FairSense: Long-Term Fairness Analysis of ML-Enabled Systems

Yining She
Carnegie Mellon University
yiningsh@andrew.cmu.edu

Sumon Biswas
Case Western Reserve University
sumon@case.edu

Christian Kästner
Carnegie Mellon University

Eunsuk Kang
Carnegie Mellon University
eunsukk@andrew.cmu.edu

***Abstract*—Algorithmic fairness of machine learning (ML) models has raised significant concern in the recent years. Many testing, verification, and bias mitigation techniques have been proposed to identify and reduce fairness issues in ML models. The existing methods are *model-centric* and designed to detect fairness issues under *static settings*. However, many ML-enabled systems operate in a dynamic environment where the predictive decisions made by the system *impact* the environment, which in turn affects future decision-making. Such a self-reinforcing *feedback loop* can cause fairness violations in the long term, even if the immediate outcomes are fair. In this paper, we propose a simulation-based framework called FAIRSENSE to detect and analyze long-term unfairness in ML-enabled systems. Given a fairness requirement, FAIRSENSE performs *Monte-Carlo simulation* to enumerate evolution traces for each system configuration. Then, FAIRSENSE performs *sensitivity analysis* on the space of possible configurations to understand the impact of design options and environmental factors on the long-term fairness of the system. We demonstrate FAIRSENSE's potential utility through three real-world case studies: Loan lending, opioids risk scoring, and predictive policing.**

I. Introduction

Socio-technical systems are increasingly using machine learning (ML) models to automate high-stakes decisions such as loan lending, drug risk scoring, predictive policing, college admission, and vaccine allocation [1–4]. As unfair decisions made by such systems can cause harm to users and our society, approaches to developing fair systems have gathered significant interest in recent years. For example, researchers have developed various methods for fairness measures and identification [5–7], verification and testing [5, 8–11], and bias mitigation [12–14].

Most of the existing work on fairness is *model-centric* under *static* settings, that is, it evaluates and improves fairness of a given model at a particular point in time. However, even if a system appears to be fair initially, fairness issues may arise after it has been deployed for a period of time; we call these *long-term fairness* issues. A long-term fairness issue arises as a result of a *feedback loop* between a system and its *environment* [15, 16]: Decisions made by an ML-enabled system induce certain changes or *shifts* in the environment, which, in turn, can influence the ensuing system behavior. For example, when an ML-enabled lending application declines a request for a bank loan, this decision may reduce the credit score of an individual, which further damages their chance of a future loan approval. If left unattended over a long period of time, such a *self-reinforcing* feedback loop can result in discrimination against certain groups of individuals [17].

Unintended consequences from feedback loops are being recognized as an emerging problem in ML-enabled systems [2, 3, 17–20]. Identifying long-term fairness issues poses new challenges beyond static fairness analysis, as it requires looking beyond the boundary of an ML model and analyzing possible *interactions between the system and its environment* [21–23]. The behavior of a system is influenced not only by ML system design decisions (e.g., data collection, agent policies, hyperparameters, optimization metrics, and retraining criteria) but also depends strongly on the dynamics of the surrounding environment (e.g., how people react to and adapt to the system and its decisions). Depending on the combination of these decisions and possible environmental dynamics, the system can evolve in numerous ways, some of which may result in an undesirable feedback loop. Thus, explicit consideration of the environment and its interactions with the system [24, 25] is crucial for identifying long-term fairness issues.

In this paper, we propose FAIRSENSE, a tool-assisted approach for **proactive analysis of long-term fairness issues that specifically considers a model of the environment and uncertainty about the environment**. A key distinction of our approach is that FAIRSENSE is designed to aid developers early *in the requirements and design stages*, so that they can focus on the design decisions and environmental factors that most impact long-term fairness, and avoid deliberating on others. That is, FAIRSENSE helps to proactively analyze requirements and consequences of different designs before the system is implemented and deployed, which helps not only to create better initial designs, but also to plan for mitigations and monitoring to maintain fairness when the system is deployed in the real world.

We show an overview of the proposed framework in Figure 1. A developer using FAIRSENSE specifies three types of inputs: (1) *system parameters*, which describe the space of configuration options (e.g., type of ML model and agent policies), to be explored, (2) desired *fairness criteria*, and (3) an *environmental model*, where the *environmental parameters* control the dynamics of environmental changes that are induced by the system's decisions. The latter model itself consists of (i) a *target dataset* that represents the target population and (ii) a *distribution-shift model*, which describes how the system outcome may cause changes in the dataset. The distribution-

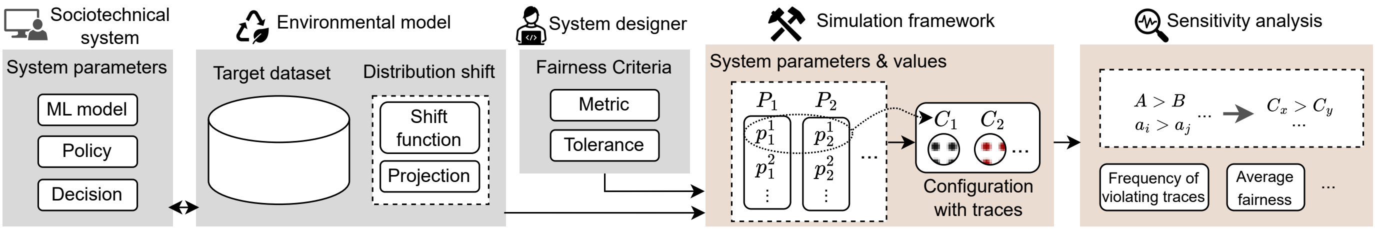


Fig. 1: An overview of the FairSense approach

shift model is stochastic, explicitly encoding uncertainty about how the environment may evolve in response to system output.

Given these inputs, FairSense performs **Monte-Carlo simulation** [26] to systematically generate *traces* that show how the system and the environment may evolve together over time for a given *configuration* (i.e., an assignment of values to the system and environmental parameters). The desired fairness criteria are then evaluated over these traces, assigning each configuration a fairness metric that represents the level of unfairness that might arise over time. System designers are often overwhelmed with many design choices and can spend a lot of time negotiating a choice that ultimately matters little for fairness. FairSense adopts **sensitivity analysis** [27] to identify which system parameters and environmental parameters are the most influential to shape long-term fairness. This helps developers to focus their time effectively on reasoning about options that provide the highest leverage, e.g., monitor critical environmental parameters closely to reduce uncertainty and react in a timely fashion or invest in system design options that have the largest potential impact. In addition, FairSense enables a **trade-off analysis** between system utility and long-term fairness metrics, to aid developers in selecting decisions that achieve desired levels of utility and long-term fairness.

To demonstrate its potential utility, we have applied FairSense on three real-world case studies, built on models and data from prior research: Loan lending [2, 19], opioids risk scoring [1, 28], and predictive policing [3, 29]. Our case studies show that FairSense can be used to systematically analyze and understand the impact of design options on long-term fairness. The main contributions of the paper are:

- A conceptual model of feedback loops and their impact on long-term fairness of ML-enabled systems (Section IV).
- A simulation-based framework that systematically explores possible evolution traces (Section V) and performs sensitivity analysis to rank system parameters in terms of their impact on long-term fairness (Section VI).
- A prototype implementation of FairSense and its demonstration on three real-world case studies (Section VIII).

II. Background

In this paper, we focused on the long-term fairness of ML-enabled sociotechnical systems. These *systems* are software solutions that closely interact with humans and society in different domains, such as education, finance, and the judiciary. The system consists of several components, such as the collected data, one or more ML models, and the decision-making entity. The system operates in a certain social context, which we refer to as the *environment*. The system and the environment interact continuously during its operation. The environment dynamics can usually be decomposed further, such as the population distribution and human behaviors. The involvement of many *agents*, such as system users and policymakers, affects various dynamics, which leads to uncertain evolution of the system and the environment over time.

A *feedback loop* occurs when the system induces certain changes to the environment, which impacts the decision-making of the system through its input [16]. A *balancing* feedback (or *negative* feedback) loop is created by system structures that are sources of both stability and resistance. On the other hand, a *reinforcing* feedback (or *positive* feedback) loop causes divergence and continuously shifts the environment toward a risky outcome, commonly seen in various domains, such as biology, electronics, economics, and sociology [20, 30]. An example of such a feedback loop is described in detail in Section III in the context of sociotechnical systems. Identifying the cause of the feedback loop before deployment can help design interventions such as creating artificial feedback to mitigate existing one or choosing the best system parameter [21]. We proposed modeling the system, environment, and their interactions, which we call a *feedback loop model*, to understand the system design space.

Various fairness criteria exist [31], with common metrics including including *demographic parity* and *equal opportunity* [32, 33]. For example, in a loan lending system, serving two groups a and b, demographic parity measures the difference in the approval rates between individuals in a and b. Formally, given the predictive outcome ($\hat{Y}$) and group membership feature (A), the demographic parity requirement is given by: $|P[\hat{Y} = 1|A = a] - P[\hat{Y} = 1|A = b]| < \epsilon$ for some threshold ϵ. In contrast, equal opportunity measures the difference of true positive rates between the groups; formally, with Y representing the actual outcome, $|P[\hat{Y} = 1|A = a, Y = 1] - P[\hat{Y} = 1|A = b, Y = 1]| < \epsilon$.

