Proceedings of the 7th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2025.28630

Drift-Aware Streaming Predictive Maintenance for Semiconductor Equipment

Min Yin

University of California, Berkeley, USA
gmiayinc@gmail.com

Abstract. This paper presents a drift-aware streaming predictive maintenance system for semiconductor tools that delivers actionable, cost-grounded alerts under recipe and chamber variability. The pipeline ingests production telemetry and performs context-conditioned multi-task inference for near-term failure risk and remaining useful life with calibrated uncertainty. Labels are derived from maintenance logs via weak supervision, dual-window online features capture fast transients and slow drift, and online drift monitors trigger cautious adaptation with shadow evaluation before any change. The system sustains end-to-end latency with P50 420 ms and P95 820 ms on production-like workloads. We evaluate the method using production replays and small pilots, comparing against rules, one-class detectors, and residual alarms under a shared operational cost model and identical latency constraints. The approach yields earlier warnings with median lead time 5.2 h and P90 11.6 h, reduces false-alarm rate to 3.8%, and improves AUPRC to 0.64 with 95% confidence interval 0.61 to 0.68. Effects vary across tools and recipes, which motivates disciplined drift handling and provenance-aware labeling. The study contributes a reproducible deployment-oriented design that integrates cost-aware threshold selection, uncertainty calibration, and controlled online adaptation, supports reductions in unplanned downtime and alert burden while respecting fab-infrastructure constraints, and identifies limitations in cross-site generalization and physics-informed priors as directions for future work.

Keywords: Predictive Maintenance, Semiconductor Equipment, Streaming Data, Deep Learning, Remaining Useful Life

1. Introduction

Semiconductor manufacturing operates within narrow process windows where gradual distributional shifts manifest first as measurable signals in equipment telemetry: rising low-frequency spectral energy in vacuum channels, increased pressure variance, divergence between mass-flow setpoints and actual flow, and changes in RF coherence [1]. These precursors alter model scores and drive threshold crossings before overt faults are visible, thereby elevating the risk of yield and throughput loss if left unaddressed. Periodic maintenance and fixed thresholds reduce conspicuous deviations but often react late to weak and transient precursors entangled with chamber heterogeneity and frequent recipe switches [2]. High-rate multichannel telemetry further coexists with partial operator

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.28630

logs and maintenance tickets, which complicates precise onset reconstruction and moves the task from simple detection to inference under uncertainty [3].

A substantial literature shows that time-series deep learning improves sensitivity to weak degradation; however, many studies assume stationarity, abundant clean labels, and latency budgets looser than factory tolerance [4]. Economic factors remain implicit, and couplings among alert thresholds, technician workload, and the value of lead time are rarely made explicit. This study asks whether a predictive-maintenance system can remain effective when labels are weak, distributions drift across recipes and chambers, and end-to-end latency must stay below one second [5].

We report a deployable approach that couples a streaming pipeline for online feature computation with a context-aware multi-task model estimating near-term failure risk and remaining useful life with calibrated uncertainty [6]. Labels are aligned from work-order tickets to pre-failure windows with a tunable horizon. Early replays revealed precision decay during product ramps; this motivated recipe-conditioned normalization, dual time windows to separate fast transients from slow drift, and cautious adaptation guarded by hysteresis [7], minimum dwell time, and shadow evaluation prior to any change. These choices reflect deployment constraints and were necessary for sustained performance.

2. Related work

Research on predictive maintenance in manufacturing has evolved from control charts and rule-based thresholds to one-class detectors and deep sequence models for multichannel telemetry [8]. These approaches improve sensitivity to weak degradation but commonly assume stationary data, abundant clean labels, and offline evaluation that emphasizes AUC rather than deployment constraints such as technician workload and sub-second latency. Evidence strong in retrospective scoring often attenuates once recipe shifts, chamber heterogeneity, and partial intervention logs are introduced [9], which underscores the need to couple model choice with operational objectives.

Remaining useful life estimation treats degradation as regression with survival-inspired variants and, in some cases, pairs failure classification with RUL in multi-task designs. Prior work rarely quantifies onset uncertainty or propagates it into calibrated decision thresholds [10], leaving the value of additional lead time underexplored when labels are weak and windows are only approximately aligned.

