Correctability-Aware Drift Mitigation via Feasible Setpoint Adaptation: A steel industry use case

Paolo Catti, Vaggelis Lakkas-Pyknis, Kosmas Alexopoulos, Nikolaos Nikolakis

Abstract— **Zero Defect Manufacturing (ZDM) requires detecting quality drifts early and adapting process settings before defects occur. This paper proposes a closed-loop setpoint adaptation method that couples online multivariate anomaly detection with constrained derivative-free optimization. Streaming process signals yield a real-time drift-risk index, while actionability is enforced via feasibility filtering and a correctability-aware acceptance score that estimates achievable drift-risk reduction under admissible setpoint changes and actuator limits. At each control cycle, Pattern Search explores candidate adjustments around the current operating point, while a physics-informed feasibility layer filters candidates by operating bounds and rate-of-change constraints. The selected feasible update is then applied to the asset controller, and the process response is evaluated in the next window, iterating online. The proposed method converts drift detection into constraint-feasible setpoint adaptations without requiring labeled defect data or an explicit defect model. A prototype was evaluated on a steel hot-forming line acting on a rolling setpoint, achieving up to 5.5% reduction in drift-risk index and a 40% lower defective-window count relative to a baseline period.**

I. Introduction

In modern manufacturing, maintaining quality and low defects increasingly requires early detection and timely mitigation of process deviations [1]. In this direction, Zero Defect Manufacturing (ZDM) promotes a proactive quality paradigm, where streaming process data are exploited to identify early signs of degradation and adapt process settings before defects materialize [2].

Achieving ZDM in real production remains challenging because quality degradation often appears as a gradual drift rather than an abrupt fault [2]. Manufacturing assets operate under varying loads, material conditions, and environmental influences, and early drift detection requires joint interpretation of multivariate sensor signals to capture subtle deviations [3]. Moreover, once drift is detected, any intervention on control parameters must respect tight operational boundaries and physical constraints, including actuator limitations and rate-of-change restrictions [3].

Corrective actions in ZDM may take several forms, including material re-routing, scheduling changes and operator advisories. This work focuses on supervisory setpoint adaptation on a single controllable process parameter, which is applicable when the drift can be compensated via a continuous-valued actuator, and the actuator's admissible range and rate-of-change limits are known.

Supervised defect prediction can support drift identification, but it depends on labeled defect data, which are scarce in ZDM-oriented environments [4]. Unsupervised anomaly detection, in contrast, can detect drift-related deviations without labeled defects. However, such approaches often stop at generating alarms and do not prescribe corrective actions [5]. Conversely, process optimization can recommend parameter settings, but it is typically applied offline and targets static objectives, rather than responding to streaming drift signals while enforcing feasibility in real time.

A severity-only drift indicator exposes the controller to two failure modes: intervening when no feasible setpoint change can meaningfully reduce drift-risk, and over-intervening when surrogate gain is noise-dominated rather than systematic [5]. Quantifying correctability, decouples actioning from drift severity. Such a correctability-aware measure can support real-time decision-making by indicating when intervention is both feasible and beneficial.

This paper proposes a closed-loop pipeline for quality drift mitigation. Streaming process variables feed an online multivariate anomaly detector that outputs a real-time drift-risk index, while a constrained derivative-free Pattern Search explores candidate setpoint adjustments. A physics-informed feasibility layer enforces operating bounds and actuator rate limits, and the best feasible update is issued as a PLC-ready command. The loop evaluates the process response each sampling window and iterates online.

The contribution is a correctability-aware acceptance score distinguishing this work from prior detectors that only raise alarms. While individual components are established, the novelty lies in their composition into the FTT-MAE drift-risk model, the acceptance score, and the closed-loop feasibility-filtered supervisory layer.

II. Literature review

To reduce reliance on defect labels, unsupervised anomaly detection learns nominal behaviour and flags deviations as potential quality drifts [6], [7]. Autoencoder-based approaches are widely used due to their ability to learn compact representations [8] and quantify deviations via reconstruction errors [9]. Furthermore, autoencoder-based approaches and their variants are typically used for multistage manufacturing quality faults detection, given their ability in learning compact representations of nominal behavior and quantify deviations via reconstruction errors [10], [11]. More recent multivariate methods, including attention-based architectures and masked reconstruction

P. Catti, V. L. Pyknis, K. Alexopoulos, N. Nikolakis are with the Laboratory for Manufacturing Systems & Automation, Department of Mechanical Engineering and Aeronautics, University of Patras Rio, Patras, Greece (corresponding author: nikolakis@lms.mech.upatras.gr)