III. Motivating Example

Unfairness in bank loan approvals and credit scoring has been investigated extensively in prior work [11, 34, 35]. For example, City National Bank was recently fined over $31 million for discriminatory lending against Black and Latinos [36]; and the Apple Card joint venture of Apple and Goldman Sachs has been accused of discriminating against female applicants for credit approval [37]. In the US, Equal Credit Opportunity Act of 1974 (ECOA) requires non-discrimination in lending.

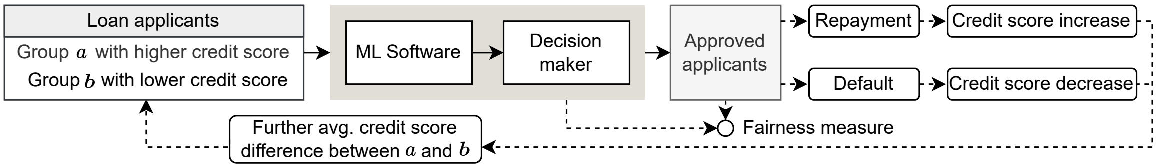


Fig. 2: A feedback loop created by ML-enabled loan lending system

A developer of a loan lending application can leverage existing fairness metrics or testing methods [8, 33, 38] to analyze whether the system *statically* satisfies a fairness requirement at the model level. However, a seemingly fair loan lending system (e.g., satisfying demographic parity at the time of deployment) may begin to exhibit unfair behaviors over time. Figure 2 depicts a possible feedback-driven interaction between the ML-enabled system and the environment. The ML model here uses the applicants' credit score to predict the likelihood of on-time loan repayment; only if this predicted value is above a certain threshold, the applicant is granted the loan. Suppose that group a historically has a higher average credit score than group b. To reduce this gap, a policy may deliberately approve a higher number of applications from group b. If, however, individuals in group b are more inclined to a loan default, their average score may decrease at a higher rate than those in group a. Furthermore, an individual whose application gets rejected may begin to apply for other loans, incurring multiple *hard inquiries* that further decrease their credit score [39]. Thus, a feedback loop will influence the credit scores of members of b to decrease over time. Even if the ML model satisfies demographic parity in the short-term, the system could begin to show unfair behavior over time due to the shifting distribution of credit scores.

The intensity of the feedback loop also depends on many factors, such as the magnitude of credit score decrease for a default and the loan approval threshold. Some of these are configurable system parameters (e.g., the loan-approval threshold), and some are uncontrollable environmental parameters (e.g., credit score update model). At design time, it may be challenging to understand how these different parameters might give rise to a feedback loop and negatively impact the long-term fairness of a system. In the following sections, we describe our approach for explicitly modeling feedback interactions between the ML-based system and the environment, and a simulation-based analysis to understand the impact of both system and environmental parameters on long-term fairness.

IV. Modeling Feedback Loops

We propose a conceptual framework to specify the structure and key elements of feedback loops in an ML-enabled system. Then, we describe how this conceptual framework can be used to develop a design-time analysis for long-term unfairness.

A. Feedback Loop Model

We illustrate the conceptual model of a feedback loop in an ML-enabled system in Figure 3: The system encompasses the ML model and any other components needed to produce the decisions; here, we modeled the system with the two components, the ML model and the *decision maker*. Feedback loops are system-level phenomena, and hence modeling the entire system, the environment, and their interactions is necessary to analyze these phenomena. mGiven input data (population sample X_{in}), the ML model M generates predictions o. For example, in a loan lending system, M takes the applicants' data as input, and outputs the probability of repayment. The decision maker entity D (e.g., the bank) then makes decision d (e.g., loan approval or rejection) based on the predictions o.

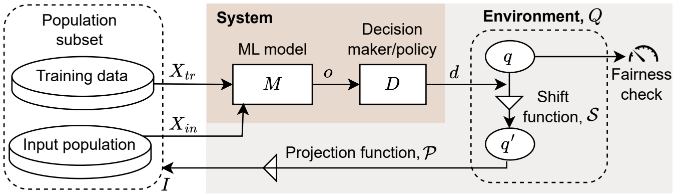


Fig. 3: Feedback loop model of ML-enabled system

The environment is modeled as the stateful entity Q, where each state $q \in Q$ captures the relevant properties of the population at a certain point in time. We model X_q as the population characteristic of the state q such that each individual $x \in X_q$ has the attributes $\langle x_1, x_2, ...\rangle$. For the loan lending application, x_i can be the protected attribute (e.g., age, gender, race) or non-protected attribute (e.g., credit score, income, education) of the applicants.

A system decision can affect the environment and *change* certain attributes, such as credit scores. The change is modeled by a *distribution-shift function* $\mathcal{S} : Q \times D \rightarrow \Delta(Q)$, where $\mathcal{S}$ is a stochastic function and $\Delta(Q)$ represents the probability distributions over the possible states Q. *The stochastic nature of this function captures uncertainty about the way in which the environment evolves given a system decision.* For some decision d, the environment shifts from q to q', where $q' \sim \mathcal{S}(q, d)$ represents the resulting probability distribution over the environmental states. Over time, one possible evolution of states is $\langle q_0, q_1, q_2, ...\rangle$, which are impacted by the series of system decisions $\langle d_0, d_1, d_2, ...\rangle$ and where q_n is sampled from the distribution $\mathcal{S}(q_{n-1}, d_{n-1})$.

The environmental state might not be fully observable by the system. In the feedback loop model, a *projection function* $\mathcal{P} : Q \rightarrow I$ determines the observable parts, where I serves as the input to the system. In practice, the developer may choose input data X_{in} from I in the next step. The training data X_{tr} is optionally set aside from I for updating the ML model periodically; i.e., M can be left static or retrained over time. In FairSense, we model $\mathcal{P}$ as a stochastic sampling function, since it can depend on various uncertain factors, such as human behavior, economic condition, and geographic location. While the system is in deployment, the environment evolves through a series of distribution shifts and may reach a state that can cause M to exhibit a certain level of unfairness.

Example: In the loan lending example, the shift function (S) can be modeled using a stochastic function that changes the credit score of the individuals based on the decision, following a Normal distribution $\mathcal{N}(\mu,\, \sigma^2)$. FAIRSENSE employs separate distributions, $\mathcal{N}_1$ and $\mathcal{N}_2$ for approval and rejection decisions. In addition, FAIRSENSE allows developers to conduct analysis for multiple environmental models by enumerating various options for the μ and σ, to reflect aggressive or conservative updates in the credit score. Similarly, $\mathcal{P}$ samples the input population by using a normal distribution $\mathcal{N}'(\mu,\, \sigma^2)$.

B. Feedback Loop Analysis

Given an instantiation of the above feedback loop model for a particular system, FAIRSENSE provides an analysis for understanding how (1) different ML system design options and (2) the dynamics of the environment may impact the long-term fairness of a system. The output of this analysis could be used by developers to (1) identify and select design options that improve long-term fairness (while considering trade-offs against other quality attributes, such as system utility) and (2) monitor the environment for the actual dynamics and apply interventions when necessary (e.g., modifying the decision-making policy) [4].

Note that the analyst does not need to provide a perfectly accurate model of the environment for the analysis to be useful. The main objective is to identify what design decisions and environmental factors are important, not what exactly will happen in a particular environment. Where uncertainty exists (for instance, if it is unclear how strongly credit scores are impacted by declined loans), the analyst can model uncertain factors explicitly as *parameters* to be explored by FAIRSENSE.

For its analysis, FAIRSENSE conducts a type of simulation-based *configuration analysis*. Each component of a feedback loop model (i.e., $M, D, Q, \mathcal{S}, \mathcal{P}$) contains one or more of *system* or *environmental parameters*. System parameters (such as the choice of ML models and the approval threshold in loan lending) are decisions that are configurable by the developer, while environmental parameters (such as the credit score change mechanism for a loan default) are assumed to be uncontrollable but observable by the ML system. Each parameter is associated with a set of *parameter values*; for example, the approval threshold for the loan lending policy may take on a value from a given range of parameter values (e.g., a credit score of 600).

Then, as an input to this analysis, the developer identifies a set of relevant system parameters (denoted $\mathbb{P}^s = P^s_1 \times P^s_2 \times \ldots P^s_m$) and environmental parameters ($\mathbb{P}^e = P^e_1 \times P^e_2 \times \ldots P^e_n$). The latter set of parameters and their values are typically elicited through a discussion with stakeholders (e.g. policy makers) or estimated based on prior data (e.g., a historical analysis of lending decisions and their impact on credit scores). The combinations of different parameter values lead to a large number of *configurations* $(C_1, C_2, ...)$ for the feedback loop model, where each configuration $C_i = \langle p_1, p_2, \ldots p_{m+n} \rangle$ is a member of the space $\mathbb{C} = \mathbb{P}^s \times \mathbb{P}^e$. The idea behind FAIRSENSE then is to simulate the feedback loop model under all possible configurations and extract relationships between the parameters and a desired long-term fairness measure. More precisely, we state the goal of the feedback loop analysis as: *Given a feedback loop model of a system $(M, D, Q, \mathcal{S}, \mathcal{P})$ and its possible configurations ($\mathbb{C}$), which of the system and environmental parameters have the most impact on its long-term fairness, negatively or positively?*

V. SIMULATION FRAMEWORK

In this section, we describe how the feedback loop model is simulated for long-term fairness analysis.

