Towards Safe ML-Based Systems in Presence of Feedback Loops

Sumon Biswas
sumonb@cs.cmu.edu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Yining She
yiningsh@cs.cmu.edu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

Eunsuk Kang
eunsukk@andrew.cmu.edu
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA

ABSTRACT

Machine learning (ML) based software is increasingly being deployed in a myriad of socio-technical systems, such as drug monitoring, loan lending, and predictive policing. Although not commonly considered safety-critical, these systems have a potential to cause serious, long-lasting harm to users and the environment due to their close proximity and effect on the society. One type of emerging problem in these systems is *unintended side effects from a feedback loop*; the decision of ML-based system induces certain changes in the environment, which, in turn, generates observations that are fed back into the system for further decision-making. When this cyclic interaction between the system and the environment repeats over time, its effect may be amplified and ultimately result in an undesirable. In this position paper, we bring attention to the safety risks that are introduced by feedback loops in ML-based systems, and the challenges of identifying and addressing them. In particular, due to their gradual and long-term impact, we argue that feedback loops are difficult to detect and diagnose using existing techniques in software engineering. We propose a set of research problems in modeling, analyzing, and testing ML-based systems to identify, monitor, and mitigate the effects of an undesirable feedback loop.

CCS CONCEPTS

• **Software and its engineering → Software creation and management**; • **Computing methodologies → Machine learning**.

KEYWORDS

Feedback loop, safety, machine learning

ACM Reference Format:
Sumon Biswas, Yining She, and Eunsuk Kang. 2023. Towards Safe ML-Based Systems in Presence of Feedback Loops. In *Proceedings of the 1st International Workshop on Dependability and Trustworthiness of Safety-Critical Systems with Machine Learned Components (SE4SafeML '23), December 4, 2023, San Francisco, CA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3617574.3617861

1 INTRODUCTION

Software with machine learning (ML) components is widely being deployed as part of *socio-technical systems*, such as drug monitoring, loan lending, and predictive policing. Although not traditionally considered safety-critical, these systems are posing increasing safety risks to our society, as they have a potential to cause direct or indirect harm in long-term. For example, a buggy or biased ML-based medical diagnosis system may deny timely access to healthcare for certain population, causing serious physical harm.

One of the emerging problems in ML-based systems is *unintended side effects from feedback loops* [5, 9, 27]. A feedback loop occurs when a system makes a decision that induces certain changes in the environment, which, in turn, influences the system's future behaviors through its input. Not all feedback loops are undesirable; for example, in control engineering [6, 13], feedback loops play a crucial role by providing an explicit mechanism to monitor and regulate the system's output to a desirable level. However, certain *self-reinforcing* feedback loops, where an increase (decrease) in the input results in an increase (decrease) in the output, can have an undesirable effect if left uncontrolled over time.

For example, consider a predictive policing system [20] that uses ML-based predictions about crime rates (based on historical data in different neighborhoods) to determine allocation of police patrol. Suppose that a high amount of patrol is assigned to a particular neighborhood based on the initial ML prediction; since more police are present, it is likely to lead to an increase in the number of arrests in that neighborhood; this, in turn, generates more arrest records to be fed back into the system for further decision making or model retraining. When this pattern of interaction repeats over time, that neighborhood may become unfairly perceived as a crime hotspot, while other neighborhoods that could actually benefit from increased patrol remain unattended [10]. There are other well-known examples of harmful side effects caused by self-reinforcing feedback loops in ML-based systems [25].

Safety risks due to feedback loops have been studied extensively in system engineering [6], control applications [13], and social sciences [8] but have received relatively little attention in software engineering. Recent works by the ML community on distribution shifts [19, 26] focus almost exclusively on model accuracy, and do not consider long-term effect on *system-level* properties such as safety or fairness. We believe that software engineers share the responsibility to explicitly account for the possibility of feedback loops in the system being developed, consider potential harmful effects, and build in mechanisms to detect and mitigate those effects.

