Manipulation-Proof Machine Learning*

Daniel Björkegren† Joshua E. Blumenstock‡ Samsun Knight§

Brown University U.C. Berkeley Brown University

This version: April 6, 2022
First version: November 30, 2018

Abstract

An increasing number of decisions are guided by machine learning algorithms. In many settings, from consumer credit to criminal justice, those decisions are made by applying an estimator to data on an individual's observed behavior. But when consequential decisions are encoded in rules, individuals may strategically alter their behavior to achieve desired outcomes. This paper develops a class of algorithm that is stable under manipulation, even when the decision rule is fully transparent. We explicitly model the costs of manipulating different behaviors, and identify decision rules that are stable in equilibrium. This approach also makes it possible to quantify the performance cost of making a decision algorithm transparent. Through a large field experiment in Kenya, we show that decision rules estimated with our strategy-robust method outperform those based on standard machine learning approaches.

Keywords: machine learning, manipulation, decisionmaking, targeting

*We are grateful for helpful conversations with Susan Athey, Jon Bittner, John Friedman, Greg Lewis, Ben Roth, and Jesse Shapiro. This project would not have been possible without the creative work of Chaning Jang, Simon Muthusi, Nicholas Owsley, and the rest of the team at the Busara Center for Behavioral Economics. We thank Jolie Wei for excellent research assistance, and numerous audiences for helpful feedback. 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 University, and the Kenya Medical Research Institute.

†dan@bjorkegren.com
‡jblumenstock@berkeley.edu
§samsun_knight@brown.edu

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 an estimator that determines future decisions.

However, these applications of machine intelligence raise an important problem. 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, which can undermine the use of machine learning in critical applications.

The problem of manipulation stems from the fact that the standard estimators used to construct decision rules assume that the relationship between the outcome of interest and human behaviors is 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).

There are two common approaches to deal with this problem. The first, familiar to economists, uses common models but restricts them to predictors that are presumed to have a theoretical relationship to the outcome of interest.[1] This simple intuitive approach amounts to having a dogmatic prior that the cost of manipulation is either infinite (for included predictors) or zero (for excluded predictors). However, most behaviors are manipulable at some cost, and it may be difficult to assess manipulability in new domains, or in modern contexts that can have thousands of predictors. Thus in practice many implementations use a second approach, which we refer to as the 'industry approach.' This keeps decision rules secret, and periodically updates the model to account for changes in the relationship between features and outcomes (Bruckner and Scheffer, 2011). However, such 'security through obscurity' exposes current applications to substantial risk (NIST 2008). If the stakes are high enough,

[1] An extreme version may restrict to predictors that causally affect the outcome of interest (Kleinberg and Raghavan, 2019; Milli et al., 2019). This may make manipulation desirable (for example, an exam may induce students to study and learn general knowledge) but can reduce predictive performance.

people eventually learn—and exploit—a system's weaknesses, and may cause great harm at unanticipated times. This approach also limits the use of machine learning in settings where feedback is noisy or delayed (e.g., it may take years for a social media platform to learn that its recommendation algorithm was gamed by foreign actors). Additionally, societies increasingly demand a 'right to explanation' about how algorithmic decisions are made (Goodman and Flaxman, 2016; Barocas et al., 2018).[2] As a result, it may be untenable to maintain the secrecy that this approach relies on: the more clearly that agents know how their behavior affects a decision, the easier it is to manipulate. And in general, there is no guarantee that the back-and-forth between estimation and agents will reach equilibrium, or if it does, that such an equilibrium will be desirable.

This paper develops a different approach, which builds on recent insights in the computer science literature (Bruckner and Scheffer, 2011; Hardt et al., 2016; Perdomo et al., 2020). We explicitly model the costs that agents incur to manipulate their behavior, and embed the resulting game theoretic model within a machine learning estimator. This allows us to construct decision rules that anticipate strategic agents, and which make good decisions even when the rule is fully transparent. We demonstrate, using Monte Carlo simulations, that our 'strategy-robust' decision rules performs better than standard models when these costs are known, even if costs are mis-specified. We then test the theory in a real world environment, through an incentivized field experiment with 1,557 people in Kenya. We use the experiment to elicit costs of manipulating behavior, and to show that the strategy-robust approach leads to more robust machine decisions.