A. Monte-Carlo Simulation

First, we obtain a target dataset that represents the environment. Taking the loan lending system as an example, the environment state can be represented by a snapshot of the dataset containing the credit scores of all loan applicants. The effect of the system decisions would be reflected in the changes in the dataset. For each configuration, we simulate the feedback loop model for k time-steps and record one trace. In every time-step t, the system takes the inputs from the current state of the environment. The decisions of the system bring certain changes to the environment in the subsequent time-step ($t' = t + 1$). Then, the system makes a new set of decisions based on the new inputs from the updated environment in step t'. We record the inputs (X_{in}), outputs (o, d), and the environmental state (q) in each step, together constituting a snapshot s. Thus, after k simulation steps, we generate a trace $T = \langle s_1, s_2, ...s_k \rangle$.

However, the feedback model may evolve in numerous ways for the same configuration, due to the uncertainty of the environmental parameters and the interactions between the system and the environment. To systematically account for the uncertainty, we conduct a *Monte-Carlo simulation* for each configuration. The Monte-Carlo simulation is a computational technique that uses random sampling to model complex systems and assess the impact of uncertainty. By generating a wide range of possible scenarios, it allows for the analysis of outcomes under varied conditions. Here, for each configuration C_i, we repeatedly conduct random simulation and collect a set of traces $\mathcal{T}_i = \{T_1, T_2, .., T_{m_i}\}$. The number of times the simulation needs to be conducted depends on the variance of the generated traces. The goal is to get a stable distribution of traces with respect to long-term fairness, and stop early for efficiency purposes. Specifically, we want to ensure that the estimated mean falls within the 5% confidence interval of the true value with a probability of over 95% [40]. Therefore, the simulation for one configuration will be run repeatedly until the recorded set of traces $\mathcal{T} = \{T_1, T_2, .., T_m\}$ satisfies $\frac{Z_{0.95} Std(\mathcal{LF}(\mathcal{T}))}{Mean(\mathcal{LF}(\mathcal{T}))\sqrt{m}} < 0.05$, where $Z_{0.95}$ is the coefficient for 95%-confidence level which is equal to 1.96 and $\mathcal{LF}$ is the selected long-term fairness metric.

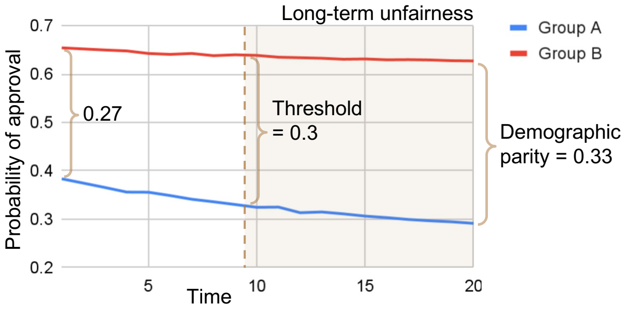


Fig. 4: An evolution trace of loan lending system showing long-term unfairness.

B. Long-term Fairness Evaluation

Given a fairness criteria, we perform long-term fairness evaluation on all the traces. For example, one possible simulation trace for a configuration of the loan-lending system is shown in Figure 4; here, the feedback loop causes a divergence of fairness between two population groups and eventually results in a violation of the demographic parity requirements.

We propose two different types of *long-term fairness criteria* that can be used to evaluate a trace for the presence of unfairness that arises over time:

- **Average increase in unfairness:** This long-term fairness of a trace $T = \langle s_1, s_2, ...s_k \rangle$ is computed by measuring the average fairness of a trace using fairness criteria $\mathcal{F}$, e.g., demographic parity, and then subtracting the unfairness of the initial state. When the goal is to analyze the trend of long-term fairness (e.g., equilibrium, oscillation), this criterion would be a suitable choice:
$$AvgInc_{\mathcal{F}}(T) = \frac{1}{k} \sum_{i=1}^{i=k} \mathcal{F}(s_i) - \mathcal{F}(s_1) \tag{1}$$
- **Maximum increase in unfairness:** This is measured by subtracting the unfairness of the initial state from the maximum unfairness exhibited by a trace for the given configuration. When the goal is to avoid any severe unfairness in the future, analyzing this criterion would be useful:
$$MaxInc_{\mathcal{F}}(T) = \max_{x \in T} \mathcal{F}(x) - \mathcal{F}(s_1) \tag{2}$$

While we focus mainly on the above two criteria, the developer may plug in other criteria into FAIRSENSE, such as *bias promptness* (i.e., how quickly the system reaches a biased state) or *violation frequency* (i.e., how many violations occur in a given time period).

VI. Sensitivity Analysis

Global sensitivity analysis technique has been successfully applied to investigate how uncertainty in the output of a model is attributed to different sources of model input [41]. This technique has also been applied to interpret the fairness in ML models [42, 43]. However, prior work focused on studying stationary ML models and evaluating the sensitivity of the training features on static fairness. Our approach, on the other hand, focuses on identifying parameters that are more likely to influence long-term fairness. The sensitivity analysis on simulation results could be used by the developer to understand the long-term fairness impact of the entire *design space* of system configurations. For example, if it is not clear how the environment behaves, different alternatives can be encoded as parameters, and a sensitivity analysis would identify whether that uncertainty in the environment is actually an important factor in influencing the long-term fairness of the model.

Exhaustively simulating the large space of configurations can become computationally prohibitive as the number of parameters and their possible values increases. Hence, we propose a sampling heuristic to reduce the number of configurations to explore while preserving the accuracy required for effective sensitivity analysis. In this section, we first describe the sensitivity analysis method and then the sampling heuristic.

A. Sensitivity Analysis with Regression Modeling

Regression analysis has been shown to be a good match for investigating parametric importance and sensitivity [27]. Compared with other sensitivity analysis methods like elementary effects methods and variance-based methods [44, 45], the coefficient for each input variable in a regression model can be directly interpreted as the exact effect that the input variable brings to the output variable.

The idea is to learn a regression model that explains how the response is influenced by different options. In our case, the response is the long-term fairness measured in simulation and the options are possible values for the system and environment parameters. The model, a *standardized multiple linear regression model*, is trained based on the measured fairness results of simulations with different configurations. The model's coefficients then indicate the influence of each parameter on the fairness result of the simulation.

The regression model has the structure shown below in Equation 3, where y is the output variable, i.e., the long-term unfairness score. As introduced in previous section, p_i is the option for the ith parameter in a configuration $C = \langle p_1, p_2, p_3, ..., p_n \rangle$. β_i and $\beta_{i,j}$ are the coefficients for the terms. We include both individual terms, which assess the impact of each parameter (P_i) in isolation, and pair-wise interaction terms, which explore the combined effects of any two parameters (e.g., the interaction of P_i and P_j). This dual approach also allows us to understand the independent contribution of every single parameter as well as the interplay between different parameters that influence long-term fairness. While interactions might be possible among more than two parameters at the same time in some case studies, our current focus is on the most salient pair-wise interactions, balancing the depth of analysis with model interpretability.

$$y = \sum_{i=1}^{n} \beta_i \cdot p_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \beta_{i,j} \cdot p_i \cdot p_j + \epsilon \tag{3}$$

We use Analysis of Variance (ANOVA) [46] for the model, which allows us to identify which parameters contribute in a statistically significant way (using the common significance threshold of $p < 0.05$). Then, we quantify the effect sizes of the statistically significant coefficients using the sums of squares and the eta-squared (η^2) derived from ANOVA. We evaluate whether an individual parameter (or its interaction with another parameter) is impactful based on Cohen's well-established guidelines [47] ($\eta^2 \geq 0.01$, ≥ 0.06, and ≥ 0.14 indicate a small, medium, and large effect, respectively), and further rank them according to their effect sizes. We additionally report the model's R^2 value as a measure of fit, that indicates how much variance in the long-term fairness the model can explain in terms of the parameters and their interactions.

B. Sampling Heuristic

The presence of numerous parameters and their potential values lead to an exponentially large configuration space. Given the time-intensive nature of Monte-Carlo simulations and the typical constraints of real-world development, simulation of all configurations can be challenging. To address this, we propose *covering array sampling* on the configuration space to reduce the number of configurations to be explored, while ensuring a diverse and comprehensive coverage of the parameter values.

Covering array sampling technique is widely used in software testing [48] to achieve an adequate coverage of program behavior with a small number of carefully selected inputs. A covering array, characterized by a coverage number g, is a structured method to select combinations of a set of n parameters' values. The key feature of a covering array is that it guarantees the inclusion of every possible combination of any set of g factors' values from these n parameters at least once; i.e., all possible g-factor interactions are covered within the array. For example, in a 2-coverage array (i.e., $g = 2$), an array of combinations is created such that every possible pair of parameter values is included. This approach effectively minimizes the number of configurations FAIRSENSE needs to simulate. For example, a 2-coverage array for ten binary parameters can cover all pairwise interactions with only 12 configurations – a significant reduction compared to the $2^{10} = 1024$ configurations needed when enumerating all.

VII. EXPERIMENTAL SETUP

To demonstrate the utility and applicability of FAIRSENSE, we conducted three case studies and answered the following research questions:

- **RQ1.** What are system and environmental parameters that significantly impact the long-term fairness of a system?
- **RQ2.** What trade-offs among the parameters does FAIRSENSE identify?
- **RQ3.** How effective is the sampling heuristics in reducing the number of configurations explored while retaining the accuracy?

RQ1 is intended to demonstrate that FAIRSENSE can potentially reduce the system developer's effort by identifying parameters that have significant influence on long-term fairness. RQ2 shows that FAIRSENSE can be used by the developer to navigate the trade-off space between fairness and utility, to identify a design solution that acceptably meets both qualities. RQ3 evaluates the efficiency of FAIRSENSE when the sampling heuristics is used.

