludwig, A., & Leopold, H. (2021). Real-time temperature prediction in a cold supply chain based on Newton's law of cooling. *Decision Support Systems*. https://doi.org/10.1016/j.dss.2020.113451

Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11, 31866-31879. https://doi.org/10.1109/ACCESS.2023.3262138

Lim, B., Arik, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4), 1748-1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Lin, H., Lin, J., & Wang, F. (2022). An innovative machine learning model for supply chain management. *Journal of Innovation & Knowledge*, 7(4), 100276.

Liu, Q., Boniol, P., Palpanas, T., & Paparrizos, J. (2024). Time-series anomaly detection: Overview and new trends. *Proceedings of the VLDB Endowment*, 17(12), 4229-4232. https://doi.org/10.14778/3685800.3685842

Liu, R., Park, K., Psallidas, F., Zhu, X., Mo, J., Sen, R., Interlandi, M., Karanasos, K., Tian, Y., & Camacho-Rodríguez, J. (2023). Optimizing data pipelines for machine learning in feature stores. *Proceedings of the VLDB Endowment*, 16(13), 4230-4239. https://doi.org/10.14778/3625054.3625060

Liu, Y., Wu, F., Lyu, C., Li, S., Ye, J., & Qu, X. (2022). Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. *Transportation Research Part E: Logistics and Transportation Review*, 161, 102694. https://doi.org/10.1016/j.tre.2022.102694

Madaio, M. A., Stark, L., Wortman Vaughan, J., & Wallach, H. (2020). Co-designing checklists to understand organizational challenges and opportunities around fairness in AI. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM. https://doi.org/10.1145/3313831.3376445

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, 38(4), 1346-1364. https://doi.org/10.1016/j.ijforecast.2021.11.013

Mesa, J. P., Montoya, A., Ramos-Pollán, R., & Toro, M. (2023). A two-stage data-driven metaheuristic to predict last-mile delivery route sequences. *Engineering Applications of Artificial Intelligence*. https://doi.org/10.1016/j.engappai.2023.106653

Mitchell, M., Wu, S., Zaldivar, A., Barnes, P., Vasserman, L., Hutchinson, B., Spitzer, E., Raji, I. D., & Gebru, T. (2019). Model cards for model reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency* (pp. 220-229). ACM. https://doi.org/10.1145/3287560.3287596

Montiel, J., Halford, M., Mastelini, S. M., Bolmier, G., Sourty, R., Vaysse, R., Zouitine, A., Gomes, H. M., Read, J., Abdessalem, T., & Bifet, A. (2021). River: Machine learning for streaming data in Python. *Journal of Machine Learning Research*, 22(110), 1-8.

Montiel, J., Read, J., Bifet, A., & Abdessalem, T. (2018). Scikit-Multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72), 1-5.

Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., ... & Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, 372, n71. https://doi.org/10.1136/bmj.n71

Paleyes, A., Urma, R. G., & Lawrence, N. D. (2022). Challenges in deploying machine learning: A survey of case studies. *ACM Computing Surveys*, 55(6), 1-29. https://doi.org/10.1145/3533378

Phumchusri, N., Chewcharat, T., & Kanokpongsakorn, S. (2024). Price promotion optimization model for multiperiod planning: A case study of beauty category products sold in a convenience store chain. *Journal of Revenue and Pricing Management*, 23(2), 164-178. https://doi.org/10.1057/s41272-023-00438-6

Polyzotis, N., Zinkevich, M., Roy, S., Breck, E., & Whang, S. (2019). Data validation for machine learning. *Proceedings of Machine Learning and Systems*, 1, 334-347.

Qi, M., Shi, Y., Qi, Y., Ma, C., Yuan, R., Wu, D., & Shen, Z. J. (2023). A practical end-to-end inventory management model with deep learning. *Management Science*, 69(2), 759-773. https://doi.org/10.1287/mnsc.2022.4564

Qin, Z., Tang, X., Jiao, Y., Zhang, F., Xu, Z., Zhu, H., & Ye, J. (2020). Ride-hailing order dispatching at DiDi via reinforcement learning. *INFORMS Journal on Applied Analytics*, 50(5), 272-286. https://doi.org/10.1287/inte.2020.1047

Rabanser, S., Günnemann, S., & Lipton, Z. (2019). Failing loudly: An empirical study of methods for detecting dataset shift. *Advances in Neural Information Processing Systems*, 32.

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., Smith-Loud, J., Theron, D., & Barnes, P. (2020). Closing the AI accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency* (pp. 33-44). ACM. https://doi.org/10.1145/3351095.3372873

Rokoss, A., Syberg, M., Tomidei, L., Hulsing, C., Deuse, J., & Schmidt, M. (2024). Case study on delivery time determination using a machine learning approach in small batch production companies. *Journal of Intelligent Manufacturing*, 35(8), 3937-3958. https://doi.org/10.1007/s10845-023-02290-2

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (pp. 1-15). ACM. https://doi.org/10.1145/3411764.3445518

Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11(12), 1781-1794. https://doi.org/10.14778/3229863.3229867

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.

Vijay, A., Thompson, R. G., Nassir, N., & Zhang, J. (2024). Machine learning based ETA prediction for dock rescheduling in hyperconnected city logistics based PI-hub facilities. *Transportation Research Procedia*, 79, 202-209. https://doi.org/10.1016/j.trpro.2024.03.028

Wang, L., Zhang, X., Su, H., & Zhu, J. (2024). A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8), 5362-5383. https://doi.org/10.1109/TPAMI.2024.3367329

Webb, G. I., Hyde, R., Cao, H., Nguyen, H. L., & Petitjean, F. (2016). Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4), 964-994. https://doi.org/10.1007/s10618-015-0448-4

Yuan, G., Wu, S., & Wang, B. (2023). Supply chain management model based on machine learning. *Neural Computing and Applications*, 35(6), 4319-4335. https://doi.org/10.1007/s00521-022-07954-9