The paper is organized into two main parts. The first part develops a method to estimate strategy-robust decision rules that are stable under manipulation. We consider a supervised machine learning framework for a policymaker making a decision y_i for each individual i. Each individual prefers a larger decision y_i. We observe a *training* subset of cases that possess both features $\mathbf{x}_i$ and optimal decisions y_i. The policymaker seeks to estimate a decision rule $\hat{y}(\mathbf{x}_i)$ for cases in an *implementation* subset where only features $\mathbf{x}_i$ are observed. Standard methods assume that $\mathbf{x}_i$'s are fixed: training

[2]For instance, 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).

and implementation samples of $(\mathbf{x}_i, y_i)$ are drawn from same distribution. Our method allows individuals to adjust behavior in response to the incentives generated by the decision rule; that is, $\mathbf{x}_i(\hat{y}(\cdot))$ is a function of the decision rule. Thus, while the training samples come from an unincentivized distribution $(\mathbf{x}_i(0), y_i)$; implementation samples come from $(\mathbf{x}_i(\hat{y}(\cdot)), y_i)$. Characterizing this latter distribution requires a new object: the distribution of the elasticity of behaviors $\mathbf{x}_i$ when incentivized, which is described by costs $\mathbb{C}$. We describe several methods to estimate manipulation costs.

We develop a framework that can be applied to general functional forms and to a variety of applications where predictions are used to derive a decision rule. To sharpen intuition, we derive specific results for linear decision rules of the form $\hat{y}(\mathbf{x}) = \boldsymbol{\beta}\mathbf{x}$ and quadratic manipulation costs. The resulting solution takes a simple nonlinear least squares form. Our method introduces a new notion of fit, which has analogues to other common linear regression approaches. Ordinary least squares (OLS) maximizes fit within sample; two stage least squares (2SLS) sacrifices fit within sample to estimate coefficients that have causal interpretations; 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 method sacrifices fit within sample to maximize fit in the counterfactual where the decision rule is used to allocate resources, 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.[3] Our method nests standard linear estimators (OLS and penalized least squares), and we regularize towards them based on the amount of manipulation observed in the data. While most of our analysis focuses on the simple linear case, we also show how this approach can be generalized beyond the linear case to train a strategy-robust decision tree.

We use Monte Carlo simulations to compare this strategy-robust approach to common alternatives. OLS can perform extremely poorly when agents behave strategically. The industry approach, which periodically retrains the model, may not converge, or if it does, may do so slowly or to an undesirable equilibrium. By contrast, our method adjusts the model to anticipate manipulation, in each subgame. In simulations where agents respond to the decision rule and manipulation costs are known, our approach

[3]This is analogous to the concept of performative prediction suggested by Perdomo et al. (2020).

exceeds the performance of other solutions. Our approach can exceed the performance of others even if manipulation costs are misspecified for some cases. Under certain parameters, the presence of manipulation can *improve* predictive performance, if it signals unobservables associated with the outcome of interest (in the spirit of Spence, 1973). In these cases, our method emphasizes features that are manipulable by the types to be screened in, but not by those to be screened out.

In the second part of the paper, we implement and test our method in the context of a field experiment in Kenya. This experiment allows us to compare the performance of the strategy-robust decision rule to standard machine learning decision rules in a real-world environment. Specifically, we built a new smartphone app that passively collects data on how people use their phones, and disburses monetary rewards to users based on the data collected. The app is designed to mimic 'digital credit' products that are spreading dramatically throughout the developing world (Bharadwaj et al., 2019; Brailovskaya et al., 2021). Digital credit products similarly collect user data, and convert it into a credit score using machine learning, based on the insight that historical patterns of mobile phone use can predict loan repayment (Björkegren, 2010; Francis et al., 2017; Björkegren and Grissen, 2019). However, as these systems have scaled, manipulation has become commonplace as borrowers learn what behaviors will increase their credit limits (McCaffrey et al., 2013; Bloomberg, 2015).[4] After being hit by manipulation, several lenders have restricted lending. This setting has two additional useful features that make it particularly attractive to study manipulation. First, because monitoring is passive, we can measure behavior both when agents are incentivized and when they are not. Second, because the payoff to consumers is monetary, manipulation costs can be reported naturally in monetary units.