The remainder of this section describes the case study systems and the experimental settings.[1] The space of possible configurations for each system is shown in Table I: In total, *Loan lending* has 768 possible configurations, *Opioid risk scoring* has 168, and *Predictive policing* has 105. The code of FAIRSENSE and the results of the case studies are available in our replication package.[2]

A. Loan Lending

ML-enabled systems are used to predict the creditworthiness of people and approve or reject loan applications. An ML model is trained on personal data (e.g., education, income, sex, race, credit score) of individuals, and then predicts the probability of an individual to repay or default. The decision-making system adopts a policy (e.g., a threshold) that approves or rejects loans based on the predictions. **Potential feedback loop:** Section III presented how a possible feedback loop might exhibit long-term unfairness against a group. **ML-enabled system:** We leverage the predictive models defined by [19], which determine credit score thresholds dynamically for different groups, and then employ the thresholds for loan approval and refusal. **Environment:** We use the FICO dataset [33] to present all the potential loan applicants. Detailed dynamics are described in Section IV.A.

Experimental settings: Following Liu et al. [2] and D'Amour et al. [19], we evaluated fairness between two racial groups – White and Black. The fairness metric is given by the demographic parity and the mean credit score difference between the groups. For long-term fairness metric, we leveraged the average increase (Eq. (1)) and the maximum increase (Eq. (2)). The developer may choose to apply FAIRSENSE using any of the defined long-term fairness metric; we used the maximum increase of demographic parity for the sensitivity analysis. To further explore the trade-off between long-term fairness and utility, we defined the profit of the bank as the utility metric.

B. Opioid Risk Scoring

Opioids are a class of medicine that are frequently used for pain management. Common opioids such as oxycodone, morphine, fentanyl, and methadone, are used to reduce pain, but can cause overreliance and addiction, which is called Opioid Use Disorder (OUD). Recent data shows a worsening situation with over 100,000 annual deaths from OUD for the first time in 2021 [49]. To reduce OUD, a Prescription Drug Monitoring Program (PDMP) is mandated in each state, which measures the opioid risk score of individuals. Many PDMPs use an ML-based application called NarxCare, which produces a numeric risk score (000-999) for individuals [1]. The risk score is shown to the doctors and pharmacists, based on which they can modify the prescription.

Potential feedback loop: The NarxCare ML model is trained using patients' medical records such as past opioid

[1] Further details of the case studies (i.e., ML model, decision maker, environmental dynamics, and configuration parameters) are presented in our supplemental material shared in the artifact: https://github.com/cmu-soda/FairSense/tree/main/SupplementalMaterial

[2] https://github.com/cmu-soda/FairSense/

TABLE I: The parameters and their possible values. The middle and right column contain the configuration parameters taking part in the systems' decision-making and the interactions between systems and environments respectively.

Case study	System parameter	Environmental parameters
Loan lending	Agent: max-util, eq-op Bank utility func param: -10, -9 ... -3	Score update-repay: 8, 12 ... 20 Score update-default: -40, -32 ... -16 Shift function mode: expected, normal, aggressive
Opioid risk scoring	Model type: XGBoost, MLP Doctor threshold: 0.3, 0.4, 0.45, 0.5, 0.55, 0.6, 0.7	Shift function mode - hospital visits: expected, equal, normal, aggressive Shift function mode - prescription: expected, normal, aggressive
Predictive policing	Model type: SEPP Hotspot number: 50	Discovery rate - hotspot cell: 0.8, 0.85 ... 1.0 Discovery rate - other cell: 0.2, 0.25 ... 0.5 Hot spot effect area range: 1, 2, 3

usage, number of pharmacies visited, number of prescribers, overlap from different prescribers, etc., [50]. The NarxCare software uses these attributes to predict a risk score [50]. Underrepresented patients including women and racial groups with complex medical conditions can have artificially inflated scores. Consequently, they can be denied a legitimate amount of opioids, suffer physical or mental debilitation, and be coerced into illegal activities [1], which may further increase their risk score in the long term [51]. **ML-enabled system:** The ML model adopted from [52] uses patients' medical records to predict opioid risk scores and then doctors make prescriptions based on scores. **Environment:** We use the publicly available Mimic-IV v2.2 dataset [53] to represent the potential patients. Patients will visit hospitals to get prescriptions; when a patient get an insufficient prescription, their further doctor visits may be affected. The shift function is defined to update the number of hospital visits by patients based on their risk scores; higher the risk score a patient has, more visits they likely need [1].

Experimental settings: We computed the fairness between two gender groups – male and female. The long-term fairness metrics are the maximum increase (Eq. (2)) in the gap between the mean opioid disorder risk scores of two groups, and the average increase (Eq. (1)) of the gap between ML model's performance metrics for two groups: Prediction accuracy and F1 score. The first one is used for sensitivity analysis of the case study, and others are used in the trade-off analysis. The utility metrics are defined by the daily average of ML model's performance metrics: prediction accuracy and F1 score.

C. Predictive Policing

Police departments are widely using ML algorithms to predict crime hotspots and deploy police based on the prediction. In a survey by the National Institute of Justice, over 70% of agencies have reported to use predictive crime maps [54]. Lum and Isaac experimented on the self-exciting point process (SEPP) model and data collected from Oakland, CA, to demonstrate over-policing in minority neighborhoods [55].

Potential feedback loop: The model predicts crime hotspots based on the crime incident records from the past [29]. Because of the crime increase for a short period or inaccurate prediction, more police may be sent to a certain location. This causes more crime discovery in that region, which, in turn, may influence the ML model to predict those regions as hotspots in the future. **ML-enabled system:** The crime prediction model is adopted from [29], which predicts crime intensity of each location for the next day. Based on the intensities, the decision maker allocates more police force to the top 50 cells (i.e., locations) with the highest intensities. **Environment:** Adopted from [29], we synthesized all the crime incidents that will take place. The shift function is used to derive the incidents occurring each day. The projection function is defined as a stochastic function that determines which incidents would be discovered based on hotspot allocation; the incidents in the neighborhoods of hotspots are more likely to be discovered.

Experimental settings: We measured the fairness of police allocation across districts by calculating the average pairwise Relative Percentage Difference (RPD) between districts' overpolicing scores. Following Akpinar et al. [29], a district's overpolicing score is determined by the relative number of predicted hotspots. For the long-term fairness metrics, we computed the maximum and average increase of the district-wise allocation unfairness. The former metric is used for sensitivity analysis in our evaluation. For system utility, we considered three metrics: The total number of discovered incidents, the mean of daily percentages of discovered incidents, and the number of correct predicted hotspots.

VIII. Evaluation Results

A. RQ1: Sensitivity Analysis to Identify Impactful Parameters

We answer RQ1 through the sensitivity analysis results on the three case studies. We identify the most impactful parameters and their interactions, and show that only a small subset of the parameters are influential on long-term fairness.

a) Loan Lending: FairSense collected 6,844 traces in total for 768 possible configurations. On average, every trace has 3.2% increase in the unfairness (demographic parity) at the final time step compared to the initial step. Around 45% of the configurations have more than 5% increase on average, and

TABLE II: Top 5 impactful regression terms for loan lending. A parameter can be numerical or categorical. Categorical parameters are transformed into binary variables using one-hot encoding. The specific values of the categorical parameter are listed in the second column. The pair of parameters represent interaction terms.

Terms	Dummies	Coefficient	Sum Sq.	$\eta^2(\%)$
1 Agent	max-util	-2.51E-02***	1.19E-01***	76.52
2 Bank utility param		8.45E-03***	1.35E-02***	8.72
3 (Agent, bank utility p.)	max-util	-8.38E-03***	1.35E-02***	8.69
4 (Score update–d., agent)	max-util	3.35E-03***	2.16E-03***	1.39
5 Score update–d.		-3.38E-03***	2.16E-03***	1.39

p-values: ***$p < .001$; **$p < .01$; *$p < .05$

TABLE III: Top 5 impactful terms for opioid risk scoring.

Terms	Dummies	Coefficient	Sum Sq.	η^2(%)
1 ML model	XGBoost	1.12E-02***	6.35E-03***	96.80
2 (ML model, Shift function–prscrptn.)	(XGBoost, normal) (XGBoost, aggressive)	2.53E-03*** 2.65E-03***	6.28E-05***	0.96
3 Shift function–hospital	normal aggressive equal	3.00E-04 5.70E-05 -3.03E-04	3.92E-05***	0.60
4 Shift function – prescription	normal aggressive	-2.37E-03*** -2.38E-03***	3.82E-05***	0.59
5 (ML model, Shift function–hospital)	(XGBoost, normal) (XGBoost, aggressive) (XGBoost, equal)	-4.15E-04* -3.21E-04 -1.77E-03***	1.92E-05***	0.29

p-values: ***p < .001; **p < .01; *p < .05

TABLE IV: Top 5 impactful terms for predictive policing.

Terms	Coefficient	Sum Sq.	η^2(%)
1 Discovery rate–other	-3.46E-02***	1.26E-01***	68.92
2 Discovery rate–hotspot	8.99E-03***	8.48E-03***	4.64
3 (Discovery rt–hotspot, discovery rt–o.)	1.75E-03	3.21E-04	0.18
4 (Discovery rt–o., hotspot area range)	9.40E-04	9.30E-05	0.05
5 Hotspot area range	-7.18E-04	5.40E-05	0.03

p-values: ***p < .001; **p < .01; *p < .05

around 17% of configurations have $\geq$ 10% increase on average, demonstrating long-term fairness issues in loan lending.

