Invisible Feedback Loops: Detecting Passive Bias in User-Facing ML Models

Kapil Kumar Goyal

Alumnus, MBA (Strategy and Negotiation), University of California, Irvine (UCI), Irvine, USA

Alumnus, B.Tech in Computer Science and Engineering, Amity University, Noida, India

Publication Date: 2025/05/27

Abstract: Machine Learning (ML) models integrated into user-facing systems are extremely well-regarded for their ability to automate and personalize experiences. But lying beneath the surface is a nefarious problem: the growth of silent feedback loops. These loops, formed when model outputs quietly influence user behavior, can in turn perpetuate existing model assumptions, leading to passive bias over time. In this paper, we propose an end-to-end system to detect, analyze, and mitigate passive bias due to such feedback loops. We introduce a feedback-aware monitoring system architecture, describe real-world application scenarios, and provide empirical methods to quantify bias propagation. Our approach highlights the performance and ethical consequences of neglecting latent model feedback and suggests deployment guidelines for responsible deployment.

Keywords: *Feedback Loops, Machine Learning, Passive Bias, Responsible AI, User Interaction, Model Drift, Bias Detection, Recommender Systems*

How to Cite: Kapil Kumar Goyal (2025) Invisible Feedback Loops: Detecting Passive Bias in User-Facing ML Models. *International Journal of Innovative Science and Research Technology*, 10(5), 1934-1938.
https://doi.org/10.38124/ijisrt/25may1549

I. INTRODUCTION

Machine learning algorithms are ubiquitous on digital platforms, especially in relation to customized systems such as recommendation systems, virtual assistants, job recruitment tools, and predictive text completion. These models condition user behavior, which consequently gives rise to new information that gets fed back into the model. This bidirectional influence often creates feedback loops where the model's predictions shape user behavior, which then influences subsequent model training.

Although positive feedback loop can optimize user experience, it also unintentionally serves to support existing biases, limit diversity, and disrupt fairness. The process — called "invisible feedback loops" — quietly continues in the background, passively distorting future predictions. An example is a suggestion model that constantly suggests popular content, which discourages users from experimenting with niche content, leading to homogenization of tastes.

This work investigates the issue of hidden feedback loops and their effects in user-oriented ML applications. It presents a framework for identifying and measuring passive bias and suggests a feedback-monitoring system that can be integrated into ML pipelines. Our goal is to render such feedback patterns visible, measurable, and eventually controllable.

II. PROBLEM DEFINITION

Undetectable feedback loops occur when a machine learning model affects the decisions that users make. Those decisions, in turn, become part of the training data for the next version of the model. It's especially worrisome when the system in question is continuously retraining based on user interactions, such as click-through rates, engagement scores, or conversions.

There are quite a few harmful ways in which these loops manifest:

- Accumulation of Bias in the Model: The initial model assumptions may not be accurate and may reflect not-so-desirable user behavior, thereby reinforcing not-so-desirable behavior in the model, too. And this accumulating bias tends not to be kind. It skews undesirable patterns in the model, and it is not right. It also increases the number of bad instances that happen with underperforming models.
- Content Diversity Loss: Recommender systems might tend to give excessive promotion to content that matches past trends, and in the process, they might end up underpromoting options that happen to be new or less represented.
- Limited Model Generalization: Feedback loops can restrict the patterns a model learns. Overfit behavior mirrors the kind of behavior you'd expect when a model learns from feedback loops. We want our models to have an ability to

generalize, which means they've learned real insights instead of just the details of the datasets they've seen.

- Bias Obfuscation: Feedback loops work on an entity by parting it from the surface. They act in most cases without our even knowing it. They are therefore very hard to see. And because most of us prefer to think in terms of individual decision-making, when we don't see bias in the data, we assume it isn't there, even though it might be present in the feedback loops.

Grasping and interrupting this loop necessitates tools and measurement of the ways in which user behavior changes in response to model outputs. This is not yet a fully explored territory in ML system design.

In recommendation systems, how users click on recommendations is often taken to indicate what their interests are. But how users click for recommendations also depends on where the recommendation itself is placed. Voice assistants adjust to user corrections; however, they may infer incorrect pronunciation standards if a user does not correct the voice assistant's mistakes.

