JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 2

Feedback-Driven Autonomy: Learning Safe Automation Boundaries In Large-Scale Decision Systems

Suganya Nagarajan

Independent Researcher, USA

Abstract

While automated decision systems are increasingly deployed at unbounded scales, autonomy is necessary for performance and availability. Autonomy is controlled only through thresholds, manual tuning, or infrequent human review, leading to automation boundaries that drift out of alignment with real-world risk once systems are in production. While previous work considers when to enable autonomy at runtime, there is relatively little work on how autonomy should adapt after deployment, during ongoing production operation. Feedback-driven autonomy is a system-level model where the boundaries of autonomy are adjusted in production not only based on prediction accuracy but also on observed operational performance. The control plane also manages an autonomy learning loop, which consumes asynchronous and aggregated feedback (e.g., rollback costs, incident correlation, human interventions) from live systems and adjusts the autonomy level of the system over time. Rather than learning an agent's decision-making like a reinforcement learning algorithm would do, the governance layer learns how autonomous the system should be in its current operational context. Examples in recommendation, offer vending, and notification systems inform the exploration of feedback-driven autonomy as a strategy for safe, scalable, and attribution-tolerant automation in high-throughput production environments. Uncovering autonomy as a learned property of the system provides a practical focus for designing adaptive, resilient, automated decision systems that can respond to a dynamic, real-world operational context after deployment.

Keywords: Autonomous Decision Systems, Adaptive Governance, Operational Feedback, Automation Boundaries, Production Systems, Safe Automation.

1. Introduction

Automated decision systems underpin large-scale digital platforms such as recommendation engines, offer selection services, and notification systems. Operating under strict latency, availability, and throughput constraints, these systems execute decisions at volumes that make continuous human supervision infeasible. Human-in-the-loop execution cannot scale to modern production demands, making autonomy essential for performance and reliability. Yet autonomy introduces operational risks that existing governance practices do not adequately address. In most organizations, autonomy boundaries are defined at deployment and rarely revisited systematically once systems are live. This creates a persistent tension between the need for scalable automation and the need for safety, reliability, and accountability in evolving production environments.

This section examines the system-level challenges of large-scale autonomy (Section 1.1) and the limitations of static autonomy in live operation (Section 1.2).

1.1 Autonomy and modern systems-related issues

Once deployed, autonomous decision systems operate continuously under evolving traffic patterns, user behavior, and business objectives. In such environments, autonomy introduces failure modes that are difficult to detect and costly to mitigate. Automated systems may generate outcomes whose negative impact is not immediately recognized as system failures, particularly when operating at scale and without continuous human supervision [1]. Small misalignments can accumulate before operators detect their systemic effect.

In practice, autonomy is frequently governed by static safeguards—such as hardcoded thresholds or manual overrides—that do not adapt to changing operational contexts. Prior work has shown that production ML systems often accumulate brittle controls and hidden technical debt as they evolve [2]. As a result, mitigation typically occurs reactively through ad hoc constraint tightening or manual intervention. While such measures may address immediate issues, they increase operational complexity and fail to provide a systematic mechanism for aligning autonomy with real-time system conditions.

1.2 Static Autonomy Limitations

In most systems, autonomy is a static property. Once a decision pathway has been authorized for automation, it will remain autonomous until human review triggers reconfiguration. This governance model creates a mismatch between the system's evolving capabilities and its permitted level of autonomy in production. Changes in models, traffic patterns, or operating conditions can shift the system's risk profile, causing initially appropriate autonomy boundaries to become either overly restrictive or unsafe.

Traditional delegation mechanisms determine whether a system may act autonomously in specific situations, but they do not address how much autonomy the system should sustain over its operational lifetime. Feedback-driven autonomy instead models autonomy as a continuously learned property, dynamically regulating the degree of automation to align with real-time operational conditions.

Existing automated decision systems rely on static thresholds, feature flags, or fully autonomous execution paths, which fail to adapt safely to dynamic operating environments. This paper introduces a feedback-driven governance framework that dynamically adjusts autonomy boundaries using operational safety signals, enabling controlled automation expansion while preserving human oversight, auditability, and rollback safety. Unlike prior approaches that treat automation as a binary state, the proposed model operationalizes autonomy as a continuously governed boundary that evolves based on observed system reliability and risk conditions

2. Background and Problem Formulation

2.1 Current Governance Practices

Operational experience with large-scale automated systems shows that operators continuously generate valuable feedback through maintenance activities (e.g., undoing risky changes), intervening in operational issues (preventing failures), and the tightening of safeguards to increase trust[1]. These corrective actions provide insight into perceived risk, trust calibration, and system maturity. However, most contemporary systems do not systematically incorporate this feedback into automation governance[3].