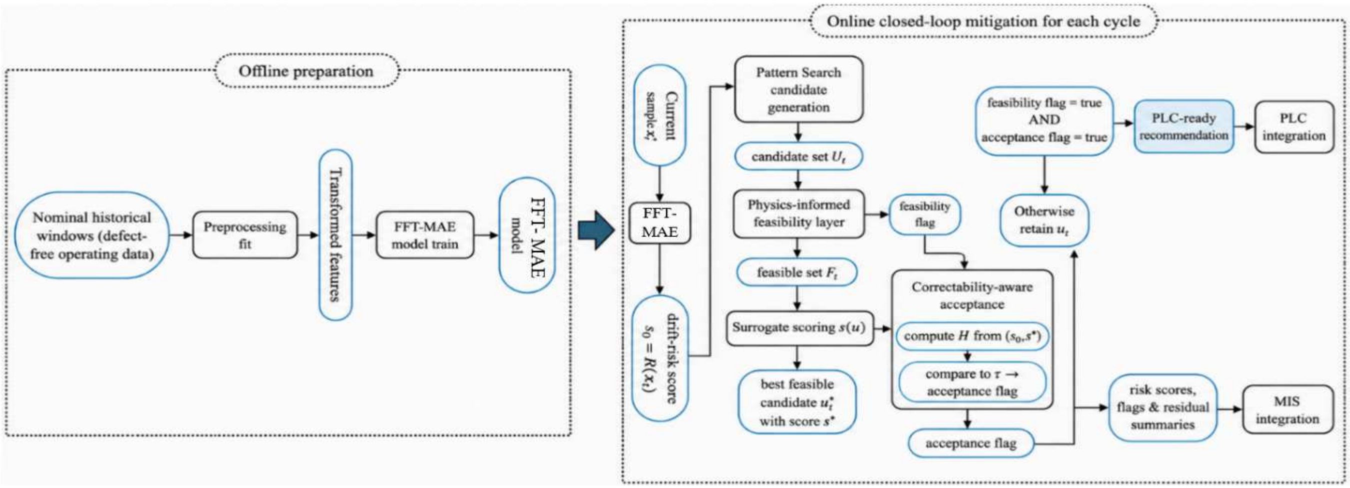

TABLE I. FIGURE 1. THE HIGH-LEVEL ARCHITECTURE OF THE PROPOSED APPROACH.

objectives, improve representation learning by capturing cross-sensor dependencies and reducing reliance on individual channels [12], [13]. Nevertheless, these approaches typically remain diagnostic, producing alarms or scores without prescribing feasible corrective actions for controller actuation [14].

Complementarily, process optimization can recommend parameter settings, but metaheuristics operate offline and do not explicitly encode real-time feasibility constraints [15]. Derivative-free local search, such as Pattern Search, is suitable for supervisory setpoint adaptation with black-box objectives derived from process signals and can be coupled with explicit feasibility handling to avoid non-actionable candidates [16]. Feasibility models, including physics-informed constraint layers, can enforce operating bounds and actuator limits to ensure safe recommendations [17], [18].

Beyond shop-floor actuation, ZDM frameworks emphasize that quality indicators should be consumable at the managerial layer, enabling traceability, governance and continuous improvement [19]. AI deployments therefore benefit from integrating drift-risk indices and recommended actions with management information systems (MIS) [20] so that local interventions are aligned with planning [21].

Overall, the literature shows that drift detection can flag degradation but is rarely embedded in loops that convert drift into feasible setpoint updates. This motivates a correctability-aware closed-loop strategy coupling multivariate drift-risk scoring with feasibility-filtered derivative-free optimization.

III. Approach

The proposed approach operates as a closed-loop setpoint adaptation schema that couples online drift-risk scoring with constraint-feasible local optimization. At control cycle *t*, the system acquires a vector of process measurements and decides whether to update a controllable setpoint.

Let t in {1, 2, ..., n} index uniform control cycles of duration Δt, x_t in R^d the window-aggregated feature vector at cycle t with d features, and u_t in $[u_{min}, u_{max}]$ the current value of a single controllable setpoint with known admissible range and rate-of-change limit Δu_{max}. The drift-risk scoring function R: $R^d \rightarrow R \geq 0$ is learned offline on nominal data D_{nom} and held fixed during deployment, and the current drift-risk score is $R_t = R(x_t)$. The control task at cycle t is to choose $u_{t+1} \in [u_{min}, u_{max}]$ such that minimises the expected drift-risk on the next window, subject to $u_{t+1}^* = \arg\min E[R(x_{t+1}) | x_t, u_{t+1}]$, the actuator-rate constraint $|u_{t+1} - u_t| \leq \Delta u_{max}$ and to operating constraints $g_i(u_{t+1}) \leq 0$ imposed by the physics-informed layer. The cycle duration Δt is shorter than the process settling time, which forms the response of R to local setpoint changes within $u_{t+1} \in [u_t - \Delta u_{max},\ u_t + \Delta u_{max}]$ dominated by short-horizon dynamics. During operation, defect labels are not available, with the real-time drift-risk estimate $R(x_t)$ being observed. Since unnecessary interventions introduce a non-zero cost due to actuator wear and controller perturbation, the policy is designed as a gated actuation policy rather than a continuously active control policy.

