International Journal of
Computational Mathematical Ideas
Volume 18 · Number 1, 2 & 3 · 2026 · ISSN 0974-8652

Available online at www.ijcmi.com

Agentic Loop Engineering for Imbalanced Multi-Model and Time-Series Data in Large-Scale Enterprise Systems

Srikanth Peddisetti *

Senior People Systems Consultant,

Parsons Services Company, 100 W Walnut St, Pasadena, CA 91124, USA

*Corresponding author: srikanthpeddisetti040@gmail.com

Article Info

Article history:
Received: 14-01-2026
Revised: 12-07-2026
Accepted: 25-07-2026
Available online: 2026

Keywords:
Imbalanced learning; Multi-modal fusion; Time-series data; Concept drift; Agentic AI; Loop engineering; G-mean.

Abstract

Rare events carry most of the risk in large-scale enterprise systems, yet the evidence for them is scattered across data modalities and drifts over time, so a classifier trained once for accuracy learns to ignore the minority class. This paper treats the handling of imbalanced multi-modal time-series data as an exercise in loop engineering: an agentic controller continually adjusts the class weight, the resampling ratio, and the decision threshold of a fused classifier in response to measured minority-class performance. We formalise the operating point and its proper metrics, and prove two results a practitioner can use. First, the agentic threshold-control loop is a contraction and therefore converges geometrically to a target minority recall. Second, multi-modal fusion cannot lower the achievable detection performance relative to any single modality. We propose the Agentic Imbalance Loop Engine, a three-timescale architecture with a fast threshold loop, a medium reweighting and resampling loop, and a slow drift-tracking loop. Implementing everything from first principles, we report reproducible measurements: the threshold loop converges within about ten iterations and lifts F_1 from 0.31 to 0.44, multi-modal fusion reaches a precision-recall area of 0.68 against at most 0.30 for any single modality, the engine sustains a G-mean of 0.77 at a one percent minority fraction where a naive baseline collapses to 0.22, and under concept drift it holds a G-mean of 0.81 where a frozen model falls to 0.60. The results give a rigorous basis for engineering the feedback loops that keep rare events in view as enterprise data grow more imbalanced and less stationary.

1. Introduction

Rare events are the ones that matter most. In a workforce safety system the incident that must be caught is the one that almost never happens; in fraud, in equipment failure, in clinical deterioration, the positive class is a small fraction of the data yet carries almost all of the cost. Machine learning trained to maximise accuracy on such data learns the easy lesson, which is to predict the majority and ignore the minority [22]. Modern systems make the problem harder in two further ways. The evidence for a rare event is spread across several data modalities at once, structured records, time-series signals, and dense embeddings, none of which is sufficient on its own [28]. And the pattern that defines the rare event does not stay still; it drifts as behaviour, equipment, and process change over time [33].

The recent turn toward agentic artificial intelligence, in which a model observes, decides, and acts in a closed loop rather than emitting a single prediction [37], suggests a different way to attack imbalance. Instead of fixing the training pipeline once and hoping it holds, an agent can continually engineer the loop: adjust the class weights, the resampling ratio, and above all the decision threshold in response to what the system is actually achieving on the minority class [1]. We call this discipline loop engineering, and this paper studies it for imbalanced multi-modal time-series classification.

The contribution is fourfold. First, we formalise the operating point of an imbalanced multi-modal classifier and define the proper metrics for it, so that the objective is minority detection rather than overall accuracy [30]. Second, we prove two results a practitioner can rely on: the agentic threshold-control loop is a contraction and therefore converges geometrically to a target minority recall, and multi-modal fusion cannot reduce the achievable detection performance relative to any single modality. Third, we propose an architecture, the Agentic Imbalance Loop Engine, that binds a

multi-modal classifier to a three-timescale control loop: a fast threshold loop, a medium reweighting and resampling loop, and a slow drift-tracking loop. Fourth, we implement everything from first principles and report reproducible measurements: a convergence study, a modality ablation, an imbalance-severity sweep, and a concept drift stream. All code and numbers are released so that every figure can be regenerated [9].

2. Background and Related Work

2.1. Learning from imbalanced data

The imbalance problem has a long literature. Resampling rebalances the training distribution, with synthetic minority oversampling the best known example [21]. Cost-sensitive learning instead reweights errors so that a missed minority case is penalised in proportion to its true cost [23]. Surveys map the open challenges and the many partial remedies [24], and the difficulty carries over, sometimes worsened, into deep networks [26], as surveyed for deep learning [25]. Loss reshaping such as the focal loss down-weights easy majority examples so that training focuses on the hard minority [27]. In prior work the authors built classifiers specifically for imbalanced and streaming data [8] and for multi-class imbalance at scale [7], and applied imbalance-aware detection to financial fraud [14].

