Training Machine Learning to Anticipate Manipulation*

Daniel Björkegren† Columbia University

Joshua E. Blumenstock‡ U.C. Berkeley

Samsun Knight§ University of Toronto

This version: July 28, 2023
First version: November 30, 2018

Abstract

An increasing number of decisions are guided by machine learning algorithms. But when consequential decisions are encoded in algorithms, individuals may strategically alter their behavior to achieve desired outcomes. This paper develops an empirical approach to adjust decision algorithms to anticipate manipulation. By explicitly modeling incentives to manipulate, our approach produces decision rules that are stable under manipulation, even when the rules are fully transparent. We stress test this approach through a large field experiment in Kenya. When implemented, decision rules estimated with our strategy-robust approach outperform those based on standard machine learning approaches.

Keywords: machine learning, manipulation, decisionmaking, digital credit, targeting

*We are grateful for helpful conversations with Susan Athey, Jon Bittner, John Friedman, Greg Lewis, Ted Miguel, Paul Niehaus, Ben Roth, and Jesse Shapiro; and for feedback from seminar audiences at Brown, BREAD, Columbia, U Chicago, ETH Zürich, Harvard, Imperial College London, Microsoft Research, MIT, NYU, NBER AI, NBER IO, NBER DEV, Oxford, Stanford, Tufts, World Bank, UC Berkeley, UCSB, and the Simons Institute. We thank Jolie Wei for research assistance, and Chaning Jang, Simon Muthusi, Nicholas Owsley, and the Busara team for their collaboration. We are grateful for funding from the Brown University Seed Fund, the Bill and Melinda Gates Foundation, and the Center for Effective Global Action. Björkegren thanks the W. Glenn Campbell and Rita Ricardo-Campbell National Fellowship at Stanford University, and Microsoft Research for support. Blumenstock thanks the National Science Foundation for support under CAREER Grant IIS-1942702. This study was pre-registered with the AEA RCT Registry (AEARCTR-0004649) and approved by the IRBs of UC Berkeley, Brown, and the Kenya Medical Research Institute.

†dan@bjorkegren.com
‡jblumenstock@berkeley.edu
§samsundknight@gmail.com

1 Introduction

An increasing number of important decisions are being made by machine learning algorithms. Algorithms determine what information we see online; who is hired, fired, and promoted; who gets a loan, and whether to give bail and parole. In the typical machine learning deployment, an individual's observed behavior is used as input to a decision rule.

However, when algorithms are used to make consequential decisions, they create incentives for people to 'game' the rule. When agents understand how their behavior affects decisions, they may alter their behavior to achieve the outcome they desire. When decision rules are gamed, they can produce decisions that are arbitrarily poor or unsafe. The problem of manipulation stems from the fact that the standard approach to training decision rules assumes that the relationship between the outcome of interest and human behaviors will remain stable. But this assumption tends to be violated as soon as a decision rule is implemented and agents have incentives to change their behavior to achieve more favored outcomes (Lucas, 1976; Goodhart, 1975).

A classical solution to this problem in economics involves modeling behavioral responses when designing policies; this approach has been integral to work in canonical settings like taxation and mechanism design (Mirrlees, 1971; Akerlof, 1978; Ramsey, 1927; Agarwal and Budish, 2021). However, this insight is not typically used when training the type of decision rules that have proliferated in society in recent years that instead rely on atheoretic estimators to uncover high-dimensional correlations in data (Breiman, 2001).

Instead, real-world applications of machine learning commonly use one of two alternate approaches to deal with this problem. The first is to restrict the decision rule to predictors that are thought to be more stable – an approach that amounts to having a dogmatic prior that the cost of manipulation is either infinite for all people (for included predictors) or zero (for excluded predictors). However, people can manipulate most predictors (i.e., their behaviors) at some cost, and those costs may be heterogeneous and difficult to assess in modern contexts that can have thousands of predictors. Thus, many real-world deployments use a second approach, which we refer to as the 'industry approach.' This approach relies on 'security through obscurity' (NIST 2008) to make the decision rule more difficult to guess, and ad-hoc

re-training to respond to the changing relationship between predictors and outcomes (Bruckner and Scheffer, 2011). However, this approach learns about manipulation by making decisions that turn out to be mistakes. Such mistakes may be tolerable in low-stakes settings, but can cause great harm at unanticipated times in high stakes settings like finance or governance. This approach also requires decision rules be kept secret: the more clearly that agents know how their behavior affects a decision, the easier it becomes to manipulate. This need for secrecy stands in sharp contrast to the increasing societal demand for a 'right to explanation' about how algorithmic decisions are made (Goodman and Flaxman, 2016; Barocas et al., 2018). Together, risk and secrecy have become central concerns in active policy debates about regulation for machine learning and artificial intelligence.[1]

This paper considers an alternate approach. We extend the classical economic approach of modeling behavioral responses to the machine learning setting. We show how machine learning models can be trained to anticipate manipulation, which leads to better decisions when implemented in a real-world setting. Our approach explicitly models agents' incentives to manipulate, and illustrates how this game-theoretic model can be embedded within the loss function of a machine learning decision rule. By anticipating how agents will be strategic, the method produces decisions that perform well when implemented—even when the decision logic is made fully transparent. We then implement and evaluate the performance of empirical decision rules adjusted in this manner through a real-world field experiment in Kenya. We use the experiment to elicit costs of manipulating behavior, and show that the 'strategy-robust' approach better anticipates real-world performance, and leads to more robust empirical decision rules. Each component of the paper – theory, estimator, and experiment – is designed to be relatively simple. Our main contribution is to integrate these components in a single end-to-end example which demonstrates how a classical economic approach can be used in a machine learning setting.

The paper is organized into two main parts. The first part develops a simple adjustment to training machine learning models which produces strategy-robust

[1]For example, the White House's Blueprint for an AI Bill of Rights (OSTP, 2022) lists safety as the first concern and calls for risks to be anticipated and mitigated *before* deployment. Likewise, the European Union's General Data Protection Regulation mandates that "meaningful information about the logic" of automated systems be available to data subjects (European Union, 2016).

decision rules that are stable under manipulation. We consider a framework where a policymaker wishes to make a decision y_i for each individual i. They observe *training data*: a subset of instances that possess both features $\mathbf{x}_i$ and optimal decisions y_i. The policymaker seeks to estimate a decision rule $\hat{y}(\mathbf{x}_i)$ that will be applied to *implementation* instances where only features $\mathbf{x}_i$ will be observed. Whereas the standard approach selects a decision rule that is optimal for the distribution $(\mathbf{x}_i, y_i)$ observed in training, our approach anticipates how individuals will adjust behavior in response to the incentives generated by a decision rule; that is, it models $\mathbf{x}_i(\hat{y}(\cdot))$. Characterizing how behavior will respond to the rule requires a structural model, which we embed within the machine learning estimator. While this general approach can be applied to arbitrary estimators, we focus primarily on linear decision rules of the form $\hat{y}(\mathbf{x}) = \boldsymbol{\beta}\mathbf{x}$, with quadratic manipulation costs. This allows us to derive analytic results that take a simple nonlinear least squares form (we later show how this approach can be generalized beyond the linear case to train strategy-robust decision trees). In Monte Carlo simulations, we show that this strategy-robust approach often outperforms common alternatives when agents behave strategically.[2]

The second major part of the paper illustrates how this framework can be applied in a real-world environment, using a field experiment in Kenya that we designed specifically to stress-test strategy-robust decision rules. This experiment allows us to train strategy-robust and standard decision rules, and evaluate their real-world performance when implemented. Specifically, we built a new smartphone app that passively collects data on how people use their phones, and disburses rewards to users based on predictions formed from the data collected. The app is designed to mimic 'digital credit' products that are spreading dramatically and quickly transforming consumer credit in the developing world (Bharadwaj and Suri, 2020; Suri et al., 2021). Digital credit products similarly collect user data and use machine learning algorithms to convert that data into a credit score, based on the insight that historical patterns of mobile phone use can predict credit repayment (Björkegren, 2010; Björkegren and

[2]These simulations also show how, by contrast, models trained in the standard manner can perform very poorly when agents behave strategically. The simple industry approach may not converge; even if it does, convergence can be slow and lead to undesirable equilibria. We also observe how, in settings where the ability to manipulate is a signal of the outcome (in the spirit of Spence, 1973), our method can implicitly exploit this correlation by *increasing* the weight on features that are manipulable by the types to be screened in, but not by those to be screened out.

Grissen, 2020). However, as these systems have scaled, they increasingly face fraud resulting from manipulation, as borrowers learn which behaviors will increase their credit limits (McCaffrey et al., 2013; Bloomberg, 2015; Crosman, 2017).[3] After being hit by manipulation, several lenders have restricted lending.

This field experiment produces several results. First, consistent with prior work showing that mobile phone data can predict credit repayment, we find that the data collected through our smartphone app ($\mathbf{x}_i(\mathbf{0})$) can be used to predict the phone owners' characteristics, such as income and intelligence.[4]

Second, during the training period, we structurally estimate $\mathbf{x}_i(\hat{y}(\cdot))$ in our model; that is, how the distribution of behaviors tends to shift when decision rule $\hat{y}(\cdot)$ is implemented. These estimates are identified through a series of randomly assigned experiments that offer simplified decision rules, each offering financial rewards based on behaviors observed through the app. For example, participants may face decision rules that reward them based on frequency of outgoing calls in a given week, or the number of text messages they receive. The average weekly payouts are designed to be similar in magnitude to the typical digital credit loans in Kenya at the time ($4.80 in Bharadwaj and Suri (2020)). The general shifts in behavior that we estimate are intuitive: for instance, outgoing communications are less costly to manipulate than incoming communications, and text messages, which are relatively cheap to send, are more easily manipulated than calls. We also find that complex behaviors (such as the standard deviation of talk time) are less manipulable than simpler behaviors (such as the average duration of talk time). We also estimate substantial heterogeneity in ability to manipulate between people; much of this heterogeneity arises from unobservables, but we find that people who self-identify as tech savvy find it easier to manipulate behavior.

Third, we evaluate the trained decision rules $\hat{y}(\mathbf{x})$ in an implementation phase of the experiment where we observe only participants' current behavior $\mathbf{x}$. We find that, when actually implemented on decisions that affect people, 'strategy-robust'

[3]For instance, a recent survey in Kenya and Tanzania found that one of the top five reasons people report saving money in digital accounts is to increase the loan amount they qualified for (FSD Kenya, 2018).

[4]Related work has used mobile phone data to predict income and wealth (Blumenstock et al., 2015; Blumenstock, 2018; Aiken et al., 2021), gender (Blumenstock et al., 2010), and employment (Sundsøy et al., 2016).

decision rules perform substantially better than standard machine learning algorithms. We make this comparison by exposing participants to predictive decision rules that offer financial rewards if they use their phones like a person of a particular type. For instance, some people receive a message that says, "Earn up to 1000 Ksh if the app guesses that you are a high income earner, based on how you use your phone," while others receive messages that offer rewards for acting like an 'intelligent' person, and so forth. Across a variety of such decision rules, we show that classifications made by the algorithm trained with the strategy-robust approach are more accurate than classifications from the standard approach. Additionally, the strategy-robust adjustment more accurately predicts the real-world performance that decision rules will attain when they are implemented and made transparent.

Finally, we use our method to estimate the performance cost of algorithmic transparency: the loss associated with disclosing the details of the decision rule. In the experiment, we experimentally vary the amount of information subjects have about the decision rule $\hat{y}$, and show that the relative performance of the strategy-robust decision rule increases with transparency. While transparency reduces the predictive performance of standard decision rules by 17% on average, it reduces the strategy-robust rule's performance by only 6%. In our setting, the performance cost of moving from an equilibrium where decision rules are secret to an equilibrium where they are disclosed is less than 8%. Our model also allows policymakers to bound this equilibrium cost of transparency even without disclosing decision rules to the world.

Taken as a whole, our paper provides a framework for implementing empirical decision rules that are robust to manipulation. It introduces a new notion of fit, which has analogues to other common linear regression approaches. For instance, ordinary least squares (OLS) maximizes fit within sample; two stage least squares (2SLS) sacrifices fit within sample to estimate coefficients that can be interpreted causally; penalized least squares (such as LASSO and ridge) sacrifice within-sample fit to generate simpler models that may better generalize to other samples drawn from the same population. Our approach sacrifices fit within sample to maximize fit in the counterfactual where the decision rule is used and agents manipulate against it. Our solution is an example of a class of estimator that maximizes *counterfactual fit* — predictive fit in a counterfactual state of the world. This is similar to the notion

of fit across different domains as discussed by Andrews et al. (2023): our framework optimizes models to perform well *in the domains they induce*.

We anticipate that approaches like the one we propose will be beneficial across a variety of domains as human and machine intelligence increasingly interact. Our approach combines experiments that measure how behavior responds to perturbations in a decision rule, with a structural model to anticipate the response to any rule, and embeds this behavioral response in an estimator that can be applied to high dimensional data. Similar approaches are likely to be relevant in a range of applied settings – especially when stakes are high or decision rules cannot be kept secret, in new implementations where there is limited evidence of historical manipulation, and when updating decision rules is costly or slow.

1.1 Connection to Literature

The conceptual problem of manipulation is not new. Goodhart (1975), in what has since become referred to as 'Goodhart's Law', noted that once a measure becomes a target, it ceases to be a good measure. Lucas (1976) also famously observed that historical patterns can deviate when economic policy changes. Empirically, agents have been observed to attempt to game decision rules in a wide range of settings, including New York high school exit exams (Dee et al., 2019), health provider report cards (Dranove et al., 2003), pollution monitoring in China (Greenstone et al., 2019), fish vendors in Chile (Gonzalez-Lira and Mobarak, 2019), and survey respondents in Indonesia (Banerjee et al., 2018).

But concerns about manipulation have become more pronounced as an increasing number of consequential decisions are automated based on complex correlations with behavioral 'big data'. In the online advertising industry, firms spend many millions of dollars each year on search engine optimization, manipulating their websites in order to be ranked higher by search engine algorithms (Borrell Associates, 2016). A quick Google search suggests over 50 thousand different websites (and 3,000 YouTube videos) contain the phrase "hack your credit score." In consumer credit – and in our main experimental application – digital behavioral data is now routinely used to assess repayment risk, as exemplified by the digital credit products that are now widespread in developing countries (Bharadwaj and Suri, 2020; Suri et al., 2021). In low-income

countries, a similar combination of machine learning and behavioral data is now being used to determine eligibility for social assistance and humanitarian aid (Aiken et al., 2021; Mukherjee et al., 2023).

In the research literature, there are two main paradigms for thinking about how to adapt decision rules to manipulation. In economics, the prevailing paradigm develops canonical models that account for behavioral responses in specific settings. In particular, the problem we study relates closely to the mechanism design literature – we explore this connection in greater detail when discussing our model in Section 2.3. Our paper is also related to the question in public finance of how to set taxes in environments where agents adapt their behaviors. Mirrlees (1971) considers taxes as a function of earnings, and faces the problem that taxation induces a behavioral response. Akerlof (1978) suggests that conditioning on additional attributes that are harder to manipulate ('tags') can improve efficiency. Ramsey (1927) suggests taxes be set using the inverse of the matrix of the costs of manipulating behavior. The market design literature has also considered designing allocation algorithms in the face of strategic reporting (e.g. Agarwal and Budish, 2021).[5] While these ideas are well studied in canonical economics settings, we are not aware of a general framework for addressing manipulation in empirical decision rules trained based on correlations in high-dimensional data, the type of rules that have proliferated in society in recent years.

By contrast, the modern machine learning paradigm typically relies on a variety of atheoretic methods to train (and re-train) decision rules based on high-dimensional correlations (Breiman, 2001). To address the manipulation that arises during implementation, a literature in statistics has considered variants of the 'industry' approach that use iterated retraining to adapt to covariate shift (cf. Sayed-Mouchaweh and Lughofer, 2012). These approaches are typically agnostic about the forces that lead to shifts, instead learning from mistakes. However, the shifts induced by manipulation have a predictable structure, so some of these realized mistakes – and latent vulnerabilities – are unnecessary. Our approach demonstrates how these forces can be

[5]Also related, Bryan et al. (2015) experimentally estimates a model where individuals' propensity to repay a loan depends on a fixed type as well as a heterogeneous susceptibility to social pressure, and Hussam et al. (2017) finds that people manipulate reports to favor friends and family when they believe the reports will be used to allocate grants, and explores methods to reduce this tendency.

modeled explicitly within a machine learning paradigm – a shift in perspective that we hope can help bridge these two paradigms and lead to more robust systems.

In this respect, our paper relates to a budding theoretical literature that has started to bridge these approaches, suggesting that behavioral responses might be incorporated in decision rules trained from data. Recent work in mechanism design (Frankel and Kartik, 2019, 2020; Ball, 2019; Hennessy and Goodhart, 2023) develops conceptual foundations, and shows that in settings like ours, the revelation principle can fail. In computer science, work on 'strategic classification' illustrates simple theoretical cases. Bruckner and Scheffer (2011) and Hardt et al. (2016) compute Stackelberg equilibria in linear classification settings where agents are strategic, with known costs that are the same for all people; Dong et al. (2018) extend this approach to an iterative environment. Perdomo et al. (2020) characterize general settings and is agnostic about the functional form of the cost function. Kleinberg and Raghavan (2019) and Milli et al. (2019) consider restricting to predictors that have a causal impact on the outcome, which can result in productive manipulation (for example, an exam induces students to study and learn general knowledge) but can reduce predictive performance. A different strand on adversarial machine learning considers the case where agents' objectives are exactly opposed to the policymaker's (cf. Huang et al., 2011).[6]

While this theoretical literature has begun to suggest how behavioral responses might be incorporated in machine decision algorithms, we are unaware of prior work that demonstrates how to estimate such algorithms, or that implements and evaluates the performance of such algorithms under real manipulation. Our paper thus makes two main contributions. First, we develop an estimable approach to adjust how decision rules are trained to anticipate manipulation. This yields rules that function well under manipulation even when fully transparent. And second, to our knowledge for the first time in any literature, we design and implement a field experiment that estimates and evaluates such decision rules in a real-world setting.