A bounded local search evaluates a finite set of candidate setpoints $u \in \mathcal{U}_t$ around the current value u_t, where U_t denotes the candidate set generated at cycle t. Each candidate is screened for feasibility first by a physics-informed constraint layer that enforces operating bounds and actuator rate-of-change limits. For each feasible candidate, a short-horizon surrogate score $s(u)$ ranks the expected drift-risk improvement. Then, the best feasible candidate is selected only if the acceptance score H exceeds a calibrated threshold τ. If accepted, the update is written to the PLC. Otherwise u_t is retained. The realized effect is then assessed on the subsequent window or after a fixed delay aligned with process dynamics, and the loop iterates (Figure 1).

Drift-risk is estimated using the Feature-Token [22] Transformer Masked Autoencoder (FTT-MAE), an unsupervised masked-reconstruction model trained on nominal operating data. Each input $x_{t,j}$ is mapped to a learned embedding $e_{t,j} = W_j x_{t,j} + b_j \in \mathbb{R}^h$ through a per-feature linear projection. The feature-token sequence is processed by an encoder under a Bernoulli mask, with mask variables $m_{t,j} \sim$ Bernoulli(p) [23]. The FTT-MAE outputs per-feature reconstruction parameters under a heavy-tailed likelihood: the reconstructed mean $\mu_t \in \mathbb{R}^d$, the reconstructed scale $\sigma_t \in \mathbb{R}^d$ with $\sigma_t > 0$ element-wise, obtained via Softplus($\cdot$), and the Student-t degrees of freedom $\nu \in \mathbb{R}$ with $\nu > 0$ [24].

The drift-risk score is computed as the aggregate negative log-likelihood across features, based on (1).

$$R_t = \sum_{j=1}^{d} -\log p(x_{t,j}; \mu_{t,j}, \sigma_{t,j}, \nu) \tag{1}$$

Where $p(\cdot; \mu_{t,j}, \sigma_{t,j}, \nu)$ is the Student-t density with location μ, scale σ, and degrees of freedom ν. Higher R_t indicates lower compatibility with the learned regime. Masked reconstruction via a Bernoulli mask forces the model to exploit cross-feature dependencies.

Given u_t, the local search generates a finite neighborhood $U_t = \{u_t + \delta_i\}_{i=1}^{K}$ with $|\delta_i| \leq \Delta_u$. The neighborhood uses K=5 directional probes $D = \{-1, -0.5, 0, 0.5, 1\}$. At cycle time t the candidate control values are generated as $u_{t+1}^{(k)} \in \{u_t - \Delta u,\ u_t - 0.5\Delta u,\ u_t,\ u_t + 0.5\Delta u,\ u_t + \Delta u\}$. The initial step size Δu is set to 5% of the admissible control range $\Delta u_0 = 0.05(u_{max} - u_{min})$. After any cycle in which no candidate satisfies the gating condition, the step size is constrained by a factor of 0.5. This is applied only down to a bound equal to 0.5% of $\Delta u_{min} = 0.005(u_{max} - u_{min})$. Therefore, the update rule for the neighbourhood size is $\Delta u \leftarrow \max(0.5\Delta u, 0.005(u_{max} - u_{min}))$. For each candidate $u \in U_t$, the algorithm forms $x_t(u)$ by replacing the setpoint component of x_t with u, while all other components remain fixed, and computes $s(u) = R(x_t(u))$. Then, acceptance is validated by the realized drift-risk after actuation. The control problem is therefore approached as a one-step, next window optimization where at each cycle the surrogate $s(u)$ approximates the expected drift-risk on the next window under a candidate setpoint, the selected update is actuated.

A discrete neighbourhood is used because the PLC accepts discrete write commands, and the FTT-MAE serves as a gradient-free black-box surrogate at deployment, making derivative-free local search appropriate. Lastly, restricting $\mathcal{U}_t$ to a finite set bounds the per-cycle computation to K forward passes of the FTT-MAE; necessary to meet the PLC cycle.