Because governance requires constraints, autonomy boundaries are typically defined at deployment based on initial risk assessment and testing outcomes. Updates to these boundaries are infrequent and usually triggered by significant incidents. As a result, autonomy constraints often drift out of alignment with operational costs and risk exposure, particularly as systems evolve and environments change[2].

Table 1 summarizes common autonomy governance practices and their limitations in large-scale decision environments.

Table 1: Governance Practice Characteristics and Associated Limitations [1]–[3]

Governance Practice	Current Implementation	Operational Limitation
Autonomy Boundary Definition	Determined at deployment time based on initial risk assessment	Boundaries become misaligned as system context evolves through updates and environmental changes
Boundary Update Mechanism	Manual adjustments triggered by significant incidents	Infrequent updates create weak correlation between constraints and actual operational cost
Risk Calibration Approach	Static thresholds established during testing phase	Unable to respond to rapid system evolution and changing business requirements
Feedback Integration	Ad-hoc incorporation through reactive interventions	Valuable operational signals from rollbacks and overrides remain systematically unutilized

2.2 The Learning Gap

A fundamental limitation of current governance approaches is that autonomy boundaries do not learn from operational experience.While systems optimize decision quality through model training and business metrics through online experimentation, they do not explicitly optimize for the safety of autonomous execution under evolving conditions or for minimizing the cost of incorrect autonomous actions[2]. Nor do they adapt autonomy scope in response to changing organizational risk tolerance [3].

These governance limitations can produce significant operational and reputational risk. Fraud detection systems may block legitimate transactions during seasonal demand spikes, pricing engines can generate extreme outputs when exposed to anomalous data, content moderation models may suppress legitimate speech due to drift, and credit decision pipelines can deny qualified applicants when data quality degrades. Such failures reveal the limits of static autonomy constraints and underscore the need for governance mechanisms that adapt safely to evolving operational conditions.

This gap manifests in two common failure patterns. In early autonomy expansion, decision-quality improvements or optimization artifacts lead to rapid increases in automation scope without corresponding safeguards. In contrast, autonomy stagnation occurs when systems fail to incorporate evidence that increased safety and reliability could justify expanding automation. Existing literature has focused on prediction accuracy and decision optimization, leaving the governance of autonomy scope over time largely unaddressed. This work builds on prior research in human-in-the-loop decision systems, supervisory control, and trustworthy AI governance. While existing approaches emphasize prediction accuracy and decision optimization, the governance of autonomy scope over time remains underexplored. The proposed framework advances this area by treating adaptive autonomy boundaries as a first-class operational control problem.

3. The Autonomy Learning Loop

3.1 Core Mechanism

The Autonomy Learning Loop (ALL) is a runtime governance mechanism that continuously calibrates the degree of autonomy a system is permitted to exercise in production. Rather than treating autonomy as a static configuration established at deployment, ALL enables autonomy to evolve based on observed operational outcomes.This mechanism enables autonomy to adapt continuously to operational conditions while preserving decision-path performance.

At its core, ALL operates as a control-plane feedback loop in which autonomy is progressively earned through demonstrated reliability, consistent with self-managing system paradigms that use monitoring and adaptation to regulate system behavior [4]. Automated decisions are executed within the authorized autonomy scope and monitored through existing telemetry, logging, and incident management systems.

Operational signals — including rollbacks, manual overrides, incident correlations, and policy interventions — are captured through monitoring telemetry and aggregated for reliability pattern analysis. Studying such runtime behavioral signals is essential for understanding and governing the behavior of deployed autonomous systems [5].

The loop operates through recurring decision cycles in which the system may execute actions autonomously or be subject to restrictions defined by delegated autonomy permissions. Following execution, operational outcomes — including rollback events, escalation patterns, and incident correlation markers are analyzed to identify reliability trends and emerging risk patterns. Unlike conventional reliability monitoring, these signals inform governance decisions that continuously calibrate the permissible scope of autonomous operation.