[6]Also related is Eliaz and Spiegler (2019), which shows that incentive problems can theoretically arise even in a setting where agents and the policymaker have identical objective functions, if the policymaker adjusts their objective function with regularization.

2 Model

This section introduces the model underlying our strategy-robust adjustment. We focus on the stylized case where the decision rule is linear and costs are quadratic, which allows us to derive solutions that directly map to the field experiment.[7] Later, in Section 5, we illustrate how the approach can be extended to a non-linear (decision tree) setting.

2.1 Setting

A policymaker observes a *training sample*, i.e., a subset of instances that possess both features $\mathbf{x}_i$ and preferred decisions y_i. The policymaker also obtains information on the costs of manipulating features, which will be detailed later. The policymaker would like to estimate the parameters of a decision rule $\hat{y}(\mathbf{x}_i)$ that will then be applied to a different *implementation* subset where only features $\mathbf{x}_i$ are observed, and may be manipulated. The decision rule $\hat{y}(\mathbf{x}_i)$ could represent, for example, the amount of aid or credit to grant based on a person's visible assets or digital behavior; how much a social network will prioritize a piece of content based on its characteristics and initial engagement; whether to interview an individual based on the text in their resume; and so forth.

The policymaker has a preferred action y_i for each individual i. The action $y_i = y(\underline{\mathbf{x}}_i, e_i)$ can be expressed as a function of i's bliss behavior $\underline{\mathbf{x}}_i$ and an idiosyncratic preference e_i. However, the implemented policy $\hat{y}(\mathbf{x}_i)$ may only be a function of the individual's *behavior* $\mathbf{x}_i$; in the linear case

$$\hat{y}(\mathbf{x}_i) = \alpha + \boldsymbol{\beta}'\mathbf{x}_i.$$

Individuals can manipulate their behavior $\mathbf{x}_i$ away from their bliss level $\underline{\mathbf{x}}_i$ at some cost, for instance by hiding assets (Camacho and Conover, 2011) or changing the keywords on their resume (Caprino, 2019). Each individual receives utility from the

[7] Linear models are relevant to the empirical setting we explore: in the context of digital credit, Björkegren and Grissen (2015) and Björkegren and Grissen (2020) find that models linear in parameters achieve comparable or better performance than (nonlinear) random forests.

policy's decision, minus their own cost of manipulation[8]

$$u_i(\hat{y}, \mathbf{x}_i) = \hat{y}(\mathbf{x}_i) - c_i(\mathbf{x}_i, \underline{\mathbf{x}}_i).$$

Individuals i are heterogeneous in two main respects: bliss behaviors $\underline{\mathbf{x}}_i$ and manipulation costs $c_i(\cdot)$. When manipulation costs are quadratic,

$$c_i(\mathbf{x}_i, \underline{\mathbf{x}}_i) = \frac{1}{2}(\mathbf{x}_i - \underline{\mathbf{x}}_i)' C_i (\mathbf{x}_i - \underline{\mathbf{x}}_i)$$

for cost matrix

$$C_i = \begin{bmatrix} c_{11i} & \cdots & c_{1Ki} \\ \vdots & \ddots & \vdots \\ c_{K1i} & \cdots & c_{KKi} \end{bmatrix}.$$

This parameterization allows for heterogeneity by behavior (indices jk) and person (index i). Some behaviors may be harder to manipulate than others, either by themselves (the diagonal) or in conjunction with other behaviors (the off-diagonals). Different types of individuals may also find it more or less costly to manipulate behaviors; for example, very clever people and or those with low opportunity costs might have lower costs.

Optimal behavior

i chooses optimal behavior $\mathbf{x}_i^*$ to maximize utility

$$\mathbf{x}_i^*(\hat{y}(\cdot)) = \arg\max_{\mathbf{x}_i} \left[u_i(\hat{y}, \mathbf{x}_i) \right]. \tag{1}$$

When the decision rule is linear and costs are quadratic, optimal behavior is

$$\mathbf{x}_i^*(\boldsymbol{\beta}) = \arg\max_{\mathbf{x}_i} \left[\alpha + \boldsymbol{\beta}' \mathbf{x}_i - c_i(\mathbf{x}_i, \underline{\mathbf{x}}_i) \right] = \underline{\mathbf{x}}_i + C_i^{-1} \boldsymbol{\beta}.$$

When the decision does not depend on behavior ($\boldsymbol{\beta} = \mathbf{0}$), i's optimal behavior equals his bliss level ($\mathbf{x}_i^*(\mathbf{0}) = \underline{\mathbf{x}}_i$). However, as $\boldsymbol{\beta}$ moves away from zero, i's behavior moves in the same direction, down-weighted by his cost of manipulation (highlighted in blue).

[8] We focus on the benchmark case where the utility from the decision exactly coincides with the policymaker's prediction $u_i(\hat{y}) \equiv \hat{y}$.

2.2 Decision rules

If the decision rule could condition on the cost function c_i of each individual i it faced, it could in principle invert manipulation to infer each individual's type. However, cost functions are not observed during implementation. Even during training, the policymaker typically has limited information about them: we assume they have beliefs about i's costs denoted with subscript q, $C_{iq} \sim \mathbb{C}_i$. We later discuss how the policymaker can obtain these beliefs.

If the policymaker faces loss $L(\mathbf{y}, [\hat{y}_i]_i)$ for classifying each individual i as $\hat{y}_i$, then a **strategy-robust decision rule** is given by

$$\hat{y}(\cdot) = \arg\min_{\tilde{y}(\cdot)} \mathbb{E}\left[L\left(\mathbf{y}, \left[\tilde{y}(\hat{\mathbf{x}}^*_{iq}\left(\tilde{y}\left(\cdot\right)\right)\right]_i\right)\right] \tag{2}$$

which deviates from standard supervised learning because it replaces i's sample behavior $\mathbf{x}_i$ with their anticipated manipulated behavior $\hat{\mathbf{x}}^*_{iq}(\hat{y}(\cdot))$ if their costs were C_{iq}.

In the linear setting, the strategy-robust decision rule is given by

$$\alpha^{SR}, \boldsymbol{\beta}^{SR} = \arg\min_{\alpha,\boldsymbol{\beta}} \mathbb{E}\left[\frac{1}{N}\sum_i \left(y_i - \alpha - \boldsymbol{\beta}'(\underline{\mathbf{x}}_i + C_{iq}^{-1}\boldsymbol{\beta})\right)^2 + \ldots\right] \tag{3}$$

which deviates from ordinary least squares by the manipulation term $C_{iq}^{-1}\boldsymbol{\beta}$. The term '…', discussed in Section 2.3, may include regularization terms $R_\lambda(\cdot)$ that penalize model complexity, or, if the policymaker cares about the costs that individuals incur manipulating behavior, an additional term to represent those costs $M(\cdot)$.

Estimation

We estimate the primitives describing behavioral responses to the decision rule $\mathbf{x}^*_i(\hat{y}_i(\cdot))$ in two steps. First, in a training sample, we observe labels $\mathbf{y}$, and behavior $\mathbf{x}$, for a subset of the population. We estimate the distribution of manipulation costs $\mathbb{C}$ from this subset. Second, these cost estimates are used in equation (3) to estimate $\boldsymbol{\beta}^{SR}$ – the strategy-robust decision rule. This decision rule can then be deployed to the full population, including individuals for whom no labels $\mathbf{y}$ are observed.

There are a variety of ways one could imagine estimating the parameters that

describe behavioral responses $\mathbf{x}_i^*(\hat{y}_i(\cdot))$. The most direct approach – and the main approach we test empirically – uses an experiment to randomly assign individuals in a training sample to one of several decision rules $\hat{y}_i$, communicates the decision rule to i, observes the resulting behavior $\mathbf{x}_i^*(\hat{y}_i(\cdot))$, and estimates parameters of $\mathbb{C}$ from those responses.[9] In the linear setting, all parameters can be estimated with simple perturbations from a base model $\boldsymbol{\beta}_0$. Consider assigning either this base model, or a perturbed model $\boldsymbol{\beta}_k = \boldsymbol{\beta}_0 + \beta\boldsymbol{\delta}_k$ that has been perturbed by amount β along dimension k, where $\boldsymbol{\delta}_k$ represents the kth unit vector. One will then observe the resulting vector of manipulated behavior

$$\mathbf{x}_i^*(\boldsymbol{\beta}_k) = \underline{\mathbf{x}}_i + C_i^{-1}\boldsymbol{\beta}_0 + \beta C_i^{-1}\boldsymbol{\delta}_k$$

where experimental variation in the last term identifies $\mathbb{C}$.

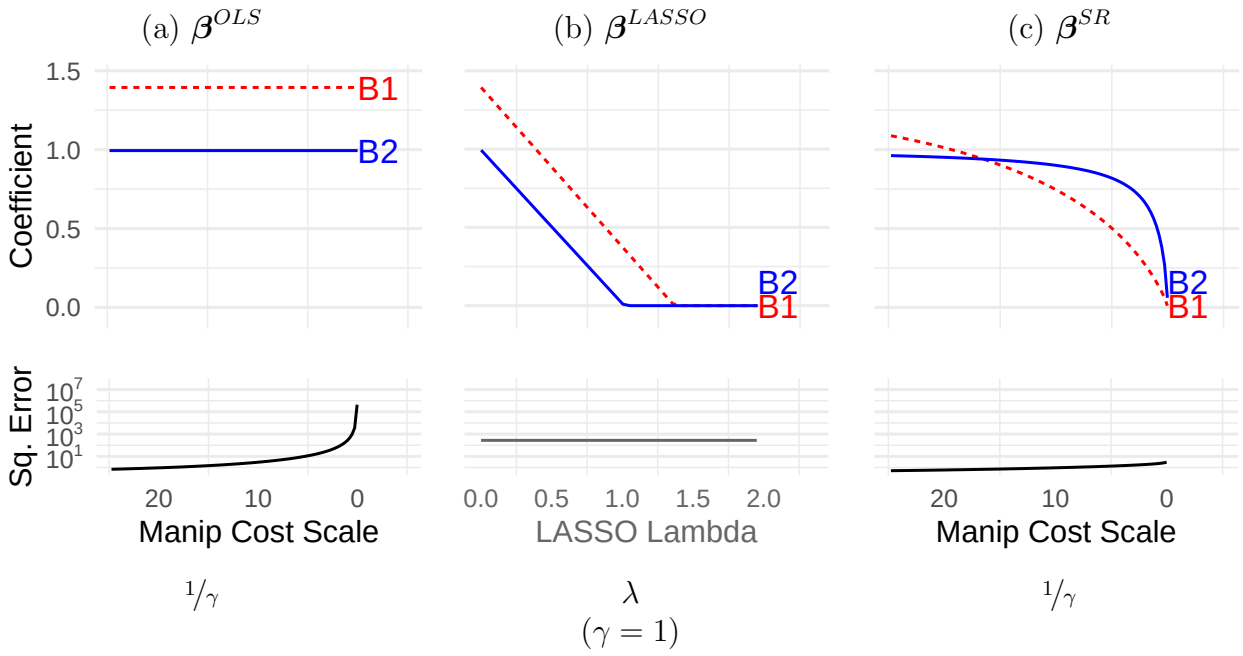
In our field experiment (Section 3), we make additional assumptions about the structure of costs (particularly, we will assume that off diagonal costs are zero, $c_{jki} \equiv 0$ for $j \neq k$, and that a person's relative gaming ability is the same across all behaviors, $c_{jkiq} \equiv c_{jk} \cdot \frac{1}{\gamma_{iq}}$) and then use a general method of moments (GMM) loss function for estimation. We provide details on the estimation procedure, as well as how estimation would change in other settings, after describing the field experiment. We use a base model $\boldsymbol{\beta}_0 = \mathbf{0}$ to mimic '*greenfield*' settings before a decision rule has been implemented. In real-world applications where a decision rule is already in use and behavior is already manipulated (i.e., a '*brownfield*' environment), one may wish to instead assess deviations from a status quo or proposed naïve model $\boldsymbol{\beta}_0$.

2.3 Intuition and Discussion

We provide intuition for how the method works using Monte Carlo simulations. These simulations involve a policymaker who implements a linear decision rule ($y = \beta_1 x_1 + \beta_2 x_2 + \alpha$) that is based on two observed behaviors, x_1 and x_2. In the simulations, x_1 is initially more predictive of i's type than x_2, but it is also more susceptible to manipulation ($c_{11i} \ll c_{22i}$). Figure 1 compares the performance of three different approaches to designing decision rules in this setting.

[9] This approach makes several assumptions, including that costs of manipulation are stable over

Figure 1: Common vs. Strategy-Robust Decision Rules



Notes: The policymaker's desired allocation is $\mathbf{y} = a + \mathbf{b}'\underline{\mathbf{x}} + \boldsymbol{e}$. The first behavior is more predictive ($b_1 > b_2$), but is easily manipulable ($c_{11i} \ll c_{22i}$) and has more manipulation noise. **(a)** OLS performance deteriorates when behavior can be manipulated. (**b**) LASSO penalization favors x_1, which will be manipulated as soon as the decision rule is implemented. **(c)** Our method anticipates that x_1 will be manipulated, and shifts weight to x_2 as behavior becomes manipulable.

$\underline{\mathbf{x}}_i \overset{iid}{\sim} N\left(\mathbf{0}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)$, $\mathbf{b} = \begin{bmatrix} 1.4 \\ 1 \end{bmatrix}$, $\mathbf{C}_i = \frac{1}{\gamma\gamma_i^{het}} \begin{bmatrix} 4 & 0 \\ 0 & 32 \end{bmatrix}$, $\frac{1}{\gamma_i^{het}} \overset{iid}{\sim} Uniform\,[0, 10]$,
$e_i \overset{iid}{\sim} N\,(0, 0.25)$. Squared error measured on an out of sample draw from the same population, incentivized to that decision rule.

Panel (a) of Figure 1 illustrates the parameters selected by OLS. These parameters do not depend on either the relative (c_{11i} vs. c_{22i}) or absolute (γ) costs of manipulation. Instead, OLS maximizes predicted performance within the unincentivized sample $(\mathbf{x}_i(\mathbf{0}), y_i)$. As shown in the lower portion, of Figure 1a, OLS performs poorly as manipulation becomes easier.

Our approach is related to regularization in that it systematically alters how features are expressed in a model. LASSO regularization (panel b) likewise places the most weight on x_1, since it is most predictive of y in the unincentivized sample. As the regularization penalty increases, both parameters are similarly penalized. However, LASSO selection does the wrong thing: it kicks x_2 out of the regression first. Regularization is commonly used to reduce overfitting when drawing different samples from the same state of the world. However, regularization does not consider how a model would best fit a different state of the world – as occurs when a decision rule is implemented and behavior becomes manipulated.

In contrast, the strategy-robust approach (panel c) adjusts the decision rule to account for how features $(\mathbf{x}_i(\boldsymbol{\beta}), y_i)$ will shift in the counterfactual state of the world where people manipulate their features in response to the model. As shown in Figure 1c, when manipulation costs are high, our method approaches OLS; as manipulation becomes easier, our method adjusts the model, in this case by penalizing x_1, and thus attains better performance. In general, this adjustment can either increase or decrease the weight on manipulated features. Adjusting for strategy-robustness may be needed even if there is no concern about overfitting (e.g., the sample is fixed, or one observes the entire population).[10]

Understanding the strategy-robust adjustment

To better understand the strategy-robust solution, we step through moment conditions in a simple case and illustrate comparative statics with simulations. The strategy-robust solution for $\boldsymbol{\beta}$ coincides with a nonlinear least squares estimator in the simple case where the policymaker knows the distribution of costs in the training sample, only

time. In Section 4.3 we discuss alternative assumptions and approaches to estimating these responses.

[10]The method can also exploit cost interactions, adjusting behaviors that make it easier to shift other predictive behaviors (akin to Ramsey (1927) taxation). When manipulating x_1 makes it easier to manipulate x_2 (c_{12i} sufficiently negative), our method further reduces weight on x_1.

cares about predictive accuracy ($M(\cdot) \equiv 0$), and there are no additional regularization terms ($R_\lambda(\cdot) \equiv 0$). First order conditions for $\boldsymbol{\beta}$ are then given by:

$$\mathbb{E}\left[\varepsilon_i\left(\boldsymbol{\beta}, \hat{\mathbf{x}}_{iq}(\boldsymbol{\beta})\right)\left(\frac{\partial \varepsilon_i}{\partial \boldsymbol{\beta}}' + \frac{\partial \varepsilon_i}{\partial \mathbf{x}}' \frac{\partial \hat{\mathbf{x}}_{iq}}{\partial \boldsymbol{\beta}}\right)\right] = \mathbf{0}$$

$$\varepsilon_i(\boldsymbol{\beta}, \mathbf{x}) = y_i - \alpha - \boldsymbol{\beta}'\mathbf{x}$$

The first term captures how $\boldsymbol{\beta}$ affects fit holding $\mathbf{x}_i$ constant, and the second term accounts for manipulation: the influence of $\boldsymbol{\beta}$ on $\mathbf{x}_i$. This results in moment condition:

$$\mathbb{E}\left[\hat{\mathbf{x}}_{iq}(\boldsymbol{\beta}) \cdot \varepsilon_i(\boldsymbol{\beta}, \hat{\mathbf{x}}_{iq}(\boldsymbol{\beta}))\right] = -\mathbb{E}\left[C_{iq}^{-1}\boldsymbol{\beta} \cdot \varepsilon_i(\boldsymbol{\beta}, \hat{\mathbf{x}}_{iq}(\boldsymbol{\beta}))\right] \tag{4}$$

When estimated on unmanipulated behaviors ($\underline{\mathbf{x}}_i$), this differs from standard estimators due to three forces.