Feasibility is enforced via $L = 3$ constraint families: (i) lower and upper operating bounds, $g_1(u) = u_{\min} - u$ and $g_2(u) = u - u_{\max}$, which prevent non-admissible excursions; (ii) actuator limit, $g_3(u) = |u - u_t| - \Delta u_{\max}$, which caps the per-cycle excursion; and (iii) any additional process-specific bounds supplied by the physics-informed layer, although in this work no additional bounds beyond (i)-(ii) were activated on the steel hot-forming line. Constraint violations are expressed as residuals $r_\ell(u) = \max(0, g_\ell(u))$ so a candidate u is feasible if and only if $r_\ell(u) = 0$ for every ℓ. The feasible set at t is $F_t = \{u \in U_t \mid r_\ell(u) = 0,\ \forall \ell\}$. At each cycle, all candidates in U_t are first screened against constraints (i)-(iii); infeasible candidates are rejected. If $F_t \neq \varnothing$, the selected setpoint is $u_t^* = \mathrm{argmin}_{u \in F_t} s(u)$; otherwise, the current setpoint u_t is retained. Acceptance is then evaluated using $H > \tau$, as described in (2)-(4), and the PLC actuates u_t^* only when both the feasibility flag, $F_t \neq \varnothing$ and $u_t^* \in F_t$, and the acceptance flag, $H > \tau$, are true.

To ensure correctability-aware acceptance, let $s_0 = s(u_t)$ be the drift-risk at the current setpoint and $s^* = s(u_t^*)$ be the predicted drift-risk at the best feasible candidate. Two improvements are computed based on (2) and (3).

$$\Delta s_{log} = \log \frac{max(s_0, \varepsilon)}{max(s^*, \varepsilon)} \tag{2}$$

$$\Delta s_{abs} = s_0 - s^* \tag{3}$$

Where $\log(\cdot)$ is the natural logarithm and $\varepsilon > 0$, a constant to avoid numerical instability when scores are near zero. As the magnitude of $s(\cdot)$ can vary across regimes, both improvements are standardized using statistics computed on historical nominal data, including $\tilde{m}_{log}$ and $\tilde{\sigma}_{log}$ the median and median absolute deviation of Δs_{log} and $\tilde{m}_{abs}$ and $\tilde{\sigma}_{abs}$ the median and median absolute deviation of Δs_{abs}. Therefore, the final acceptance score is provided based on (4).

$$H = w \frac{\Delta s_{log} - \tilde{m}_{log}}{\tilde{\sigma}_{log}} + (1 - w) \frac{\Delta s_{abs} - \tilde{m}_{abs}}{\tilde{\sigma}_{abs}} \tag{4}$$

Where $w \in [0, 1]$. The weight w trades off the log-ratio term against the absolute term. Both τ and w are calibrated jointly on a labelled historical split that is disjoint from the unlabelled dataset D_{nom} training FTT-MAE. Calibration is performed using a grid search over $w \in \mathcal{W} = \{0.25, 0.5, 0.75\}$ and $\tau \in \mathcal{T} = \{0.24, 0.40, 0.55, 0.69, 0.85\}$. Defect labels are used only for offline calibration. For each pair (w, τ), the closed loop is replayed offline on the calibration split; the resulting acceptance flags are matched against the defects. The selected operating point is $(w^\star, \tau^\star) = \arg\max_{(w,\tau) \in \mathcal{W} \times \mathcal{T}} F_1(w, \tau)$, and this pair is retained and used unchanged during deployment. Using (2) and (3), the approach avoids emphasising small changes.

The PLC performs deterministic cyclic communication with the proposed approach, exchanging the required process tags and receiving the candidate setpoint, a feasibility flag from the physics-informed constraint layer and an acceptance flag based on $H > \tau$. The PLC updates the actuator only when both flags are true, otherwise it retains u_t.

By design, the method is a bounded supervisory heuristic: every update is feasible by construction, lying within actuator rate and range limits, and the candidates evaluated per cycle are bounded, as the neighbourhood contains exactly K probes. Actuation is controlled by the calibrated predicted-improvement score, so the expected one-step drift-risk under the trained model is non-increasing whenever $H > \tau$.

For MIS integration, each cycle logs $\{R_t, u_t, u_t^*, H\}$ with constraint residual summaries and flags. Such records support traceability and shift-level KPI aggregation such as intervention counts, constraint-violation rates and drift-risk trends, without requiring storage of raw high-rate signals.

IV. IMPLEMENTATION

The closed-loop pipeline is implemented in Python, with FTT-MAE (PyTorch) and the Pattern Search, feasibility, and acceptance layers (NumPy/pandas). Hyperparameters were selected by 5-fold cross-validation on a labelled historical split. FTT-MAE was then trained from scratch on the nominal-only set D_{nom} using the selected hyperparameters, with the architecture kept compact for edge deployment.