However, these are challenging tasks to carry out using the existing tools and techniques that are available to software engineers. The impact of a feedback loop is often gradual and may not be evident until the system has been deployed for a long period, but most techniques in software testing and verification are designed to reason about the system as *it is*, not about how the system and the environment might evolve. Even domain experts may not be aware of potential harms that may emerge, so new methods for requirements elicitation, specification, and validation for feedback-driven

SE4SafeML '23, December 4, 2023, San Francisco, CA, USA

ACM ISBN 979-8-4007-0379-9/23/12.
https://doi.org/10.1145/3617574.3617861

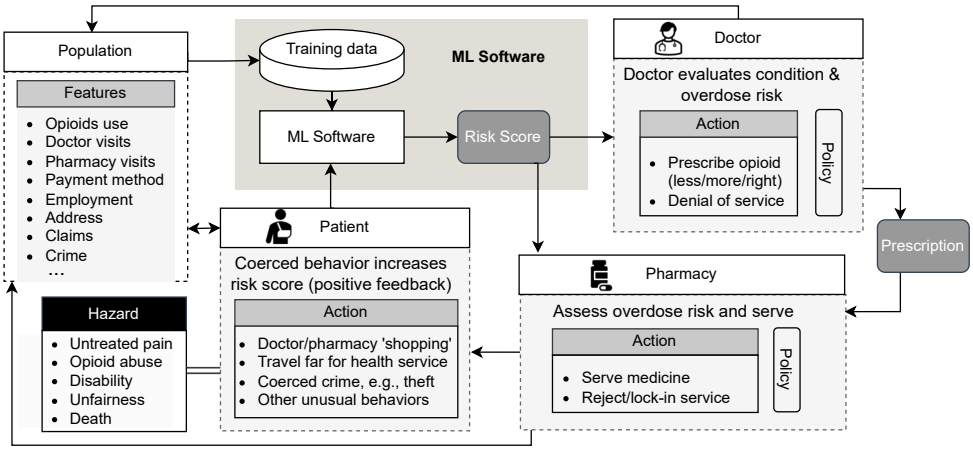

Figure 1: ML-based risk scoring for opioid overdose can lead to several safety hazards in the long run

systems are needed. Furthermore, monitoring and intervening the system (e.g., through retraining or updating decision-making policy) to mitigate the effects of an ongoing feedback loop is an important step that could benefit from improved developer tools.

In this position paper, we aim to bring attention to emerging safety risks that are introduced by feedback loops in an ML-based system, and the challenges of identifying and addressing their negative impact throughout the software development life-cycle. We begin by introducing an ML-based drug monitoring system as an example and potential harmful effects that can arise from a feedback loop. We then propose a conceptual framework for modeling and analyzing feedback loops, and conclude by proposing a set of research challenges and opportunities.

2 FEEDBACK LOOPS IN ML: AN EXAMPLE

In the United States, prescription opioid abuse is the leading cause of death for adults under the age of 50, even more than car crashes or gun violence [12, 23]. Common opioids such as methadone, morphine, fentanyl, etc. are frequently used for pain management that can result in overreliance and addiction, which is called Opioid Use Disorder (OUD). A Study showed that 79.9% of abusers had an opioid prescription before their first abuse [12]. With the growing number of OUD, the US Department of Health and Human Services declared prescription opioid drug abuse an 'epidemic' in 2015. Recent data show a worsening situation with an estimate of over 100,000 annual deaths caused by OUD for the first time in 2021 [1].

Despite extensive research on the prevalence and causes of OUD, an appropriate method of opioid prescription for preventing misuse, abuse or addiction is not well understood because of variations in medical conditions, age, and policies. To reduce overdose, a predictive surveillance platform called Prescription Drug Monitoring Program (PDMP) is mandated in each state, which measures an overdose risk score for opioid prescription. NarxCare is widely-used ML-based software in PDMP that produces a numeric risk score (000-999; a higher score implies a higher risk) for doctors and pharmacists [2]. Potential hazards brought by the PDMP risk scoring system are shown in Figure 1. The ML algorithm is trained using PDMP data of patients, including features such as past opioid usage, number of pharmacies visited, number of prescribers, overlap from different prescribers, etc., [3]. The score can directly impact doctors' evaluation of opioid prescriptions and pharmacists' decisions to allow medicine purchase. Their decision may induce changes in the patient behavior, which, in turn, result in new observations being generated and fed back into the system.