Policing tools that forecast where crime will occur next may be misrepresenting crime patterns because of historical bias in law enforcement. They are trained on arrest data, and if that data reflects a certain bias, then so too will the predictive tools.

- *Effects:*

- Loss of model generality: The model overfits to biased interaction histories.
- Echo chambers of algorithms: Systems that recommend content make the already-consumed content even more pronounced.
- Diminished originality and variety: The systems do not offer unexamined or less common viewpoints.

Without any malicious intent, these issues come up, which makes it harder to detect and justify passive bias to stakeholders.

III. RELATED WORK

Work done previously in the ML fairness domain and concerning feedback effects encompasses studies that look at various kinds of feedback effects. For instance, some studies focus specifically on bias amplification in recommender systems [1].

Other work looks at the exploration-exploitation tradeoffs in reinforcement learning [2] and considers the implications of those tradeoffs for fairness. Still other studies concern themselves with the social impacts of echo chambers in social media algorithms [3] and hold those algorithms accountable for their effects.

Meanwhile, responsible AI initiatives like Fairness, Accountability, and Transparency in Machine Learning (FAT-ML) [4, 5] push the conversation even further and indeed closer toward an ideal outcome. Amershi et al. [4] and Holstein et al. [5] have called for human-in-the-loop and feedback-aware ML development practices.

The unique contribution of this paper is a systematized approach to monitor and analyze feedback loops post-deployment using contextual behavior data.

IV. PROPOSED FRAMEWORK: FEEDBACK LOOP DETECTION SYSTEM (FLDS)

The Feedback Loop Disentanglement Engine (FLDE) framework has been put forward to expose, dissect, and reduce the power of passive feedback loops in ML systems. It has a natural modularity and layer-wise design that lets it dovetail into not only the model evaluation pipeline but also the post-deployment monitoring of applications like search engines, recommendation systems, hiring algorithms, etc.

- *Layer of Observation*

- Logging of Input: Tracks continuously user interaction data such as clicks, skips, ratings, dwell time, and more
- Context of Environment: Captures in real-time metadata like device, location, time of day, and user demographic information (when available and collected ethically)
- Traceability of Feature: Captures how features change over time and evolve with user interaction and model outputs.

- *Bias Signal Detection Layer*

- Drift Detection: This tracks the distribution of user behavior and looks for changes that can't be attributed to normal variation in the data. When such changes are detected, they are further examined to see if they are harmless or may lead to bias in model predictions.
- Reinforcement Signature Analysis: The actions taken by users in response to a model's prediction can be used to retrain the model (i.e., to reinforce the model's structure). When such actions form certain patterns that could lead to future bias, the model (or the Bias Signal Detection Layer) flags these patterns for human review.
- User Diversity Index (UDI): Just as the actions taken by users can be predicted by the model, so too can the types of users that the model engages. If the model is mainly engaging one type of user (subgroup), then it might be a sign that the model is becoming biased.

- Loop Attribution Engine

- Causal Inference Module: Estimates the causal impact of model output on user behavior by using techniques that include difference-in-differences, counterfactuals, or instrumental variables.
- Temporal Decay Modeling: Isolates first-time versus repetitive interactions by assigning time-decayed weights to feedback.
- Interaction Graph Analyzer: Constructs a bipartite graph of users and model outputs over time to find concentrated feedback zones.

- *Bias Mitigation Layer*

- Prompt Diversification Engine: Injects deliberate variety or exploration into outputs (e.g., multi-armed bandit style).

- Debiasing Filters: Reweights incoming feedback during training to prevent overfitting to historical artifacts.
- User Experience Balancer: Adjusts exposure frequency and tail-content surfacing based on fairness objectives.

➢ Governance of Feedback Dashboards

- Transparency Reports: Dashboards that visually explain the loop risks, the degree of bias, and the audit trails of the patterns that were detected.
- Human-in-the-Loop Annotations: The ability for reviewers of the system to flag suspected passive loops for further investigation
- Alerts and Guardrails: The system can suggest to practitioners that certain emerging metrics might mean that that specific component is starting to become biased.

FLDE is designed to operate in a time-bound fashion. Its modules work together in a way that can be made to seem near-real-time depending on the application at hand, the application's needs, and certain adjustable tolerances. In terms of governance, using FLDE to observe feedback loops results in a more understandable model. It makes the appearance of the loop, and the risk associated with it a visible and understandable phenomenon.