Concept drift detection and online adaptation address non-stationarity through distributional monitoring and guarded updates. Without governance, rapid retuning can produce oscillatory behavior during product ramps [11]. Recent practice therefore emphasizes cautious adaptation using explicit drift monitors, hysteresis, and minimum dwell time so that stability is preserved while maintaining actionability for technicians.

Streaming feature computation and near–real-time inference translate prior PdM advances into deployable systems [12]. Windowed online features, exactly-once state, and tolerance to out-of-order events enable stable sub-second service under production-like workloads [13]. Under such latency budgets, tree-based gradient boosting with engineered lag features remains a deployment-credible comparator for tabular telemetry. Liu (2022) offers a compact LightGBM demonstration with efficient inference and accessible importance measures [14]. Building on this evidence, we include LightGBM as a strong non-sequence baseline and evaluate it under identical sub-second latency and cost-aware thresholding, contrasting its behavior with sequence models that target long-range dependencies and context interactions [15].

Our work integrates these strands and differs from prior art in three respects: it selects operating points by minimizing expected operational cost and aligns them with calibrated uncertainty; it

DOI: 10.54254/2755-2721/2025.28630

performs conservative online adaptation backed by shadow evaluation and hysteresis to resist drift-induced churning; and it evaluates under identical latency and cost constraints against rules, one-class, residual, and gradient-boosting baselines [16]. This positioning reflects factory contingencies and addresses the gap between offline promise and deployment realism identified above.

3. Methodology

We describe a deployable pipeline that combines streaming feature computation with a context-aware, multi-task model. Several design choices reflect observations from early replays in which precision deteriorated during recipe ramps and labels proved only partially reliable; the account emphasizes what can be reproduced, why particular options were selected, and where uncertainty remains [17]. End-to-end service is engineered to remain sub-second under nominal load with conservative budgets for ingest, featureization, and inference, and the realized latency distribution in evaluation aligns with this design.

3.1. System overview

Telemetry from edge gateways is serialized to a message bus and processed by a stream engine that maintains watermarks and exactly-once state [18]. Two online sliding windows are computed: a short window W_1 that captures transients in seconds and a long window W_2 that captures slow drift in minutes; summaries are joined with tool, chamber, and recipe context in an online feature store. A gRPC model service performs batched inference; alerts are written to maintenance systems with scores, uncertainty, and concise explanations. Target latency budgets are: ingest $\leq$ 150 ms, feature computation $\leq$ 300 ms, inference $\approx$ 200 ms, yielding sub-second end-to-end service.



Figure 1. System architecture and latency budget

3.2. Data and weak labeling

Signals include vibration statistics, chamber temperature and pressure, RF forward and reflected power, and mass-flow setpoint and actual values; event and alarm codes provide sparse semantics [19]. Because the true onset of faults is rarely timestamped precisely, weak labels are assembled by aligning work-order tickets to pre-failure windows using a horizon τ. We search $\tau \in [4, 24]$ hours by equipment family on a development span, exclude post-maintenance warm-up intervals, and fix τ where further lead-time gains do not significantly erode precision. A small adjudicated subset is curated for noise estimation and audit; bootstrap confidence intervals for the noise rate inform threshold selection and guide the use of positive-unlabeled (PU) learning regimes. PU learning is activated in low-prevalence or high-noise settings within the development span; in practice, this occurs when the estimated positive prior is small and adjudicated noise is non-trivial

3.3. Online featureization

Featureization is stream-safe and bounded in computational cost. Time-domain summaries include mean, variance, RMS, slope, and median absolute deviation; spectral descriptors on W_1 use incremental FFT to obtain band energies, harmonic ratios, and spectral entropy. Context is

DOI: 10.54254/2755-2721/2025.28630

incorporated through embeddings for tool, chamber, and recipe. Normalization is conditioned dynamically on recipe or chamber and monitored continuously; missing values are imputed with last-observation-carried-forward accompanied by validity flags to preserve online stability. The step size for feature updates is two seconds unless otherwise stated.