While the actual goal of the tool is to curtail opioid overdose, the risk scores could be interpreted in various ways and result in unwarranted denial of service from the healthcare providers [4]. The problem firstly is the misguided usage of the features like the number of prescriptions, which can flag chronic pain and cancer patients as high-risk. Second, the success of the tool is assessed based on the decrease in overdose deaths; however, while this may reduce overall OUD cases, numerous legitimate users can suffer from physical debilitation, untreated pain, social and mental damage, or be coerced into illegitimate activities [17]. Here, we describe an instance of a feedback loop that might result in a safety hazard.

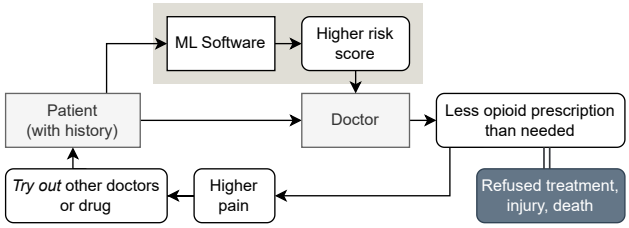

Figure 2: A feedback loop reinforcing risky patient behavior

Example: Feedback loop in PDMP. The genetic variance in metabolic rates for people can vary up to 17-fold. Therefore, the risk score is suggested to be used at the user's (e.g., doctor) discretion. Depending on the doctor's decision, some patients may receive a lesser amount of opioids than needed, which, in turn, is likely to cause higher pain. These patients can go to another doctor, try other drugs in pharmacies, or exhibit other unusual behavior, e.g., identity theft, cash payments, etc. As described by NarxCare, any of these behaviors can influence the ML software to increase the risk score of the patients in their subsequent doctor or pharmacy visits. Thus, an increasing risk score may eventually cause the patient to be 'locked-in' by the provider and result in untreated pain. Figure 2 depicts the specific feedback loop created by the PDMP system.

3 MODELING AND REASONING ABOUT FEEDBACK LOOPS

To mitigate potential safety hazards due to feedback loops in an ML-based system, developers may wish to ask questions such as:

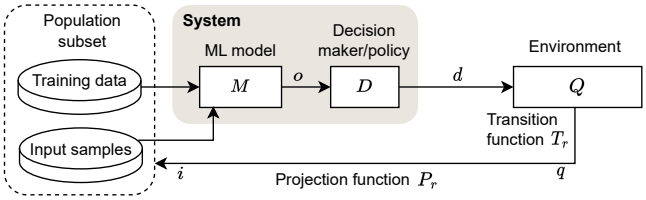

Figure 3: A framework for ML-driven feedback loops

Are there possible self-reinforcing feedback loops in this system? What interactions between the system and environment could give rise to such feedback? What are some harmful effects? What type of intervention is needed to mitigate the negative impacts? To support approaches for systematically answering these questions, we propose a conceptual framework for capturing interactions between the system and the environment, and the feedback loop.

Framework: Figure 3 illustrates the proposed framework. The ML model M is trained using an initial set of training data that is gathered from the environment. Given some input (i) to the system (e.g., patients' information in PDMP), M performs inference to generate a prediction output (e.g., risk score). This prediction (o) is then passed onto a decision-making entity D, which makes a further decision (d) based on a policy, i.e., doctors or pharmacies that use the risk score to determine the prescription amount.