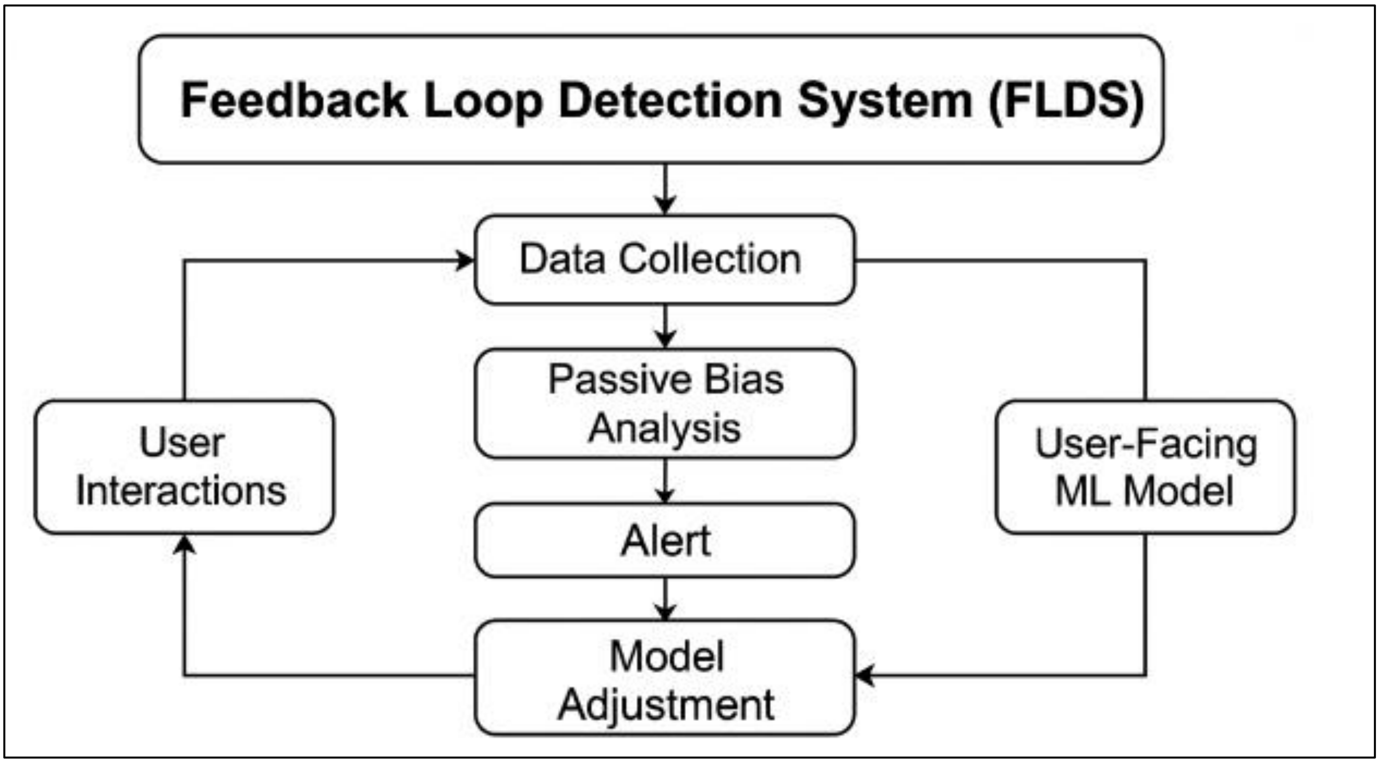

Fig 1 Feedback Loop Detection System (FLDS)

V. USE CASE 1: E-LEARNING RECOMMENDATION SYSTEM

➢ Context: An edtech platform employs machine learning to recommend the next courses for students based on their behaviors and performances.

➢ Feedback Loop Scenario: The platform accumulates signals favoring lower-complexity courses that students strongly prefer to click on and enroll in courses that are easier for them, and hence, in the eyes of the ML model, slightly more likely to be recommended by the model.