During the adjustment phase, the autonomy boundary is evaluated to determine whether the delegation scope should be expanded, constrained, or maintained based on observed operational trends. This calibration reflects a governance process in which operational experience informs the safe expansion or restriction of automation authority over time, drawing on established practices for human-in-the-loop governance in AI systems [6]. The resulting autonomy boundaries are then applied to subsequent decision cycles, establishing a closed feedback loop that aligns autonomy scope with observed operational reliability. Each stage in Table 2 contributes operational feedback that enables the Autonomy Learning Loop to calibrate autonomy boundaries based on demonstrated reliability.

Table 2: Autonomy Learning Loop Operational Stages and Functions, adapted from feedback-loop governance and self-managing system principles [4, 5].

Stage	Primary Function	Output
Decision Execution	Execute decisions autonomously or under constraint based on current delegation rules	Autonomous actions reflecting earned autonomy levels
Outcome Observation	Monitor operational results through existing infrastructure	Captured signals, including rollback events and escalation patterns
Feedback Signal Capture	Aggregate and analyze outcome signals by decision class and temporal window	Reliability trends and anomaly indicators
Autonomy Boundary Adjustment	Determine whether to expand, tighten, or maintain delegation boundaries	Updated autonomy constraints reflecting operational evidence
Autonomy Enforcement	Apply updated boundaries to subsequent autonomous operations	Closed feedback loop aligning scope with system capabilities

3.2 Design Principles

The Autonomy Learning Loop incorporates safety constraints into its design, allowing it to function in high-throughput production environments. Outcome capture is performed asynchronously and decoupled from request execution paths to avoid impacting decision latency and system throughput. Feedback signals are captured via control-plane telemetry that observes system behavior without being on critical paths [5].

Learning occurs through bounded temporal windows: feedback signals are accumulated within each window, and autonomy boundary adjustments are evaluated only at window boundaries. This windowed aggregation filters transient noise and short-lived anomalies that should not influence autonomy governance. Additionally, adjustments to the granted autonomy are rate-limited and applied conservatively to prevent oscillatory behavior and to ensure that changes remain observable and operationally interpretable at runtime.

3.3 Distinguishing Characteristics

The Autonomy Learning Loop is not a reinforcement learning mechanism and does not attempt to optimize decision policies or maximize reward outcomes. Reinforcement learning searches for optimal actions within the decision space, whereas the Autonomy Learning Loop governs the permissible scope of automation within a governance space. The mechanism does not retrain predictive models or alter their optimization of input-to-prediction mappings. Instead, it regulates when automated decisions may be executed autonomously and when constraints should apply. In this sense, the learning objective is not decision accuracy, but the safe calibration of autonomy boundaries. By optimizing the transition between constrained and autonomous operation, the Autonomy Learning Loop improves safety, reliability, and operational trust without modifying the underlying decision models.

3.4 Operational Architecture of the Autonomy Learning Loop

The autonomy learning loop is operationalized through a modular control architecture that enables adaptive governance of automated decisions in production environments. A decision engine executes rule-based or machine learning–driven actions, while a safety signal aggregator continuously evaluates operational indicators such as error rates, anomaly detections, override frequency, and policy violations. These signals are assessed by an autonomy boundary controller, which dynamically adjusts the authorized scope of automation in response to observed system reliability and risk conditions. When decisions exceed the permitted autonomy boundary or safety thresholds are breached, a human oversight interface enables review, override, and escalation workflows. All decisions, boundary adjustments, and overrides are recorded within an audit layer to ensure traceability and compliance.

In deployment, the controller may operate inline for latency-sensitive decisions or asynchronously in high-throughput environments where safety evaluation and corrective actions occur post-decision. Potential failure modes include delayed or degraded safety signals, monitoring outages, model drift producing unstable outputs, and overload of human review queues. Safeguards such as boundary freeze, conservative fallback thresholds, escalation to manual review, and prioritization mechanisms ensure resilience during anomalies or signal degradation. Rollback mechanisms allow restoration of prior boundary states and reversion to human-review workflows, ensuring safe recovery without service disruption.