Shifts First, our solution anticipates that *levels* of behaviors will best respond to the choice of $\boldsymbol{\beta}$. The left side of equation (4) is akin to OLS except with counterfactual behaviors $\hat{\mathbf{x}}_{iq}(\boldsymbol{\beta})$. When these behaviors cannot be manipulated ($C_i \to \boldsymbol{\infty}$), our solution corresponds to OLS. If each behavior j has the same manipulation cost for all people (i.e., $c_{jki} \equiv c_{jk}$), each person will shift their behavior the same amount in response to a given decision rule $\boldsymbol{\beta}$. As a result, the method will maintain the same coefficient on the manipulated behavior, regardless of how easy it is to manipulate. In that case, only the constant term need be adjusted. Although this does not affect predictive performance, individuals may incur substantial costs manipulating. We demonstrate this in a simulation in the Supplemental Online Appendix (section S4).[11]

Signaling and noise Second, our solution anticipates that manipulation may be *heterogeneous* across people: our moment expectations are taken over the distribution of gaming ability. This typically accounts for most of the adjustment of our method, and can affect the decision rule in two ways. If the people who find it easier to manipulate are differentially likely to have higher values of the outcome (y_i), manipulation represents a signal of the underlying type (as in Spence (1973) and Nichols and

[11] Please note that this paper contains two distinct appendices: The main Appendix included at the end of this manuscript (where sections are prefaced with "A"), and the Supplemental Online Appendix (where sections are prefaced with "S").

Zeckhauser (1982)). Our method will exploit these correlations, and will tend to *increase* the weight on behaviors that are easy for the targeted people to manipulate (demonstrated in a simulation in the Supplemental Online Appendix Section S4.2). On the other hand, there is often additional variance in manipulation ability between people that is not observably related to the outcome. Unlike revelation mechanisms where a person's type can be inferred from their behavior, in our setting, like the theoretical settings of Frankel and Kartik (2019, 2020) and Ball (2019), individuals have both heterogeneous types and heterogeneous ability to shift their behaviors. As a result, the people with more desirable behaviors are a combination of those with more desirable types and those with higher ability to game, regardless of type. This unobserved variance will tend to 'muddle' the relationship between a behavior x_i and type $\underline{\mathrm{x}}_i$ (Frankel and Kartik, 2019). Because the mapping may not be one-to-one, types may not be fully revealed. In these cases, our method will tend to attenuate the coefficients of behaviors for which there is a lot of unobserved variance. When the method attenuates the coefficients on manipulable behaviors, this force is typically the major reason.

Subgame perfection Third, our solution anticipates the *gradient* of those behaviors: how $\mathbf{x}_i$ would respond if $\boldsymbol{\beta}$ were to deviate off path. This is because the right-hand side of equation (4) differs from orthogonality, and results in a Stackelberg or subgame-perfect equilibrium. In contrast, standard estimators assume that if $\boldsymbol{\beta}$ were to deviate, $\mathbf{x}_i$ would remain fixed. In that sense, standard approaches compute a one-step best response. Thus, even if one trained a standard decision rule on data from a strategy-robust equilibrium $(y_i, \mathbf{x}_i^*(\boldsymbol{\beta}^{SR}))$, it would result in a different decision rule that would escape the equilibrium.[12] In similar settings, Ball (2019) and Frankel and Kartik (2020) show theoretically that a Stackelberg solution like ours that anticipates reactions (and thus commits to not exploit partial equilibrium correlations) can lead to better predictive performance than repeated best responses.

[12] $\boldsymbol{\beta} = \boldsymbol{\beta}^{SR}$ is a solution to $\mathbb{E}_i\left[\mathbf{x}_i^*(\boldsymbol{\beta}^{SR})\varepsilon_i(\boldsymbol{\beta}, \mathbf{x}_i^*(\boldsymbol{\beta}^{SR}))\right] = \mathbf{0}$ only if the right hand side of equation (4) is zero.

Performance

Through further simulations, we illustrate how the strategy-robust approach can produce better decisions when people manipulate behavior. Table 1 simulates a scenario with three behavioral features, where x_1 is initially more predictive of the individual's type ($b_1 = 3$) than the other two features ($b_2 = b_3 = 0.1$); however, x_1 is also easier to manipulate ($c_{22i} = 2 \cdot c_{11i}$ and $c_{33i} = 4 \cdot c_{11i}$) and is subject to more noise. (We assume that the noise is proportional to the manipulation cost.) Through further simulations, we illustrate how the strategy-robust approach can produce better decisions when people manipulate behavior. Table 1 simulates a scenario with three behavioral features, where x_1 is initially more predictive of the individual's type ($b_1 = 3$) than the other two features ($b_2 = b_3 = 0.1$); however, x_1 is also easier to manipulate ($c_{22i} = 2 \cdot c_{11i}$ and $c_{33i} = 4 \cdot c_{11i}$) and is subject to more noise. (Here the noise is proportional to the manipulation cost.)

Panel B of Table 1 illustrates the performance of two status-quo approaches to constructing decision rules. In the first row, OLS models the static relationship between features and the outcome. This approach would perform well if behavior were fixed (as indicated by the low squared loss in training data); however, once people respond to the decision rule, the OLS rule leads to very poor decisions (the loss 'when implemented'). The next set of rows in Panel B illustrate a common 'industry' approach, in which the OLS model is periodically retrained. For instance, after observing behavior in the first period (when the rule $\boldsymbol{\beta}^{OLS}$ is active), the model is then re-trained to obtain $\boldsymbol{\beta}^{OLS(2)}$, which places lower weight on the manipulated x_1. However, once people respond to this new rule, it also performs poorly. As this process continues, the rule always appears to predict well on the training sample but makes poor decisions when actually implemented. In this case, the process does not converge; it alternates between decision rules that place high and low weight on x_1.[13] Thus, standard approaches can perform poorly even in stable settings with perfect information. In settings with noise or frictions in learning, a system might unexpectedly and catastrophically fail when the other side discovers how to exploit it.

In contrast, the strategy-robust decision rule ($\boldsymbol{\beta}^{SR}$ in Panel C) adjusts the coef-

[13] These oscillations can be dampened by using cumulative data from all prior periods, as shown in the Supplemental Online Appendix (section S4.2.1). However, that still takes several iterations to converge, and the resulting equilibrium is inferior to that of the strategy-robust approach.

Table 1: Manipulation Can Harm Prediction (Monte Carlo)

	Decision Rule				**Performance** (squared loss)	
	β_1	β_2	β_3	α	On training data	When implemented
Panel A: **Data Generating Process (Unmanipulated)**						
$\boldsymbol{b}^{DGP}$	3.00	0.10	0.10	0.20	0.27	3745.05
Panel B: **Standard Approaches**						
$\boldsymbol{\beta}^{OLS}$	3.04	0.06	0.12	0.21	0.27	3961.23
Industry Approach						
$\boldsymbol{\beta}^{OLS(2)}$	0.06	2.09	−1.68	−0.80	3.28	625.76
$\boldsymbol{\beta}^{OLS(3)}$	3.11	−0.04	0.22	0.17	0.27	4332.21
$\boldsymbol{\beta}^{OLS(4)}$	0.12	2.08	−1.67	−0.76	3.07	619.06
⋮						
$\boldsymbol{\beta}^{OLS(1001)}$	3.74	−1.34	1.57	−0.39	1.38	11 611.88
$\boldsymbol{\beta}^{OLS(1002)}$	0.70	1.86	−1.53	−0.40	1.67	565.38
Panel C: **Strategy-Robust Method**						
$\boldsymbol{\beta}^{SR}$	0.50	0.54	−0.10	−1.81	9.16	1.94
If policymaker knows only the distribution of costs between individuals:						
$\boldsymbol{\beta}^{SR}_{C_{iq}=bootstrap_i(C_i)}$	0.31	0.49	0.15	−0.74	7.00	3.38
If costs are misestimated:						
$\boldsymbol{\beta}^{SR}_{C_{iq}\equiv 2\cdot diag(C_i)}$	0.66	0.72	−0.35	−1.57	6.89	10.83

Notes: Monte Carlo simulation results. Panel A shows the coefficients that relate the outcome (y) to behaviors ($\mathbf{x}$) under the data generating process (DGP). Panel B shows coefficients from OLS. For the industry approach, the training data for $\boldsymbol{\beta}^{OLS(r)}$ is the manipulated data from when $\boldsymbol{\beta}^{OLS(r-1)}$ is assigned. Panel C shows coefficients estimated with the strategy-robust method with costs known during training ($C_{iq} \equiv C_i$); with heterogeneous costs bootstrapped between individuals over 10 draws; and with costs mis-estimated to be double and to omit off-diagonals. Performance is assessed on the same sample of individuals under the training data, and when the data is manipulated. Parameters:

$$C_i = \tfrac{1}{\gamma_i}\begin{bmatrix} 1.0 & 0.1 & 0.2 \\ 0.1 & 2.0 & 0.8 \\ 0.2 & 0.8 & 4.0 \end{bmatrix}, \underline{\mathrm{x}} \overset{iid}{\sim} N\left(\mathbf{0}, \begin{bmatrix} 1.0 & 1.0 & 0.1 \\ 1.0 & 2.0 & 1.0 \\ 0.1 & 1.0 & 1.0 \end{bmatrix}\right), \gamma_i = \begin{cases} 1 & \underline{\mathrm{x}}_{i1} \leq 0.2 \\ 10 & \underline{\mathrm{x}}_{i1} > 0.2 \end{cases}, e_i \overset{iid}{\sim} N(0, 0.25)$$

ficients by penalizing the behavior x_1 which has more manipulation noise, instead shifting weight to behaviors that are harder to manipulate (x_2 and x_3). It anticipates manipulation off-path, sacrificing performance in the environment in which it is trained (in-sample, no manipulation) for performance in the counterfactual implementation environment where there will be manipulation. When individuals manipulate as described in the model, the strategy-robust decision rule exceeds the performance of standard estimators.[14]

Our method performs similarly well when the policymaker knows only the distribution of costs $\mathbb{C}$ and not the cost of each individual in its training sample (second to last row of Table 1). The method can also reduce risk when behavioral responses are misestimated. For instance, the last row considers the case where all off-diagonal elements are erroneously set to zero, and the estimated costs of manipulation are two times too large. Performance deteriorates relative to the case where we know the true cost matrix, but our method still outperforms OLS in the presence of manipulation.

Regularization and the social cost of manipulation

One may wish to add additional terms into '...' in equation (3), for two reasons. First, strategy-robust rules may be estimated on samples, in which case it can improve *out-of-sample* counterfactual fit to combine the strategy-robust adjustment with a regularization term such as $R_\lambda^{LASSO}(\boldsymbol{\beta}) = \lambda \sum_k |\beta_k|$. We use this approach in our experiment (Section 3) and demonstrate in a simulation (Supplemental Online Appendix S4).

Second, when the policymaker cares about not only the resulting allocation, but also the costs that individuals incur manipulating, one may include a function describing this cost, $M(\cdot)$. A policymaker that is narrowly concerned with their own objective may thus select different decision rules from one that cares about social welfare: for instance, a profit-maximizing firm may be satisfied with an equilibrium where all individuals expend welfare gaming a test; a social planner may not be.

[14]The strategy-robust approach can also be combined with the industry approach by using the strategy-robust approach first, then iteratively retraining – see Supplemental Online Appendix (Section S4).

3 Field Experiment in Kenya

We designed a field experiment to test the performance of strategy-robust decision rules in a real-world setting. Working with the Busara Center for Behavioral Economics in Nairobi, we developed and deployed a new smartphone-based application ('app') to 1,557 research subjects.

The app was designed to mimic key features of 'digital credit' applications that have become wildly popular in recent years and which are transforming how consumers in developing countries access credit (Bharadwaj and Suri, 2020; Suri et al., 2021). In a typical digital credit application, lending decisions are based on an 'alternative credit score' that is constructed by applying machine learning algorithms to data on how the loan applicant uses their phone (Björkegren, 2010; Francis et al., 2017; Björkegren and Grissen, 2020). At the time of our field experiment, CGAP (2018) estimated that 27% of Kenyan adults had an outstanding 'digital credit' loan. Yet, there is mounting evidence that digital credit is a domain where manipulation is problematic. In one example, Bloomberg covered a story where "a scam artist studied the loan-approval patterns for several months, using 30 different sim cards to generate data sets and deciphering the lender's algorithms. He fleeced the firm of $30,000 in one day and then vanished."[15] The potential for manipulation is also salient to everyday customers: in a survey conducted in Kenya and Tanzania, respondents listed the desire to obtain larger digital loans as one of the top-five reasons for saving money in their mobile money accounts (FSD Kenya, 2018).

This section describes the app and experimental design; estimates costs of manipulation and derives strategy-robust decision rules using our method; and compares the performance of these new algorithms to traditional learning algorithms. Our design was pre-specified in a pre-analysis plan registered in the AEA RCT registry under AEARCTR-0004649.

[15] Bloomberg Technology, Sep. 22, 2015. American Banker similarly describes how fraudsters have learned to take out large loans by manipulating a sequence of loan applications (`https://www.americanbanker.com/news/how-fraudsters-are-gaming-online-lenders`).

3.1 Experimental design and smartphone app

We designed our experiment to create incentives similar to those of a digital credit lending app. These apps run in the background on a smartphone, and collect data on phone use (including data on communications, mobility, social media behavior, and much more). Digital credit apps use this information to allocate loans to people who appear creditworthy (i.e., for whom $\hat{y}_i$ exceeds some threshold). Since financial regulations prevented us from actually underwriting loans to research subjects, we instead focused on analogous problems where a decisionmaker wishes to allocate resources to individuals with specific characteristics—for instance, by paying individuals who have a certain income level, or other characteristic (e.g., level of intelligence or education).[16]

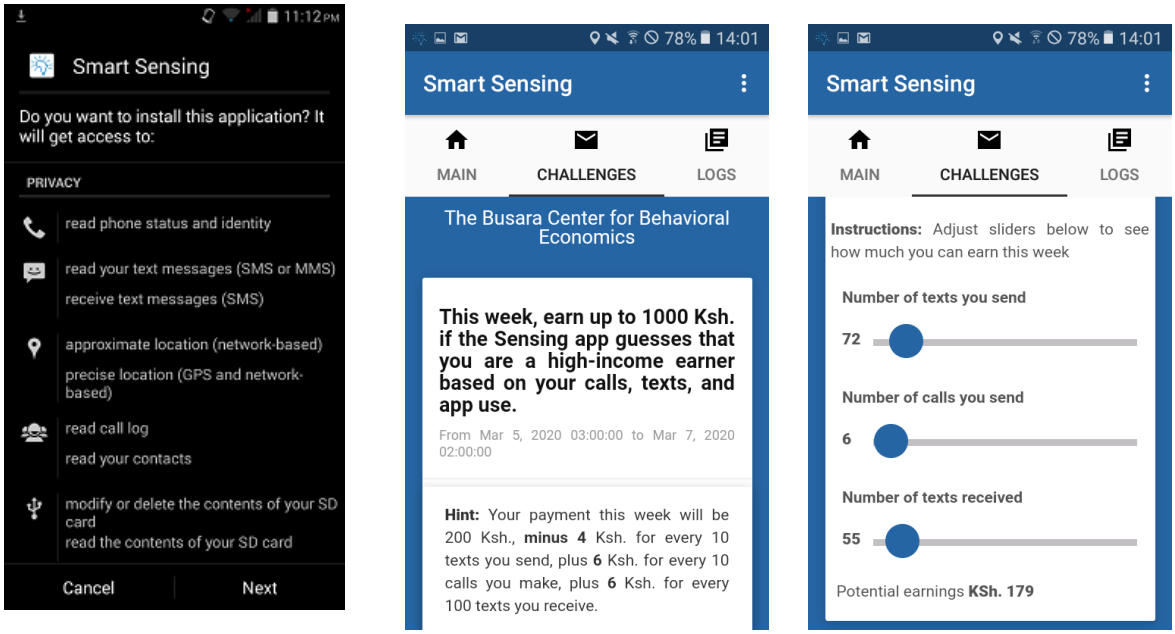
Smartphone app The 'Smart Sensing' app we created has two key features. First, it ran in the background on the phone to capture anonymized metadata on how individuals use their phones, such as when calls or texts were placed, which apps were installed and used, geolocation, battery usage, wifi connections, and when the screen was on. In total, we extract over 1,000 behavioral indicators ("features"). Second, the app delivered weekly "challenges" to participants, which appeared on the user's phone, and which provided financial rewards based on the user's behavior (see Figure 2). We describe these challenges in greater detail below. Participants were paid a base amount of 100 Ksh. for uploading data, plus any challenge winnings, directly via mobile money at the end of each week.

Study population and recruitment The subject population consists of Kenyans aged 18 years or older who own a smartphone and could travel to the Busara center in Nairobi. Participants were recruited in person in public spaces in Nairobi, and were invited for an enrollment session at the Busara center. During enrollment, participants completed a baseline survey, and were asked to install and keep the Smart Sensing app on their phones for roughly 16 weeks. During the consenting process, participants were told the dimensions of behavior that would be recorded by the app, and were

[16] While these prediction targets differ from credit-worthiness, there are many settings where similar characteristics are inferred by digital traces (for example, social assistance programs that target the poor (Aiken et al., 2021), or digital advertisers who target college students).

Figure 2: Smart Sensing App

(a) Installation Screen (b) Challenge under Transparency (c) Earnings Calculator



given the opportunity to ask questions. Participants had the opportunity to view the Android permissions required for the app to function properly, and generally appeared to understand the privacy tradeoffs involved in participation. Our sample includes only participants who opted in. 83% of participants elected to receive challenges in English, 16% in Swahili, and 1% in both languages.