This field experiment produces several results. First, consistent with prior work, we find that a person's mobile phone usage behaviors ($\mathbf{x}_i(0)$) have a predictive relationship with their characteristics, such as income and intelligence.[5]

Second, we structurally estimate $\mathbb{C}$ in our model; that is, the distribution of costs of manipulating a variety of observed behaviors $\mathbf{x}_i$. These estimates are identified

[4] 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 qualified for (FSD Kenya, 2018).

[5] 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).

through a series of randomly assigned experiments that offer financial incentives to alter behaviors observed through the app. For example, participants may be paid to increase the frequency of outgoing calls in a given week, or decrease the number 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 et al. (2019)). The pattern of costs we estimate is 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).

Third, we find that 'strategy-robust' decision rules, which account for the costs of manipulation, perform substantially better than standard machine learning algorithms. We make this comparison by providing participants financial incentives to 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 with the strategy-robust algorithm are more accurate than classifications from standard algorithms.

Finally, we use our method to estimate the performance cost of algorithmic transparency, incurred to the policymaker for disclosing the details of the decision rule. In the experiment, we experimentally vary the amount of information subjects have about the decision rule, and show that the relative performance of the strategy-robust decision rule increases with transparency. Transparency reduces the predictive performance of standard decision rules by 17% on average, but 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 designing decision rules that are robust to manipulation. To focus attention on the core tension between standard

decision rules and those that account for strategic behavior, our implementation makes several simplifications: our main results are based on linear decision rules (with a brief extension to decision trees); we model manipulation costs as quadratic with a particular form of heterogeneity; we do not allow manipulation costs to change over time; and we focus on the contrast between full knowledge and no knowledge of the decision rule. This leaves many interesting avenues for future work: on adapting this approach to more sophisticated algorithms, such as random forests or neural networks; alternative methods for eliciting and modeling manipulation costs; settings where agents learn about the decision rule or their costs of manipulation; and dynamic, iterative estimation. We briefly sketch several of these extensions in Sections 5 and 6. More broadly, our approach to using experiments to gauge manipulation – and in particular the structural approach to accounting for that response – is 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

Agents game decision rules in a wide variety of empirical settings. Manipulation has been documented in contexts ranging from New York high school exit exams (Dee et al., 2019) and health provider report cards (Dranove et al., 2003), to pollution monitoring in China (Greenstone et al., 2019), to fish vendors in Chile (Gonzalez-Lira and Mobarak, 2019). 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."

Indeed, the dilemma 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. More broadly, our approach connects with literatures in both economics and computer science.

In particular, the problem we study relates closely to the mechanism design

literature – we explore these connections in greater detail after presenting our model in Section 2.2. 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. Our method evaluates predictors with the inverse of the matrix of the costs of manipulating them, in a manner similar to Ramsey (1927). The market design literature has also considered designing allocation algorithms in the face of strategic reporting (Agarwal and Budish, 2021).

Our main experimental application, as well as one of our extensions, connects to work on poverty targeting. In developing countries and other settings where income is difficult to observe, policymakers commonly determine program eligibility (y_i) based on easily observable characteristics or behaviors ($\mathbf{x}_i$) (Hanna and Olken, 2018), and more recently, based on how people use their phones (Aiken et al., 2021). The policymaker may infer a household's type based on the levels of these variables, or, implicitly, on how they change in response to incentives.[6] There is evidence that such decision rules induce households to manipulate their observable features. For instance, Banerjee et al. (2018) find that adding a question about flat screen TV ownership to a census caused people to under-report ownership by 16% on a follow-up survey, making them appear less wealthy.[7]