At deployment, the pipeline runs containerised on an edge host co-located with the PLC. Sensor packets arrive over MQTT, are persisted in PostgreSQL and exposed via HTTP REST. At each cycle, Node-RED pulls the measurement window from REST, invokes the closed loop, and forwards the resulting tuple [u_t, feasibility flag, acceptance flag] to the PLC via OPC UA. The PLC writes the setpoint only when both flags are true, otherwise it retains u_t. Cycle-level outputs $\{R_t$, u_t, u_t^*, H, constraint-residual summaries$\}$ are persisted in the database for MIS dashboards and traceability.

V. USE CASE

The proposed closed-loop approach was evaluated on a steel-industry hot forming line consisting of a heating, oxide removal and rolling step. The objective was to recommend updates to a single rolling setpoint u_t to mitigate emerging quality drift before defects occur. The FTT-MAE drift risk model is trained using only nominal data, and defect labels are not used for training and are used only for offline evaluation and reporting of intervention trade-offs.

The dataset comprised 115,000 observations over 3 months with 0.85% defective samples. FTT-MAE training used only nominal windows; defect labels served offline evaluation only. A variable subset with summary statistics is shown in Table I. To stabilize training and ensure comparable feature scales, each feature was transformed with Yeo-Johnson [25]. Data was split using a stratified 80% and 20% split. The 80% labelled training split was used for FTT-MAE hyperparameter selection and for (w, τ) calibration. FTT-MAE was trained on the nominal subset. Defect labels were used for selection, not training, and 20% was held out for testing. The dataset shows raw, pre-transformation signals, with the target rolling parameter being an offset (mm from nominal reference), hence negative values are possible.

TABLE I. TABLE I. DATASET SAMPLE.

Feature	Mean	[Min, Max]	Skewness
Target rolling parameter [mm]	3.60	[-39, 22]	-0.98
Rolling temperature [°C]	275.14	[51, 571]	-0.02
Product temperature after heating [°C]	1275.73	[998, 1363]	-19.76
Heating process's energy consumption [kW]	738.68	[733, 1072]	-2.07

Given the defect rarity, model evaluation prioritised the Area Under the Precision-Recall Curve (AUPRC), defined as the integral of precision over recall as the decision threshold varies. The Area Under the Receiver Operating Characteristic Curve (AUROC), defined as the probability that a randomly chosen positive is ranked above a randomly chosen negative, was also reported as a threshold-independent secondary measure. To demonstrate the applicability of FTT-MAE, its performance was compared against reconstruction-based baselines that included Conditional Variational Autoencoders (CVAE), Hierarchical Variational Autoencoders (HVAE) and Recurrent Variational Autoencoders (RVAE). All models were tuned using 5-fold cross-validation on the training set. As depicted in Table II, FTT-MAE achieves the highest mean AUPRC, justifying its selection against similar models.

TABLE II. TABLE II. EVALUATION METRICS ACROSS THE 5-FOLD CROSS-VALIDATION TEST-FOLDS.

Model	Mean AUPRC (%)	Mean AUROC (%)
FTT-MAE	6.18 (±1.01pp)	74.66 (±6.43pp)
RVAE [26]	3.44 (±0.41pp)	66.01 (±3.05pp)
CVAE [27]	3.25 (±1.13pp)	66.52 (±3.05pp)
HVAE [28]	3.09 (±0.75pp)	65.84 (±1.83pp)

A 2-week real-world validation experiment was performed. At each cycle, pattern search evaluates candidate setpoints in a bounded U_t and computes score *H*. Corrective action is triggered when $H > \tau$, where τ is selected from historical data to balance responsiveness to defects. Table III reports the F1-score of accepted interventions against offline defect labels to identify the optimal combination of w and τ, while Table IV reports the trade-off of the acceptance threshold τ on the calibration split. Defect-window reduction is the fraction of windows later labelled defective that are eliminated in the offline counterfactual replay when the closed loop applies u_t^*, and non-defect intervention rate is the fraction of windows later labelled non-defective on which the method would trigger an intervention ($H > \tau$). We selected $\tau = 0.69$ and $w = 0.5$ as a balanced operating point.

TABLE III. TABLE III. F1-SCORE OF ACCEPTED INTERVENTIONS AGAINST OFFLINE DEFECT LABELS ON THE CALIBRATION SPLIT.

w \ τ	0.24	0.40	0.55	0.69	0.85
0.25	0.071	0.084	0.092	0.095	0.078
0.5	0.074	0.090	0.101	0.112	0.089
0.75	0.069	0.083	0.094	0.103	0.081

TABLE IV. TABLE IV. APPLIED THRESHOLD.