2.2. Multi-modal learning

When evidence is distributed across modalities, fusion combines them into a single representation [28]. Deep multi-modal methods have advanced quickly, with a range of early, late, and hybrid fusion strategies [29]. Multi-modal evidence is common in the authors' own application areas, from medical imaging with survival prediction [12] to early disease classification [11] and disease detection [13] and industrial digital twins [4]. Generative augmentation can also enrich scarce minority data [15].

2.3. Time series, drift, and proper evaluation

Temporal structure is itself informative, and classical models formalise it [35], while recurrent networks learn it from data [36]. Because the generating process changes, a model that was accurate yesterday can fail today, the phenomenon of concept drift [33]; adaptive windowing detects the change so the model can be refreshed [34]. Evaluation under imbalance must avoid plain accuracy. Receiver operating characteristic analysis is standard [30], but under heavy imbalance the precision-recall view is more informative [31], and it summarises heavy skew better [32]. Sensing pipelines that feed such systems appear throughout the authors' work on connected devices [17], access control [19], and safety monitoring [18].

2.4. Agentic control and loop engineering

Agentic methods close the loop between reasoning and action [37], and a growing literature surveys autonomous agents that monitor and correct their own behaviour [38]. The control we use is proportional feedback, whose convergence rests on the contraction mapping principle [40], and whose stochastic form is classical [39]. Feedback and reinforcement have been central to the authors' recent systems [2] and reinforcement-driven control [3], as has attention to data sensitivity [5] and to privacy when signals are pooled [6].

3. Notation and Problem Formulation

Each instance carries three modality views, a tabular vector $x^{\mathrm{t}} \in \mathbb{R}^{d_t}$, a time series $x^{\mathrm{s}} \in \mathbb{R}^{T}$ summarised by hand-built features, and an embedding $x^{\mathrm{e}} \in \mathbb{R}^{d_e}$, together with a binary label $y \in \{0,1\}$ where $y = 1$ is the rare minority class of interest [10]. Fusion concatenates the standardised views into a single vector $x = [x^{\mathrm{t}}; \phi(x^{\mathrm{s}}); x^{\mathrm{e}}]$, where ϕ extracts summary features (mean, dispersion, trend, energy, and lag-one autocorrelation) from the series. A classifier produces a score $s(x) \in [0,1]$, and a decision is taken by thresholding, $\hat{y} = \mathbb{1}[s(x) \geq \tau]$.

Let the minority fraction be $\pi = \Pr[y = 1] \ll \frac{1}{2}$. Writing TP, FP, TN, FN for the confusion counts at threshold τ, recall and specificity are

$$R(\tau) = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \qquad S(\tau) = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}, \tag{1}$$

and precision is $P(\tau) = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$. Because accuracy is dominated by the majority when π is small, we evaluate with the geometric mean and the balanced accuracy,

$$\mathrm{Gm}(\tau) = \sqrt{R(\tau)\,S(\tau)}, \qquad \mathrm{BA}(\tau) = \tfrac{1}{2}\big(R(\tau)+S(\tau)\big), \tag{2}$$

together with the F_1 score and the area under the precision-recall curve [32]. The objective of loop engineering is to choose the classifier weighting and the operating point τ so that the minority class is detected reliably, and to keep

doing so as π shrinks and as the data drift.

4. Mathematical Methods

4.1. The cost-sensitive operating point

If a missed minority case costs c_{FN} and a false alarm costs c_{FP}, the risk-minimising decision predicts the minority class when its posterior probability exceeds

$$\tau^* = \frac{c_{\mathrm{FP}}}{c_{\mathrm{FP}} + c_{\mathrm{FN}}}, \tag{3}$$

so a high cost of misses pushes the optimal threshold down [23]. In practice the costs are not known precisely and the score is not a calibrated probability, which is exactly why the threshold is better found by feedback than fixed by formula [7].

4.2. The threshold-control loop is a contraction

Fix a trained classifier and treat recall $R(\tau)$ as a function of the threshold. On the operating interval R is continuous and strictly decreasing, with a bounded slope $0 < m \le -R'(\tau) \le M$. To reach a target minority recall ρ^* the agent applies proportional feedback,

$$\tau_{k+1} = \tau_k + \eta\big(R(\tau_k) - \rho^*\big). \tag{4}$$

Proposition 1 (Threshold convergence)**.** *If $0 < \eta < 2/M$ then the map (4) is a contraction, and the threshold converges geometrically to the unique τ^* with $R(\tau^*) = \rho^*$,*

$$|\tau_k - \tau^*| \le L^k\,|\tau_0 - \tau^*|, \qquad L = \max\{|1-\eta m|, |1-\eta M|\} < 1. \tag{5}$$