Finally, our approach relates to recent theoretical work on 'strategic classification' in the computer science literature. Bruckner and Scheffer (2011) and Hardt et al. (2016) develop algorithms to compute Stackelberg equilibria of one-shot (linear) classification settings where agents are strategic and have known, homogeneous costs of specific forms; Dong et al. (2018) extends this approach to an iterative, online environment. Perdomo et al. (2020) characterize general settings that do not depend on a specific

[6] Our method thus includes this latter case of self-targeting (Nichols and Zeckhauser, 1982; Alatas et al., 2016), which identifies beneficiaries based on willingness to engage with a costly "ordeal."

[7] In other examples from the development literature, Camacho and Conover (2011) find that after a program eligibility decision rule was made transparent, it was manipulated by an amount corresponding to 7% of the National Health and Social Security budget. They note, "there is anecdotal evidence of people moving or hiding their assets, or of borrowing and lending children." And Niehaus et al. (2013) find that when implementing agents can be corrupted, considering additional poverty indicators can worsen the targeting of benefits, by making it more difficult to verify eligibility.

cost function for changing individual features, and Kleinberg and Raghavan (2019) explores the distinction between productive and unproductive manipulation. More broadly, this paper relates to machine learning literature on covariate shift (cf. Sayed-Mouchaweh and Lughofer, 2012) and work from the computer security community on adversarial machine learning (cf. Huang et al., 2011).

Thus, papers from a variety of sub-literatures have confronted the notion that agents will act strategically when their actions are used to determine allocations. Relative to prior work, our paper makes two main contributions. First, we develop an equilibrium model of decision rule manipulation that can be estimated, which 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 stress-tests such decision rules in a real-world setting with incentivized agents.

2 Theory

This section introduces our approach by first outlining a general framework for strategy-robust decision rules. We then focus on the special case where the decision rule is linear and costs are quadratic, and derive solutions that we illustrate with simulations and later test in a field experiment.

2.1 General model

A policymaker observes a *training sample*, i.e., a subset of instances that possess both features $\mathbf{x}_i$ and optimal 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 $\hat{y}(\mathbf{x}_i)$ for instances in an *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, credit, or discount 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 policymaker only observes the individual's behavior $\mathbf{x}_i$, which may differ from their bliss level $\underline{\mathbf{x}}_i$. They select a deterministic decision rule $\hat{y}(\mathbf{x}_i)$.

Individuals can manipulate their behavior $\mathbf{x}_i$ away from their bliss level $\underline{\mathbf{x}}_i$ at some cost. Each individual earns utility from the policymaker's decision, minus some cost:

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

Where manipulation could entail hiding assets (Camacho and Conover, 2011), altering digital behavior, tweaking the characteristics of a piece of content or seeding its initial interactions (Barnhart, 2021), or adjusting the keywords in a resume (Caprino, 2019).

Instances i are heterogeneous in two main respects: bliss behaviors $\underline{\mathbf{x}}_i$ and manipulation costs $c_i(\cdot)$. When i knows the decision rule $\hat{y}(\cdot)$ and receives benefits according to it, optimal behavior is given by

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

When $v_i(\cdot)$, $\hat{y}(\cdot)$, and $c_i(\cdot, \cdot)$ are differentiable, first order conditions imply that the optimal $\mathbf{x}_i$ equates the marginal gain and marginal cost of manipulation:

$$\left(\nabla_{\mathbf{x}_i} \hat{y}(\mathbf{x}_i)\right) \frac{\partial v_i}{\partial \hat{y}} = \nabla_{\mathbf{x}_i} c_i(\mathbf{x}_i, \underline{\mathbf{x}}_i)$$

Decision rules

If, during implementation, the policymaker knew each individual i's cost function c_i, they could invert manipulation to infer the individual's type. However, each individual's cost function is not typically known, which leads to information loss (Frankel and Kartik, 2019). We assume that during implementation the policymaker only observes behavior $\mathbf{x}_i$, but during model training the policymaker also obtains some knowledge about costs, believing i's costs are distributed $c_{iq} \sim \mathbb{C}_i$ for a random draw q. We demonstrate a way to recover these beliefs using experiments, and in later sections consider alternate approaches to estimating these costs (such as polling

domain experts or deriving from first principles).