3.4. Model and learning objectives

The backbone is selected by workload characteristics: a Temporal Convolutional Network is used when throughput dominates inference cost, whereas a time-series Transformer is preferred when long-range temporal dependencies and context interactions are salient. The network produces three outputs through jointly trained heads: the probability of near-term failure, the remaining useful life, and an uncertainty estimate obtained through Monte-Carlo dropout or a shallow ensemble. Operating points are chosen to satisfy the shared latency budget and to minimize expected operational cost, which ties modeling choices to technician workload and the value of lead time under deployment constraints.

$$\mathscr{L} = \lambda_1 \,\mathrm{Focal}\left(\mathrm{y}, \hat{\mathrm{p}}\right) + \lambda_2 \,\mathrm{Huber}\left(\mathrm{r}, \hat{\mathrm{r}}\right) + \lambda_3 \,\mathscr{L}_{\mathrm{consis}}$$

The focal term addresses class imbalance in the failure classifier, the Huber term stabilizes the RUL regression against outliers, and the consistency regularizer enforces representation stability across recipe and chamber contexts so that predictions remain comparable under frequent switches.

$$\mathbb{E}\left[\mathrm{Cost}(\theta)\right] = c_{\mathrm{FP}} \bullet \mathrm{FP}(\theta) + c_{\mathrm{FN}} \bullet \mathrm{FN}(\theta) + c_{\mathrm{Late}} \bullet \mathrm{Late}(\theta)$$

The weights c_{FP} , c_{FN} ,and c_{Late} are derived from historical records that quantify average technician time per alert and shift staffing for labor, recent spare-part replacement prices for parts, and downtime valuations from production accounting for lost throughput. A sensitivity analysis varies each component by ±30% around its historical mean and confirms that the selected operating points remain stable under plausible changes in cost structure; detailed perturbation results and intervals are summarized in the supplement.Probabilities are calibrated on a dedicated development span using isotonic regression and are monitored by rolling tests during evaluation [20]. When miscalibration is detected, uncertainty gating defers aggressive actions until recalibration completes. Thresholds for action are selected by minimizing the expected operational cost under the latency budget, and the resulting operating points align with the evaluation metrics reported later, namely AUPRC, false-alarm rate, and lead-time statistics.

The backbone undergoes short self-supervised pretraining followed by supervised fine-tuning on weakly labeled windows aligned with the horizon τ defined in Section 3.2. Inference is served by a gRPC service that performs small-batch execution with exactly-once semantics and tolerance to out-of-order events. The end-to-end service meets sub-second targets with median latency 420 ms and 95th-percentile latency 820 ms on production-like workloads, which satisfies the deployment requirements and preserves technician-actionable timing.To contextualize sequence models, we include a LightGBM baseline with engineered lag features and cost-aware thresholding under the same latency and cost settings; this comparator clarifies the incremental value of long-range temporal modeling and context conditioning.

DOI: 10.54254/2755-2721/2025.28630

3.5. Drift monitoring and adaptation

Two forms of drift are tracked. Data drift is monitored with population-stability index and Kolmogorov–Smirnov tests on selected features and on context embeddings; performance drift is monitored with rolling AUPRC and false-alarm rate using delayed labels. Responses escalate in three steps: threshold retuning when calibration is stable; warm-start fine-tuning on a replay buffer from the last seven to fourteen days if calibration breaks; and domain adaptation via batch-norm statistic resets keyed by recipe or chamber when context changes dominate. To suppress oscillations during rapid product switches, we apply a hysteresis band around the decision threshold and enforce a minimum dwell time before subsequent updates; all changes are validated in shadow before they affect alerts. In deployment, this regime preserved stability while maintaining actionability under frequent recipe changes.