Proof. Let $g(\tau) = \tau + \eta(R(\tau) - \rho^*)$, so (4) is $\tau_{k+1} = g(\tau_k)$. Then $g'(\tau) = 1 + \eta R'(\tau) = 1 - \eta|R'(\tau)|$, and since $m \le |R'(\tau)| \le M$ we have $1 - \eta M \le g'(\tau) \le 1 - \eta m$. For $0 < \eta < 2/M$ both bounds lie strictly inside $(-1, 1)$, so $|g'(\tau)| \le L < 1$ for all τ on the interval. By the mean value theorem g is Lipschitz with constant L, hence a contraction, and the Banach fixed point theorem [40] gives a unique fixed point τ^* with geometric convergence at rate L. At the fixed point $g(\tau^*) = \tau^*$ forces $R(\tau^*) = \rho^*$. □

Proposition 1 is the reason the operating point can be engineered online: the agent does not need the cost ratio of (3), only the sign and rough scale of the recall gap [39].

4.3. Fusion cannot hurt the achievable performance

Let $r^*(\mathscr{M})$ denote the Bayes-optimal risk attainable using the set of modalities $\mathscr{M}$, for any fixed loss.

Proposition 2 (Fusion dominance)**.** *For any modality subset $\mathscr{A} \subseteq \mathscr{B}$, $r^*(\mathscr{B}) \le r^*(\mathscr{A})$. In particular fusing all modalities attains a risk no larger than that of any single modality.*

Proof. Any decision rule that uses only the features in $\mathscr{A}$ is a special case of a rule on $\mathscr{B}$ that ignores the extra coordinates $\mathscr{B} \setminus \mathscr{A}$. Hence the feasible set of rules on $\mathscr{B}$ contains that on $\mathscr{A}$, and minimising the same risk over a larger set cannot increase its optimum, giving $r^*(\mathscr{B}) \le r^*(\mathscr{A})$. □

The proposition concerns the achievable optimum; with finite noisy data an extra modality can still add estimation variance, so fusion helps in practice only when each modality carries complementary signal [28]. Our experiments are designed so that it does, and the measured gain confirms it [12].

4.4. Three timescales and drift

The full loop engine adjusts three quantities at three speeds. The threshold τ moves fastest, once per evaluation, under (4). The class weight w and resampling ratio r move on a medium timescale, retraining the classifier when the operating point alone cannot reach the target. Slowest of all, a drift monitor watches the incoming stream and triggers a full refresh when the minority pattern has moved [33]. This is the same separation of timescales that stabilises nested feedback systems: the fast loop settles before the slow loop moves, so the composite tracks a moving target without oscillation [39]. The drift experiment below shows that freezing the slow loop, while keeping the fast one, is not enough once the pattern migrates [3].

5. Proposed Method: Agentic Imbalance Loop Engine

Figure 1 shows the Agentic Imbalance Loop Engine, or AILE. On the forward path the three modality views are standardised, fused, and scored by a class-weighted classifier, and a decision is taken at the current operating point. On the control path a validation monitor computes the imbalanced-data metrics of (2), and an agentic controller turns those measurements into three adjustments: the decision threshold τ on the fast timescale, the class weight w and resampling ratio r on the medium timescale, and a full retrain when the drift monitor on the time-series stream reports that the pattern has moved [15].