The fitted regression model explains the variance well ($R^2 = 0.970$). Table II shows the terms (i.e., individual parameters and their interactions) with the top 5 effect sizes (sum of squares) in the regression model, all statistically significant.

Regarding RQ1, there are only 5 out of 15 terms that can be considered as impactful ($\eta^2 \geq 0.01$). Among them, the choice of *agent* is the most dominant factor ($\eta^2 = 0.76$) influencing long-term fairness; max-utility agent can greatly improve long-term fairness. The *bank utility parameter* and its interaction with the *agent* together explain $\sim$17% of the variance in the total sum of squares; both have a moderate influence on long-term fairness. One standard deviation increase in *bank utility parameter* can decrease long-term fairness by 0.00845, indicating that it is fairer if a bank makes loan decisions conservatively (i.e., smaller *bank utility parameter*). However, with the existence of max-utility agent, the individual effects of *bank utility parameter*, *score update parameters*, and *shift function mode* will be offset. This illustrates how sensitivity analysis can highlight the small number of decisions (or sources of uncertainty of environment parameters) that require careful attention, where one factor dominates, four more have moderate influences, and 10 more are largely negligible.

b) Opioid Risk Scoring: We collected 1,780 traces in total for 168 possible configurations. The average initial unfairness score (average risk gap between groups) for all configurations is 0.253%. However, in the final step of the simulation, the average unfairness score increases to 2.859%. More than 25% of configurations have an increase of more than 3.2%.

The fitted regression model again explains the variance very well ($R^2 = 0.995$). We show the ranking of the terms for this case study in Table III. **Regarding RQ1**, only 1 out of 10 parameters is impactful ($\eta^2 \geq 0.01$): the choice of *ML model*, explaining around 97% of all variance. Choosing XGBoost model would significantly amplify long-term unfairness. Although the other terms have much smaller effects ($\eta^2 < 0.01$), we noticed that two environmental parameters in the configuration, *Shift function–hospital* and *Shift function–prescription* and their interaction terms with *ML model* also have small but statistically significant influences, that developers might want to consider when tuning the model.

c) Predictive Policing: We explored 2,304 traces in total for 105 possible configurations. The average initial unfairness score is 0.739%. In the last step of the simulation, the average score for all configurations increases to 21.522%. Around 22% of all configurations see an increase of more than 25%.

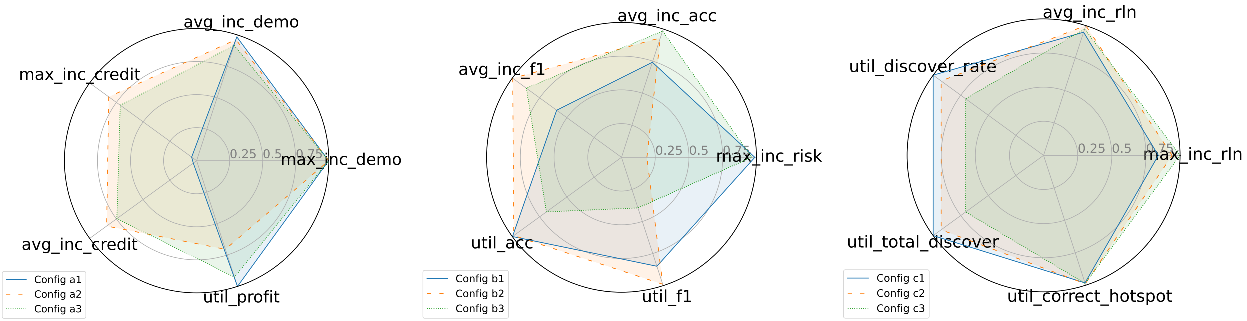
The fitted regression model again explains most of the variance ($R^2 = 0.738$, considered high in sensitivity analysis). Table IV summarizes the ranking of the terms' effect sizes for predictive policing. **Regarding RQ1**, 2 of 5 terms ($\eta^2 \geq 0.01$) are considered impactful. The incident discovery rate for areas except the hotspots (*Discovery rate–other*) has the greatest impact on long-term fairness, explaining 69% of the variance. The increase in *Discovery rate–other* can greatly enhance long-term fairness. The second impactful parameter is the incident discovery rate for hotspot areas (*Discovery rate–hotspot*), explaining about 5% of the variance. Compared to *Discovery rate–other*, the increase in the discovery rate for hotspot areas has the opposite effect, it reduces long-term fairness. This suggests long-term fairness might be improved if the police force is not only concentrated in hotspot areas predicted by the SEPP model but also partially distributed to other regions.

Summary. We observe that only a small subset (10-40%) of the parameters impact ($\eta^2 > 0.01$) long-term fairness. These results demonstrate that FAIRSENSE can be used to identify the most impactful parameters and allow the developer to allocate their design effort on them.

B. RQ2: Trade-off Between Long-Term Fairness and Utility

The fairness requirement of the system can be defined using multiple metrics. In addition, the developer would typically care about the utility of the system, e.g., financial profit of the bank. Optimizing exclusively for one long-term fairness metric may overlook this aspect, as the configuration yielding the lowest unfairness scores does not necessarily guarantee optimal utility. Therefore, it is essential to identify the trade-off between multiple fairness criteria and the utility of the system. To this end, given the long-term fairness metric and utility metric, FAIRSENSE finds a set of Pareto-optimal configurations. These configurations represent scenarios where any improvement in utility would result in a decrease in long-term fairness, and vice versa. In this section, we demonstrate the trade-offs that exist in the case studies.

a) Loan Lending.: Figure 5a shows the trade-off among 4 long-term fairness metrics and 1 utility metric (defined in



(a) Loan lending case study.

(b) Opioid risk prediction case study.

(c) Predictive policing case study.

Fig. 5: The radar plots visualizing trade-offs in three Pareto-optimal configurations for each case study. All values were scaled to [0,1]. A higher value implies better performance.

Section VII) for 3 Pareto-optimal configurations.[3] Configuration *a1* optimizes maximum increase of demographic parity (*max_inc_demo*) and financial profit of the bank (*util_profit*), while configuration *a2* and *a3* optimize average increase of credit score gap (*avg_inc_credit*) and *util_profit*. A detailed examination of *a2* and *a3* reveals a discernible conflict between *avg_inc_credit* and *util_profit*, highlighting the inherent trade-offs in optimizing these two metrics simultaneously. Interestingly, configuration *a1* excels in both *max_inc_demo* and *util_profit*, suggesting that there might be no evident conflict between them. If the developer places a higher emphasis on utility while optimizing all fairness metrics, *a3* could be chosen, as it has nearly optimal utility and good fairness.

b) Opioid Risk Scoring.: Figure 5b shows the trade-offs for 3 Pareto-optimal configurations. Configuration *b1* optimizes maximum increase in risk gap (*max_inc_risk*) and utility of accuracy (*util_acc*). *b2* is Pareto-optimal in both (i) optimizing average increase in accuracy gap (*avg_inc_acc*) and *util_acc*, (ii) optimizing average increase in f1 gap (*avg_inc_f1*) and *util_acc*. Notably, *b3* optimizes two long-term fairness metrics *max_inc_risk* and *avg_inc_acc*.

Looking at *b1* and *b2*, we find that achieving the highest *util_acc* is possible while addressing any of the fairness metrics involved. Furthermore, *b3* demonstrates that all three fairness metrics can attain favorable scores concurrently. Beyond these three Pareto-optimal configurations, it is surprising to find that there always exists a nearly optimal configuration for any pair of metrics. This observation suggests an absence of significant pairwise conflicts in this case study.

c) Predictive Policing.: Figure 5c illustrates the trade-offs for 3 Pareto-optimal configurations that optimize maximum increase of relative number of hotspots (*max_inc_rln*) and total number of discovered incidents (*util_total_discover*). The comparison of these 3 configurations reveals a conflict between the fairness metric (*max_inc_rln*) and utility metrics (*util_total_discover* and *util_discover_rate*). Notably, as we move from *c1* to *c2* and then to *c3*, there is a monotonic increase in *max_inc_rln*, accompanied by a corresponding decrease in both utility metrics. Furthermore, the magnitude of changes observed in these metrics appears to align with the principle of diminishing marginal utility. While *c3* shows only a slight improvement in *max_inc_rln* over *c2*, it experiences a substantial reduction in the utility metrics. Conversely, *c1* marginally outperforms *c2* in utility, but at the cost of a significant reduction in *max_inc_rln*.

Summary. Overall, regarding RQ2, these observations highlight the complex and often delicate balance between maximizing long-term fairness and utility in system design, demonstrating the need for careful investigation of the trade-off between specific metrics during the design stage.

C. RQ3: Performance Evaluation

In this section, we evaluate the efficiency of FairSense in exploring a potentially large space of configurations for simulation through sampling.

Our hypothesis is that by applying covering array sampling, FairSense can avoid simulating every configuration while maintaining a high accuracy of regression analysis and the identification of the most significant variables or interaction terms. To test this, we consider the baseline as the ranking of the statistically significant (i.e., $p \leq 0.05$) terms, which is computed by analyzing every possible configuration from the given configuration space. Then, we compute the ranking of the same terms after applying both 2-coverage and 3-coverage sampling. We measure the time consumption of each sampling strategy, and computed their effectiveness using ranking similarity between the baseline ranks and the ranks found using sampling. The ranking similarity metrics we used are Rank Biased Overlap (RBO) with persistence 0.8 [56] and Kendall Tau (Tau) [57].