The environment is modeled as an entity that is associated with some state $q \in Q$, which captures relevant properties of a population at a particular point in time. In the PDMP example, each state q may itself be a complex entity that captures the properties of patients (e.g., age, gender, opioid usage, pharmacy visits). When the system makes a decision, the environment transitions from one state to another, during which the properties of some subset of the population may change, depending on how the system's decision influences the subset. For example, in PDMP, if the system decides to deny medication to a group of patients, those patients may choose to visit another pharmacy; this behavior would then be reflected through a change in their pharmacy visit records. In our framework, we model these changes as a *transition function* $T_r : Q \times D \rightarrow Q$, where the environment moves from q to $q' = T_r(q, d)$ for given system decision d[1]. As the system makes a series of decisions $\langle d_0, d_1, ... \rangle$ over time, the environment also evolves through a corresponding sequence of states $\langle q_0, q_1, ... \rangle$, where $q_n = T_r(q_{n-1}, d_{n-1})$.

As the environment evolves, the ML model M is again fed input data that is gathered by sampling the population. Occasionally, some part of this data may be set aside as new training data for retraining the model. In our framework, a *projection function* $P_r : Q \rightarrow I$ is used to determine the observable parts of the environment state q that M uses for prediction (i.e., the set of input features).

Analysis: We envision that this framework could be used to support both informal and formal reasoning about feedback loops in ML-based systems. Having identified the major elements of the framework (i.e., M, D, Q, T_r, and P_r) for a specific ML-based system, one could simulate the framework under different initial states (i.e., $q_0 \in Q$) or transition functions (T_r) for some number of steps, to identify possible feedback loops and gain insights about how system evolves with a series of interactions, i.e., oscillating or converging feedback loop in mid- or long-term.

[1]For simplicity, T_r here is assumed to be a deterministic function. In general, a stochastic model of the environment is likely to be more suitable.

To understand potential harmful effects of a feedback loop, one could monitor how particular properties of the environment change over time. For example, in PDMP, we could define a variable that corresponds to the percentage of the population from a certain neighborhood that is denied a prescription; if this variable gradually increases and exceeds a threshold, the subset of the population would be at safety risk as a result of a feedback loop.

If the elements of the framework can be encoded as formal artifacts, it may be possible to (semi-)automate the simulation-based analysis. For instance, M itself could be substituted by a real ML model; Q could be represented as a set of records that encode relevant properties of a population; and T_r could be defined using a set of manually-devised, domain-specific rules or possibly learned from real-world data. Then, the entire framework could be simulated to explore a large number of possible system traces and detect harmful feedback loops. Such simulation might not always generate accurate evolution of system if impactful exogenous variables (e.g., transition rules) are not captured. However, even with the models of Q or T_r that are incomplete or rough approximation of the real-world entities, such analysis can reveal valuable insights as it is often done in fields such as econometrics, social sciences, and system dynamics [22, 29].

4 RESEARCH CHALLENGES & DIRECTIONS

Impact analysis for safety: Retrospective analyses and repair, after a safety violation has already occurred, can be too costly. We advocate proactive analysis (such as simulation) to understand the effects of possible feedback loops. Even if the environment model is an approximation of the real world, such analyses could provide useful insights about safety. One challenge with simulation is a large (and potentially infinite) number of simulation traces to explore and evaluate. In each step of the simulation, uncertainties in the environment model and the dynamic behavior of agents can yield many different choices for the system trace to evolve. A naive random or exhaustive search is likely to be impractical, and further research is needed on techniques for efficiently exploring a diverse set of traces within a limited amount of analysis resources.

ML system design and repair strategies: We advocate considering feedback loops as a first-class concept in ML component design. When selecting a feature or optimization function, we should evaluate how predictions may influence the environment. For example, is it appropriate to optimize the ML model for lowering the number of OUD cases in a given area? It has been shown that an ML agent may exploit reward hacking to maximize its objective by gradually limiting the access of opioids to chronic patients or individuals from specific neighborhoods [5]. The NarxCare ML model selects 12 features from 70 PDMP variables [3], which should be evaluated for their possible contributions to a feedback loop. Furthermore, continuous efforts to explore unobserved parts of the environment through data collection and simulation would be helpful. In addition, recent ML approaches such as meta-learning [24] and population-based training [16] could be adopted to find hyperparameters that minimize the effect of a harmful feedback loop.