Boundary expansion policies operate under governance controls aligned with organizational risk tolerance and regulatory obligations, and recorded overrides and boundary adjustments support audit review, compliance verification, and post-incident analysis. The effectiveness of the architecture can be evaluated through simulation, shadow-mode deployment, and phased rollout strategies, using operational metrics such as override frequency, escalation volume, incident reduction, and error-rate trends to assess governance performance and safety improvements.

4. Feedback Signals and Scalability

4.1 Operational Consequence Signals

Feedback-driven autonomy is governed not by prediction accuracy alone [2], but by the operational consequences of autonomous actions in production environments. The cost and risk of executing actions autonomously are shaped by system dynamics, user behavior, and operational conditions that are not captured by offline evaluation metrics or model accuracy measures.

One indicator of autonomy risk is rollback frequency. When system behavior requires frequent manual intervention, configuration rollbacks, or emergency mitigation actions, the system has exceeded acceptable autonomy levels for its operational context. Recovery cost reflects the time, effort, and resources required to restore normal operation, reflecting system health indicators commonly used in reliability monitoring practices [7].

Incident correlation refers to the temporal proximity between autonomous decisions and operational incidents such as service degradation, instability, or outages [8]. Human override rate measures how frequently operators intervene to prevent or reverse automated actions, providing a direct signal of misalignment between automated behavior and operational judgment.

A confidence–outcome divergence may occur when system confidence or expected performance remains high while operational outcomes degrade. This divergence signals that predictive performance alone is insufficient to govern safe autonomy [2]. User trust signals provide additional indicators of autonomy risk. Sustained increases in complaints, abandonment, opt-outs, or disengagement may reflect erosion of trust in automated decisions, even when short-term performance metrics remain favorable.
Unlike conventional reliability monitoring, these signals are not used solely for incident response or diagnostics; instead, they serve as governance inputs that continuously calibrate the permissible scope of autonomous operation, enabling autonomy boundaries to evolve with demonstrated system reliability.

Table 3 summarizes principal operational consequence signals and the system behaviors they capture.

Table 3: Feedback Signal Categories with operational consequence signals used in reliability monitoring and observability practices [7, 8]

Signal Category	Functional Role	System Behavior Captured
Rollback Frequency	Direct indicator of autonomy boundary exceedance	Manual interventions and emergency mitigations reversing autonomous decisions
Recovery Cost	Measure of failure impact severity	Time and resources required to restore stable operation after undesirable outcomes
Incident Correlation	Detection of temporal risk associations	Service degradation and customer impact events clustering with autonomous decisions
Human Override Rate	Gauge of alignment with operator judgment	Manual interventions modifying or reversing autonomous behavior
Confidence-OutcoDivergence	Assessment of self-calibration accuracy	Mismatches between internal confidence levels and observed decision results
Trust Indicators	Measurement of user relationship health	Complaint rates, disengagement patterns, and escalation behaviors from users

4.2 Scalability Architecture

While comprehensive feedback observation is essential, high-throughput decision systems impose strict limits on telemetry volume, processing overhead, and storage costs. Capturing outcomes at fine granularity across all decisions can introduce prohibitive computational and operational burdens. To preserve system performance, feedback collection and learning must scale independently from request execution paths, consistent with large-scale observability architectures designed to minimize performance overhead [8].
Scalable outcome capture is achieved through control-plane telemetry that observes system behavior without residing in request paths. Outcome signals are aggregated at the decision-class and contextual-window levels, significantly reducing data volume while preserving trend-level information necessary for autonomy learning.
A central architectural principle is attribution-tolerant design. In real-world production systems, it is rarely possible to attribute operational outcomes such as rollbacks, incidents, or overrides to a single causal decision. Instead of requiring precise causal attribution, the Autonomy Learning Loop identifies correlated risk patterns across decision cohorts and contextual windows. When adverse patterns emerge, the system responds conservatively by tightening autonomy boundaries for affected decision classes. This correlation-based approach enables reliable governance under uncertainty while maintaining scalability and operational resilience.

5. Formal Logic of Boundary Adaptation

To transition the Autonomy Learning Loop (ALL) from a conceptual framework to an executable governance mechanism, adaptation of autonomy boundaries can be modeled as a state transition process. Let $B_t \in [0,1]$ represent the authorized scope of autonomy for a specific decision class during time window t, where higher values correspond to broader autonomous operation.