Defect-window reduction (%)	Non-defect intervention rate (%)	τ value
39.99	29.99	0.69
55.10	35.00	0.24

Reductions of a few percentage points in R_t are consistent with the scope of the problem where only a single setpoint is actuated, the admissible step is small, and the dominant share of R_t is driven by upstream features that the actuator cannot influence. As an upper-bound reference, the maximum reduction reachable by an approach that selects the best feasible candidate at each cycle on the calibration split is 7.2%. The proposed method achieves 5.5% maximum and 4.93% mean on defective windows, capturing 68% of that bound. The 6.18% AUPRC reflects the irreducible overlap of nominal and defective regimes at 0.85% prevalence. The AUPRC of a random ranker on this dataset is 0.85%, thus, the FTT-MAE delivers a lift of 7.3 times over chance.

For accepted interventions, the approach yields about 5% drift-risk reduction while respecting feasibility constraints. To quantify mitigation in the real-world deployment, for each accepted intervention at cycle *t*, the realized relative reduction was computed as $\Delta R(\%) = \frac{R_t - R_{t+1}}{R_t}$, where R_t is the drift-risk score before actuation and R_{t+1} is the score recomputed on the next sampling window after the PLC-applied update. Table V summarizes the results over the 2-week experiment for $\tau = 0.69$. Interventions associated with windows later labelled as defective yield a mean ΔR of approximately 5.0% (median 5.0%, maximum 5.5%), while non-defective windows exhibit smaller improvements of 2.1% on average, indicating that the acceptance thresholds concentrate stronger corrections when the process deviates toward defect regimes. No feasibility violations were observed in the PLC/edge logs during the experiment; feasibility filtering is enforced prior to PLC write.

TABLE V. TABLE V. VALIDATION OF REALIZED DRIFT-RISK REDUCTION.

Group	N	Mean R_t	Mean R_{t+1}	Mean $\Delta R(\%)$	Median $\Delta R(\%)$	Max $\Delta R(\%)$
Defective windows	72	313.38	297.94	4.93	5.03	5.50
Non-defective windows	348	286.29	280.20	2.13	2.13	4.49
All accepted	420	290.94	283.24	2.61	2.42	5.50

Among the $L = 3$ constraint families, namely the lower and upper operating bounds and the actuator rate-of-change limit, the constraint $g_3(u) = |u - u_t| - \Delta u_{\max}$ was the only activated during the deployment, binding the selected setpoint u_t^* at one extreme of the Δu neighbourhood in 312 of the 420 accepted interventions, corresponding to 74.3%. This is consistent with the slow nature of the drift, since meaningful corrections accumulate over multiple cycles. The bounds $u_{\min}$ and $u_{\max}$ were never active during the two-week experiment, which is consistent with the asset operating well within its nominal range; the bounds are nevertheless retained for safety in the face of future regime changes.

In addition, the per-intervention ΔR distribution exhibits a long-tailed shape with an interquartile range of [1.4%, 3.9%]. The upper has 14 interventions (3.3% of the accepted total), with $\Delta R \in [5.0\%, 5.5\%]$, all associated with windows later labelled defective and concentrated near the start of drift. No interventions produced a negative ΔR. No accepted intervention was removed as an outlier. The high-response cases were retained and characterised separately, since they corresponded to early drift episodes later labelled defective.

To isolate the contribution of the correctability-aware acceptance score, two ablations were run (Table VI). The first baseline was feasibility-only, which always applies the best feasible candidate u_t^* whenever the feasible set $\mathcal{F}_t$ is non-empty, that is, whenever $\mathcal{F}_t \neq \varnothing$, while bypassing the acceptance score H. The second baseline is improvement-only, which accepts u_t^* whenever the surrogate score improves over the current setpoint, $s^* < s_0$, by any margin; this bypasses the standardised acceptance score H but still requires a predicted improvement over the current setpoint. Defect-window reduction is computed against a no-intervention reference of setpoints over the same period. Lastly, the non-defect intervention rate has been computed as $\frac{N_{\text{int,nondef}}}{N_{\text{nondef}}}$, where $N_{\text{int,nondef}}$ the accepted interventions on windows later labelled non-defective, and N_{nondef} the total non-defective windows in the 2-week validation stream.

TABLE VI. TABLE VI. ABLATION OF ACCEPTANCE POLICY.