Weekly rhythm The study followed a weekly rhythm. Each Wednesday at noon, each participant received a generic notification on their phone that said, 'Opt in to see this week's challenge!' If the participant opened the app and opted in, they were shown information about the decision rule they faced that week (see Figure 2). Challenges were valid until 1pm Tuesday. At the conclusion of the challenge, participants had 21 hours to ensure that their data was uploaded (i.e., until 10am Wednesday). Busara then determined how much each participant should be paid, and payments were sent via mobile money by noon Wednesday, at which point the next week's cycle would begin.[17]

[17]Participants could attrite by not opting in to the weekly challenge or by not uploading their data. In either case, the Busara center attempted to contact such individuals via text message and

Randomization of decision rules Each week, each participant was randomly assigned to one of three types of decision rules: control, simple, or complex. The control decision rules ($\boldsymbol{\beta}_0 \equiv \mathbf{0}$), which were deployed during the first few weeks of the experiment, did not require any action from participants; each individual who successfully uploaded their data received the same reward irrespective of how they used their phone in that week.[18] Each simple rule made decisions based on one specific behavior ($\boldsymbol{\beta}_k = \beta_k \boldsymbol{\delta}_k$), and were of the form, 'We'll pay you β_k for each x_k you do', where behavior k and amount β_k were assigned randomly. Most incentive amounts were positive but some were negative (participants were incentivized to reduce behavior).[19] For example, one simple challenge was, "You will receive 12 Ksh. for every incoming call you receive this week, up to Ksh. 250." The control and simple decision rules were used to collect training data.

Finally, in the last part of the study, we assign complex decision rules. These were designed to mimic real-world applications of machine learning, in which people can receive a desirable benefit based on how they are classified. An example is depicted in Figure 2. The complex decision rules were of the form, 'We'll pay you m if you are classified as $\hat{y}$.' Our main analysis focuses primarily on responses to one such challenge, 'Earn up to 1000 Ksh. if the Sensing app guesses you are a high-income earner'; the results pooled over all complex challenges are provided in the Supplemental Online Appendix (Table S1).

Predicting user characteristics from app data

Using data from the 'control' weeks, where the app collected data on user behavior but did not provide incentives for people to change their behavior, we assess the

phone call, following an attrition protocol detailed in the Supplemental Online Appendix (Section S1.4). We include in our analysis only participant-weeks where the participant opted in and uploaded during the end-of-week upload window.

[18] Specifically, the subject received a challenge of the form, 'Dear user, you do not have to do anything for this week's challenge. You will receive an extra Ksh 100 for accepting this challenge.'

[19] Each individual's payment level for k was drawn from $\{-2r_k, -r_k, r_k, 2r_k, 4r_k, 8r_k\}$, for scalar r_k. We scaled the payout for each behavior so that the maximum payout could be achieved by someone reaching the 90th percentile of baseline behavior. Given budget constraints we were not able to assign simple challenges for all measurable behaviors, so we assigned simple challenges for behaviors k that were predictive of main outcomes in control weeks, or similar to a predictive behavior. For example, if outgoing calls were predictive, we also include a corresponding measure based on incoming calls. See Supplemental Online Appendix Section S1.5.

Table 2: Behavior Predicts Individual Characteristics

		Monthly Income			**Intelligence** (Above Median Ravens)	
Mean Duration of Evening Calls		-0.559	(3.702)		0.0001	(0.0002)
Mean Duration of Outgoing Calls		-1.770	(8.965)		-0.0007	(0.0004)*
Calls with Non-Contacts		-42.023	(14.033)***	••	-0.002	(0.0006)***
Outgoing Text Count	••	10.211	(12.396)		0.0004	(0.0006)
Incoming Text Count	•	3.888	(7.974)	••	-0.0002	(0.0004)
Evening Text Count	•	-9.029	(7.815)		-0.0002	(0.0003)
Outgoing Call Count	••	76.752	(18.133)***		0.002	(0.0008)*
Missed Outgoing Call Count		-84.533	(31.636)***	•	-0.003	(0.0014)**
Outgoing Texts on Weekdays		-15.023	(15.210)		-0.001	(0.0007)
Max Daily Incoming Text Count		2.901	(21.212)	•	0.003	(0.0009)***
Intercept		5651.04	(430.141)***		0.480	(0.019)***
N (individuals)		1539			1557	
R^2		0.026			0.027	

Notes: Each column represents a regression of the outcome characteristics (column header) on behaviors measured through the Sensing app (rows) Observations include data collected during the first week the participant used the sensing app. Standard errors in parentheses. * = 10 percent significance, ** = 5 percent significance, *** = 1 percent significance. • : included in incentivized naive LASSO model, • : included in incentivized strategy-robust (SR) model.

extent to which a user's characteristics can be predicted based on how they use their phone. In Table 2, we observe that phone data can weakly predict monthly income and intelligence (above-median performance on Raven's matrices).[20]

3.2 Evidence that simple decision rules induce manipulation

Participants change behavior when facing simple algorithms. We show this using data from the 'simple' rules that form decisions based on one specific aspect of phone use (such as increasing the number of incoming calls). Table 3 presents a regression of each participant's weekly level of different behaviors (columns) on randomly assigned incentives to change specific behaviors (rows). There are three takeaways. First, individuals manipulate the behaviors that are incentivized, as shown by the diagonal, which is positive and significant for most behaviors. Second, some behaviors are

[20]The relatively low predictive power ($R^2 \approx 0.03$) is likely due to the fact that we have a small sample of relatively homogeneous users that are observed for short time spans. We estimate the regression model over the subset of predictors which were selected as predictive by LASSO, and for which we estimate costs in the experiment (Section 3.3).

more manipulable than others. For example, the number of texts sent was 49 times more responsive to incentives than the number of people called during the workday. And finally, incentivizing one behavior can affect others, as shown in the off diagonal elements. For example, incentivizing missed incoming calls also increased the number of texts sent (it may be that people sent messages to ask their contacts to call them back). In theory, our method can exploit these cross-elasticities, though many are noisily estimated in our data.

Table 3: Behavior Changes when Incentivized

Behavior incentivized	**Behavior observed** (change per ¢ of incentive)				
	# Texts sent	# Missed calls (incoming)	# Missed calls (outgoing)	# People called (Workdays, i.e. M-F, 9am-5pm)	# Calls w non-contacts (weekends)
# Texts sent	**24.51**	-0.052	-0.836	-0.305	-0.022
	(3.202)***	(0.588)	(0.87)	(0.217)	(0.368)
# Missed incoming calls	4.15	**0.708**	0.825	0.128	-0.002
	(2.196)*	(0.403)*	(0.597)	(0.149)	(0.252)
# Missed outgoing calls	-0.213	0.324	**1.187**	0.22	0.502
	(2.856)	(0.524)	(0.776)	(0.194)	(0.328)
# People called (workday)	2.308	0.156	0.679	**0.497**	0.108
	(2.505)	(0.46)	(0.681)	(0.17)***	(0.288)
# Calls w non-contacts (weekends)	-2.019	-0.056	1.234	0.015	**1.233**
	(2.866)	(0.526)	(0.779)	(0.194)	(0.329)***
Individual Fixed Effects	X	X	X	X	X
Week Fixed Effects	X	X	X	X	X
N (person-weeks)	7966	7966	7966	7966	7966
R^2	0.704	0.552	0.637	0.604	0.491

Notes: Standard errors in parentheses. Bold indicates diagonal: effect on behavior j when behavior j is incentivized. Each column represents a separate regression over the full set of behaviors assigned; only the first five coefficients reported here. N represents person-weeks during which 'simple' (single behavior) challenges were issued. * $p < 0.1$, ** $p < 0.05$, *** $p < 0.01$.

Separately, we find that behaviors that are to easier to manipulate on average also have a higher variance between people when incentivized. That is, behaviors that are hard to manipulate tend to be hard for everyone, but behaviors that are easy to manipulate on average tend to be differentially manipulable for different people. This

relationship is nearly proportionate (shown in Supplemental Online Appendix Section S2.1), suggesting an approximation allowing separable heterogeneity by behavior and by individual.

3.3 Parameterization and estimation

Having confirmed that, as expected, participants manipulate behavior when facing a decision rule, we next show how the data from the control and simple decision rules can be used to estimate the primitives of our model.In our context, given experimental variation in the decision rules during training, we first impose structure on the behavioral response (i.e., on the manipulation costs $\mathbb{C}$), and then use GMM to estimate those structural parameters from 'training' data (control and simple weeks). We show how those parameter estimates can be used to construct strategy-robust decision rules, and then, in Section 3.4, empirically evaluate those rules using separate implementation data collected during the final stage of the experiment.

Parameterizing behavioral response

We parameterize manipulation costs to make better use of our limited sample. We assume that manipulation costs take a quadratic form.[21] We allow for separable heterogeneity by behavior (jk) and person (iq), parameterized as

$$c_{jkiq} = \frac{1}{\gamma_{iq}} \cdot c_{jk}.$$

We allow individual gaming ability, $\gamma_{iq} = e^{-\boldsymbol{\omega}' z_i} + v_q$, to vary with self-reported tech skills $z_i \in \{0, 1\}$, and with unobserved heterogeneity $v_q \sim V$ (where $\mathbb{E}v_q = 0$).[22] Separability in heterogeneity implies that the noise induced by manipulation in behavior j will be proportional to its cost of manipulation c_{jj}. As a result, this expects that behaviors that are more manipulable on average are more subject to manipulation noise, and our method will tend to attenuate their expression in decision rules.

[21] In the Supplemental Online Appendix (Section S2.1), we show that the quadratic cost assumption is a reasonable (if imperfect) approximation of how people respond to variable incentive amounts.

[22] Tech skills explained the most heterogeneity in preliminary analysis. It is known only for the sample used to estimate primitives, but not for the implementation sample. Spence signaling will only be captured in this dimension of heterogeneity.

Estimating costs and other primitives

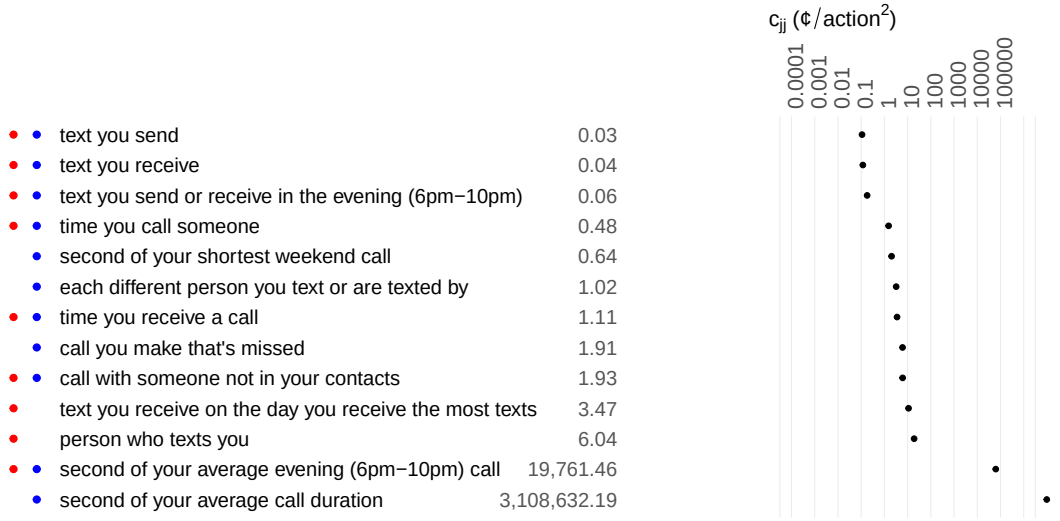
We use the simple and control challenges that were deployed during the first few weeks of the experiment to obtain training data to estimate manipulation costs and other primitives ($\underline{\mathbf{x}}_i$). Full details of the estimation procedure are provided in Appendix A1; the Appendix also describes how estimation would work in a brownfield setting where primitives are estimated after decision rules have already been implemented. It also shows how an adapted procedure can be used in a 'one shot' setting where each individual's behavior is observed only once (in our case, given the cost of onboarding new participants, we use multiple observations per individual to increase statistical power). When estimating costs, we regularize towards standard methods (such as OLS and LASSO). In particular, we penalize off-diagonal elements to zero because they are otherwise noisily estimated in our sample; this results in a diagonal cost matrix C.

The estimated costs of manipulation, for the main behaviors selected by our models, are summarized in Table 4; additional behaviors are contained in Appendix Table A1). Several intuitive patterns to the costs of manipulation can be seen in the top panel of Table 4. Outgoing communications are less costly to manipulate than incoming communications. Text messages, which are relatively cheap to send, are more manipulable than calls, which are relatively expensive. Simpler behaviors (such as the number of texts sent) are more manipulable than complex behaviors (such as the standard deviation of texts sent by day; see Appendix Table A1).

Costs are also heterogeneous across people, as shown in the bottom panel of Table 4. On average it is 9% easier for individuals who report advanced or higher tech skills to manipulate behaviors. Including unobserved heterogeneity, the 90th percentile of gaming ability finds it twice as easy to game as the 10th percentile. As noted, this spread in manipulation ability, and how it correlates with the outcome, is a primary reason that our method generates different decision rules.

Table 4: Estimated Manipulation Costs

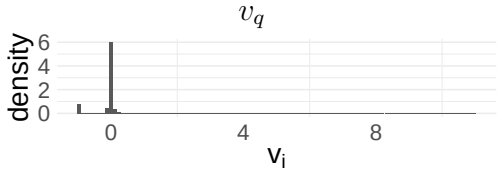
Heterogeneity by Behavior (**C** diagonal; subset of behaviors selected by models)



Heterogeneity by Person (γ_{iq})

$$\gamma_{iq} = e^{-\omega z_i} + v_q$$

Low tech skills	1.000
High tech skills	1.087



Parameters estimated using GMM. Top panel shows only behaviors used in models (• : naive LASSO, • : strategy-robust); for all behaviors see Appendix Table A1. In cost matrix, off-diagonal elements are regularized to zero ($\lambda^{costs}_{offdiagonal} \to \infty$), diagonal elements are regularized with $\lambda^{costs}_{diagonal} = 1.0$, set via cross validation. v_q plot omits top 5 percent of observations.

Estimating Decision Rules

Once the primitives are estimated, the strategy-robust decision rules can be constructed using the empirical analogue of equation (3):

$$\alpha^{SR}, \boldsymbol{\beta}^{SR} = \arg\min_{\alpha,\boldsymbol{\beta}} \left[\frac{1}{N} \sum_i \left[\frac{1}{Q} \sum_q \left[y_i - \alpha - \boldsymbol{\beta}'(\underline{\mathbf{x}}_i + C_{iq}^{-1}\boldsymbol{\beta}) \right]^2 + R_\lambda(\boldsymbol{\beta}, \mathbf{y}, \mathbf{C}_q) \right] \right]$$

which uses estimates of $\underline{\mathbf{x}}_i$ and Q random draws of the cost matrix C_{iq}, where draws $v_q \sim V$ are treated as random effects.[23] In order to keep decision rules simple and interpretable for our participants, we use LASSO regularization in $R_\lambda(\cdot)$ to restrict decision rules to have at most three predictors.[24] The distribution of unobserved gaming ability V is affected by a shrinkage parameter, which we calibrate based on performance on the first few weeks of decision rules (see Appendix A1).

3.4 Results: Naïve vs. Robust Decisions

The final and most important stage of the experiment compares decisions made by algorithms trained with the standard approach to those made by the strategy-robust approach that anticipates manipulation. As discussed above, the decision rules themselves are constructed using estimates of primitives obtained during the first few weeks of the experiment. We evaluate the performance of those decision rules using data collected during the latter stages of the experiment. Importantly, there is a clear line between these two stages; aside from the decision rule itself, no information collected during the first stage of the experiment (such as responses to the baseline survey, or any other individual-specific information) is considered by the decision rule

[23] We estimate $\underline{\mathbf{x}}_i$ as the simple average of $\boldsymbol{x}_i$ during control weeks (without week fixed effects). Due to the tight experimental timeline, the implemented decision rules were derived from preliminary estimates of $\mathbf{C}_i$. The main tables report the decision rules as assessed by final cost estimates; as shown in the Supplemental Online Appendix (Section S2.3) decision rules resulting from preliminary and final cost estimates are similar. The main analysis further omits select weeks when upload servers were offline and there was a mistake in computing the heterogeneity parameter; the Supplemental Online Appendix (Section S2.3) shows that our results are robust to their inclusion.

[24] Specifically, we regularized naïve LASSO decision rules with $\lambda = \max(\lambda^{cv}, \underline{\lambda}^{3var})$, where λ^{cv} is the cross-validated penalty parameter and $\underline{\lambda}^{3var}$ is the smallest that resulted in a 3-variable model. We set the regularization hyperparameter λ with cross validation in the unmanipulated baseline sample, and use the same λ to penalize our strategy-robust decision rule.

during the evaluation.[25]

Main Experimental Treatments

Participants were randomly assigned into different target outcomes (y), decision rules (standard $\boldsymbol{\beta}^{LASSO}$, or strategy-robust $\boldsymbol{\beta}^{SR}$), and whether the decision rule was opaque or transparent to the user. Under the opaque treatment, users were told only the target outcome and the reward (e.g., Figure 2b without the Hint). Under the transparent treatment, users saw the coefficients of the decision rule, which revealed how much they would be rewarded for each behavior, and an interactive earnings calculator (e.g., all of Figure 2b and c). Because the transparent treatment revealed information about potential decision rules, after a person had seen a transparent challenge for a given outcome, we did not assign them to an opaque challenge for the same outcome.

Table 5 provides suggestive evidence of how decision rules affect behavior. The first panel simply indicates the naïve estimated decision rule: high-income people make more outgoing calls, send fewer texts, and receive more texts. In the second panel, we see that if people are rewarded when they 'act like a high-income earner' but are not told the decision rule, the response is not statistically significant and often in the wrong direction on average (i.e., participants place fewer calls and send more texts). However, participants assigned the transparent treatment change their behavior broadly in the direction rewarded by the algorithm, though the response is measured with noise.