➢ Outcome: The model is now basing its recommendations on features that are flagged because of an unfortunate overfitting situation, which is also correlated to the now less safe (but still somewhat safe) space of showing students courses that are very much in their zone of proximal development.

➢ FLDE Application: By simulating what students would choose under different conditions and applying causal tracing, the platform identified that course difficulty was unduly weighted in the ranking model. The engine helped reweight features to de-bias course exposure.

VI. USE CASE 2: RESUME SCREENING IN RECRUITMENT

➢ Context: A talent acquisition team employs ML to conduct pre-screens of resumes for engineering roles.

➢ Feedback Loop Scenario: The model learns to favor resumes from universities or with past employers who are also favored by the model. As it becomes more and more skewed to this kind of profile, it appears to be improving based on reinforcement from human reviewers, after all, it is picking resumes that the real humans reviewing them are often picking, too.

➢ Outcome: The model gradually reinforces selective patterns, privileging profiles associated with narrow bands of socioeconomic status, diverse talent pools with unconventional backgrounds, or any number of other kinds of underrepresented groups that aren't underrepresented by virtue of being unconventional.

- FLDE Application: Here again, the counterfactual simulator shows just how much signal is in the non-inverted feature. The causality estimator also detects strong model influence on recruiter choices. Both are very good signs for the use of FLDE.

VII. USE CASE 3: VIRTUAL HEALTH SYMPTON CHECKER

- Context: A virtual health assistant employs a large language model (LLM) to sort out symptoms and recommend appropriate next steps in care.
- Feedback Loop Scenario: Prompts used early in the assistant's life were conservatively framed for risk mitigation (e.g., "consult a doctor"). These led users to report their health condition as not improved due to the influence of our study on the risk feedback they provided. In turn, our underreported health improvement reinforced our bias against recommending the assistant as an alternative to speaking with a doctor in person.
- Outcome: The LLM health assistant became more and more cautious in its recommendations, leading some users to not trust it as an alternative to speaking with a doctor in person.
- FLDE Application: Interaction tracking and bias diagnostics revealed a tone imbalance across patient types. Causal modeling confirmed influence from early prompts. Prompt shaping and user-segment tuning improved patient outcomes and restored balance.

VIII. EVALUATION METRICS

We propose a systematic method for feedback loop evaluation.

- Feedback Amplification Rate (FAR): This metric assesses how much a certain output, like a prediction, is boosted by what users did in the past. If a user did something that was valuable to the system—like being right in their choice, which is infrequent—FAR would want to give that a lot of value. If a user is being right a lot (which, again, is not too common), then they should be really pushing the system forward in terms of the kind of predictions it is making.
- Divergence in User Behavior (DUB): Measures how much different the users act from how they previously acted, before and after a measure was taken.
- Entropy-Based Diversity Score (EBDS): Measures the niceness gradients in the diversity of recommendations over time.
- Temporal Bias Index (TBI): Assesses the sustained directionality of bias over moving time windows.
- User Retention Shift (URS): Monitors the loss of user engagement that occurs when personalization is too narrow.

IX. LIMITATION

Our proposed system shines a light on the hidden feedback biases of models; however, we must acknowledge some limitations.

- Metadata Dependence: FLDS performance is heavily dependent on rich interaction metadata.
- Challenges of Interpretability: It is still hard to causally attribute the shift in user behavior.
- Real-Time Scalability: If FLDS must be used at large scale, latency and infrastructure costs could be considerably increased.
- Domain-Specific Calibration: Metrics like FAR and REI need to be well-calibrated per domain, thus limiting generalization.

X. ETHICAL CONSIDERATIONS

Disregarding feedback loops is risky; it may reinforce discrimination, narrow exposure, and degrade fairness. The ethical deployment of AI necessitates the following:

- Clarity: Tutoring users about the ways they are profiled based on their feedback.
- Preservation of Diversity: Working to make sure that model predictions are balanced, so that the content they produce is diverse, and the outcomes—at least in intent, if not in observable effects—are also diverse.
- Ongoing Surveillance: Feedback bias should be watched post-launch, not just when modeling.
- User Empowerment: Interfaces should enable users to countermand or modify model-guided decisions.