Policy	Interventions	Mean $\Delta R(\%)$	Defect reduction (%)	Non-defect intervention rate
Feasibility	1,184	1.42	19.6	0.82
Improvement	763	2.08	27.4	0.61
Proposed approach	420	2.61	40.0	0.30

Defect-generation reduction is defined as $DGR = 1 - \frac{N_{def}^{post}}{N_{def}^{pre}}$, where N_{def}^{pre} the number of defective windows observed in a matched 2-week period under the previous policy and N_{def}^{post} the count during the 2-week deployment. On the 2-week stream, $N_{def}^{pre} = 163$ and $N_{def}^{post} = 98$ resulting in a $DGR = 0.399 = {\sim}40\%$. The gap between the 72 accepted interventions on later-defective windows (Table V) and the 65 defective windows eliminated according to the DGR calculation reflects 7 interventions on windows for which the applied u_t^* did not move the next-window below the defect threshold, and is therefore expected.

Lastly, nine stakeholders (quality and process managers) rated an MIS-oriented dashboard built from cycle-level decision logs on a 5-point Likert scale: workflow fit 4.4, traceability/audit readiness 4.2, shift reporting clarity 4.3, and dashboard integration ease 4.1. High scores were attributed to surfacing actionable, low-volume decision events aggregable into KPIs; lowest scores concerned governance and alignment with existing reporting policies.

VI. CONCLUSIONS

This work proposes a closed-loop ZDM drift mitigation strategy that turns drift detection into constraint-feasible setpoint updates. It combines an unsupervised transformer autoencoder for drift-risk scoring with derivative-free pattern search, while a physics-informed constraint layer enforces operating constraints. Setpoint updates are sent as PLC-ready messages, and decisions are logged to MIS.

The approach was evaluated on a hot-forming line and achieved up to 5.5% drift-risk reduction and a 40% lower defective-window count relative to a matched period, while MIS outputs remained compact and directly consumable, supported by a formative manager review with average scores above 4/5 across usability criteria.

A limitation is the modest discriminative power of scores under extreme defect rarity, with a mean AUPRC of 6.18%, reflecting overlap between. Future work will therefore focus on improved representation learning and calibration under rare-event conditions, broader validation across operating regimes, and long-horizon deployment considerations.

Acknowledgment

This work was partially supported by the HORIZON-CL4-2021-TWIN-TRANSITION-01 openZDM project under Grant Agreement No. 101058673

References

[1] N. Leberruyer, J. Bruch, M. Ahlskog, and S. Afshar, 'Toward Zero Defect Manufacturing with the support of Artificial Intelligence—Insights from an industrial application', Computers in Industry, vol. 147, p. 103877, May 2023, doi: 10.1016/j.compind.2023.103877.

[2] E. Verna, G. Genta, M. Galetto, and F. Franceschini, 'Zero defect manufacturing: a self-adaptive defect prediction model based on assembly complexity', International Journal of Computer Integrated Manufacturing, vol. 36, no. 1, pp. 155–168, Jan. 2023, doi: 10.1080/0951192X.2022.2081360.

[3] F. Kibrete, D. Engida Woldemichael, and H. Shimels Gebremedhen, "Multi-Sensor data fusion in intelligent fault diagnosis of rotating machines: A comprehensive review," Measurement, vol. 232, p. 114658, Jun. 2024, doi: 10.1016/j.measurement.2024.114658.

[4] S. Ghansiyal et al., "Anomaly detection towards zero defect manufacturing using generative adversarial networks," Procedia CIRP, vol. 120, pp. 1457–1462, Jan. 2023, doi: 10.1016/j.procir.2023.09.193.

[5] O. Abdelrahman and P. Keikhosrokiani, "Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning," IEEE Access, vol. 8, pp. 189661–189672, Jan. 2020, doi: 10.1109/ACCESS.2020.3029826.

[6] C. Eze, A. Ezema, L. Roth, Z. Pan, F. Ponci, and A. Monti, "AI Bias in Power Systems Domain—Exemplary Cases and Approaches," Energies, vol. 18, no. 18, p. 4819, Sep. 2025, doi: 10.3390/en18184819.

[7] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series," IEEE Access, vol. 7, pp. 1991–2005, 2019, doi: 10.1109/ACCESS.2018.2886457.

[8] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Fault-Attention Generative Probabilistic Adversarial Autoencoder for Machine Anomaly Detection," IEEE Transactions on Industrial Informatics, vol. 16, no. 12, pp. 7479–7488, Dec. 2020, doi: 10.1109/TII.2020.2976752.

[9] R. Kermenov, G. Nabissi, S. Longhi, and A. Bonci, "Anomaly Detection and Concept Drift Adaptation for Dynamic Systems: A General Method with Practical Implementation Using an Industrial Collaborative Robot," Sensors, vol. 23, no. 6, p. 3260, Mar. 2023, doi: 10.3390/s23063260.