Performance of decision rules

Our main empirical results, shown in Table 6, compare the performance of naïve and strategy-robust decision rules. The first two columns (under 'Income') show results for the challenge that rewarded participants for using their phones like a high-income earner; the last two columns show the performance averaged across both the income challenge and an intelligence challenge (measured using Raven's matrices).

[25] One could be concerned that despite this, performance assessed in our final stage may be artificially high because it uses average costs estimated based on the responses to incentives of a sample that includes some of the same individuals. In a robustness test, we evaluate the performance of our decision rules on a sample that excludes individuals who received a decision rule during the first stage that provided incentives to change a behavior that was selected in the second state decision rule. Results are similar; see Supplemental Online Appendix Table S1, columns 3-4.

Table 5: Agents Game Algorithms

	Calls (outgoing)	Texts (outgoing)	Texts (incoming)	Calls w con-contacts (incoming + outgoing)	Avg call length (evening, seconds)
Panel A: **Incentives generated by algorithm (¢/action)**					
$\boldsymbol{\beta}^{LASSO}$	0.625	-0.395	0.065	0	0
Panel B: **Regression of $\mathbf{x}_{it}$ (column label) on treatment assignment (row label)**					
Opaque challenge	-4.7	12.5	11.1	0.8	-4.3
	(8.6)	(17.2)	(20.7)	(3.4)	(7.1)
Transparent challenge	13.7	-17.5	-6.5	0.3	-2.1
	(7.9)*	(15.7)	(19.0)	(3.1)	(6.5)
N (Person-weeks)	1651	1651	1651	1651	1651

Notes: Panel A reports the decision rule associated with the challenge, 'Earn up to 1000 Ksh. if the Sensing app guesses you are a high-income earner!'. Panel B reports how behaviors (indicated by columns) changed when participants were randomly assigned to the opaque challenge (which provided no information about the decision rule) or the transparent challenge (which revealed the details of the decision rule).The sample includes all people who were assigned the income challenge (either opaque, or the transparent LASSO model), in control weeks and the week they were assigned that challenge. Standard errors in parentheses. * $p < 0.1$.

The decision rules and associated manipulation costs are shown in the top panel ("Decision Rules"); the relative performance of the different decision rules is shown below (under "Prediction Error"). We note several results.

First, in Panel A, we observe important differences in the decision rules. LASSO places weight on the behaviors that were most correlated at baseline: outgoing calls, outgoing texts, and incoming texts. However, some of these behaviors, particularly text messaging, are quite manipulable (as shown in the 'Costs' column) and subject to manipulation noise. Although in general the adjustments made by the strategy-robust approach can be subtle depending on how gaming ability correlates with the outcome, here the decision rule attenuates or drops behaviors that are more manipulable (i.e., it drops incoming texts in favor of evening texts).

We evaluate predictive performance using root mean squared error (RMSE), in units of US dollars, in Panel B. This measures how far off the payments we gave to people (based on the model and their behavior that week) were from what we desired to give to them (based on their fixed characteristic that was targeted). The first pair

Table 6: Strategy-Robust vs. Standard Decision Rules

	Income		*Costs*	**Pooled: Income & Intelligence**	
Panel A: **Decision Rule**	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{SR}$	c_{jj}	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{SR}$
	¢/action		¢/action^2		
# Texts (outgoing)	-0.395	-0.107	0.035	.	.
# Texts (incoming)	0.065	0	0.037	.	.
# Texts (6pm-10pm)	0	-0.121	0.057	.	.
# Calls (outgoing)	0.625	0.542	0.480	.	.
Intercept (α)	301.071	304.622		.	.
Panel B: **Prediction Error**	RMSE ($)			RMSE ($)	
Baseline Data: Control	3.574	3.583		4.273	4.278
Baseline Data: Predicted Transparent	3.672	3.585		4.328	4.279
Implemented: Opaque	3.549	3.525		4.224	4.216
Implemented: Transparent	3.675	3.484		4.356	4.189
Average Payout ($)	3.34	3.25		4.21	4.18
N (Control Individuals)	1376	1376		1391	1391
N (Treatment Person-Weeks, Opaque)	75	75		156	156
N (Treatment Person-Weeks, Trans.)	90	74		166	154

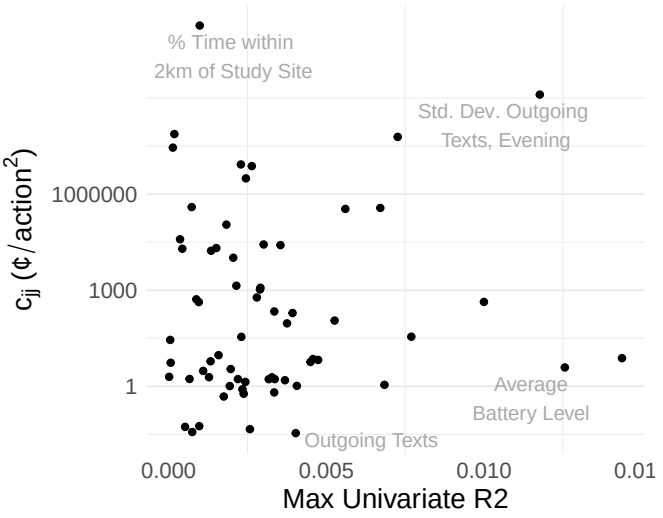
Notes: Panel A reports the decision rule associated with the challenge, and the costs associated with manipulating these behaviors. Panel B reports the performance of each decision rule by outcome, root mean squared error (RMSE) at the week-model level. Pooled metrics present the mean RMSE across models. Predicted Transparent represents the average expected performance of models given the theoretical model, behavior incentives, and estimated costs. Implemented Transparent/Opaque represents the average performance of models when assigned with/without transparency hints. Average payout represents the average payout to recipients based on model coefficients, given observed behavior. SR model estimated using preliminary costs estimates. Full results reported in Supplemental Online Appendix Table S1.

of rows report the prediction error that would be expected ex ante if behavior were the same as the control weeks. The first row shows that if there were no manipulation, LASSO would be expected to perform marginally better than our strategy-robust estimator (by \$0.01 for income; \$0.005 for income and intelligence pooled). The second row shows the error predicted by our model if the rule were made transparent and people were manipulating behavior: here, the strategy-robust method is expected to perform better (by \$0.09 for income; \$0.05 pooled).

The next pair of rows report the prediction error that we actually obtained when the decision rules were implemented experimentally. These may differ from the expected prediction error either if people respond differently than anticipated by our model, or because of noise from week to week. Here, we find that the strategy-robust (SR) method performs better than LASSO when participants are given full information about the decision rule (by \$0.19 / 5% for income; \$0.17 / 4% pooled). The strategy-robust method also performs slightly better when the decision rule is opaque (by \$0.02 / 0.6% for income; \$0.01 / 0.2% pooled) — possibly because of increased shrinkage relative to standard LASSO. Table A2 shows detailed results for both the income and intelligence outcomes, and the Supplemental Online Appendix shows that the performance improvements are even larger when all outcomes are considered: under full information, SR outperforms LASSO by 12%; under opacity, SR outperforms LASSO by 1% (see Supplementary Online Appendix Table S1).

Overall, the strategy-robust adjustment provides two benefits. When applied during training, it produces models that perform better when implemented and made transparent. And when applied during evaluation, it better anticipates the performance a model will achieve in the real world. Even if a policymaker intended to keep the decision rule opaque, using the strategy-robust method can reduce systematic risk in the chance that agents discover the decision rule. In practical implementations, policymakers could adaptively tweak the level of robustness to match the level of manipulation.

Figure 3: Manipulation Costs vs. Baseline Predictive Power



Each dot is a feature (i.e., a behavior recorded in the smartphone app). The x-axis indicates the highest R^2 across income and intelligence; the y-axis indicates the estimated manipulation cost. A subset of illustrative features are labeled in gray.

4 Discussion

4.1 Contrast to standard approaches

The standard approach to evaluating machine learning estimators evaluates each predictor based on its correlation with the outcome within a training dataset. However, as can be seen in Figure 3, features that appear equally predictive in a training dataset can have wildly different manipulability. The figure compares the average estimated cost of manipulation (y-axis) to the baseline predictive power (x-axis) of several dozen features from our experiment. We observe that some of the most predictive features (like the average battery level on the person's phone) are also very easy to manipulate: a standard model that selects those features could perform very poorly when people manipulate behavior.

We also compare our method to two common approaches to manipulation, simulating performance using our experimentally estimated model of behavior.

Contrast with the 'intuitive' approach

An alternate intuitive approach would be to train a standard estimator but simply omit behaviors that are most manipulable (e.g., by only considering features above some y-axis threshold on Figure 3). We assess this approach in the Supplemental Online Appendix (section S4.3). This 'intuitive' approach reduces the predicted manipulability of models, but – as suggested by Figure 3 – also removes from consideration useful predictors, in some cases by so much that it decreases the predicted performance. In extreme cases, regularized models such as LASSO can be left with no behaviors that are predictive enough to include in the regression. In contrast, our approach can extract signal even from manipulable behaviors, and performs better in these simulations.

Contrast with the iterative 'industry' approach

A second approach involves iteratively re-training a naïve machine learning estimator after people have responded to the previous decision rule. With both income and intelligence, we observe that the simulated performance of this method approaches the strategy-robust method after approximately 4 iterations of consumers being made aware of a new rule, adapting behavior, and then the policymaker retraining the algorithm (see Supplemental Online Appendix, Section S4.3). However, simulated performance of this iterative approach then begins to deteriorate. When predicting income this deterioration is small, but for intelligence, performance eventually falls below the performance obtained before any retraining.[26]

4.2 Performance cost of transparency

While society increasingly demands transparency in machine decisions, transparency can facilitate manipulation, which may reduce the quality of those decisions.

Our setting allows us to estimate this performance cost of transparency by comparing the performance of the naïve rule under opacity to the strategy-robust transparent rule. The latter approximates the optimal performance that can be attained under

[26]This is foreshadowed by the difference in moment conditions between the methods (equation (4)): even when trained on data from a strategy-robust equilibrium, standard methods may leave the committed optimal decision rule, because they do not anticipate that agents will respond.

transparency when resulting equilibrium manipulation is anticipated; the former approximates the optimal if opacity prevents all manipulation. Because the opaque rule also faces the threat of manipulation, this difference represents an upper bound of the true performance cost. Crucially, with our model, this quantity can be estimated without revealing the decision rule: it only requires the estimation of the primitives from the first part of our experiment.[27]

We estimate this cost of transparency in two ways: with our model and with our experiment, shown in the final rows of Panel B of Table 6. Our model predicts that transparency will reduce the performance of naive models by $\$4.328 - \$4.273 = \$0.055$ (1.2%) on average across income and intelligence, but that strategy-robust models will perform similarly whether transparent or opaque. These predictions are similar to the actual change in performance due to transparency that we find in our experiment: $\$4.356 - \$4.224 = \$0.132$ (3%) for naive models, and negligible for our strategy-robust models.[28] The income and intelligence outcomes had a lower cost of transparency, on average, than the other outcomes tested in our experiment; when we pool all outcomes together we find that transparency reduced performance of naive models by 17% and strategy-robust models by only 6%.

4.3 Alternate methods to estimate manipulation costs

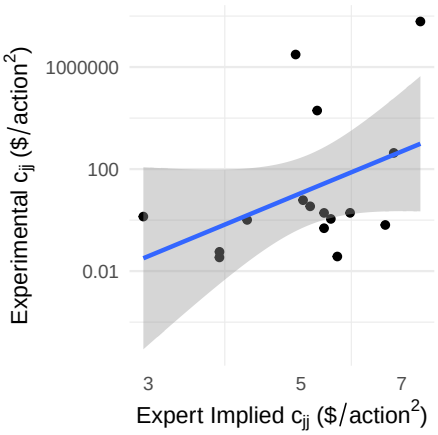
Estimating a strategy-robust decision rule requires beliefs about the costs of manipulating different behaviors. This paper demonstrates an experimental approach to eliciting those costs, but alternative approaches may be better suited to other settings:

Expert elicitations. We evaluate how well experts can predict the costs of manipulating different behaviors, in the spirit of DellaVigna and Pope (2016). We surveyed experts with different backgrounds (PhDs from different fields, research assistants, Busara staff who had not worked on the experiment, and Mechanical Turk workers in the U.S.) to predict how Kenyans would manipulate different phone behaviors when incentivized. We then infer the structural cost parameters implied by the predictions of the 171 respondents. Results are shown in Figure 4.

[27] Our method of estimating costs does requires revealing the existence of features to users, but does not require specifying whether those features are included in the model, or with what weights.

[28] Our results suggest that the cost of transparency is actually negative when the decision rule targets high-income individuals, which is theoretically possible.

Figure 4: Costs Elicited from Experts and Costs Measured in Experiment



Notes: Each dot represents a behavior captured by the Sensing App. Y-axis indicates the cost of manipulating that behavior, estimated through our experiment (Table 4). X-axis indicates costs elicited from expert surveys, inferred as $\hat{c}_{jj} = \frac{1}{N_{survey}} \sum_i \frac{\beta_j}{\max(0.001, \Delta_{jji})}$ for each i surveyed.

Although experts generally predict that costs are too low, the correlation is 0.30. If we use expert predictions of manipulation costs to train our model, and then assess predicted performance with the experimentally estimated model, even these noisy estimates improve simulated performance substantially for one outcome, and have an inconsequential negative effect on the other, as shown in Table A3. This suggests that expert elicitations show promise as a low-cost way to estimate manipulation costs. See Supplemental Online Appendix Section S3.

First principles/structural approach. In some cases, it may be possible to estimate the cost of underlying manipulations from market prices and first principles.[29] A structural model of costs would allow an implementer to account for changes in these underlying parameters, suggesting how manipulation will change if, for example, the price of calls changed, or a service emerged that made it easy to send automatic

[29]For example, the dark net price index (Gomez, 2020) reports the going price for online manipulations from an investigation on web forums: the average rate for 1,000 Instagram likes is $6; 1,000 Twitter retweets go for $25, suggesting they are more costly to manipulate. One can also cost out manipulation strategies: one can increase the number of non-contacts spoken with by randomly dialing 10 digit numbers and hanging up after the recipient picks up. That costs the call price of $0.04/minute plus the value of the time to dial a 10 digit number, divided by the fraction of such numbers that are valid and pick up, which can be valued at the going wage.

messages.

4.4 Social costs of manipulation

Our main specifications consider a narrow-minded policymaker who seeks only to maximize predictive accuracy, such that $M(\cdot) \equiv 0$ in equation (3). A socially-minded policymaker may also weigh the costs that agents incur manipulating behavior. Appendix Table A4 shows that as the loss function places more weight on the welfare costs that agents incur manipulating, our method adjusts models, typically towards even less manipulable behaviors.

4.5 Learning

While our main results compare performance when individuals have full knowledge or no knowledge of the decision rule, in many settings the individual will have noisy beliefs about how decisions are made. Here, we generalize to the case where individual i believes the prediction function will be $\tilde{\hat{y}}_i \sim G_i(\hat{y})$ when the actual decision rule is $\hat{y}(\cdot)$. Behavior follows the generalization of equation (1),

$$\mathbf{x}_i^*(\hat{y}(\cdot)) = \arg\max_{\mathbf{x}_i} \left[\mathbb{E}_{\tilde{\hat{y}}_i \sim G_i(\hat{y})} \left[v_i \left(\tilde{\hat{y}}_i(\mathbf{x}_i) \right) \right] - c(\mathbf{x}_i, \underline{\mathbf{x}}_i) \right]. \tag{5}$$

That is, agents balance the cost of manipulation against its *expected* utility gain.

Our main model is linear, which has no risk aversion ($v_i(\hat{y}) = \hat{y}$), so that uncertainty would not affect expected behavior. However, if individuals were risk averse ($\frac{\partial^2 v_i}{\partial y^2} < 0$), then uncertainty about $\hat{y}$ would reduce the incentive to manipulate (the first term). A designer could then reduce manipulation by either obfuscating or by introducing randomness into the decision rule. Although these approaches may be appropriate in some settings (as with the drunk driving checkpoints described in Banerjee et al. (2019)), they undermine a major goal of transparency: that people know how they are evaluated.

In settings where risk aversion and uncertainty are important, one could explicitly model these two objects with equation (5). Alternatively, the estimates of the linear model may represent reasonable local approximations of the distribution of beliefs and risk aversion in the sample. In particular, individuals often have difficulty

understanding the complex functional forms that arise from modern machine learning (Du et al., 2019; Poursabzi-Sangdeh et al., 2021), and commonly respond to linear heuristics when facing nonlinear functions (Liebman and Zeckhauser, 2004; Rees-Jones and Taubinsky, 2020). To make a decision rule robust to manipulation, it may be sufficient to make it robust to these heuristic responses. In that sense, our linear model may be viewed as an approximation of these beliefs.

5 Extension to Other Machine Learning Algorithms

This paper focuses on linear decision rules to sharpen intuition, but the core insight is also relevant in nonlinear settings. Here we show how it could apply to a decision tree, a class of model that can capture nonlinearities and high-dimensional interactions that are also common in other machine learning models.

5.1 Strategy-Robust Decision Tree

We derive a model that generates data well described by a tree, develop an estimation procedure, and then demonstrate in a simple example.

Model

Each individual i has some baseline behavior $\underline{\mathbf{x}}_i$, and a true binary classification

$$y_i = e_i \cdot (\underline{\mathbf{x}}_i \in R) + (1 - e_i) \cdot (\underline{\mathbf{x}}_i \notin R)$$

for some region R defined by interactions of $\underline{\mathbf{x}}_i$, and noise $e_i \sim Bernoulli(p)$. For simplicity, assume $v_i(\hat{y}) \equiv \hat{y}$.