Results for the three case studies, shown in Table V, demonstrate that both 2-coverage and 3-coverage sampling require only a significantly smaller number of configuration simulations to obtain a regression model with little loss of model fit (R^2). The models trained on samples also identify impactful terms with a ranking very similar to the baseline derived from analyzing all configurations. Specifically, the analysis results of 3-coverage sampling for loan lending and opioid risk scoring benchmarks are almost as good as the baseline while saving 80% and 50% simulation effort (the

[3] For visualization, we omitted showing all the Pareto-optimal configurations for the case studies in the plot. The assignments of all the configurations used in this section are presented in supplemental material.

TABLE V: Comparison of efficiency of FairSense with baseline method.

Case Study	2-coverage					3-coverage					Baseline (no sampling)		
	R^2	*RBO*	*Tau*	*# configs*	*time*	R^2	*RBO*	*Tau*	*# configs*	*time*	R^2	*# configs*	*time*
Loan lending	0.916	0.913	0.778	33	6min	0.968	0.967	0.889	135	24min	0.970	768	2.3hr
Opioid risk scoring	0.976	0.867	0.619	28	39min	0.994	0.883	0.714	85	2hr	0.995	168	3.9hr
Predictive policing	0.631	1.0	1.0	35	≈39.5hr	/	/	/	105	/	0.738	105	≈118.6hr

predictive policing case study only has three parameters, so 3-way coverage samples all possible configurations). With 2-coverage sampling, simulation effort is reduced to 5%, 16%, 33% compared to analyzing all configurations, while the resulting models still explain the variance similarly well.

In summary, to answer RQ3, the results show that the impactful parameters are still identifiable in the sensitivity analysis with carefully sampled configurations. The results in Table V demonstrate that this heuristics can eliminate a significant number of configurations (and traces associated with them) while retaining the performance of fitted models.

IX. Threats to Validity

Fidelity of the environment models. The validity of simulation results depends on the fidelity of the environment model used for simulation. Instead of coming up with environmental parameters and dynamics on our own, we inferred these from the existing studies and analyses of these systems (loan lending: [2, 19]; opioid risk scoring [1, 28, 52]; predictive policing: [3, 29]). Although a process for developing environmental models is beyond the scope of this paper, we provide a discussion of existing methodologies that could be adopted for this purpose in Section XI.

Validity of simulation results with respect to the real world. Simulation in FairSense is a type of "what-if" analysis, estimating the potential impact of options for various systems and environmental parameters on fairness over time.

In this paper, we do not validate the accuracy of the simulation with regard to the real world. Such validation would be very difficult and is orthogonal to the contributions of this paper. In theory, one way to validate the accuracy of our analysis results would be to deploy and execute a system under all possible configurations, and collect the resulting traces as the ground-truth data. Doing so, however, would be extremely challenging (and ethically questionable) [58], especially for socio-technical systems where the system directly interacts with and influences users in the real world, sometimes negatively. To the best of our knowledge, no such ground-truth data is available for the kind of systems we study in this paper.

Instead, we provide a comparison against other prior analyses of long-term fairness for these systems. *The comparison shows that the conclusions drawn from the simulation about the real world are consistent with those from the other analyses.* Although this does not offer the same level of validation of the simulator's ability to correctly model real-world behaviors against real-world observations, we believe that it provides evidence that the simulation is able to produce meaningful predictions about the potential impact of the parameters.

Loan lending: Liu et al. [2] propose an analytical model that estimates the impact of different ML policies (e.g., equal opportunity vs. maximizing utility) on long-term fairness, studying a loan lending system based on the same dataset as the one used in this paper. Their model shows that the *eq-op* agent, which optimizes for equality among different groups in the short-term, may actually result in long-term unfairness, in a somewhat counter-intuitive and surprising result [2, Theorem 3.4]. Consistent with their conclusion, our sensitivity analysis results (Table II) also show that *max-util* agent is the most effective in reducing long-term unfairness, and hence fairer than the *eq-op* agent. Moreover, our analysis presents additional information, such as how the choice of agent together with other parameters, e.g., utility of the bank (Table-II, term 3), affect long-term fairness.

Predictive policing: Ensign et al. [3] propose an analytical model that estimates the occurrence of a feedback loop in the same predictive policing system that we studied. In particular, their model shows that over time, the system converges to a biased allocation scheme that assigns police only to the neighborhoods with the highest number of observed incidents and ignores those that are historically not hotspots. Their model also indicates that this bias can be mitigated by deliberately allocating resources to those non-hotspots, to increase the number of observations in those neighborhoods. These findings are consistent with our analysis results: The positive coefficient of discovery rate-hot spot (Table-IV, term 2) confirms that the dominant observation of incidents in hotspots would exacerbate unfairness [3, Sec. 4.2.1] On the other hand, the large negative coefficient of discovery rate-other (Table-IV, term 1) marks the importance of improving discovery rates of non-hot spot area for long-term fairness [3, Sec. 4.2.2].

Opioid scoring: Although fairness in ML-based opioid risk scoring has been studied [28, 51, 59], to the best of our knowledge, no prior work studies long-term fairness issues. Adam et al. [60] conducted a simulation study on the same dataset that we used; specifically, they created artificial data drift and investigated the impact of different ML model retaining methods on performance. However, their work did not consider the impact of the system-decision on the environment and thus is not comparable to ours.

X. Related Work

Development of Fair ML-Enabled Systems. Understanding and improving algorithmic fairness in ML models has received significant attention in the recent past [6, 7, 31, 61–64]. Many bias mitigation techniques have been proposed for ML algorithms [12, 13, 33, 65–70]. The mitigation techniques can be categorized into preprocessing [12, 66, 68], in-processing [69–71], and post-processing methods [33, 42], depending on where the mitigation is applied. However, several challenges remain for the development of fair ML-enabled systems [6, 61].

Prior works showed that fairness-enhancing interventions can fail due to fluctuations in dataset characteristics, preprocessing methods, etc. [7, 14, 72, 73]. Holstein et al. outlined the challenges industry product teams face in developing fair systems [74]. Thus, several software engineering techniques have been proposed for testing [5, 8, 9, 38, 75–78], verifying [10, 11, 79–82], and achieving the accuracy-fairness trade-offs [14, 83]. However, these works focus on fairness under static settings and do not consider long-term fairness.

Long-term Fairness. Gohar et al. [84] conducted a survey on different notions of long-term fairness and created a taxonomy. D'Amour et al. [19] conducted simulations to show that static analysis is not sufficient to capture long-term fairness issues. Researchers focused on the predictive policing model to investigate the divergence of fairness over time [3, 29, 55]. Algorithmic solutions have also been proposed by considering the temporal factor of fairness in the sequential selection process [85, 85–87]. Albarghouthi and Vinitsky [88] proposed a runtime specification language to monitor fairness statistics and provide warnings for violations. Henzinger et al. [20, 89] built retrospective analysis and proposed a runtime statistical estimator to avoid long-term unfairness. Several ML algorithms have been proposed for optimizing a long-term fairness objective under certain assumptions or fixed environmental dynamics [58, 90, 91]. However, no prior work has focused on analyzing the influence of system parameters on long-term fairness. Understanding the dynamics of fairness requires modeling the system and its context [4, 92, 93], and difficult to achieve through static analysis [20].

Feedback Loops. An emerging problem for ML systems is to ensure robustness in presence of feedback loops [18, 23, 60, 94, 95]. O'Neil [17] explained several harmful feedback loops in sociotechnical systems at length. Pagan et al. [22] classified the different types of feedback loops in ML-enabled systems. With an emphasis on accuracy, most of the ML research in the area focuses on data bias and distribution shifts induced from feedback [96–98]. However, designing an ML system for long-term fairness would need adaptive design and mitigation strategies [99]. Recently, Reader et al. [100] proposed a system theory-based approach to quantify feedback in sociotechnical systems. Martin Jr et al. [21] also recommend system-level analysis and in-depth understanding of the societal context to identify feedback loops. To that end, we have built a simulation-based framework to analyze long-term fairness issues.

XI. Discussions

The simulation-based analysis in FairSense relies on an environmental model that describes how the environment evolves in response to the system output. FairSense is specifically designed to enable reasoning about interactions between the system and the environment when certain details about the environment are unknown at the design time; these details can be encoded as environmental parameters, which are then explored by the tool to provide insights into their impact. This allows the system designer to identify which uncertainty in the environment is most important to focus their efforts on, rather than wasting efforts on aspects of the environment that matter little for fairness. That is, the challenge of creating an accurate environment model is not a limiting factor, but a key motivation for sensitivity analysis in FairSense.

Although our environmental models were derived from prior work (as described in Section VII), in practice, creating environment models would involve a requirements engineering process understanding relevant environment behaviors from stakeholders and domain experts. Beyond requirements engineering techniques, the *system dynamics* community has long studied a rich set of methods for building, simulating and analyzing environmental models in socio-technical systems [15, 16]. One promising method is a type of modeling notation called *causal loop diagram (CLD)*, which is used to model the environment as a set of *variables* (parameters in our terminology), relationships between those variables (i.e., shift dynamics), and possible feedback loops that arise from them [101]. CLDs have been used to model and simulate the environment in a wide range of domains, such as economics, social sciences, ecology, and public policy; methodologies for developing CLDs are also well-studied [102–105]. CLDs have also been adopted in requirements engineering (to model the impact of software on sustainability, for example [106]), although not yet in the context of fairness, as far as we know.