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

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

which deviates from standard supervised learning because it replaces the sample behavior $\mathbf{x}_i$ with anticipated manipulated behavior $\hat{\mathbf{x}}^*_{iq}(\hat{y}(\cdot))$ given manipulation costs c_{iq}.

2.2 Linear model

To more crisply illustrate how the model works in a setting where analytic solutions can be easily derived, we focus on the case where the decision rule is linear and costs are quadratic. While the experiment we develop in Section 4 is designed to match this linear setting, Section 6 considers the nonlinear case of strategy-robust decision trees.

We first consider the case where the utility from the decision exactly coincides with the policymaker's prediction $v_i(\hat{y}) \equiv \hat{y}$. This is a useful benchmark that could, for instance, reflect a scenario where a policymaker is targeting monetary benefits, or a social media platform is determining how much exposure to give a piece of content. Predictions are linear:

$$\begin{aligned} y_i &= a + \boldsymbol{b}' \underline{\mathbf{x}}_i + e_i \\ \hat{y}(\mathbf{x}_i) &= \alpha + \boldsymbol{\beta}' \mathbf{x}_i \end{aligned}$$

with $e_i \perp \underline{\mathbf{x}}_i$.

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 matrix C_i:

$$C_i = \frac{1}{\gamma_i} \begin{bmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{K1} & \cdots & c_{KK} \end{bmatrix}$$

This implies that some behaviors may be harder to manipulate than others, either by themselves (the diagonal c_{kk}) or in conjunction with other behaviors (the off-diagonals c_{kj}). Different types of instances may also find it more or less costly to manipulate (γ_i) – for example, very clever people and automatically-generated content might have larger γ_i.

Given these functional forms, manipulation shifts behavior:

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

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

The strategy-robust decision rule is given by

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

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

Discussion

Our solution coincides with a nonlinear least squares estimator in the simple case where the policymaker knows costs in the training sample, only 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:

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

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

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:

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

Contrast with standard estimators. When estimated on unmanipulated behaviors ($\underline{\mathbf{x}}_i$), this solution differs in three ways from standard estimators. 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 all people have the same gaming ability, manipulation may simply shift behaviors, without damaging their information content.

Second, it anticipates that these best responses may have some *spread*. Heterogeneity in gaming ability that is unobserved will tend to 'muddle' the relationship between a behavior x_i and type $\underline{\mathrm{x}}_i$, as described in Frankel and Kartik (2019). Our moment expectations are taken over the distribution of gaming ability, so such variance will tend to attenuate coefficients.

Third, it 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. For this reason, even if one obtained data from a strategy-robust equilibrium $(y_i, \mathbf{x}_i^*(\boldsymbol{\beta}^{SR}))$, OLS will not generally yield the strategy-robust decision rule.[8] In similar settings, Ball (2019) and Frankel and Kartik (2020) show theoretically that a Stackelberg solution that anticipates reactions (and thus commits to not exploit partial equilibrium correlations) can lead to better predictive performance than repeated best responses.

In later sections, we empirically contrast strategy-robust and standard decision rules. Section 2.3 makes this comparison with simulations. Section 4 compares behavior in an incentivized field experiment, where both decision rules are estimated on unmanipulated behavior.

[8] $\boldsymbol{\beta} = \boldsymbol{\beta}^{SR}$ 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.

Connection to mechanism design. Our problem relates to revelation mechanisms, which attempt to learn a person's type from their behavior. If the mapping between types and behavior is one-to-one, types can be fully revealed (e.g., Spence, 1973). 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. Because the mapping may not be one-to-one, types may not be fully revealed. Our contribution is to develop and test an empirical method that probabilistically separates types and manipulation costs. Recent empirical studies have considered similar problems. For instance, 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. Also related, Eliaz and Spiegler (2019) show 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.