We seek to classify individuals using some rule $\hat{y}(\mathbf{x}_i)$ where observed behavior $\mathbf{x}_i$ may be manipulated at cost $c_i(\mathbf{x}_i, \underline{\mathbf{x}}_i) = \sum_k c_{ik} 1\{x_{ik} \neq \underline{\mathrm{x}}_{ik}\}$. There is a fixed cost c_{ik} to change each behavior k at all; if that cost is incurred, x_{ik} can be changed to anything. Then observed behavior $\mathbf{x}_i^*(\hat{y}(\cdot))$ will be consistent with

$$x_{ik}^*(\hat{y}(\cdot)) = \begin{cases} \underline{\mathrm{x}}_{ik} & \text{if } \Delta_{ik} \leq c_{ik} \\ \arg\max_{x_{ik}} \hat{y}([\mathbf{x}_{i,-k}^*, x_{ik}]) & \text{if } \Delta_{ik} > c_{ik} \end{cases}$$

where $[\mathbf{x}^*_{i,-k}, x_{ik}]$ represents the vector given by $\mathbf{x}^*_i$ with the kth position replaced by x_{ik}, and the returns to manipulating dimension k are given by $\Delta_{ik} = \max_{x_{ik}}(\hat{y}([\mathbf{x}^*_{i,-k}, x_{ik}])) - \hat{y}([\mathbf{x}^*_{i,-k}, \underline{x}_{ik}])$.

Estimation

Given this model of behavior, how should predictions be formed? We demonstrate adding strategy-robustness to a textbook greedy tree algorithm (cf. Hastie et al., 2016, section 9.2). This algorithm recursively applies binary partitions to $\mathbf{x}$ to obtain a prediction rule of the form $\hat{y}(\mathbf{x}_i) = \sum_m y_m I(\mathbf{x}_i \in R_m)$, such that each region R_m has an associated prediction y_m.

To start, consider a candidate splitting variable k and split point s, and define the half-planes $R_1(k,s) = \{\mathbf{x}|x_{ik} \leq s\}$ and $R_2(k,s) = \{\mathbf{x}|x_{ik} > s\}$. Then we seek the splitting variable k and split point s that solve

$$\min_{k,s} \left[Q_m(R_1(k,s)) + Q_m(R_2(k,s))\right]$$

given a Gini index measure of error

$$Q_m(R) = \frac{2 \cdot ||\{y_i \in Y^*(R) : y_i = 1\}|| \cdot ||\{y_i \in Y^*(R) : y_i = 0\}||}{||Y^*(R)||^2}$$

where $Y^*(R) = \{y_i|\mathbf{x}^*_i(\hat{y}_{k,s}(\cdot)) \in R\}$ is the set of associated true labels for region R. $\hat{y}_{k,s}(\cdot)$ represents the decision boundary that would result from holding all earlier splits constant and adding a split at (k,s). Note that this anticipates that individual i may manipulate $\mathbf{x}_i$ to alter which region they are assigned to. In contrast, a naïve tree would be identical but would replace $Y^*(R)$ with $Y(R) = \{y_i|\mathbf{x}_i \in R\}$, i.e., it assumes behavior is fixed.

After the first split is determined, one can recursively apply this rule in each of the resulting regions, until a stopping condition applies. The associated prediction for each region is given by $y_m = \text{mode}(Y^*(R_m))$ (or $y_m = \text{mode}(Y(R_m))$ for the naïve tree).

Example

Figure 5 compares strategy-robust trees to standard decision trees, in examples with two features.

In Figure 5a, both features x_1 and x_2 appear to be predictive of the binary label in baseline data. But x_1 is manipulable at low cost and differentiates only a small part of the population.[30] If we train a standard decision tree, as shown in Figure 5a (i), it will split on both variables. However, given this tree, agents who would receive a negative outcome (i.e., the red circles) will manipulate x_1. This standard decision tree is not stable: everyone will be classified as positive. In contrast, in Figure 5a (ii) we see that after the strategy-robust tree splits on x_2, it anticipates that if it were to split on x_1, that behavior would be manipulated and would not be predictive. It results in a simpler tree. In particular, the original tree adds interactions that improves decisions for a small group of people, but makes the entire algorithm vulnerable to manipulation from a much larger part of the population. In this case, strategy-robustness attenuates the tree, reducing the number of feature interactions and thus the model variance.

In Figure 5b, we provide an example where *neither* x_1 nor x_2 appear to be very predictive of the binary label in the baseline data, but are differentially manipulable by different types. A naïve tree as shown in Figure 5b (i) will try to split on x_1, which will then be severely manipulated. It performs worse than a random guess. In contrast, in Figure 5b (ii) we see that the strategy-robust tree instead splits on behavior x_2: even though it is not informative in baseline data, when it is incentivized, the desired types will manipulate it to differentiate themselves. Thus, strategy-robustness does not attenuate the tree but rather expresses different variation, using manipulation as a signal of type.

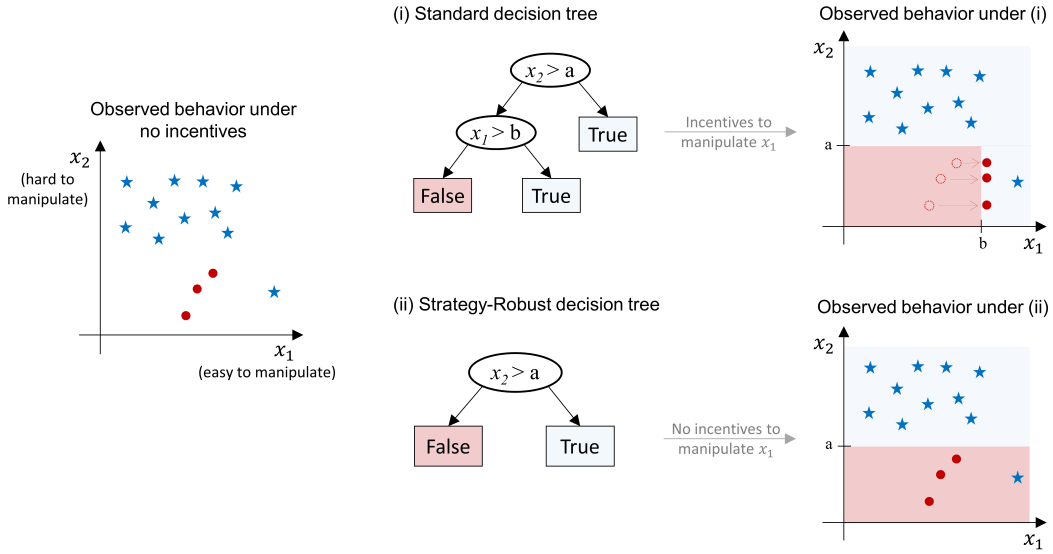
5.2 Discussion

Manipulated data generating processes

Manipulation can impact not only the parameters for a given function class, but also which function class is appropriate. A class of function that describes y well

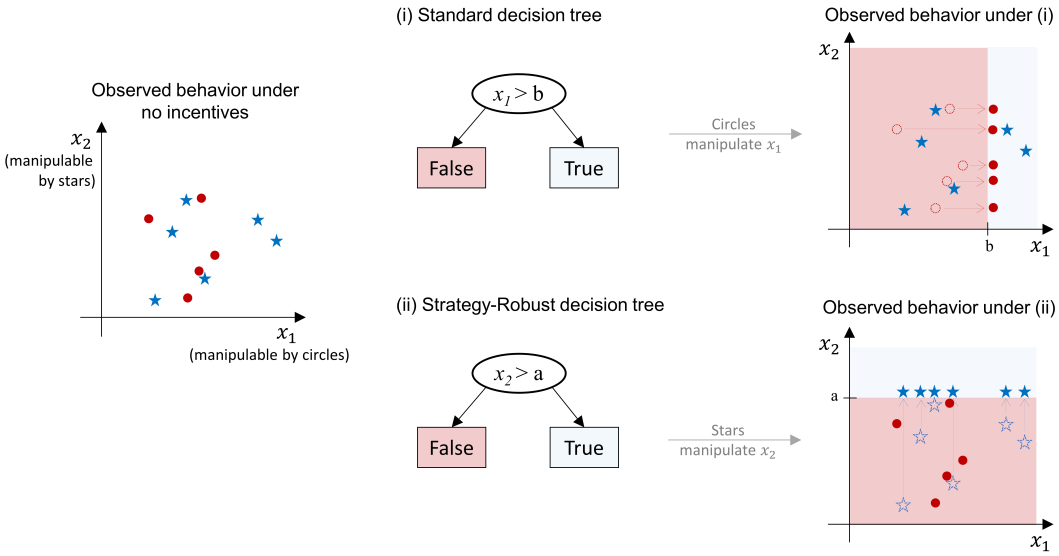
[30] For example, this setup could apply to a hiring problem like in Li et al. (2020) (where $\hat{y}$ is the decision to interview a candidate, and x_2 is hard evidence such as GPA and x_1 is soft evidence like including a keyword such as 'leadership' on a resume).

Figure 5: Standard vs. Strategy-Robust Decision Tree



(a) Example demonstrating attenuation

Notes: We seek a classification rule that will benefit the desired targets (denoted by stars). In baseline data, the first behavior appears to help classify a small part of the population, but it is easily manipulable ($c_{1i} \ll c_{2i}$). **(a)** A standard tree adds an interaction with x_1, and as a result incorrectly classifies the circles when the rule is implemented and their behavior is manipulated. **(b)** The strategy-robust tree anticipates that x_1 will be manipulated, and never adds the vulnerability to the model, resulting in better accuracy.



(b) Example demonstrating signaling

Notes: The first behavior (x_1) appears to help classify a small number of stars in baseline data, but is easily manipulable by the circles. The second behavior does not appear to be helpful in baseline data but is manipulable by stars. **(a)** A standard tree attempts to exploit the variation in x_1 in the baseline data, but as soon as the rule is implemented, the circles manipulate and only 18% of individuals are classified correctly. **(b)** Our method anticipates that circles will manipulate x_1, so does not add the vulnerability to the model. It also anticipates that stars will manipulate x_2, and exploits that to separate the types, resulting in 100% correct classification.

when there is no manipulation will continue to describe it well under manipulation only in special cases. In our linear model, since manipulation under quadratic costs induces linear shifts throughout the distribution, the manipulated data are still well described by a linear model. Likewise, our tree example admits fixed manipulation costs, so that manipulated data are still well described by a tree with adjusted cutoffs and levels. However, if manipulation costs were continuous or continuously heterogeneous, agents near the discontinuities of a tree would have strong incentives to manipulate behavior. In that case, a tree no longer would describe the data well, because optimal classification boundaries will no longer be sharp. This suggests that some of the extreme nonlinearities and discontinuities common in machine learning models may not be desirable in the presence of manipulation.[31] In general, to develop strategy-robust versions of other methods (e.g., random forests or neural nets), it may be necessary to either ensure that the functional form of manipulation cost does not shift the function class (as with our linear model and classification tree), or otherwise construct new variants that account for the shifts and smoothness arising from manipulation.[32]

The shape of manipulation costs

Experiments can reveal the shape of manipulation costs, and thus suggest appropriate functional forms. In the Supplemental Online Appendix (section S2.1), we use our experimental data to analyze how behavior responds to random variation in incentives. We find that linearity is a reasonable first approximation, though there is some evidence of diminishing returns to manipulation, and less response for negative incentives. In general, manipulation costs might include fixed costs, asymmetries (e.g., for some behaviors, increases are differentially costly than decreases), and dynamic elements (such as seasonality).[33] In cases where manipulation costs are changing, one could

[31] The mechanism design literature has noted that linear decision rules can be more robust (Holmstrom and Milgrom, 1987; Carroll, 2015).

[32] Nonlinear environments may also have many more equilibria. In such settings, if iterative learning converges, it may converge to an undesirable equilibrium, whereas an approach like ours could be used to select a global optimum.

[33] As suggested by Ball (2019), there may also be particular features that have more heterogeneity in cost between individuals. We treat these two dimensions of heterogeneity as independent. If our approach were extended to allow for this interdependence, it would down-weight indicators that have a particular spread in manipulability.

either view static estimates as an approximation, or model the process by which they change (see Section 4.3), such as might arise from changes in prices or social learning.[34]

6 Conclusion

This paper considers the possibility that machine decisions change the world in which they are deployed. We focus on the case where individuals manipulate their behavior in order to game decision rules. We derive decision rules that anticipate this manipulation, by embedding a behavioral model of how individuals will respond. This structural approach makes it possible to decompose decision rules into constituent components, and to gather data on how those components can be manipulated. From these components, our structural model allows us to understand how *any* proposed decision rule of a given form would be manipulated. This allows us to compute decision rules that are optimal in equilibrium.

We demonstrate our method in a field experiment in Kenya by deploying a tailor-made smartphone app that mimics the 'digital credit' loan products that are now commonplace in sub-Saharan Africa. We find that even some of the world's poorest users of technology – who are relatively recent adopters of smartphones and to whom whom the concept of an 'algorithm' is quite foreign (Musya and Kamau, 2018) – are savvy enough to change their behavior to game machine decisions. In this setting, we show that our strategy-robust approach outperforms standard estimators on average by 12% when individuals are given information about the scoring rule. This framework also allows us to quantify the "cost of transparency", i.e., the loss in predictive performance associated with moving from "security through obscurity" (with a naïve decision rule) to a regime of full algorithmic transparency (with our strategy-robust rule). We estimate this loss to be roughly 6% in equilibrium — substantially less than the 17% loss associated with making the naïve rule transparent.

[34] Our experiment was designed specifically to limit organic social learning. For instance, intake sessions were designed to educate users about how the Sensing app's decisions were made, so that most 'learning' would occur prior to the experimental assignment of challenges. Subsequently, individuals were randomly assigned different outcomes and decision rules on different weeks, to reduce the potential gains from communication. In practice, we observe very little communication between participants: Analyzing data collected by the app, we find that only 3% of phone-based communication was between study participants.

We focus on the simple case of linear models with a small number of predictor variables, where subjects have either no information or full information about the decision rule. We envision useful extensions to more complex models and more nuanced beliefs.

More generally, our approach that combines machine learning estimators with models of human behavior may be relevant to a wide range of contexts where machine learning systems face changing human environments. This structural approach is different from the prevailing approach to machine learning, which relies on large amounts of data and flexible functions that impose few assumptions about how the data are generated. A deep problem with the status quo approach is that these methods often perform much better in the lab than they do when implemented (cf. Lazer et al., 2014; Andrews et al., 2023). We study a particular implementation issue — strategic manipulation. Because the distribution of data changes upon implementation, the most naïve, fully unstructured approach would require implementing every possible decision rule $\hat{y}(\cdot)$ to evaluate performance on its resulting data $\mathbf{x}_i(\hat{y}(\cdot))$ in order to find a global optimum. We make the observation that the counterfactual world that occurs after implementing $\hat{y}(\cdot)$ has a predictable structure: including a particular variable in a model tends to induce manipulation and spread in that variable, of a magnitude related to its costs and benefits. These benefits can be inferred for free, because they're a direct function of the estimand itself: $\hat{y}(\cdot)$. This structure makes it possible to predict counterfactual fit, and more efficiently identify the models that will perform well when implemented. In simulations and in our experiment, we show that using this structure can improve reliability.

In this sense, our paper offers a machine learning interpretation of Lucas (1976), where algorithmic decisions change the context of the systems they model. In settings like ours, $\boldsymbol{\beta}$ determines not just predictive performance within a given world, but also which counterfactual world occurs.

References

Agarwal, Nikhil and Eric Budish, "Market Design," *Handbook of Industrial Organization*, 2021.

Aiken, Emily, Suzanne Bellue, Dean Karlan, Christopher Udry, and Joshua E Blumenstock, "Machine Learning and Mobile Phone Data Can Improve the Targeting of Humanitarian Assistance," *Working Paper*, July 2021.

Akerlof, George A., "The economics of "tagging" as applied to the optimal income tax, welfare programs, and manpower planning," *The American economic review*, 1978, *68* (1), 8–19.

Andrews, Isaiah, Drew Fudenberg, Lihua Lei, Annie Liang, and Chaofeng Wu, "The Transfer Performance of Economic Models," 2023.

Ball, Ian, "Scoring Strategic Agents," *arXiv:1909.01888 [econ]*, November 2019. arXiv: 1909.01888.

Banerjee, Abhijit, Esther Duflo, Daniel Keniston, and Nina Singh, "The Efficient Deployment of Police Resources: Theory and New Evidence from a Randomized Drunk Driving Crackdown in India," Working Paper 26224, National Bureau of Economic Research September 2019. Series: Working Paper Series.

_ **, Rema Hanna, Benjamin A Olken, and Sudarno Sumarto**, "The (lack of) Distortionary Effects of Proxy-Means Tests: Results from a Nationwide Experiment in Indonesia," Working Paper 25362, National Bureau of Economic Research December 2018.

Barocas, Solon, Moritz Hardt, and Arvind Narayanan, *Fairness and Machine Learning*, fairmlbook.org, 2018.

Bharadwaj, Prashant and Tavneet Suri, "Improving Financial Inclusion through Digital Savings and Credit," *AEA Papers and Proceedings*, May 2020, *110*, 584–588.

Björkegren, Daniel, "'Big data' for development," 2010.

_ **and Darrell Grissen**, "Behavior Revealed in Mobile Phone Usage Predicts Credit Repayment," *The World Bank Economic Review*, October 2020, *34* (3), 618–634.

Björkegren, Daniel and Darrell Grissen, "Behavior Revealed in Mobile Phone Usage Predicts Loan Repayment," *Available at SSRN 2611775*, 2015.

Bloomberg, "Phone Stats Unlock a Million Loans a Month for Africa Lender," *Bloomberg.com*, September 2015.