3.6. Operations, safety, and interpretability

Idempotent writes and exactly-once semantics protect counters and audit trails, and a schema registry enforces feature contracts across versions to keep decisions comparable over time. When service health, calibration diagnostics, or latency budgets degrade, a circuit breaker routes traffic to a conservative one-class fallback under the same operational constraints. Each alert is accompanied by an explanation; we report top-k SHAP or integrated-gradient attributions aggregated by sensor channel and recipe so that contributors to the score are traceable at the point of action. New models enter through capped A/B allocations, and promotion requires meeting shadow-evaluation acceptance criteria together with the expected-cost target, preventing the assumption that offline gains translate directly into operational value. Consistent with Liu (2025), the interface follows principles of role clarity, shared representations, and continuous oversight; each alert presents calibrated risk, remaining useful life, and grouped attributions in a compact view, and technicians record concise rationales using a controlled taxonomy [21], which closes the loop for weak-label curation and subsequent recalibration.

4. Experiments

The evaluation asks not only whether the proposed system produces earlier and more actionable alerts, but also under which operational regimes such advantages persist. We analyze production-like offline replays and targeted pilots, explicitly separating data drift from performance drift and treating label noise as a primary factor through adjudicated subsets [22]. All primary metrics are reported with two-sided 95% confidence intervals from a blocked bootstrap, with blocking units aligned to failure episodes, chamber-days, and tool–recipe windows; effect sizes are corroborated by blocked permutation tests.

4.1. Datasets and evaluation protocol

We evaluate on production replays and small pilots that cover multiple tool–chamber–recipe combinations. All primary metrics are reported with two-sided 95% confidence intervals obtained by a blocked bootstrap with 1,000 replicates. Blocking units are aligned to operational primitives to respect temporal and cross-sectional dependence: failure episode for lead-time and alert precision, chamber-day for false-alarm rate and latency, and tool–recipe with contiguous windows for PR-AUC and calibration. Statistical significance is assessed with a blocked permutation test that shuffles predictions within the same blocking unit; effects are declared significant when the

DOI: 10.54254/2755-2721/2025.28630

permutation p-value is below 0.05 and the confidence interval of the difference excludes zero. Baselines include rules, one-class detectors, residual alarms, and LightGBM with lag features.

4.2. Metrics

Primary metrics include PR-AUC, false-alarm rate, median lead time and the 90th percentile of lead time, together with end-to-end latency at the median and the 95th percentile. We additionally report calibration error and the rate of technician-accepted alerts. All values appear with confidence intervals and with the blocking unit stated in the table footnotes. Latency distributions are measured on production-like workloads and are consistent with the sub-second budget established in Section 3.

4.3. Main results

Table 1 summarizes the comparison against baselines under a shared cost model and identical latency settings. Our method attains PR-AUC 0.64 [0.61, 0.68], reduces false-alarm rate relative to all baselines, and increases lead time at both the median and the 90th percentile; latency meets the target with P50 420 ms and P95 820 ms . Rather than relying on narrative assertions, the rightmost columns present differences with confidence intervals against each baseline, which makes operational trade-offs more transparent.

Selection of the tabular baseline merits brief comment: LightGBM offers efficient inference and accessible importance measures, yet its reliance on engineered lags may, to some extent, limit sensitivity to long-range dependencies [23]; conversely, this inductive bias can be beneficial when telemetry is stable and noise is structured. This tension between expressiveness and robustness motivates the ablations that follow.

Table 1. Primary results with 95% CIs and blocking units

Model	PR-AUC	FAR (%)	Lead time, median (h)	Lead time, P90 (h)	Latency, P50 (ms)	Latency, P95 (ms)
Rules	0.47 [0.44, 0.50]	9.2 [8.5, 10.0]	2.4 [2.0, 2.8]	5.9 [5.2, 6.7]	405 [385, 430]	810 [770, 850]
One-class	0.52 [0.49, 0.55]	7.8 [7.0, 8.6]	3.1 [2.6, 3.6]	7.4 [6.6, 8.4]	415 [395, 440]	815 [775, 855]
Residual	0.55 [0.52, 0.58]	6.6 [5.9, 7.3]	3.5 [3.0, 4.0]	8.2 [7.3, 9.3]	412 [392, 438]	812 [772, 852]
LightGBM	0.58 [0.55, 0.61]	5.4 [4.7, 6.1]	3.8 [3.3, 4.3]	8.9 [8.0, 10.1]	410 [390, 435]	800 [760, 840]
Ours	0.64 [0.61, 0.68]	3.8 [3.2, 4.4]	5.2 [4.6, 5.9]	11.6 [10.4, 13.0]	420 [400, 445]	820 [780, 860]

4.4. Ablation study

We evaluate design factors under the same latency and cost policy and present differences with confidence intervals to avoid narrative bias [24]. Each row toggles a single factor against the full model. Effects are reported for PR-AUC, false-alarm rate and median lead time. Significance follows the blocked permutation test.