Connection to regularization. Our approach is related to regularization in that it systematically alters how features are expressed in a model. However, strategy-robustness alters this expression in a different way. Namely, strategy-robustness adjusts a model to account for how features will shift in a counterfactual state of the world where that model is incentivized, and as a result, features are manipulated. Such an adjustment may be needed even if there is no concern about overfitting (e.g., the sample is fixed, or one observes the entire population). In contrast, regularization adjusts a model to penalize complexity. For instance, LASSO penalizes the magnitudes of the coefficients, ridge penalizes based on the squared values, and so forth. Regularization is commonly used to reduce overfitting when drawing different samples from the same state of the world. In practice, strategy-robust rules may also be estimated on samples, so it may improve out-of-sample *counterfactual* fit to include a regularization term such as $R_\lambda^{LASSO}(\boldsymbol{\beta}) = \lambda \sum_k |\beta_k|$ in the additional terms '…' in equation (3).[9]

[9]Under these regularization terms, when $M(\cdot) \equiv 0$ and $C_i \to \boldsymbol{\infty}$ the resulting solution corresponds

Welfare cost of manipulation. When the policymaker cares about not only the resulting allocation, but also the manipulation costs that individuals incur, this may be captured by the term $M(\cdot)$. 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.

2.3 Intuition

We demonstrate the method with Monte Carlo simulations of the linear model of Section 2.2. This involves deriving desired allocations $\mathbf{y} = a + \mathbf{b}'\underline{\mathrm{x}} + \boldsymbol{e}$, then assessing different methods to generate decision rules $\hat{y}(\mathbf{x})$. To focus on the intuition, this section assumes that the policymaker is able to recover the manipulation costs of each individual in its training sample ($C_{iq} \equiv C_i$), but not in implementation.

Comparative statics

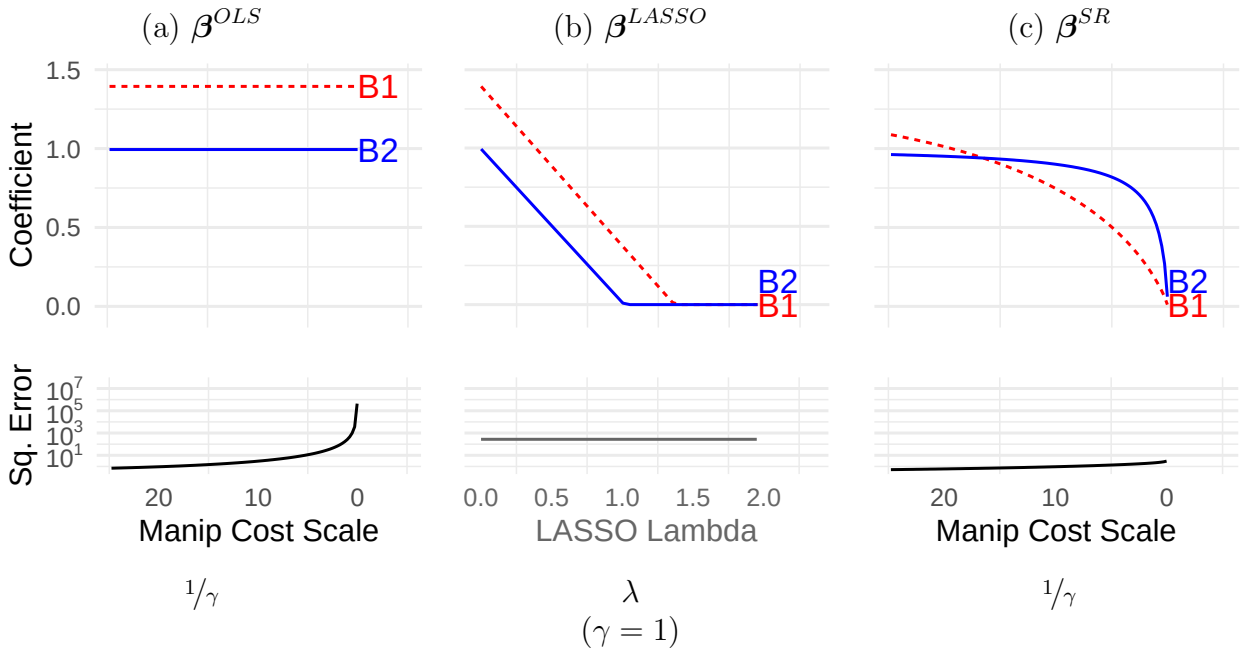
We consider a case where x_1 is more predictive than x_2 in baseline behavior, but would be easily manipulated if used in a decision rule ($b_1 > b_2$ but $c_{11} \ll c_{22}$).