A complementary approach to improving the quality of the models used in FairSense is *runtime monitoring*. Once a system is deployed, observations collected from the environment (i.e., new data samples) could be used to evaluate whether the environment model is consistent with the actual environmental behavior. If there are discrepancies (possibly due to inaccurate modeling or data drift), this information could be used to update the model and improve its fidelity. Recent work in runtime monitoring for fairness [20, 88, 89] could be adopted for this purpose as part of a framework like FairSense.

XII. Conclusions

Understanding the impact of design decisions for ML-enabled systems has received much attention recently. Many ML interventions have been proposed to improve algorithmic fairness in static settings; however, the long-term impact of such interventions is still unclear, as the system interacts with the environment over time, possibly forming unexpected feedback loops. Precisely understanding the environment to accurately predict long-term fairness is challenging. To this end, we have proposed FairSense, to aid developers in identifying and understanding design decisions and environmental factors that impact long-term fairness.

Acknowledgments

We'd like to thank our anonymous reviewers for their feedback, which greatly helped improve this paper. This work was supported in part by the National Science Foundation Awards No. 2131477, 2144860, and 2233871.

References

[1] E. Jatho, L. Mailloux, S. Rismani, E. Williams, and J. A. Kroll, “System safety engineering for social and ethical ML risks: A case study,” *arXiv preprint arXiv:2211.04602*, 2022.

[2] L. T. Liu, S. Dean, E. Rolf, M. Simchowitz, and M. Hardt, “Delayed impact of fair machine learning,” in *ICML*, 2018.

[3] D. Ensign, S. A. Friedler, S. Neville, C. Scheidegger, and S. Venkatasubramanian, “Runaway feedback loops in predictive policing,” in *Conference on Fairness, Accountability and Transparency FAT*, 2018.

[4] A. Farahani, L. Pasquale, A. Bennaceur, T. Welsh, and B. Nuseibeh, “On adaptive fairness in software systems,” in *International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2021.

[5] S. Galhotra, Y. Brun, and A. Meliou, “Fairness testing: testing software for discrimination,” in *ESEC/FSE*, 2017.

[6] S. Biswas and H. Rajan, “Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness,” in *Proceedings of the ESEC/FSE*, 2020.

[7] S. Biswas and H. Rajan, “Fair preprocessing: Towards understanding compositional fairness of data transformers in machine learning pipeline,” in *The 29th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2021.

[8] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, “Black box fairness testing of machine learning models,” in *Proceedings of the 27th ACM ESEC/FSE*, 2019.

[9] S. Udeshi, P. Arora, and S. Chattopadhyay, “Automated directed fairness testing,” in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018.

[10] P. Zhang, J. Wang, J. Sun, G. Dong, X. Wang, X. Wang, J. S. Dong, and T. Dai, “White-box fairness testing through adversarial sampling,” in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, 2020, pp. 949–960.

[11] S. Biswas and H. Rajan, “Fairify: Fairness verification of neural networks,” in *ICSE’2023: The 45th International Conference on Software Engineering*, 2023.

[12] J. Chakraborty, S. Majumder, and T. Menzies, “Bias in machine learning software: Why? how? what to do?” ser. ESEC/FSE 2021, 2021.

[13] J. Chakraborty, S. Majumder, Z. Yu, and T. Menzies, “Fairway: A way to build fair ml software,” in *Proceedings of the 28th ESEC/FSE*, 2020.

[14] G. Nguyen, S. Biswas, and H. Rajan, “Fix fairness, don’t ruin accuracy: Performance aware fairness repair using automl,” in *ESEC/FSE’2023: The 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2023.

[15] J. D. Sterman, *Business dynamics: systems thinking and modeling for a complex world*. McGraw-Hill, 2000.

[16] D. H. Meadows, *Thinking in systems: A primer*. chelsea green publishing, 2008.

[17] C. O’Neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2017.

[18] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete problems in ai safety,” 2016.

[19] A. D’Amour, H. Srinivasan, J. Atwood, P. Baljekar, D. Sculley, and Y. Halpern, “Fairness is not static: deeper understanding of long term fairness via simulation studies,” in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020.

[20] T. Henzinger, M. Karimi, K. Kueffner, and K. Mallik, “Runtime monitoring of dynamic fairness properties,” in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 2023.

[21] D. Martin Jr, V. Prabhakaran, J. Kuhlberg, A. Smart, and W. S. Isaac, “Extending the machine learning abstraction boundary: A complex systems approach to incorporate societal context,” *arXiv preprint arXiv:2006.09663*, 2020.

[22] N. Pagan, J. Baumann, E. Elokda, G. De Pasquale, S. Bolognani, and A. Hannák, “A classification of feedback loops and their relation to biases in automated decision-making systems,” in *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, 2023.

[23] S. Biswas, Y. She, and E. Kang, “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*, 2023.

[24] M. Jackson, “The world and the machine,” in *International Conference on Software Engineering (ICSE)*. ACM, 1995, pp. 283–292.

[25] C. A. Gunter, E. L. Gunter, M. Jackson, and P. Zave, “A reference model for requirements and specifications,” *IEEE Softw.*, 2000.

[26] C. Z. Mooney, *Monte Carlo Simulation*. SAGE Publications, 1997.

[27] A. McCulloch, “Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models,” *Journal of the Royal Statistical Society Series A: Statistics in Society*, 2005.

[28] A. E. Kilby, “Algorithmic fairness in predicting opioid use disorder using machine learning,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.

[29] N.-J. Akpinar, M. De-Arteaga, and A. Chouldechova, “The effect of differential victim crime reporting on predictive policing systems,” in *Proceedings of the ACM FAccT*, 2021.

[30] E. Kang, “The role of environmental deviations in engineering robust systems,” in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2021, pp. 435–438.

[31] S. Verma and J. Rubin, “Fairness definitions explained,” in *Proceedings of the International Workshop on Software Fairness, FairWare*, 2018.

[32] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, “Fairness through awareness,” in *Proceedings of the 3rd innovations in theoretical computer science conference*, 2012.

[33] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in *Advances in neural information processing systems*, 2016.

[34] N. Byrnesarchive, “Artificial intolerance,” *MIT Technology Review*, 2016. [Online]. Available: https://www.technologyreview.com/2016/03/28/246328/artificial-intolerance

[35] M. Weber, M. Yurochkin, S. Botros, and V. Markov, “Black loans matter: Distributionally robust fairness for fighting subgroup discrimination,” *arXiv preprint arXiv:2012.01193*, 2020.

[36] D. of Justice, “Justice department secures over 31 million from city national bank to address lending discrimination allegations,” *Office of Public Affairs*, 2023.

[37] E. Nedlund, “Apple card is accused of gender bias. here’s how that can happen,” *CNN Business*, 2019. [Online]. Available: https://www.cnn.com/2019/11/12/business/apple-card-gender-bias/index.html

[38] F. Tramer, V. Atlidakis, R. Geambasu, D. Hsu, J.-P. Hubaux, M. Humbert, A. Juels, and H. Lin, “Fairtest: Discovering unwarranted associations in data-driven applications,” in *2017 IEEE EuroS&P*, 2017.

[39] https://www.businessinsider.com/personal-finance/what-is-hard-inquiry-how-affect-credit-score, 2023.

[40] M. J. Gilman, “A brief survey of stopping rules in monte carlo simulations,” in *Proceedings of the Second Conference on Applications of Simulations*. Winter Simulation Conference, 1968.

[41] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola, *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[42] P. A. Grabowicz, N. Perello, and A. Mishra, “Marrying fairness and explainability in supervised learning,” in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[43] B. Ghosh, D. Basu, and K. S. Meel, ““how biased are your features?”: Computing fairness influence functions with global sensitivity analysis,” in *Proceedings of the 2023 ACM Conference on FAccT*, 2023.

[44] *Elementary Effects Method*. John Wiley & Sons, Ltd, 2007, ch. 3.

[45] *Variance-Based Methods*. John Wiley & Sons, Ltd, 2007.

[46] R. A. Fisher, “Statistical methods for research workers,” in *Breakthroughs in statistics: Methodology and distribution*. Springer, 1970.

[47] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied multiple regression/correlation analysis for the behavioral sciences*, 2013.

[48] A. Hartman, *Software and Hardware Testing Using Combinatorial Covering Suites*. Springer, 2005.

[49] https://www.cnn.com/2021/11/17/health/drug-overdose-deaths-record-high, 2021.

[50] https://www.in.gov/pla/inspect/files/Narxcare_user_guide.pdf, 2020.

[51] J. D. Oliva, “Dosing discrimination: regulating pdmp risk scores,” *Cal. L. Rev.*, vol. 110, p. 47, 2022.

[52] R. Vunikili, B. S. Glicksberg, K. W. Johnson, J. T. Dudley, L. Subramanian, and K. Shameer, “Predictive modelling of susceptibility to substance abuse, mortality and drug-drug interactions in opioid patients,” *Frontiers in Artificial Intelligence*, vol. 4, p. 742723, 2021.

[53] A. E. Johnson, L. Bulgarelli, L. Shen, A. Gayles, A. Shammout, S. Horng, T. J. Pollard, S. Hao, B. Moody, B. Gow *et al.*, “Mimic-iv, a freely accessible electronic health record dataset,” *Scientific data*, 2023.