Safety requirements elicitation for feedback loops: In a complex system with multiple stakeholders (like the one shown in Figure 1), there are often conflicting requirements. In PDMP, while

one important requirement is to provide an appropriate amount of opioids, another state-mandated regulation may impose a restriction on the maximum opioid prescription allowed for each doctor. A requirements elicitation and negotiation process that explicitly considers trade-offs between them is a crucial step in developing safe ML systems. Additionally, requirements on feedback loops would likely be defined over how the system evolves over time, instead over a static snapshot. Thus, new metrics and dynamic tolerance limits for feedback loops are needed to guide the safety evaluation of system designs and policies. Existing modeling approaches like problem frames [15], feedback modeling [27], community-based system dynamics (CBSD) [21], and Leveson's System Theoretic Process Analysis (STPA) [17] would be good starting points.

Causal reasoning: Identifying the root cause(s) of a feedback loop is challenging given a rich set of system and environmental characteristics and their possible interactions. Different factors such as the properties of the ML system M (e.g., retraining frequency, decision threshold), policies D (e.g., elders are prioritized for vaccines in certain area [11]), intensity of transitions T_r (e.g., amount of opioid adjustment) may contribute to the rise of a feedback loop. We envision that a simulation framework (like the one that we have proposed) could be used to enable a data-driven analysis of highly configurable systems and identify possible causal relationships between these factors. In particular, the analysis could be used to infer causal loop diagrams [18] or stock-and-flow diagrams [17] to inform the effect of particular configuration options and enable counterfactual reasoning. Challenges remain in modeling unobserved variables and missing causal links, which could be mitigated by engaging with domain experts and stakeholders.

Adaptive policies and intervention: An adaptive approach that dynamically adjusts the behavior of the system to maintain a desired level of fairness is another promising direction [11]. Methods and tools from self-adaptive systems could be leveraged [28]. Different adaptation tactics, such as model retraining and replacement, selectively forgetting certain inputs, amending outcomes in post-processing, and coping with covariate/concept drift, could help mitigate the effect of a feedback loop [7]. In addition, such adaptive approaches should be coupled with a runtime monitor that continuously analyzes the system behavior and detect the rise of a possible feedback loop (similar to monitors proposed in [14])

5 CONCLUSION

With the ever-increasing autonomy and data-driven world, ML-based software systems are influencing lives and society more than ever. In this paper, we bring attention to undesirable but implicit feedback loops in ML-based systems that can cause difficult-to-reverse, harmful consequences on safety. Commonly induced as side effects of opaque ML processes, positive feedback loops have been overlooked in software design and analysis. We've illustrated such feedback loops with a real-world example, that require immediate attention to avoid long-term safety impact on our society. We've proposed a simulation-based framework to identify and analyze the feedback loops in socio-technical systems. We've further outlined the associated research challenges and future directions.

ACKNOWLEDGEMENT

This work is supported in part by the National Science Foundation (NSF) Award OAC-2233871.