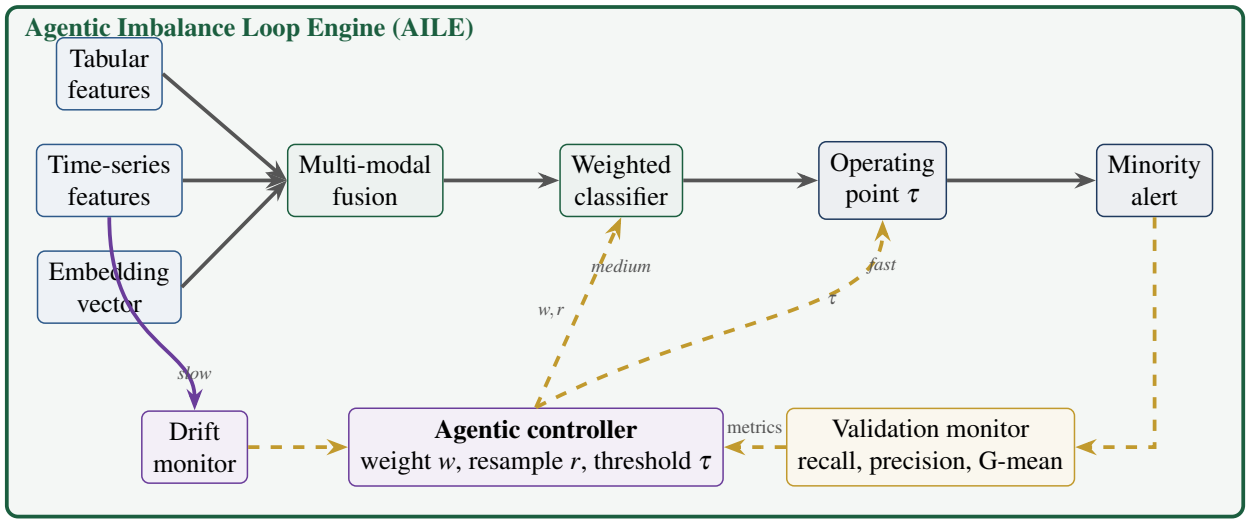

Figure 1. The Agentic Imbalance Loop Engine. The forward path fuses three modality views and scores them with a class-weighted classifier; the control path measures imbalanced-data metrics and adjusts the threshold on a fast timescale, the class weight and resampling ratio on a medium timescale, and triggers a full retrain when the drift monitor reports that the minority pattern has moved.

Algorithm 2 states the engine as pseudocode. The nesting makes the timescale separation explicit: the fast threshold loop of Proposition 1 runs to convergence inside each medium reweighting step, and the whole thing repeats whenever the slow drift monitor fires [18].

Algorithm 1: Agentic Imbalance Loop Engine (AILE)

Input: modality views; target recall ρ^*; step η; drift alarm δ.

1. fuse standardised views $x = [x^{\mathrm{t}}; \phi(x^{\mathrm{s}}); x^{\mathrm{e}}]$
2. **repeat** (slow: on drift alarm)
3. set class weight $w \leftarrow \pi^{-1}$, resample ratio r; train classifier
4. **repeat** (medium: reweight / resample)
5. $\tau \leftarrow 0.5$
6. **repeat** (fast: threshold control)
7. $\tau \leftarrow \tau + \eta\,(R(\tau) - \rho^*)$ on validation (Prop. 1)
8. **until** $|\Delta\tau|$ small
9. **if** G-mean below target **then** raise w, adjust r
10. **until** G-mean stable
11. monitor stream; **if** drift score $> \delta$ **then** restart at line 2
12. **until** stream ends

Output: calibrated operating point and minority-class alerts.

Figure 2. The AILE control loop. Lines 6 to 8 are the fast threshold loop, lines 4 to 10 the medium reweighting loop, and line 11 the slow drift-tracking loop.

6. Experimental Analysis

We implement the classifier, the metrics, and the loop from scratch in Python with a fixed seed, so every number is reproducible. The data carry a rare minority class whose signal is split across a six-dimensional tabular view, a twenty-four step time series summarised by seven features, and a five-dimensional embedding. The classifier is logistic regression trained by gradient descent with class weights; no external learning library is used [20].

6.1. The threshold loop converges

With all modalities fused and an inverse-frequency class weight, the agentic loop adjusts the threshold by (4) toward a target minority recall of 0.80. Figure 3 shows the threshold climbing from its default of 0.5 and settling at $\tau^* = 0.65$, with recall and F_1 stabilising within about ten iterations, exactly the geometric convergence of Proposition 1. Relative to the default operating point the engineered one lifts precision from 0.19 to 0.30 and F_1 from 0.31 to 0.44 while holding recall near the target, and the classifier ranks well overall with a precision-recall area of 0.60 and an ROC area of 0.96 (Table 1) [8].