[54] D. Weisburd, R. Greenspan, S. Mastrofski, and J. J. Willis, “Compstat and organizational change: A national assessment,” *National Institute of Justice*, 2008.

[55] K. Lum and W. Isaac, “To predict and serve?” *Significance*, 2016.

[56] W. Webber, A. Moffat, and J. Zobel, “A similarity measure for indefinite rankings,” *ACM Transactions on Information Systems (TOIS)*, 2010.

[57] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, 1938.

[58] T. Yin, R. Raab, M. Liu, and Y. Liu, “Long-term fairness with unknown dynamics,” *Advances in Neural Information Processing Systems*, 2024.

[59] D. C. McElfresh, L. Chen, E. Oliva, V. Joyce, S. Rose, and S. Tamang, “A call for better validation of opioid overdose risk algorithms,” *J Am Med Inform Assoc*, vol. 30, no. 10, pp. 1741–1746, Sep. 2023.

[60] G. A. Adam, C.-H. K. Chang, B. Haibe-Kains, and A. Goldenberg, “Hidden risks of machine learning applied to healthcare: unintended feedback loops between models and future data causing model degradation,” in *Machine Learning for Healthcare Conference*, 2020.

[61] S. A. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. P. Hamilton, and D. Roth, “A comparative study of fairness-enhancing interventions in machine learning,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.

[62] S. Tizpaz-Niari, A. Kumar, G. Tan, and A. Trivedi, “Fairness-aware configuration of machine learning libraries,” in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 909–920.

[63] J. M. Zhang and M. Harman, ““ignorance and prejudice” in software fairness,” in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 1436–1447.

[64] S. Majumder, J. Chakraborty, G. R. Bai, K. T. Stolee, and T. Menzies, “Fair enough: Searching for sufficient measures of fairness,” *ACM Transactions on Software Engineering and Methodology*, 2021.

[65] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, “Learning fair representations,” in *ICML*, 2013.

[66] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, “Certifying and removing disparate impact,” in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015.

[67] M. B. Zafar, I. Valera, M. G. Rogriguez, and K. P. Gummadi, “Fairness constraints: Mechanisms for fair classification,” in *Artificial Intelligence and Statistics*, 2017.

[68] F. Kamiran and T. Calders, “Data preprocessing techniques for classification without discrimination,” *Knowledge and Information Systems*, 2012.

[69] B. H. Zhang, B. Lemoine, and M. Mitchell, “Mitigating unwanted biases with adversarial learning,” in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018.

[70] T. Kamishima, S. Akaho, H. Asoh, and J. Sakuma, “Fairness-aware classifier with prejudice remover regularizer,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2012.

[71] Z. Chen, J. M. Zhang, F. Sarro, and M. Harman, “Maat: a novel ensemble approach to addressing fairness and performance bugs for machine learning software,” in *Proceedings of the ESEC/FSE*, 2022, pp. 1122–1134.

[72] S. Qian, V. H. Pham, T. Lutellier, Z. Hu, J. Kim, L. Tan, Y. Yu, J. Chen, and S. Shah, “Are my deep learning systems fair? an empirical study of fixed-seed training,” *NIPS*, 2021.

[73] M. Zhang and J. Sun, “Adaptive fairness improvement based on causality analysis,” in *Proceedings of the 30th ACM ESEC/FSE*, 2022, pp. 6–17.

[74] K. Holstein, J. Wortman Vaughan, H. Daumé III, M. Dudik, and H. Wallach, “Improving fairness in machine learning systems: What do industry practitioners need?” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[75] H. Zheng, Z. Chen, T. Du, X. Zhang, Y. Cheng, S. Ji, J. Wang, Y. Yu, and J. Chen, “Neuronfair: Interpretable white-box fairness testing through biased neuron identification,” May 21-May 29 2022.

[76] M. Fan, W. Wei, W. Jin, Z. Yang, and T. Liu, “Explanation-guided fairness testing through genetic algorithm,” in *Proceedings of the 44th International Conference on Software Engineering*, 2022, pp. 871–882.

[77] V. Monjezi, A. Trivedi, G. Tan, and S. Tizpaz-Niari, “Information-theoretic testing and debugging of fairness defects in deep neural networks,” in *Proceedings of the 45th International Conference on Software Engineering*, ser. ICSE ’23, 2023, p. 1571–1582.

[78] E. Soremekun, S. Udeshi, and S. Chattopadhyay, “Astraea: Grammar-based fairness testing,” *Transactions on Software Engineering*, 2022.

[79] A. Albarghouthi, L. D’Antoni, S. Drews, and A. V. Nori, “Fairsquare: probabilistic verification of program fairness,” *Proceedings of the ACM on Programming Languages*, 2017.

[80] O. Bastani, X. Zhang, and A. Solar-Lezama, “Probabilistic verification of fairness properties via concentration,” *Proceedings of the ACM on Programming Languages*, 2019.

[81] P. G. John, D. Vijaykeerthy, and D. Saha, “Verifying individual fairness in machine learning models,” in *Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020.

[82] Y. Li, J. Wang, and C. Wang, “Certifying the fairness of knn in the presence of dataset bias,” in *International Conference on Computer Aided Verification. Springer*, 2023.

[83] M. Hort, J. M. Zhang, F. Sarro, and M. Harman, “Fairea: A model behaviour mutation approach to benchmarking bias mitigation methods,” in *Proceedings of the 29th ACM ESEC/FSE*, 2021.

[84] U. Gohar, Z. Tang, J. Wang, K. Zhang, P. L. Spirtes, Y. Liu, and L. Cheng, “Long-term fairness inquiries and pursuits in machine learning: A survey of notions, methods, and challenges,” *arXiv preprint arXiv:2406.06736*, 2024.

[85] Y. Hu and L. Zhang, “Achieving long-term fairness in sequential decision making,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[86] M. Wen, O. Bastani, and U. Topcu, “Algorithms for fairness in sequential decision making,” in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

[87] H. Mouzannar, M. I. Ohannessian, and N. Srebro, “From fair decision making to social equality,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 359–368.

[88] A. Albarghouthi and S. Vinitsky, “Fairness-aware programming,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 211–219.

[89] T. A. Henzinger, M. Karimi, K. Kueffner, and K. Mallik, “Monitoring algorithmic fairness,” in *Computer Aided Verification: 35th International Conference, CAV 2023*. Berlin, Heidelberg: Springer-Verlag, 2023.

[90] A. Weber, B. Metevier, Y. Brun, P. S. Thomas, and B. C. da Silva, “Enforcing delayed-impact fairness guarantees,” *arXiv preprint arXiv:2208.11744*, 2022.

[91] R. Du, D. Muthirayan, P. P. Khargonekar, and Y. Shen, “Long-term fairness for real-time decision making: A constrained online optimization approach,” *arXiv preprint arXiv:2401.02552*, 2024.

[92] P. Schwöbel and P. Remmers, “The long arc of fairness: Formalisations and ethical discourse,” in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[93] A. D. Selbst, D. Boyd, S. A. Friedler, S. Venkatasubramanian, and J. Vertesi, “Fairness and abstraction in sociotechnical systems,” in *In the Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.

[94] E. Kang and R. Meira-Góes, “Requirements engineering for feedback loops in software-intensive systems,” in *2022 IEEE 30th International Requirements Engineering Conference Workshops (REW)*, 2022.

[95] J. L. Hellerstein, Y. Diao, S. Parekh, and D. M. Tilbury, *Feedback control of computing systems*. John Wiley & Sons, 2004.

[96] R. Taori and T. Hashimoto, “Data feedback loops: Model-driven amplification of dataset biases,” in *ICML*, 2023.

[97] D. Krueger, T. Maharaj, and J. Leike, “Hidden incentives for auto-induced distributional shift,” *arXiv preprint arXiv:2009.09153*, 2020.

[98] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. MIT Press, 2022.

[99] M. Casimiro, P. Romano, D. Garlan, G. A. Moreno, E. Kang, and M. Klein, “Self-adaptation for machine learning based systems,” 2021.

[100] L. Reader, P. Nokhiz, C. Power, N. Patwari, S. Venkatasubramanian, and S. A. Friedler, “Models for understanding and quantifying feedback in societal systems,” in *FAccT ’22*, 2022.

[101] J. D. Sterman, “System dynamics modeling: Tools for learning in a complex world,” *California Management Review*, pp. 8–25, 2001.

[102] N. Dhirasasna and O. Sahin, “A multi-methodology approach to creating a causal loop diagram,” *Syst.*, vol. 7, no. 3, p. 42, 2019.

[103] H. Kim and D. F. Andersen, “Building confidence in causal maps generated from purposive text data: mapping transcripts of the federal reserve,” *System Dynamics Review*, vol. 28, no. 4, pp. 311–328, 2012.

[104] J. R. Burns and P. Musa, “Structural validation of causal loop diagrams,” in *Proceedings of the 19th International Conference of the System Dynamics Society*, 2001, pp. 23–27.

[105] L. F. Luna-Reyes and D. L. Andersen, “Collecting and analyzing qualitative data for system dynamics: methods and models,” *System Dynamics Review*, vol. 19, no. 4, pp. 271–296, 2003.

[106] B. Penzenstadler, L. Duboc, C. C. Venters, S. Betz, N. Seyff, K. Wnuk, R. Chitchyan, S. M. Easterbrook, and C. Becker, “Software engineering for sustainability: Find the leverage points!” *IEEE Softw.*, 2018.