The boundary evolves according to:

$$B_{t+1} = B_t + \Delta B_t$$

where ΔB_t is determined by evaluating an aggregated operational risk signal S_t against safety thresholds

τ_{safe} and τ_{risk}. The signal S_t may incorporate rollback rates, override frequency, incident correlations, and reliability indicators observed during the window.

Autonomous Expansion (Learning Phase)

When operational risk remains below the safety threshold,

$$\Delta B_t = \alpha(1 - B_t), \text{ if } S_t < \tau_{safe}$$

where:

- α is a small positive learning rate governing how quickly autonomy is earned,
- $(1-B_t)$ ensures expansion slows as full autonomy is approached, requiring sustained reliability evidence.

Conservative Contraction (Safety Phase)
When risk signals exceed the risk threshold,

$$\Delta B_t = -\beta B_t, \text{ if } S_t \geq \tau_{risk}$$

where β is a contraction rate configured such that $\beta \gg \alpha$, ensuring rapid response to operational risk.

Threshold-Based Governance
This asymmetric update structure ensures autonomy evolves as an emergent property of demonstrated reliability rather than a static configuration. Windowed aggregation over time filters transient noise and emphasizes trend-level reliability indicators.

This formulation represents a reference control structure; practical implementations may incorporate policy guardrails, domain constraints, and safety overrides.

6. Practical Applications

6.1 Recommendation Systems

Recommendation systems represent a canonical domain for feedback-driven autonomy, where deployment scale makes human oversight of individual decisions infeasible. Large-scale recommender platforms operate under strict latency and throughput constraints while continuously adapting to evolving user behavior, requiring ongoing evaluation and operational monitoring at scale [9].

Operational feedback provides the basis for expanding autonomy boundaries. Low rollback rates indicate deployment stability, sustained engagement metrics suggest continued positive user response, and the absence of incident correlation signals operational safety. In large-scale recommender systems, behavioral feedback from live traffic is essential for evaluating decision performance and guiding safe policy evolution over time [10]. Aggregated outcomes across decision classes and time windows reveal readiness for broader autonomous operation. Through the Autonomy Learning Loop, decision autonomy can expand progressively as reliability is demonstrated, while contracting automatically if performance degrades.

6.2 Offer Vending Systems

Offer vending systems require tighter autonomy governance because decisions directly affect revenue outcomes and customer behavior exhibits inherent variability. During periods of stable traffic and predictable promotional cycles, autonomous offer selection may safely operate within broader boundaries.However, high-impact sales events introduce rapid shifts in traffic volume, user behavior, competitive dynamics, and failure costs. Under these conditions, the operational risk associated with incorrect or poorly targeted offers increases significantly. Industry experimentation platforms emphasize the importance of guardrails and staged rollout strategies to manage revenue risk and prevent large-scale negative impact during high-stakes events [11].

Because rollback costs rise during peak periods and the consequences of erroneous offers become more severe the Autonomy Learning Loop can respond by preemptively tightening autonomy boundaries. As operational conditions normalize and reliability signals stabilize, autonomy may be safely re-expanded. This context-sensitive contraction and expansion enables revenue-critical systems to balance optimization with operational safety.

6.3 Notification Systems

Notification systems illustrate the challenges of sustaining long-term trust through automated decision-making. The effectiveness of an individual notification is difficult to measure in isolation; however, persistent misalignment between notifications and user expectations can erode trust over time. Users may disengage, opt out, or develop negative perceptions of the service, a phenomenon documented in studies of notification fatigue and user response behavior [12].

Consider an autonomous frequency optimization algorithm designed to maximize short-term engagement. If increased notification volume leads to user fatigue, unsubscribe rates may rise while long-term engagement declines. Through the Autonomy Learning Loop, sustained co-occurrence of these indicators alongside autonomous frequency decisions signals erosion of trust. In response, the system tightens autonomy boundaries governing notification frequency, preventing further trust degradation while preserving operational effectiveness. Maintaining long-term trust requires ongoing monitoring of user responses and adapting automation behavior to align with user expectations and tolerance thresholds [1]

These scenarios illustrate how feedback-driven autonomy governs systems with distinct risk profiles, demonstrating the generality of the approach.