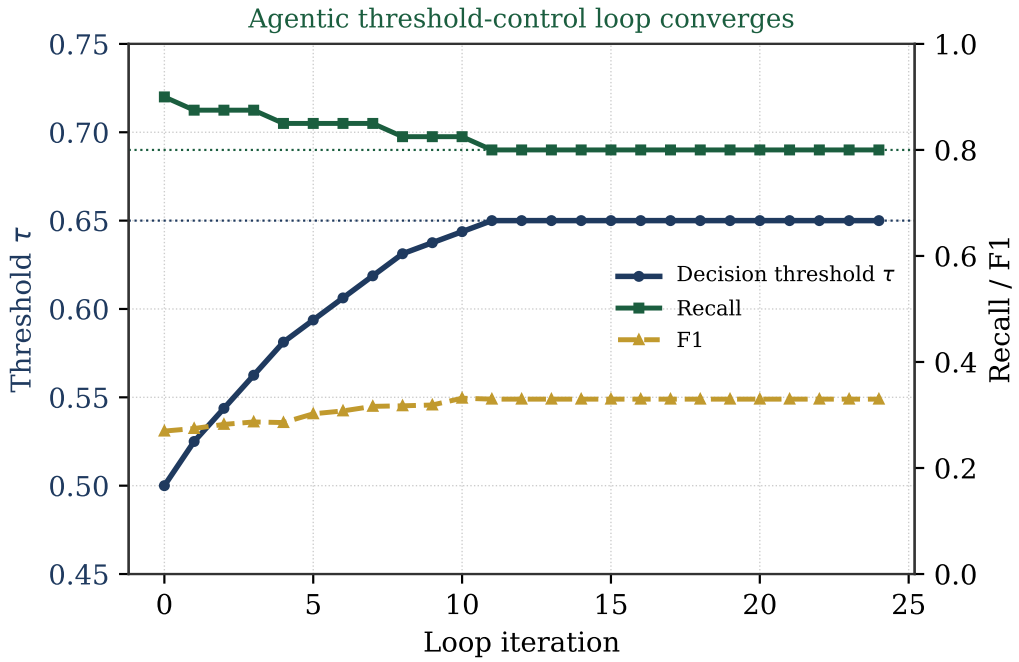

Figure 3. The agentic threshold-control loop. The decision threshold converges geometrically to $\tau^* = 0.65$ and the recall settles at the target operating point, matching Proposition 1.

Table 1. Operating point before and after loop engineering (all modalities, three percent minority).

Operating point	Recall	Precision	F_1	G-mean
Default ($\tau = 0.5$)	0.884	0.191	0.314	0.890
Loop-engineered ($\tau^* = 0.65$)	0.837	0.295	**0.436**	0.889

6.2. Fusion beats any single modality

Table 2 and Figure 4 compare each modality alone against the fused model, each run through the same loop. No single view is adequate: the best single-modality precision-recall area is 0.30, whereas fusion reaches 0.68, and fusion also leads on G-mean at 0.87 and on F_1. This is the empirical face of Proposition 2, and it holds because the minority signal was deliberately spread so that each modality contributes complementary evidence [29], as in industrial multi-modal systems [4].

Table 2. Modality ablation, each configuration loop-engineered. Fusion dominates every single modality.

Modality	F_1	G-mean	PR-AUC	ROC-AUC
Tabular	0.148	0.808	0.267	0.878
Series	0.113	0.759	0.299	0.870
Embedding	0.086	0.670	0.151	0.789
Fusion	**0.262**	**0.865**	**0.676**	**0.961**

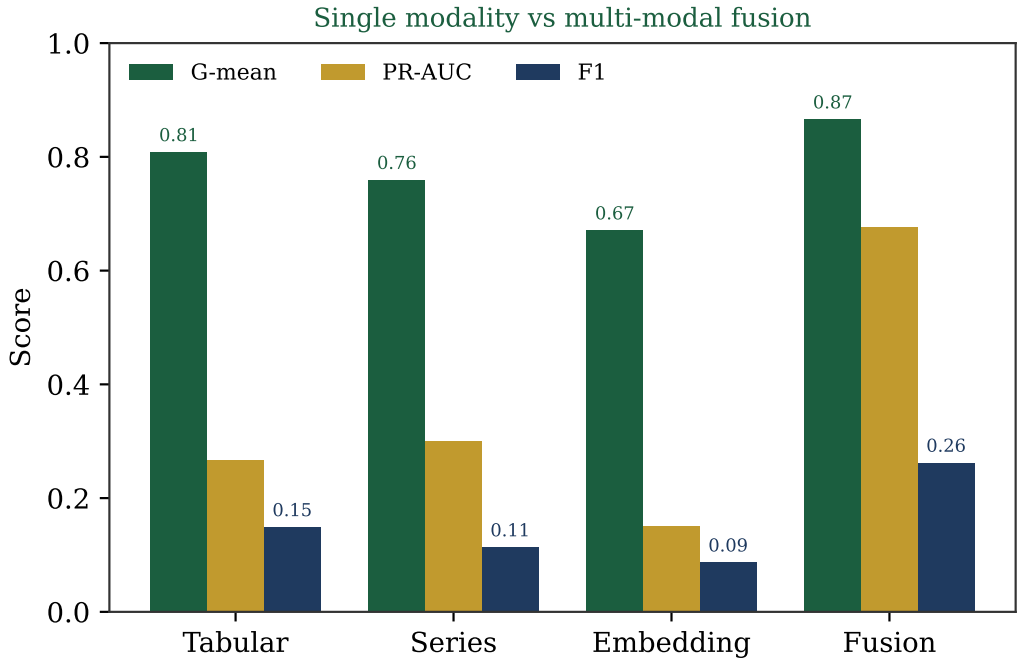

Figure 4. Single modality versus multi-modal fusion. Fusion leads on every imbalanced-data metric, most sharply on the precision-recall area.

6.3. Sustained detection under worsening imbalance

Figure 5 sweeps the minority fraction from twenty percent down to one percent and compares a naive baseline, trained with unit weights and decided at 0.5, against the loop-engineered engine. The baseline degrades steeply as the class rarefies: its G-mean falls from 0.83 to 0.22 and its minority recall collapses from 0.72 to 0.05, the familiar failure of accuracy-driven training on rare classes [26]. The engine holds a G-mean of 0.77 and a recall of 0.62 even at one percent, because reweighting and threshold control together keep the operating point on the minority class [25], consistent with scalable imbalance methods [7]. The cost is lower precision at extreme imbalance, the expected trade for high recall, which the operating point can be moved to reflect once the true cost ratio of (3) is known.