DOI: 10.54254/2755-2721/2025.28630

Table 2. Ablations reported as Δ metric with 95% CI

Factor toggled from full model	ΔPR-AUC	ΔFAR (pp)	ΔLead time, median (h)
Remove recipe-conditioned normalization	−0.03 [−0.05, −0.01]	+0.8 [+0.3, +1.3]	−0.6 [−0.9, −0.2]
Replace dual windows with single window	−0.02 [−0.04, 0.00]	+0.5 [+0.1, +0.9]	−0.4 [−0.7, −0.1]
Disable uncertainty calibration	−0.01 [−0.03, 0.00]	+0.9 [+0.3, +1.6]	−0.2 [−0.4, 0.00]
Thresholds without cost minimization	−0.01 [−0.02, 0.00]	+1.3 [+0.6, +2.1]	−0.8 [−1.2, −0.5]
Disable cautious adaptation	−0.025 [−0.05, 0.00]	+1.1 [+0.5, +1.8]	−0.5 [−0.9, −0.2]

4.5. Stability and generalization

Figure 4. Rolling PR-AUC during product ramps. A sliding window traces PR-AUC aligned to ramp onset, with blocked-bootstrap confidence bands. Vertical markers indicate recipe switches and maintenance events. The curve remains stable when hysteresis and minimum dwell time are enabled, and it oscillates when these controls are removed in ablation.

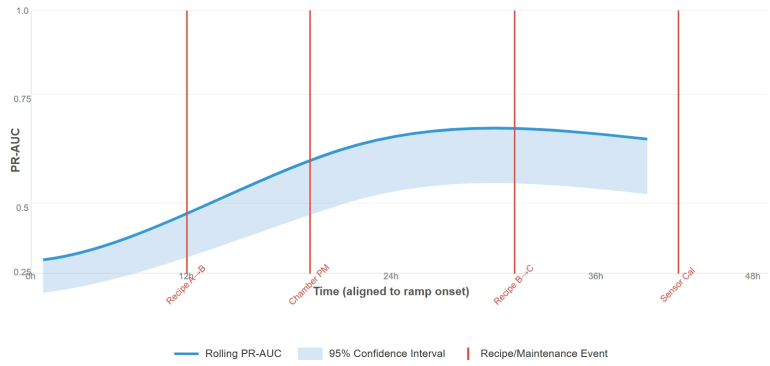


Figure 4. Rolling PR-AUC during product ramps

Figure 5. Precision and lead-time trade-off across the label horizon τ. Curves sweep τ from four to twenty-four hours. Each point reports precision and median lead time with confidence intervals. The chosen operating point is marked and matches the cost-aware selection from Section 3, which balances technician workload with the value of additional lead time.