[10] G. Provan, "Toward Explainable AutoEncoder-Based Diagnosis of Dynamical Systems," Algorithms, vol. 16, no. 4, p. 178, Mar. 2023, doi: 10.3390/a16040178.

[11] H. Kang and P. Kang, "Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism," Knowledge-Based Systems, vol. 290, p. 111507, Apr. 2024, doi: 10.1016/j.knosys.2024.111507.

[12] A. Logacjov and K. Bach, "Self-supervised learning with randomized cross-sensor masked reconstruction for human activity recognition," Engineering Applications of Artificial Intelligence, vol. 128, p. 107478, Feb. 2024, doi: 10.1016/j.engappai.2023.107478.

[13] X. Deng and Z. Zhang, "Nonlinear Chemical Process Fault Diagnosis Using Ensemble Deep Support Vector Data Description," Sensors, vol. 20, no. 16, p. 4599, Aug. 2020, doi: 10.3390/s20164599.

[14] I.-S. Gherghina, N. Bizon, G.-V. Iana, and B.-V. Vasilică, "Recent Advances in Fault Detection and Analysis of Synchronous Motors: A Review," Machines, vol. 13, no. 9, p. 815, Sep. 2025, doi: 10.3390/machines13090815.

[15] Z. Wang, J. Shao, S. Zhang, J. Zhang, and Y. Pang, "Research Progress on Application of Machine Learning in Continuous Casting," Metals, vol. 15, no. 12, p. 1383, Dec. 2025, doi: 10.3390/met15121383.

[16] J. Zhu et al., "An Intelligent Optimization Strategy for Blast Furnace Charging Operation Considering Three-Dimensional Burden Surface Shape," IEEE/CAA Journal of Automatica Sinica, vol. 12, no. 7, pp. 1445–1463, Jul. 2025, doi: 10.1109/JAS.2025.125192.

[17] F. Wahid et al., "An Enhanced Firefly Algorithm Using Pattern Search for Solving Optimization Problems," IEEE Access, vol. 8, pp. 148264–148288, 2020, doi: 10.1109/ACCESS.2020.3015206.

[18] P. Catti, N. Nikolakis, K. Sipsas, N. Picco, and K. Alexopoulos, "A hybrid digital twin approach for proactive quality control in manufacturing," Procedia Computer Science, vol. 232, pp. 3083–3091, Jan. 2024, doi: 10.1016/j.procs.2024.02.124.

[19] N. Nikolakis and P. Catti, "A Physics-Aware Latent Diffusion Framework for Mitigating Adversarial Perturbations in Manufacturing Quality Control," Future Internet, vol. 18, no. 1, p. 23, Jan. 2026, doi: 10.3390/fi18010023.

[20] Y. Qamsane et al., "A Methodology to Develop and Implement Digital Twin Solutions for Manufacturing Systems," IEEE Access, vol. 9, pp. 44247–44265, 2021, doi: 10.1109/ACCESS.2021.3065971.

[21] G. Yasmeen, L. Anthonysamy, and A. O. Ojo, "Resource-Governed BDA Adoption for Resilient Supply-Chain Operations: Qualitative Evidence from Malaysian Manufacturing Industry," Sustainability, vol. 17, no. 21, p. 9620, Oct. 2025, doi: 10.3390/su17219620.

[22] Y. Gorishniy, I. Rubachev, V. Khrulkov, and A. Babenko, "Revisiting Deep Learning Models for Tabular Data", Advances in Neural Information Processing Systems, vol. 34, 2021, pp. 18932-18943.

[23] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders Are Scalable Vision Learners," Dec. 19, 2021, arXiv: arXiv:2111.06377. doi: 10.48550/arXiv.2111.06377.

[24] I. Papastathopoulos and J. A. Tawn, "A generalised Student's t-distribution," Statistics & Probability Letters, vol. 83, no. 1, pp. 70–77, Jan. 2013, doi: 10.1016/j.spl.2012.09.002.

[25] E. Tieppo, J. P. Barddal, and J. C. Nievola, "Improving Data Stream Classification using Incremental Yeo-Johnson Power Transformation," in 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Oct. 2022, pp. 3286–3292. doi: 10.1109/SMC53654.2022.9945302.

[26] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, 'Generating sentences from a continuous space', presented at the Proceedings of the 20th SIGNLL conference on computational natural language learning, 2016, pp. 10–21.

[27] K. Sohn, H. Lee, and X. Yan, 'Learning Structured Output Representation using Deep Conditional Generative Models'.

[28] A. Vahdat and J. Kautz, 'NVAE: A deep hierarchical variational autoencoder', Advances in neural information processing systems, vol. 33, pp. 19667–19679, 2020.