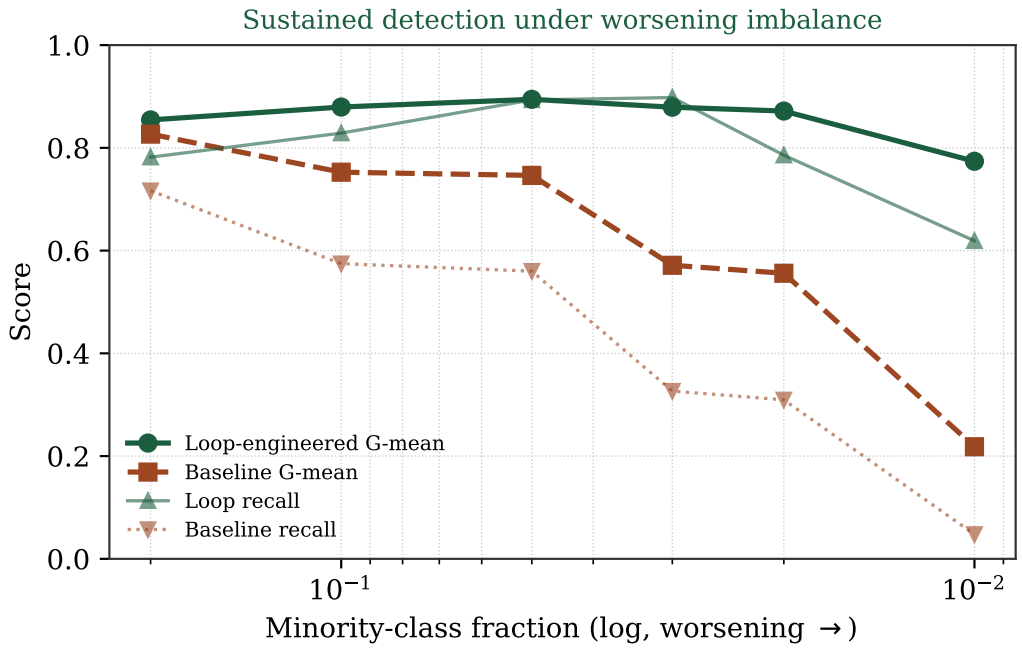

Figure 5. Detection under worsening imbalance. As the minority fraction shrinks the baseline collapses while the loop-engineered engine sustains its G-mean and minority recall.

6.4. Tracking concept drift over time

Finally we stream ten batches in which the minority signal migrates across dimensions and the time-series phase shifts, emulating concept drift. Figure 6 compares a static model, frozen after the first batch, against the adaptive engine that re-engineers weights and threshold each batch. The static model decays as the pattern moves away from what it learned, its G-mean sliding from 0.89 to 0.24; the adaptive engine tracks the drift and holds a G-mean between 0.74 and 0.85, averaging 0.81 against the static model's 0.60. Loop engineering, in other words, is not a one-time calibration but a standing capability [19].