Blumenstock, Joshua E., "Estimating Economic Characteristics with Phone Data," *AEA Papers and Proceedings*, 2018, *108*, 72–76.

Blumenstock, Joshua Evan, Dan Gillick, and Nathan Eagle, "Who's Calling? Demographics of Mobile Phone Use in Rwanda," in "2010 AAAI Spring Symposium Series" March 2010.

_ , **Gabriel Cadamuro, and Robert On**, "Predicting poverty and wealth from mobile phone metadata," *Science*, November 2015, *350* (6264), 1073–1076.

Borrell Associates, "Trends in Digital Marketing Services," 2016.

Breiman, Leo, "Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)," *Statistical Science*, August 2001, *16* (3), 199–231. Publisher: Institute of Mathematical Statistics.

Bruckner, Michael and Tobias Scheffer, "Stackelberg Games for Adversarial Prediction Problems," in "Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining" KDD '11 ACM New York, NY, USA 2011, pp. 547–555.

Bryan, Gharad, Dean Karlan, and Jonathan Zinman, "Referrals: Peer Screening and Enforcement in a Consumer Credit Field Experiment," *American Economic Journal: Microeconomics*, August 2015, *7* (3), 174–204.

Camacho, Adriana and Emily Conover, "Manipulation of Social Program Eligibility," *American Economic Journal: Economic Policy*, May 2011, *3* (2), 41–65.

Caprino, Kathy, "How To Write A Resume That Passes The Artificial Intelligence Test," 2019. Section: Careers.

Carroll, Gabriel, "Robustness and Linear Contracts," *American Economic Review*, February 2015, *105* (2), 536–563.

CGAP, "Kenya's Digital Credit Revolution Five Years On," *CGAP*, March 2018.

Crosman, Penny, "How fraudsters are gaming online lenders," *American Banker*, March 2017.

Dee, Thomas S., Will Dobbie, Brian A. Jacob, and Jonah Rockoff, "The Causes and Consequences of Test Score Manipulation: Evidence from the New York Regents Examinations," *American Economic Journal: Applied Economics*, July 2019, *11* (3), 382–423.

DellaVigna, Stefano and Devin Pope, "Predicting Experimental Results: Who Knows What?," Working Paper 22566, National Bureau of Economic Research August 2016.

Dong, Jinshuo, Aaron Roth, Zachary Schutzman, Bo Waggoner, and Zhiwei Steven Wu, "Strategic Classification from Revealed Preferences," in "Proceedings of the 2018 ACM Conference on Economics and Computation" EC '18 ACM New York, NY, USA 2018, pp. 55–70.

Dranove, David, Daniel Kessler, Mark McClellan, and Mark Satterthwaite, "Is More Information Better? The Effects of "Report Cards" on Health Care Providers," *Journal of Political Economy*, June 2003, *111* (3), 555–588.

Du, Mengnan, Ninghao Liu, and Xia Hu, "Techniques for interpretable machine learning," *Communications of the ACM*, December 2019, *63* (1), 68–77.

Eliaz, Kfir and Ran Spiegler, "The Model Selection Curse," *American Economic Review: Insights*, September 2019, *1* (2), 127–140.

European Union, "EU General Data Protection Regulation (GDPR)," 2016.

Francis, Eilin, Joshua Blumenstock, and Jonathan Robinson, "Digital Credit: A Snapshot of the Current Landscape and Open Research Questions," *CEGA White Paper*, 2017.

Frankel, Alex and Navin Kartik, "Muddled Information," *Journal of Political Economy*, August 2019, *127* (4), 1739–1776.

_ and _ , "Improving Information from Manipulable Data," *arXiv:1908.10330 [econ]*, April 2020. arXiv: 1908.10330.

FSD Kenya, "Tech-enabled lending in Africa," 2018.

Gomez, Miguel, "Dark Web Price Index," 2020. Section: SECURITY.

Gonzalez-Lira, Andres and Ahmed Mobarak, "Slippery Fish: Enforcing Regulation under Subversive Adaptation," IZA Discussion Paper 12179, Institute of Labor Economics (IZA) February 2019.

Goodhart, Charles, *Monetary Relationships: A View from Threadneedle Street*, University of Warwick, 1975. Google-Books-ID: GKwJMwEACAAJ.

Goodman, Bryce and Seth Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *arXiv:1606.08813 [cs, stat]*, June 2016. arXiv: 1606.08813.

Greenstone, Michael, Guojun He, Ruixue Jia, and Tong Liu, "Can Technology Solve the Principal-Agent Problem? Evidence from Pollution Monitoring in China," 2019.

Hardt, Moritz, Nimrod Megiddo, Christos Papadimitriou, and Mary Wootters, "Strategic Classification," in "Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science" ITCS '16 ACM New York, NY, USA 2016, pp. 111–122.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd edition ed., New York, NY: Springer, January 2016.

Hennessy, Christopher A. and Charles A. E. Goodhart, "Goodhart's Law and Machine Learning: A Structural Perspective," *International Economic Review*, 2023, *n/a* (n/a). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/iere.12633.

Holmstrom, Bengt and Paul Milgrom, "Aggregation and Linearity in the Provision of Intertemporal Incentives," *Econometrica*, 1987, *55* (2), 303–328.

Huang, Ling, Anthony D. Joseph, Blaine Nelson, Benjamin IP Rubinstein, and J. D. Tygar, "Adversarial machine learning," in "Proceedings of the 4th ACM workshop on Security and artificial intelligence" ACM 2011, pp. 43–58.

Hussam, Reshmaan, Natalia Rigol, and Benjamin N. Roth, "Targeting High Ability Entrepreneurs Using Community Information: Mechanism Design in the Field," November 2017.

Kleinberg, Jon and Manish Raghavan, "How Do Classifiers Induce Agents to Invest Effort Strategically?," in "Proceedings of the 2019 ACM Conference on Economics and Computation" EC '19 ACM New York, NY, USA 2019, pp. 825–844. event-place: Phoenix, AZ, USA.

Lazer, David, Ryan Kennedy, Gary King, and Alessandro Vespignani, "The Parable of Google Flu: Traps in Big Data Analysis," *Science*, March 2014, *343* (6176), 1203–1205.

Li, Danielle, Lindsey R. Raymond, and Peter Bergman, "Hiring as exploration," Technical Report, National Bureau of Economic Research 2020.

Liebman, Jeffrey and Richard J. Zeckhauser, "Schmeduling," 2004.

Lucas, Robert E., "Econometric policy evaluation: A critique," *Carnegie-Rochester Conference Series on Public Policy*, January 1976, *1* (Supplement C), 19–46.

McCaffrey, Mike, Olivia Obiero, and George Mugweru, "M-Shwari: Market Reactions and Potential Improvements," Technical Report 139 2013.

Milli, Smitha, John Miller, Anca D. Dragan, and Moritz Hardt, "The Social Cost of Strategic Classification," in "Proceedings of the Conference on Fairness, Accountability, and Transparency" FAT* '19 ACM New York, NY, USA 2019, pp. 230–239. event-place: Atlanta, GA, USA.

Mirrlees, J. A., "An Exploration in the Theory of Optimum Income Taxation," *The Review of Economic Studies*, 1971, *38* (2), 175–208.

Mukherjee, Anit Nath, Bermeo Rojas,Laura Ximena, Okamura,Yuko, Muhindo,Jimmy Vulembera, and Paul G. A. Bance, "Digital-first Approach to Emergency Cash Transfers: Step-kin in the Democratic Republic of Congo (English)," *World Bank*, March 2023.

Musya, Mercy and Grace Kamau, "How do you say "algorithm" in Kiswahili?," December 2018. Library Catalog: medium.com.

National Institute of Standards and Technology, "Guide to General Server Security," *NIST Special Publication*, July 2008, (800-123).

Nichols, Albert L. and Richard J. Zeckhauser, "Targeting Transfers through Restrictions on Recipients," *The American Economic Review*, 1982, *72* (2), 372–377.

OSTP, "Blueprint for an AI Bill of Rights," Technical Report 2022.

Perdomo, Juan C., Tijana Zrnic, Celestine Mendler-Dünner, and Moritz Hardt, "Performative Prediction," *arXiv:2002.06673 [cs, stat]*, June 2020. arXiv: 2002.06673.

Poursabzi-Sangdeh, Forough, Daniel G Goldstein, Jake M Hofman, Jennifer Wortman Wortman Vaughan, and Hanna Wallach, "Manipulating and Measuring Model Interpretability," in "Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems" CHI '21 Association for Computing Machinery New York, NY, USA May 2021, pp. 1–52.

Ramsey, F. P., "A Contribution to the Theory of Taxation," *The Economic Journal*, 1927, *37* (145), 47–61.

Rees-Jones, Alex and Dmitry Taubinsky, "Measuring "Schmeduling"," *The Review of Economic Studies*, October 2020, *87* (5), 2399–2438.

Sayed-Mouchaweh, Moamar and Edwin Lughofer, *Learning in Non-Stationary Environments: Methods and Applications*, Springer Science & Business Media, April 2012. Google-Books-ID: qFWM2nva7xQC.

Spence, Michael, "Job Market Signaling," *The Quarterly Journal of Economics*, 1973, *87* (3), 355–374.

Sundsøy, Pål, Johannes Bjelland, Bjørn-Atle Reme, Eaman Jahani, Erik Wetter, and Linus Bengtsson, "Estimating individual employment status using mobile phone network data," *arXiv:1612.03870 [cs]*, December 2016. arXiv: 1612.03870.

Suri, Tavneet, Prashant Bharadwaj, and William Jack, "Fintech and household resilience to shocks: Evidence from digital loans in Kenya," *Journal of Development Economics*, November 2021, *153*, 102697.

Appendices

A1 Estimation Details

In our experiment, each individual i was randomly assigned a decision rule in time period t, which provided rewards based on their behavior: $\hat{y}_{it}(\mathbf{x}_{it}) = \alpha_{it} + \boldsymbol{\beta}'_{it}\mathbf{x}_{it}$. Their resulting behavior, $\mathbf{x}^*_{it}(\boldsymbol{\beta}_{it}) = \underline{\mathbf{x}}_i + C_i^{-1}\boldsymbol{\beta}_{it} + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it}$, could have deviated from the bliss level ($\underline{\mathbf{x}}_i$) due to manipulation, or shocks that were common ($\boldsymbol{\mu}_t$) or individual-specific ($\boldsymbol{\epsilon}_{it}$).[35] Shocks are mean zero: $\mathbb{E}\boldsymbol{\mu}_t = \mathbf{0}$ and $\mathbb{E}\boldsymbol{\epsilon}_{it} = \mathbf{0}$.

We use the control and simple challenges to estimate types $\underline{\mathbf{x}}$, cost parameters related to C_{iq}, and the distribution of unobserved gaming ability V. To estimate types, we use an ordinary least squares regression,

$$\mathbf{x}_{it} = \underline{\mathbf{x}}_i + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it}, \tag{6}$$

including only time periods where $\boldsymbol{\beta} = \mathbf{0}$ (in which people act as their types). We include week fixed effects ($\boldsymbol{\mu}$) to improve precision.

When estimating manipulation costs, we impose $\underline{\mathbf{x}}$ and $\boldsymbol{\mu}$ from above. We limit the potential to overfit by constraining behaviors to move in the direction they are incentivized ($c_{jj} > 0$).[36] We recover the distribution of unobserved gaming ability V by shrinking and then shuffling the gaming ability residuals.

Moment Conditions

The following moment conditions jointly identify C and $\boldsymbol{\omega}$.

Implemented decision rules are orthogonal to idiosyncratic behavior shocks ($\mathbb{E}[\boldsymbol{\beta}_{itk}\epsilon_{itj}] =$

[35] This arises from the utility function $u_{it} = \hat{y}_{it}(\mathbf{x}_{it}) - c_i(\mathbf{x}_{it}, \underline{\mathbf{x}}_i) + (\boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it})'C_i(\mathbf{x}_{it} - \underline{\mathbf{x}}_i)$.

[36] We use LASSO penalization on the ease of manipulation (penalizing costs to infinity), allowing separate hyperparameters for diagonal and off-diagonal costs ($\boldsymbol{\lambda}^{costs} = \{\lambda^{costs}_{diagonal}, \lambda^{costs}_{offdiagonal}\}$). We use three-fold cross validation to select $\lambda^{costs}_{diagonal}$and let $\lambda^{costs}_{offdiagonal} \to \infty$.

0). For each pair of behaviors jk (including $j = k$) this yields sample moment condition

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\mathbb{T}_i}\beta_{itk}\left[x_{ijt} - \underline{\mathrm{x}}_{ij} - \mu_{jt} - e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_j\right] = 0 \tag{7}$$

where $[\mathbf{a}]_k$ indicates the kth element of $\mathbf{a}$.

Implied unobserved heterogeneity $\tilde{v}_i$ is given by

$$\tilde{v}_i = \frac{1}{\sum_{t\in\mathbb{T}_i^{treatment}}|K_{it}^{eval}|}\sum_{t\in\mathbb{T}_i^{treatment}}\sum_{k\in K_{it}^{eval}}\left[\frac{x_{ikt} - \underline{\mathrm{x}}_{ik} - \mu_{kt}}{[C^{-1}\boldsymbol{\beta}_{it}]_k} - e^{-\boldsymbol{\omega}'\mathbf{z}_i}\right], \tag{8}$$

where K_{it}^{eval} is the set of behaviors to be evaluated for i in period t.[37] Unobserved heterogeneity is mean zero, yielding moment condition, $\frac{1}{N}\sum_i \tilde{v}_i = 0$, and orthogonal to each heterogeneity characteristic z_l, yielding moment condition(s) $\frac{1}{N}\sum_i z_{li}\cdot\tilde{v}_i = 0$.

Additional Moment Conditions for Brownfield Case

In greenfield settings where the base model $\boldsymbol{\beta}_0 = \mathbf{0}$, during training it is possible to infer individual types directly from baseline data (equation (6)), prior to estimating costs. Our method can also be applied in *brownfield* settings, where a decision rule has already been implemented and baseline behavior is already manipulated.

In such a setting, one would want to append the following moment conditions to estimate $\underline{\mathbf{x}}$ and $\boldsymbol{\mu}$. Both are based on the condition $\mathbb{E}[\epsilon_{itk}] = 0$. For each individual i and behavior k, we have

$$\underline{\mathrm{x}}_{ik} = \frac{1}{|\mathbb{T}_i|}\sum_{t\in\mathbb{T}_i}\left[x_{ikt} - \mu_{kt} - e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_k\right] \tag{9}$$

For each time period t and behavior k we have

$$\mu_{kt} = \frac{1}{|\{i|\mathbb{T}_i \ni t\}|}\sum_{i|\mathbb{T}_i\ni t}\left[x_{ikt} - \underline{\mathrm{x}}_{ik} - e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_k\right] \tag{10}$$

Identification still requires observing random variation along each behavior in the

[37] We set $K_{it}^{eval} = \{k \text{ s.t. } \beta_{itk} \neq 0\}$, so that $\tilde{v}_i$ is evaluated only off shifts in the incentivized behavior. One could alternately evaluate how each incentive shifts all behaviors.

decision rule (and ensuing manipulation).[38]

Manipulation Cost Regularization

We add to our GMM loss function the regularization term:

$$R_{costs}^{\boldsymbol{\lambda}^{costs}}(\cdot) = \left[\lambda_{diagonal}^{costs}\sum_{k}\theta_{kk} + \lambda_{offdiagonal}^{costs}\sum_{j\neq k}\theta_{jk}\right]\left[\frac{1}{N}\sum_{i}e^{-2\boldsymbol{\omega}'\mathbf{z}_i}\right]$$

where θ_{jk} represents the elements of inverse costs C^{-1}.

Unobserved Gaming Ability

We recover the distribution of unobserved gaming ability V in two steps. We compute gaming ability residuals $\tilde{v}_i$ as in equation (8), which capture whether each individual manipulates more or less than predicted during incentivized periods. Then, to reduce the impact of noise and outliers, we shrink and winsorize these inferred shocks. We form the empirical distribution $V = \{\max(\phi \cdot \tilde{v}_i, \underline{v})\}_i$, where $\underline{v}$ is the lowest value of $\tilde{v}$ that leads to a nonnegative implied gaming ability, and ϕ is a shrinkage parameter calibrated to minimize overall error in observed incentivized periods (that is, $\underline{v} = \min_i(\tilde{v}_i | \phi \cdot \tilde{v}_i \geq -\min_j(e^{-\boldsymbol{\omega}'\mathbf{z}_j}))$).

We calibrated ϕ to 1e-6; For details, see Supplemental Online Appendix S2.3.2.

One-Shot Estimation

Our experiment observes each individual over multiple time periods, which allowed us additional statistical power given our limited budget. Our approach can also be applied if each individual i is observed only over one period t.

C and $\boldsymbol{\omega}$ can be recovered by adjusting the brownfield moment conditions to remove individual and time fixed effects. This entails replacing the moment condition

[38] This inversion will be more sensitive to the specification of the model than when unincentivized behavior can be observed directly in training. Because there are limits on how much you can change a model that is in production and still maintain good performance, the brownfield approach may require more data to obtain precise estimates.

in equation (7) with

$$\frac{1}{N}\sum_{i=1}^{N}\beta_{itk}\left[x_{ijt}-\chi_j-e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_j\right]=0,$$

Equation (8) with

$$\tilde{v}_i=\frac{x_{ik_it}-\chi_{k_i}}{[C^{-1}\boldsymbol{\beta}_{it}]_{k_i}}-e^{-\boldsymbol{\omega}'\mathbf{z}_i},$$

Equation (9) with

$$\chi_k=\frac{1}{N}\sum_{i=1}^{N}\left[x_{ikt}-e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_k\right],$$

and dropping equation (10), where individual types are replaced with a term representing common behavior $\boldsymbol{\chi}$ of dimension K, and where k_i is the behavior incentivized for individual i in week t.