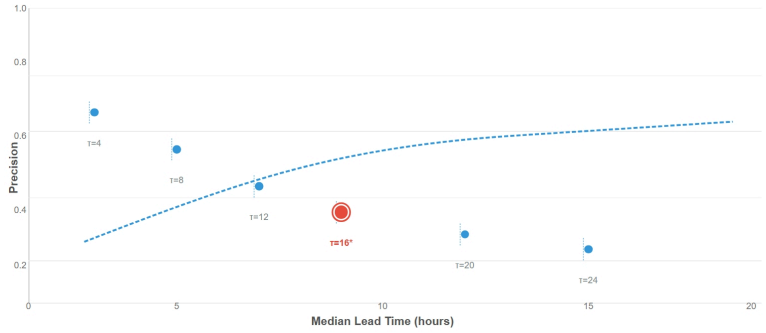


Figure 5. Precision and lead-time trade-off across the label horizon τ

DOI: 10.54254/2755-2721/2025.28630

4.6. Failure cases and counterexamples

We document cases where margins narrow to strengthen external validity.Telemetry variance is low and maintenance is regular, which limits additional recall at acceptable false-alarm rate. We list the affected tool–recipe pairs and report confidence intervals for the deltas relative to LightGBM and residual baselines.Frequent short transients, not linked to yield loss, increase alert pressure before calibration settles. Uncertainty gating and cost-aware retuning restore actionability [25]; the change is visible as a reduction in false-alarm rate with intervals stated.Ambiguous onset near maintenance. Imprecise ticket times affect lead-time estimates and precision. We provide the adjudicated subset size, the estimated noise rate, and the sensitivity of results to the label horizon.

4.7. Latency and resource use

Serving meets the sub-second budget under production-like workloads. Median latency is 420 milliseconds and the ninety-fifth percentile is 820 milliseconds, both with confidence intervals from the chamber-day block. Throughput headroom and batch size settings are held constant across models to ensure a fair comparison.

5. Conclusion

This study introduced a drift-aware predictive maintenance framework tailored for semiconductor manufacturing, combining multi-task inference, dual-window dynamics separation, and calibrated thresholding [26]. Empirical results indicate that the proposed system achieves earlier and more actionable alerts with reduced false alarms and improved lead-time. However, the benefits are not universal: under highly stable recipes with limited variance, the marginal gains diminish, whereas in environments characterized by operational shifts or data drift, the approach demonstrates clear advantages. These findings suggest that effectiveness is conditional and should be evaluated with respect to specific process regimes.

For future research, three reproducible directions are outlined. First, establishing cross-fab evaluation protocols is necessary to systematically examine the generalization capability of the framework [27]. Second, defining interpretability-based stability metrics may provide a more transparent understanding of long-term robustness and failure modes. Third, implementing annual operational threshold reviews can better align system parameters with evolving production costs and reliability requirements. These steps will facilitate more rigorous benchmarking and ensure that experimental improvements translate into sustainable industrial deployment.

References

[1] Wang J Y, Tse K T, Li S W. Integrating the effects of climate change using representative concentration pathways into typhoon wind field in Hong Kong [C]//Proceedings of the 8th European African Conference on Wind Engineering. 2022: 20-23.

[2] Tao Y. SQBA: sequential query-based blackbox attack, Fifth International Conference on Artificial Intelligence and Computer Science (AICS 2023). SPIE, 2023, 12803: 721-729.

[3] Tao Y. Meta Learning Enabled Adversarial Defense, 2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE). IEEE, 2023: 1326-1330.

[4] Luo, M., Zhang, W., Song, T., Li, K., Zhu, H., Du, B., & Wen, H. (2021, January). Rebalancing expanding EV sharing systems with deep reinforcement learning. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence (pp. 1338-1344).

[5] Xiong, X., Zhang, X., Jiang, W., Liu, T., Liu, Y., & Liu, L. (2024). Lightweight dual-stream SAR-ATR framework based on an attention mechanism-guided heterogeneous graph network. IEEE Journal of Selected Topics in Applied