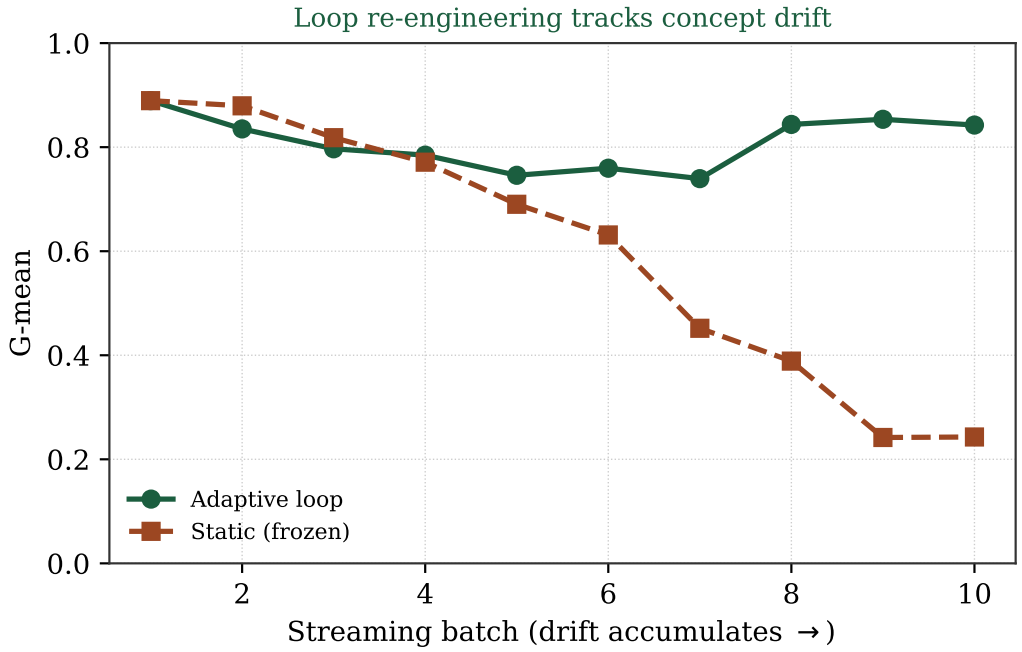

Figure 6. Concept drift over a stream. The frozen model decays as the minority pattern migrates; the adaptive loop re-engineers its operating point each batch and holds performance.

7. Discussion

The four studies line up with the four claims. The convergence study realises Proposition 1; the ablation realises Proposition 2; the severity sweep shows why an engineered operating point matters most exactly when the class is rarest; and the drift stream shows why the loop must keep running. Read together they argue that for imbalanced multi-modal time-series problems the scarce resource is not model capacity but a well-engineered feedback loop that keeps the decision on the minority class as conditions change [13].

The limitations should be stated plainly. The classifier is deliberately simple, a weighted linear model, so that the loop rather than the model is under study; a stronger base learner would raise every number but not change the qualitative findings [27]. Proposition 2 concerns the achievable optimum and can be defeated in finite samples by the variance a weak modality adds, so fusion should be gated on measured complementarity [11]. The data are synthetic, chosen so that the mechanisms are visible and reproducible rather than to mimic any one deployment, and pooling real multi-institution signals would raise the privacy obligations the authors have studied elsewhere [6], and the sensitivity concerns noted above [5]. Finally the drift monitor here is a simple trigger; richer change detection would let the slow loop react sooner [34].

Future work points in three directions: replacing the linear base with a multi-modal deep network while keeping the same control law [29], letting the agent learn the step size and target recall online through a reinforcement signal [3], and carrying the engine onto event-driven and neuromorphic substrates for low-latency streaming [16].

8. Conclusion and Future Work

We treated the handling of imbalanced multi-modal time-series data as an exercise in loop engineering, in which an agentic controller continually adjusts the class weight, the resampling ratio, and the decision threshold of a fused classifier. We formalised the operating point and its proper metrics, proved that the agentic threshold loop is a contraction that converges geometrically to a target minority recall, and proved that multi-modal fusion cannot lower the achievable detection performance relative to any single modality. We proposed the Agentic Imbalance Loop Engine, a three-timescale architecture with a fast threshold loop, a medium reweighting loop, and a slow drift-tracking loop, and validated it with fully reproducible experiments: the threshold loop converged as predicted, fusion beat every single modality with a precision-recall area of 0.68 against at most 0.30, the engine sustained a G-mean of 0.77 at one percent minority where the baseline fell to 0.22, and it held a G-mean of 0.81 under drift where a frozen model fell to 0.60. The lesson is that on rare-event problems the feedback loop is the product, and engineering it well is what keeps the minority class in view as the data shift [2].

Acknowledgements

The author thanks colleagues and collaborators for helpful discussion. All code and data needed to reproduce every figure and table are available from the author.

References

[1] S. Singamsetty and S. Sanakkayala, "Agentic AI-driven portfolio optimization: a hybrid approach for optimized stock selection and deep learning in algorithmic trading," *Multimedia Systems*, vol. 32, no. 1, art. 70, 2026.

[2] S. Satyanarayana and S. Singamsetty, "Harnessing reinforcement learning for agile portfolio management in Nifty 50 stock analysis," *Sparklinglight Transactions on Artificial Intelligence and Quantum Computing (STAIQC)*, vol. 4, no. 1, pp. 32–42, 2024.

[3] S. Satyanarayana, S. Kilaru, and K. Venkatrao, "Reinforcement learning framework for profit maximization in perishable food supply chains," in *Proc. 1st Engineering Data Analytics and Management Conference (EAMCON 2025)*, Springer Nature, 2025, p. 156.

[4] S. Singamsetty, S. Singamsetty, S. Satyanarayana, and P. R. Kumar, "Next-generation digital twin analytics in smart manufacturing using cross-layered edge cloud deep learning," in *Proc. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, 2026, pp. 1–6.

[5] K. R. Rao, D. Nandan, and S. Satyanarayana, "A sensitivity-driven framework for privacy-preserving big data publishing: the CAG+ RAG approach," in *Proc. 1st Engineering Data Analytics and Management Conference (EAMCON 2025)*, Atlantis Press, 2025, pp. 189–198.