XI. CONCLUSION

Invisible feedback loops are a critical blind spot in the deployment of responsible AI. They are not always evident during initial model evaluation but can have far-reaching implications on fairness, diversity, and trust. Our proposed framework — LoopAware — offers a proactive, modular, and enterprise-ready solution to detect and mitigate such loops.

LoopAware transforms feedback loops from invisible liabilities into actionable insights. The framework not only tracks signals comprehensively but also quantifies bias passively and intervenes responsibly. This framework empowers enterprises to go beyond static model performance metrics and adopt a more dynamic, ongoing approach to AI system monitoring.

Moreover, the expanded use cases in this paper highlight how subtle, systemic reinforcement can emerge in diverse environments — from content feeds to hiring algorithms and public safety decisions. The framework is not limited to these domains but is extensible across multiple industries.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25may1549

Ultimately, this paper contributes a scalable pathway to operationalize ethical AI practices in real-world user-facing ML systems. By actively detecting and intervening in feedback loops, organizations can align their models with broader social and regulatory expectations. Continued research and real-world deployments will further refine these mechanisms and ensure that invisible biases do not become entrenched in the fabric of AI systems.

XII. FUTURE WORK

There are many chances to build on the FLDS framework.

- Analysis of Cross-System Feedback: Knowing how conduct in one system (e.g., search) influences conduct in another (e.g., ads or recommendations).
- Real-Time Feedback Loop Visualization: Creatinguser interfaces for stakeholders to witness the development of feedback over time.
- Decentralized Feedback Analysis: Extending loop detection to federated systems using user data spread across devices.
- Simulated Feedback Environments: Leveraging reinforcement learning to mimic the sustained impacts of various damping strategies.
- User-In-The-Loop Reweighting: Users help to customize prompt weights and make model personalization decisions.
- Integration with Explainability Tools: SHAP or LIME outputs that explain the loop-induced behavior of the model integrate well with the existing infrastructure.
- Cross-Domain Adaptation: Extending the framework to such domains as legal tech, insurance, and online education.
- Bias dashboards for online systems: Creating dashboards that visualize drift, feedback strength, and risk loop scores, as well as showing strong visual bias elements in virtual systems.
- Human-in-the-Loop Audits: Allowing experts in the field to check the recommendations made by the model and to point out, if necessary, the kind of loops that the model might be prone to.
- Recommendations for Policy: Working in partnership with regulators to establish compliance benchmarks related to loops.

Investigating these avenues will increase our comprehension of bias in ML and consequently create systems that are more responsible, adaptable, and centered around user needs.

REFERENCES

[1]. A. Chaney, B. Stewart, and B. A. Resnick, "How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility," in Proc. RecSys, 2018

[2]. A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” arXiv preprint arXiv:1710.11214, Oct. 2017. [Online]. Available: https://arxiv.org/abs/1710.11214

[3]. T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay, "Accurately Interpreting Clickthrough Data as Implicit Feedback," in Proc. SIGIR, 2005.

[4]. R. Binns, M. Veale, U. Lyngs, J. Zhao, and N. Van Kleek, "‘It’s Reducing a Human Being to a Percentage’: Perceptions of Justice in Algorithmic Decisions," in Proc. CHI, 2018.

[5]. W. Wang, Y. Zhang, Z. Yan, S. Wang, J. Wu, and C. Wang, “Prompt2Model: Teaching Large Language Models to Write and Run Programs,” arXiv preprint arXiv:2505.12185, May 2025. [Online]. Available: https://arxiv.org/abs/2505.12185

[6]. R. Shah, Y. Li, H. Hu, and M. Sun, “Evidence-Informed Evaluation of Large Language Models,” arXiv preprint arXiv:2505.11509, May 2025. [Online]. Available: https://arxiv.org/abs/2505.11509

[7]. A. Wang, R. Wu, S. Lee, and Y. Xu, “Judging LLM Judges: A Study of Bias in AI Feedback,” arXiv preprint arXiv:2505.11350, May 2025. [Online]. Available: https://arxiv.org/abs/2505.11350

[8]. J. Wang, T. Zhang, Y. Zhang, and M. Wang, “LLM-Smith: Evaluating Robustness of Large Language Models with Adversarial Model Inversion,” arXiv preprint arXiv:2505.07581, May 2025. [Online]. Available: https://arxiv.org/abs/2505.07581