DOI: 10.54254/2755-2721/2025.28630

Earth Observations and Remote Sensing, 1–22.
[6] Li, K., Chen, X., Song, T., Zhou, C., Liu, Z., Zhang, Z., Guo, J., & Shan, Q. (2025a, March 24). Solving situation puzzles with large language model and external reformulation.
[7] APA Huang, J., & Qiu, Y. (2025). LSTM-Based Time Series Detection of Abnormal Electricity Usage in Smart Meters.
[8] Pang, F. (2020, November). Research on Incentive Mechanism of Teamwork Based on Unfairness Aversion Preference Model. In 2020 2nd International Conference on Economic Management and Model Engineering (ICEMME) (pp. 944-948). IEEE.
[9] Li, K., Chen, X., Song, T., Zhang, H., Zhang, W., & Shan, Q. (2024). GPTDrawer: Enhancing Visual Synthesis through ChatGPT. arXiv preprint arXiv: 2412.10429.
[10] Pang, F. (2025). Animal Spirit, Financial Shock and Business Cycle. European Journal of Business, Economics & Management, 1(2), 15-24.
[11] Luo, M., Du, B., Zhang, W., Song, T., Li, K., Zhu, H., ... & Wen, H. (2023). Fleet rebalancing for expanding shared e-Mobility systems: A multi-agent deep reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 24(4), 3868-3881.
[12] Huang, T., Cui, Z., Du, C., & Chiang, C. E. (2025). CL-ISR: A Contrastive Learning and Implicit Stance Reasoning Framework for Misleading Text Detection on Social Media. arXiv preprint arXiv: 2506.05107.
[13] Shih, K., Deng, Z., Chen, X., Zhang, Y., & Zhang, L. (2025, May). DST-GFN: A Dual-Stage Transformer Network with Gated Fusion for Pairwise User Preference Prediction in Dialogue Systems. In 2025 8th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (pp. 715-719). IEEE.
[14] Liu, Z. (2022, January 20–22). Stock volatility prediction using LightGBM based algorithm. In 2022 International Conference on Big Data, Information and Computer Network (BDICN) (pp. 283–286). IEEE.
[15] Cui, Z., Huang, T., Chiang, C. E. & Du, C. (2025). Toward Verifiable Misinformation Detection: A Multi-Tool LLM Agent Framework. arXiv preprint arXiv: 2508.03092.
[16] Yiyi Tao, Yiling Jia, Nan Wang, and Hongning Wang. 2019. The FacT: Taming Latent Factor Models for Explainability with Factorization Trees. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19). Association for Computing Machinery, New York, NY, USA, 295–304.
[17] Huang, T., Yi, J., Yu, P., & Xu, X. (2025). Unmasking digital falsehoods: A comparative analysis of llm-based misinformation detection strategies. arXiv preprint arXiv: 2503.00724.
[18] Li, X., Cao, H., Zhang, Z., Hu, J., Jin, Y., & Zhao, Z. (2024). Artistic Neural Style Transfer Algorithms with Activation Smoothing. arXiv preprint arXiv: 2411.08014. 3
[19] Yi, Q., He, Y., Wang, J., Song, X., Qian, S., Zhang, M., ... & Shi, T. (2025). SCORE: Story Coherence and Retrieval Enhancement for AI Narratives. arXiv preprint arXiv: 2503.23512.
[20] Wang J, Cao S, Tim K T, et al. A novel life-cycle analysis framework to assess the performances of tall buildings considering the climate change [J]. Engineering Structures, 2025, 323: 119258.
[21] Liu, Z. (2025). Human-AI Co-Creation: A Framework for Collaborative Design in Intelligent Systems. arXiv: 2507.17774.
[22] Yiyi Tao, Zhuoyue Wang, Hang Zhang, Lun Wang. 2024. NEVLP: Noise-Robust Framework for Efficient Vision-Language Pre-training. arXiv: 2409.09582.
[23] Li, X., Cao, H., Zhang, Z., Hu, J., Jin, Y., & Zhao, Z. (2024). Artistic Neural Style Transfer Algorithms with Activation Smoothing. arXiv preprint arXiv: 2411.08014. 3
[24] Li, X., Wang, X., Qi, Z., Cao, H., Zhang, Z., & Xiang, A. DTSGAN: Learning Dynamic Textures via Spatiotemporal Generative Adversarial Network. Academic Journal of Computing & Information Science, 7(10), 31-40. 3
[25] Zhang, L., Wang, L., Huang, Y., & Chen, H. (2019). Segmentation of Thoracic Organs at Risk in CT Images Combining Coarse and Fine Network. SegTHOR@ ISBI, 11(16), 2-4.
[26] Huang, T., Cui, Z., Du, C., & Chiang, C. E. (2025). CL-ISR: A Contrastive Learning and Implicit Stance Reasoning Framework for Misleading Text Detection on Social Media. arXiv preprint arXiv: 2506.05107.
[27] Wang, J., Chen, J., Shen, P. et al. Regional-scale intelligent optimization and topography impact in restoring global precipitation data gaps. Commun Earth Environ 6, 671 (2025).