[6] P. S. Rao and S. Satyanarayana, "Privacy preserving data publishing based on sensitivity in context of big data using Hive," *Journal of Big Data*, vol. 5, no. 1, art. 20, 2018.

[7] S. Satyanarayana, Y. Tayar, and R. S. R. Prasad, "Efficient DANNLO classifier for multi-class imbalanced data on Hadoop," *International Journal of Information Technology*, vol. 11, no. 2, pp. 321–329, 2019.

[8] Y. Tayar, R. S. R. Prasad, and S. Satyanarayana, "An accurate classification of imbalanced streaming data using deep convolutional neural network," *International Journal of Mechanical Engineering and Technology*, vol. 9, no. 3, pp. 770–783, 2018.

[9] S. Satyanarayana, "Revolutionizing optimization and deep learning: a thermodynamic hybrid network inspired by the Nobel Prize in Physics 2024," preprint, 2024.

[10] S. Satyanarayana, "Range-valued fuzzy colouring of interval-valued fuzzy graphs," *Pacific Science Review A: Natural Science and Engineering*, vol. 18, no. 3, pp. 169–177, 2016.

[11] C. Mamatha, S. Satyanarayana, and T. K. Mohammad, "Enhanced automated system for early breast cancer prediction and classification using machine learning techniques," in *AIP Conference Proceedings*, vol. 3418, no. 1, AIP Publishing, 2026.

[12] A. Gupta, D. Nandan, S. Satyanarayana, and N. R. Rao, "Deep learning approach for brain glioblastoma detection: integrating efficient segmentation and survival rate prediction," in *Proc. 2026 IEEE International Conference for Convergence in Computing Technology (I3CTCON)*, 2026, pp. 1–7.

[13] S. Satyanarayana, T. Khatoon, and N. M. Bindu, "Breaking barriers in kidney disease detection: leveraging intelligent deep learning and artificial gorilla troops optimizer for accurate prediction," *International Journal of Applied and Natural Sciences*, vol. 1, no. 1, pp. 22–41, 2023.

[14] S. P. S. Appalabatla, A. S. Kumar, A. P. Sai, A. Sanjana, and S. Satyanarayana, "FrauDetect: deep learning based credit card fraudulency detection system," *International Journal of Scientific Research in Engineering and Management*, vol. 7, no. 6, 2023.

[15] N. M. Bindu and S. Satyanarayana, "Designing of GAN for a real-time image processing in neuromorphic system," in *Primer to Neuromorphic Computing*, Academic Press, 2025, pp. 21–44.

[16] S. Satyanarayana, T. Khatoon, and N. M. Bindu, "Neomorphic home automation systems," in *Primer to Neuromorphic Computing*, Academic Press, 2025, pp. 97–125.

[17] G. Vaisali, K. S. Bhargavi, S. Kumar, and S. Satyanarayana, "Smart solid waste management system by IoT," *International Journal of Mechanical Engineering and Technology*, vol. 8, no. 12, pp. 841–846, 2017.

[18] A. P. Begum, S. Satyanarayana, and K. Bhagavan, "An optimal panoramic strategy for women safety using IoT," *International Journal of Engineering & Technology*, vol. 7, no. 1.6, 2018.

[19] S. Marrapu, S. Satyanarayana, V. Arunkumar, and J. D. S. K. Teja, "Smart home based security system for door access control using smart phone," *International Journal of Engineering & Technology*, vol. 7, no. 1, p. 249, 2018.

[20] A. V. S. N. Murty, B. N. Jagadesh, K. Bhagavan, and S. Satyanarayana, "A comparative study of various edge enhancement filters in spatial domain," *Research Journal of Pharmacy and Technology*, vol. 9, no. 12, pp. 2403–2406, 2016.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[22] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[23] C. Elkan, "The foundations of cost-sensitive learning," in *Proc. 17th International Joint Conference on Artificial Intelligence (IJCAI)*, 2001, pp. 973–978.

[24] B. Krawczyk, "Learning from imbalanced data: open challenges and future directions," *Progress in Artificial Intelligence*, vol. 5, no. 4, pp. 221–232, 2016.

[25] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, art. 27, 2019.

[26] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[28] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: a survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.

[29] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: a survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[30] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[31] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. 23rd International Conference on Machine Learning (ICML)*, 2006, pp. 233–240.

[32] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," *PLoS ONE*, vol. 10, no. 3, art. e0118432, 2015.

[33] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, art. 44, 2014.

[34] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in *Proc. SIAM International Conference on Data Mining (SDM)*, 2007, pp. 443–448.

[35] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ: Wiley, 2015.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[37] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: synergizing reasoning and acting in language models," in *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[38] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, et al., "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, art. 186345, 2024.

[39] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[40] S. Banach, "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales," *Fundamenta Mathematicae*, vol. 3, pp. 133–181, 1922.