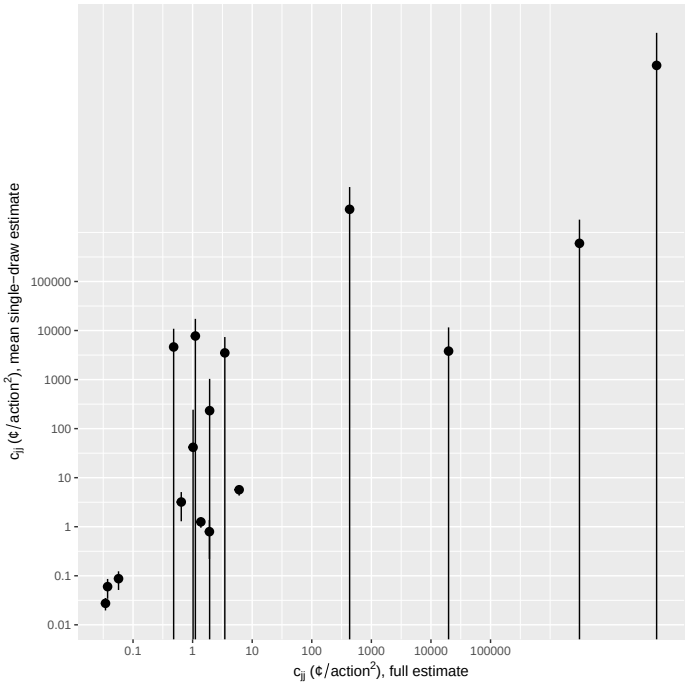
An estimate of each person's bliss behavior can then be obtained by undoing any predicted manipulation:

$$\underline{\mathrm{x}}_{ik}=x_{ikt}-e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_k$$

though with just one observation this will be more affected by idiosyncratic noise.

We demonstrate that this approach obtains similar results by mimicking the data that would result if our experiment had observed each individual for only one period. Because this will drastically reduce our sample size, we simulate this over multiple replication draws. For replication draw r, for each individual i we restrict the sample to include only one randomly selected incentivized week $t_{ir}\in\mathbb{T}_i^{treatment}$, and consider the average over replications $r\in\{1...R\}$. Figure A1 shows that the one shot estimates are similar to the full sample estimates (we report the average $\hat{c}^{-1}_{k_{ir}k_{ir}}=\frac{1}{R}\sum_r\frac{1}{\hat{c}_{k_{ir}k_{ir}}}$); the Pearson correlation coefficient between the two measures is 0.9987. The corresponding estimate for $\hat{\boldsymbol{\omega}}=\frac{1}{R}\sum_r\hat{\boldsymbol{\omega}}_r=0.22$ (standard deviation 0.98), with roughly 30% of single-draw estimates below or equal to the full-sample estimate of -0.083 and roughly 70% of estimates above.

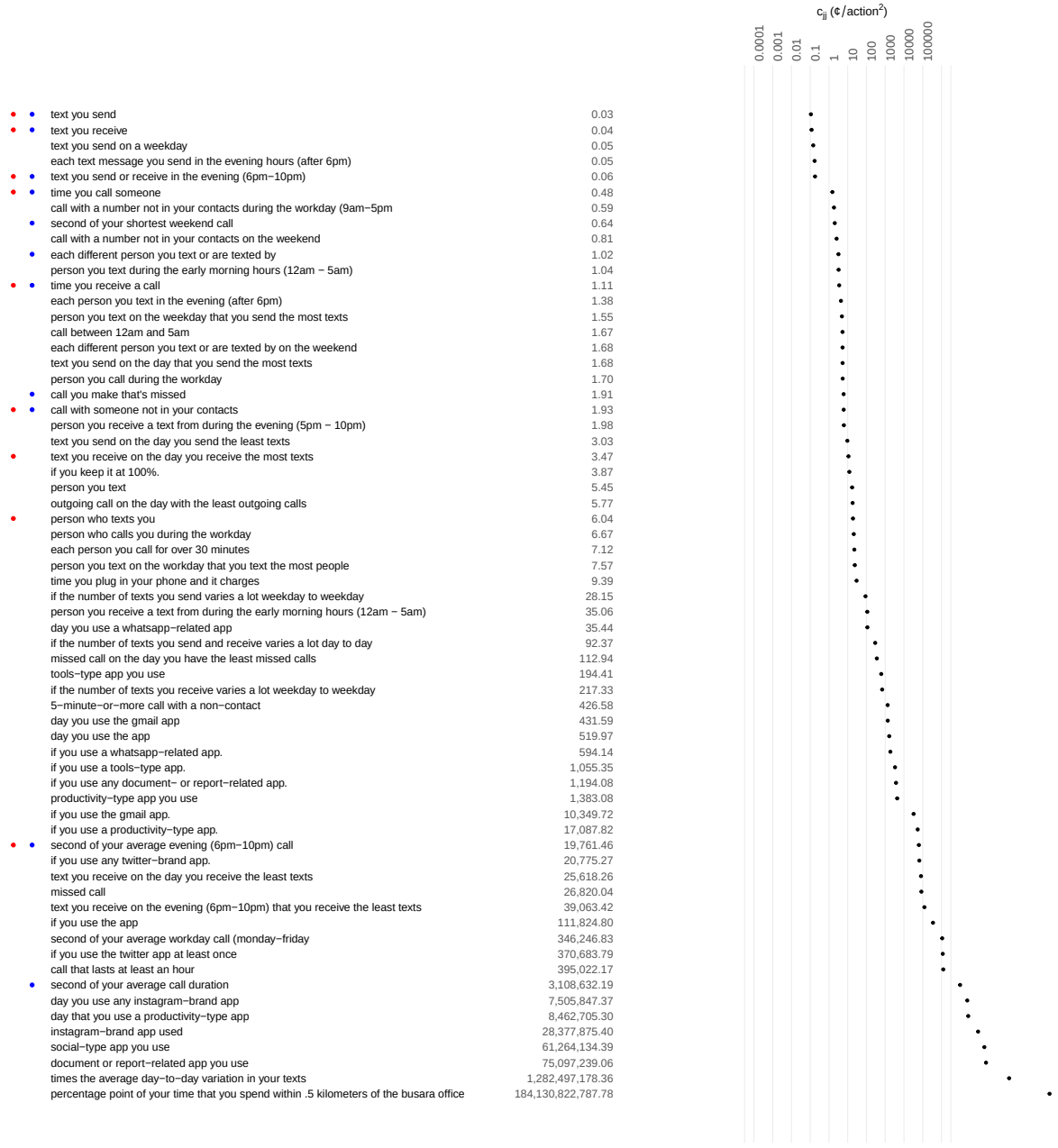
Figure A1: Manipulation Costs Estimated with only One Observation per Person



Notes: Our main estimates (with multiple observations per person) are shown on the x-axis. The average estimate obtained when each individual is observed only once is shown on the y-axis; the standard deviation across replications is shown as a whisker in either direction.

Table A1: Estimated Manipulation Costs for All Behaviors

Heterogeneity by Behavior (**C** diagonal; all incentivized behaviors)



•	text you send	0.03
•	text you receive	0.04
	text you send on a weekday	0.05
	each text message you send in the evening hours (after 6pm)	0.05
•	text you send or receive in the evening (6pm−10pm)	0.06
•	time you call someone	0.48
	call with a number not in your contacts during the workday (9am−5pm	0.59
•	second of your shortest weekend call	0.64
	call with a number not in your contacts on the weekend	0.81
•	each different person you text or are texted by	1.02
	person you text during the early morning hours (12am − 5am)	1.04
•	time you receive a call	1.11
	each person you text in the evening (after 6pm)	1.38
	person you text on the weekday that you send the most texts	1.55
	call between 12am and 5am	1.67
	each different person you text or are texted by on the weekend	1.68
	text you send on the day that you send the most texts	1.68
	person you call during the workday	1.70
•	call you make that's missed	1.91
•	call with someone not in your contacts	1.93
	person you receive a text from during the evening (5pm − 10pm)	1.98
	text you send on the day you send the least texts	3.03
•	text you receive on the day you receive the most texts	3.47
	if you keep it at 100%.	3.87
	person you text	5.45
	outgoing call on the day with the least outgoing calls	5.77
•	person who texts you	6.04
	person who calls you during the workday	6.67
	each person you call for over 30 minutes	7.12
	person you text on the workday that you text the most people	7.57
	time you plug in your phone and it charges	9.39
	if the number of texts you send varies a lot weekday to weekday	28.15
	person you receive a text from during the early morning hours (12am − 5am)	35.06
	day you use a whatsapp−related app	35.44
	if the number of texts you send and receive varies a lot day to day	92.37
	missed call on the day you have the least missed calls	112.94
	tools−type app you use	194.41
	if the number of texts you receive varies a lot weekday to weekday	217.33
	5−minute−or−more call with a non−contact	426.58
	day you use the gmail app	431.59
	day you use the app	519.97
	if you use a whatsapp−related app.	594.14
	if you use a tools−type app.	1,055.35
	if you use any document− or report−related app.	1,194.08
	productivity−type app you use	1,383.08
	if you use the gmail app.	10,349.72
	if you use a productivity−type app.	17,087.82
•	second of your average evening (6pm−10pm) call	19,761.46
	if you use any twitter−brand app.	20,775.27
	text you receive on the day you receive the least texts	25,618.26
	missed call	26,820.04
	text you receive on the evening (6pm−10pm) that you receive the least texts	39,063.42
	if you use the app	111,824.80
	second of your average workday call (monday−friday	346,246.83
	if you use the twitter app at least once	370,683.79
	call that lasts at least an hour	395,022.17
•	second of your average call duration	3,108,632.19
	day you use any instagram−brand app	7,505,847.37
	day that you use a productivity−type app	8,462,705.30
	instagram−brand app used	28,377,875.40
	social−type app you use	61,264,134.39
	document or report−related app you use	75,097,239.06
	times the average day−to−day variation in your texts	1,282,497,178.36
	percentage point of your time that you spend within .5 kilometers of the busara office	184,130,822,787.78

Notes: Parameters estimated using GMM. Red dot indicates used in a LASSO model; blue indicates used in SR model. In cost matrix, off diagonal elements c_{jk}; j$\neq k$ regularized to zero ($\lambda^{costs}_{offdiagonal} \to \infty$), diagonal elements regularized with $\lambda^{costs}_{diagonal} = 1.0$, set via 3-fold cross validation.

Table A2: Performance of Decision Rules

	Costs	**Income & Intelligence** (Pooled)		**Income**		**Intelligence** (Ravens above median)	
	c_{jj} ¢/action^2	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{SR}$	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{SR}$	$\boldsymbol{\beta}^{LASSO}$	$\boldsymbol{\beta}^{SR}$
				¢/action		¢/action	
Panel A: **Decision Rule**							
text_count_out	0.035	-	-	-0.395	-0.107		
text_count_incoming	0.037	-	-	0.065		0.278	0.145
text_count_evening	0.057	-	-		-0.121		
call_count_out	0.480	-	-	0.625	0.542		
call_count_outgoing_missed	1.91	-	-			-0.208	
calls_noncontacts	1.929	-	-			-0.606	-0.575
max_daily_texts_incoming	3.471	-	-				0.324
intercept	.	-	-	301.071	304.622	490.727	488.441
Panel B: **Prediction Error**		RMSE ($)		RMSE ($)		RMSE ($)	
Baseline Data: Control		4.273 (0.024)	4.278 (0.028)	3.574 (0.050)	3.583 (0.057)	4.971 (0.010)	4.973 (0.009)
Baseline Data: Predicted Transparent		4.328 (0.030)	4.279 (0.031)	3.672 (0.062)	3.585 (0.058)	4.984 (0.012)	4.974 (0.009)
Implemented: Opaque		4.224 (0.135)	4.216 (0.115)	3.549 (0.250)	3.525 (0.215)	4.898 (0.066)	4.906 (0.049)
Implemented: Transparent		4.356 (0.091)	4.189 (0.122)	3.675 (0.179)	3.484 (0.239)	5.037 (0.042)	4.894 (0.052)
Average Payout ($)		4.21	4.18	3.34	3.25	5.07	5.11
N (Control Individuals)		1391	1391	1376	1376	1391	1391
N (Treatment person-weeks, Opaque)		156	156	75	75	81	81
N (Treatment person-weeks, Transparent)		166	154	90	74	76	80

Notes: Panel A reports the decision rule associated with the challenge, and the costs associated with manipulating these behaviors. Panel B reports the performance of each decision rule by outcome, root mean squared error (RMSE) at the week-model level. Pooled metrics present the mean RMSE across models. Predicted Transparent represents the average expected performance of models given the theoretical model, behavior incentives, and estimated costs. Implemented Transparent/Opaque represents the average performance of models when assigned with/without transparency hints. SR model estimated using preliminary cost estimates. Bootstrapped standard errors in parentheses, which hold fixed the decision rule and resample individuals with replacement.

Table A3: SR Models Based on Expert-Estimated Costs

	Costs (Actual)	*Costs* (From Experts)	**Income**			**Intelligence** (above median Ravens)		
			$\boldsymbol{\beta}^{LASSO_{final}}$	$\boldsymbol{\beta}^{SR_{final}}_{ExpertCost}$	$\boldsymbol{\beta}^{SR_{final}}$	$\boldsymbol{\beta}^{LASSO_{final}}$	$\boldsymbol{\beta}^{SR_{final}}_{ExpertCosts}$	$\boldsymbol{\beta}^{SR_{final}}$
Panel A: **Decision Rule**								
text_count_out	0.035	3.804	-0.499	-0.329	-0.093			
text_count_incoming	0.037	5.645	0.141	0.014		0.270	0.223	0.114
text_count_evening	0.057	3.805			-0.115			
call_count_out	0.480	5.4	0.657	0.591	0.501		-0.058	
call_count_outgoing_missed	1.914	5.4				-0.156		
calls_noncontacts	1.929	5.891				-0.547		-0.518
max_daily_texts_incoming	3.471	5.155						0.421
Intercept			296.342	305.309	303.456	489.686	483.529	487.049
$\lambda^{decision}$			759.295	759.295	759.296	1032.37	1032.37	1032.37
Panel B: **Prediction Error**				RMSE ($)			RMSE ($)	
Predicted Opaque			3.572	3.577	3.586	4.972	4.982	4.973
Predicted Transparent			3.831	3.64	3.586	4.983	4.989	4.973

Notes: Panel A reports the decision rules derived from naive LASSO and our strategy-robust model, as well as strategy-robust models that use only control weeks and costs estimated from expert surveys. It also reports the costs associated with these behaviors. Panel B reports the predicted performance of these decision rules, using the experimentally estimated model. $\beta^{LASSO_{final}}$ presented in this table differs slightly from the β^{LASSO} which was implemented. The regularization protocol was updated to select penalization closer to the boundary of 3 coefficients and the sample was changed to coincide with that used for the SR model (it includes only individuals with nonmissing tech skills, dropping approximately 1.5 percent of the sample). For expert survey costs, we infer heterogeneity in gaming ability using variation in participant responses (see Supplemental Appendix).

Table A4: Models Adjusted for Welfare Costs of Manipulation

	Costs	**Income**					**Intelligence** (above median Ravens)				
	c_{jj}	$\boldsymbol{\beta}^{LASSO_{final}}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0.1}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0.5}$	$\boldsymbol{\beta}^{SR_{final}}_{w=1}$	$\boldsymbol{\beta}^{LASSO_{final}}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0.1}$	$\boldsymbol{\beta}^{SR_{final}}_{w=0.5}$	$\boldsymbol{\beta}^{SR_{final}}_{w=1}$
Panel A: **Decision Rule**											
text_count_out	0.035	-0.499	-0.093	-0.092							
text_count_incoming	0.037	0.141					0.270	0.114	0.067	0.030	0.019
text_count_out_evening	0.054										
text_count_evening	0.057		-0.115	-0.115	-0.055	-0.037					0.023
call_count_out	0.480	0.657	0.501	0.494	0.278	0.179					
max_daily_texts_out	1.683				-0.294	-0.222					
call_count_outgoing_missed	1.914						-0.156				
calls_noncontacts	1.929						-0.547	-0.518	-0.422	-0.204	
max_daily_texts_in	3.471							0.421	0.541	0.518	0.387
call_count_over_1_minute	395022										
Intercept		296.342	303.456	303.669	312.514	314.717	489.686	487.049	489.071	488.921	489.317
$\lambda^{decision}$		759.296	759.296	759.296	759.296	759.296	1032.37	1032.37	1032.37	1032.37	1032.37
Panel B: **Prediction Error**				RMSE ($)					RMSE ($)		
Predicted Opaque		3.572	3.586	3.586	3.599	3.607	4.972	4.973	4.974	4.979	4.984
Predicted Transparent		3.831	3.586	3.586	3.598	3.607	4.983	4.973	4.974	4.979	4.984

Notes: Panel A reports the decision rules derived from naive LASSO and our strategy-robust model, with varying social welfare weight w placed on the costs agents incur manipulating. Panel B reports performance, measured as root mean squared error (RMSE). $\beta^{LASSO_{final}}$ presented in this table differs slightly from the β^{LASSO} which was implemented. The regularization protocol was updated to select penalization closer to the boundary of 3 coefficients and the sample was changed to coincide with that used for the SR model (it includes only individuals with nonmissing tech skills, dropping approximately 1.5 percent of the sample). Manipulation costs included in policymaker's objective as $M(\cdot) = w \cdot \mathbb{E}_i \left[c_i(\mathbf{x}_i^*(\boldsymbol{\beta}), \underline{\mathrm{x}}_i)\right] = w \cdot \mathbb{E}_{i,q} \left[\frac{1}{2}\boldsymbol{\beta}' C_{iq}^{-1'} \boldsymbol{\beta}\right]$, for a weight w on consumer welfare.