Figure 1 compares our method to OLS and LASSO, which both place most weight on x_1. OLS maximizes predicted performance within the unincentivized sample $(\mathbf{x}_i(\mathbf{0}), y_i)$; as shown in Figure 1a, it performs poorly as manipulation becomes easier. Figure 1b shows that for a given cost of manipulation, LASSO shrinks these coefficients. However, standard regularization does not consider manipulation, and LASSO selection does the wrong thing: it kicks x_2 out of the regression first. In contrast, our method considers how predictive features will be in equilibrium when the decision rule is implemented: $(\mathbf{x}_i(\boldsymbol{\beta}), y_i)$. As shown in Figure 1c, when manipulation costs are high, our method approaches OLS; as manipulation becomes easier, our method substantially penalizes x_1.

The Supplemental Appendix (section 4.1) presents additional comparative statics. If each feature is equally costly to manipulate ($c_{jl} \equiv c_{kl}$), our method shrinks them together, similar to ridge regression. If all individuals have the same gaming ability ($\gamma_{iq} \equiv \gamma$), then manipulation shifts behavior uniformly; although this does not affect

to LASSO or ridge, respectively.

Figure 1: Common vs. Strategy-Robust Decision Rules



Note: The first behavior is more predictive ($b_1 > b_2$), but is easily manipulable ($c_{11} \ll c_{22}$). **(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.

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

predictive performance, individuals may incur substantial costs manipulating. We also include an example that combines strategy-robustness with regularization.[10]

Performance

We contrast our method with standard approaches in a Monte Carlo simulation in Table 1. In this simulation, type $\underline{x}_1$ has a large weight in the desired allocation ($b_1 = 3$) relative to the other two dimensions ($b_2 = b_3 = 0.1$); however, behavior x_1 is much easier to manipulate ($c_{11} = 1$ vs. $c_{22} = 2$ and $c_{33} = 4$). In this environment, OLS considers the static relationship in the unmanipulated data. This rule would perform well if behavior were fixed (the 'no manipulation' column); however, once consumers adjust to the rule, it makes terrible decisions (the 'manipulation' column).

A common 'industry approach' involves retraining the model periodically. Thus, as shown in Panel B of Table 1, if we observe consumers' new behavior and re-estimate OLS, we obtain $\boldsymbol{\beta}^{OLS(2)}$, which places negative weight on the manipulated x_1. However, once consumers respond, the decision rule performs poorly. If we repeatedly allow individuals to best respond, and then estimate the decision rule $\boldsymbol{\beta}^{OLS(r)}$ using data from the prior period $(r-1)$, this process continues to make poor decisions. In this case, the process does not converge; it alternates between decision rules that place high and low weight on x_1.[11] 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.[12]

In contrast, our strategy-robust decision rule ($\boldsymbol{\beta}^{SR}$) penalizes the easily manipulable behavior x_1 and shifts weight to behaviors that are harder to manipulate (x_2 and x_3). It anticipates manipulation off-path, sacrificing performance in the environment in

[10] As the cost of manipulating one particular behavior (c_{22}) decreases, it is penalized, and weight is shifted to other behaviors. The method also can exploit cost interactions, penalizing 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_{12} sufficiently negative), our method further reduces weight on x_1.