Table 4: Application domain characteristics and autonomy adaptation patterns demonstrating feedback-driven governance across operational contexts [10, 12]

Application Domain	Operational Context	Autonomy Adaptation Pattern
Recommendation Systems	New ranking models requiring validation at scale	Gradual expansion during stable outcomes followed by automatic tightening when engagement declines
Offer Vending Systems	Financial decisions with variable risk profiles	Context-sensitive contraction during high-impact events with autonomous re-expansion after normalization
Notification Systems	Long-term trust relationship management	Proactive boundary tightening detecting gradual trust erosion before critical thresholds

Cross-Domain Pattern	Shared feedback-driven governance pattern	Conservative response to trend-level feedback enabling self-correction without execution blocking

Conclusion

As automated decision systems scale beyond the limits of human oversight, static autonomy, assigned at deployment and left unchanged over a system's lifetime, becomes insufficient. Production environments evolve continuously, and autonomy must adapt alongside changes in system behavior, operational risk, and organizational tolerance for failure. This paper presents feedback-driven autonomy as a governance paradigm in which autonomy boundaries evolve based on observed operational outcomes rather than predicted performance alone. The Autonomy Learning Loop demonstrates how safe automation boundaries can be learned from asynchronous, aggregated, and attribution-tolerant operational signals. The approach is inherently scalable and suited to high-throughput production environments while preserving the safety, predictability, and accountability required to sustain institutional trust in automation.

The guiding principle is that autonomy should be earned through demonstrated reliability. Systems may expand autonomous operation when supported by sustained evidence of stability and safety, and contract autonomy when operational feedback indicates elevated risk. Through bounded adaptation cycles, autonomy becomes an emergent property shaped by real-world performance rather than static configuration. Organizations pursuing greater automation face a choice: maintain static autonomy with reactive manual intervention, or adopt feedback-driven governance that enables autonomy to evolve in alignment with operational realities. The Autonomy Learning Loop provides a practical framework for implementing adaptive autonomy governance in production systems, enabling automation that is both powerful and safe.

References

1. Saleema Amershi, et al., "Guidelines for Human-AI Interaction," ACM Digital Library, 2019. Available: https://dl.acm.org/doi/epdf/10.1145/3290605.3300233
2. D. Sculley, Gary Holt, et al., "Hidden Technical Debt in Machine Learning Systems," ACM Digital Library, 2015. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf
3. Inioluwa Deborah Raji, et al., "Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing," ACM Digital Library, 2020. Available: https://dl.acm.org/doi/epdf/10.1145/3351095.3372873
4. Jeffrey O. Kephart and David M. Chess, "The Vision of Autonomic Computing," ACM Digital Library, 2003. Available: https://dl.acm.org/doi/10.1109/MC.2003.1160055
5. Iyad Rahwan, et al., "Machine behaviour," Nature, 2019. Available: https://www.nature.com/articles/s41586-019-1138-y
6. Michael Madaio, et al., "Co-Designing Checklists to Understand Organizational Challenges and Opportunities around Fairness in AI," CHI Conference on Human Factors in Computing Systems, ACM, 2020. Available:https://dl.acm.org/doi/fullHtml/10.1145/3313831.3376445
7. Rob Ewaschuk, "Monitoring Distributed Systems," Chapter 6, O'Reilly Media, Inc, 2017. Available: https://sre.google/sre-book/monitoring-distributed-systems/
8. Benjamin H. Sigelman, et al., "Dapper, a Large-Scale Distributed Systems Tracing Infrastructure," Google Research, 2010. Available: https://static.googleusercontent.com/media/research.google.com/en//archive/papers/dapper-2010-1.pdf
9. Paul Covington, et al., "Deep Neural Networks for YouTube Recommendations," Proceedings of the 10th ACM Conference on Recommender Systems, ACM, New York, NY, USA, 2016. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45530.pdf

10. Minmin Chen, et al., “Top-K Off-Policy Correction for a REINFORCE Recommender System,” arXiv, 2021. Available: https://arxiv.org/pdf/1812.02353
11. Ron Kohavi, et al., “Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing,” Cambridge University Press, 2020. Available: https://www.researchgate.net/publication/339914315_Trustworthy_Online_Controlled_Experiments_A_Practical_Guide_to_AB_Testing
12. Mingyuan Zhong, et al., “ForceBoard: Subtle Text Entry Leveraging Pressure,” ACM Digital Library, 2018. Available: https://dl.acm.org/doi/10.1145/3173574.3174102