REFERENCES

[1] . https://www.cnn.com/2021/11/17/health/drug-overdose-deaths-record-high.
[2] . https://bamboohealth.com/solutions/narxcare.
[3] . https://www.in.gov/pla/inspect/files/Narxcare_user_guide.pdf.
[4] . https://www.wired.com/story/opioid-drug-addiction-algorithm-chronic-pain.
[5] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.
[6] Karl Johan Åström and Richard M Murray. 2021. *Feedback systems: an introduction for scientists and engineers*. Princeton university press.
[7] Maria Casimiro, Paolo Romano, David Garlan, Gabriel A Moreno, Eunsuk Kang, and Mark Klein. 2021. Self-Adaptation for Machine Learning Based Systems.
[8] Efrén Cruz Cortés, Sarah Rajtmajer, and Debashis Ghosh. 2022. Locality of Technical Objects and the Role of Structural Interventions for Systemic Change. In *2022 ACM Conference on Fairness, Accountability, and Transparency*. 2327–2341.
[9] Alexander D'Amour, Hansa Srinivasan, James Atwood, Pallavi Baljekar, D. Sculley, and Yoni Halpern. 2020. Fairness is Not Static: Deeper Understanding of Long Term Fairness via Simulation Studies. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 525–534.
[10] Danielle Ensign, Sorelle A. Friedler, Scott Neville, Carlos Scheidegger, and Suresh Venkatasubramanian. 2018. Runaway Feedback Loops in Predictive Policing. In *Conference on Fairness, Accountability and Transparency FAT*. PMLR, 160–171.
[11] Ali Farahani, Liliana Pasquale, Amel Bennaceur, Thomas Welsh, and Bashar Nuseibeh. 2021. On Adaptive Fairness in Software Systems. In *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*.
[12] Justine S Hastings, Mark Howison, and Sarah E Inman. 2020. Predicting high-risk opioid prescriptions before they are given. *Proceedings of the National Academy of Sciences* 117, 4 (2020), 1917–1923.
[13] Joseph L Hellerstein, Yixin Diao, Sujay Parekh, and Dawn M Tilbury. 2004. *Feedback control of computing systems*. John Wiley & Sons.
[14] Thomas A. Henzinger, Mahyar Karimi, Konstantin Kueffner, and Kaushik Mallik. 2023. Runtime Monitoring of Dynamic Fairness Properties. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. ACM, 604–614.
[15] Michael Jackson. 2001. *Problem frames: analysing and structuring software development problems*. Addison-Wesley.
[16] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846* (2017).
[17] Edgar Jatho, Logan Mailloux, Shalaleh Rismani, Eugene Williams, and Joshua A Kroll. 2022. System Safety Engineering for Social and Ethical ML Risks: A Case Study. In *NeurIPS ML Safety Workshop*.
[18] Eunsuk Kang and Rômulo Meira-Góes. 2022. Requirements Engineering for Feedback Loops in Software-Intensive Systems. In *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2–5.
[19] David Krueger, Tegan Maharaj, and Jan Leike. 2020. Hidden incentives for auto-induced distributional shift. *arXiv preprint arXiv:2009.09153* (2020).
[20] Kristian Lum and William Isaac. 2016. To predict and serve? *Significance* (2016).
[21] Donald Martin Jr, Vinodkumar Prabhakaran, Jill Kuhlberg, Andrew Smart, and William S Isaac. 2020. Extending the machine learning abstraction boundary: A Complex systems approach to incorporate societal context. *arXiv preprint arXiv:2006.09663* (2020).
[22] Donella H. Meadows. 2008. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Chelsea Green Publishing.
[23] Ethel AM Mensah, Musarath J Rahmathullah, Pooja Kumar, Roozbeh Sadeghian, and Siamak Aram. 2021. A Proactive Approach to Combating the Opioid Crisis Using Machine Learning Techniques. In *Advances in Computer Vision and Computational Biology*. Springer, 385–398.
[24] Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. 2018. Meta-Learning Update Rules for Unsupervised Representation Learning. In *International Conference on Learning Representations*.
[25] Cathy O'neil. 2017. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown.
[26] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. 2022. *Dataset shift in machine learning*. Mit Press.
[27] Lydia Reader, Pegah Nokhiz, Cathleen Power, Neal Patwari, Suresh Venkatasubramanian, and Sorelle A. Friedler. 2022. Models for understanding and quantifying feedback in societal systems. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*.
[28] Mazeiar Salehie and Ladan Tahvildari. 2009. Self-adaptive software: Landscape and research challenges. *TAAS* 4, 2 (2009), 14:1–14:42.
[29] John D Sterman. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill Education.

Received 2023-07-04; accepted 2023-08-10