[11] These oscillations can be dampened by using cumulative data from all prior periods, as shown in the Supplemental Appendix (section 4). That still takes several iterations to converge to a less performant equilibrium.

[12] For example, Gonzalez-Lira and Mobarak (2019) find that increased enforcement of a ban on selling endangered fish led vendors to learn about, and more effectively undermine, the decision rule.

Table 1: Manipulation Can Harm Prediction (Monte Carlo)

	Decision Rule				**Performance** (squared loss)	
	β_1	β_2	β_3	α	No manip.	Manipulation
Panel A: **Data Generating Process**						
$\boldsymbol{b}^{DGP}$	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. 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 behavior with and without manipulation. Parameters:

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

which it is trained (in-sample, no manipulation) for performance in the counterfactual where there is manipulation. When individuals manipulate as described in the model, our decision rule exceeds the performance of standard estimators.[13]

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 (Panel C of Table 1). The method can also reduce risk even if manipulation costs 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.

Manipulation can improve performance

Manipulation can *improve* performance, if ease of manipulation (γ_i) is correlated with the outcome (y_i). In that case, manipulation represents a signal of the underlying type, as in Spence (1973) and self-targeting (Nichols and Zeckhauser, 1982). The Supplemental Appendix (Sections 4.2 and 4.4) provides examples of such signaling where manipulation improves the performance even of naïve estimators. In the examples, our method *increases* the use of manipulated behaviors to better exploit the information contained in manipulation, and thus further improves performance. In that sense, if different subgroups in a population (like the rich or poor) have different ability to manipulate behavior (Hu et al., 2019), our method can utilize that correlation to bring allocations closer to the policymaker's objective.

3 Estimation

This section describes how the strategy-robust model can be estimated with experimental data. We will assume, for now, that it is possible to experimentally incentivize individuals to manipulate different behaviors. In this setting, individual i in time period t is randomly assigned a decision rule, which pays out based on their behavior: $\hat{y}_{it}(\mathbf{x}_{it}) = \alpha_{it} + \boldsymbol{\beta}'_{it}\mathbf{x}_{it}$. Their resulting behavior may deviate from the bliss level ($\underline{\mathrm{x}}_i$)

[13] The strategy-robust approach can also be combined with the industry approach by using the strategy-robust approach first, then iteratively retraining, as shown in the Supplemental Appendix (Section 4).

due to manipulation, or shocks that are common ($\boldsymbol{\mu}_t$) or individual-specific ($\boldsymbol{\epsilon}_{it}$):

$$\mathbf{x}^*_{it}(\boldsymbol{\beta}_{it}) = \underline{\mathrm{x}}_i + C_i^{-1}\boldsymbol{\beta}_{it} + \boldsymbol{\mu}_t + \boldsymbol{\epsilon}_{it} \tag{5}$$

where both components are mean zero: $\mathbb{E}\boldsymbol{\mu}_t = \mathbf{0}$ and $\mathbb{E}\boldsymbol{\epsilon}_{it} = \mathbf{0}$.[14]

In our experiment, we randomize two types of treatments such that, in a given period, i will receive one of two types of decision rules. She may be assigned a 'control' decision rule, which does not incentivize behaviors: $\boldsymbol{\beta}_{it} \equiv 0$. Alternatively, she may be assigned a 'treatment' decision rule, which incentivizes one behavior $k \in 1...K$, by paying $\beta_{itk} \neq 0$ for each unit of behavior k and nothing for other behaviors: $\beta_{itj} = 0$ for $j \neq k$.

In the controlled environment of our experiment, we observe each individual i over multiple time periods $t \in \mathbb{T}_i$. This allows us to increase statistical power given the cost of onboarding participants. However, as we show in Appendix A1, this same approach can be adapted to recover manipulation costs in a 'one shot' setting where each individual's behavior is observed only once. Likewise, though our experiment reflects a *greenfield* environment where subjects can be observed without incentives, Section 5.6 describes how a similar approach could be used in *brownfield* settings by locally randomizing around decision rules that are already implemented.

3.1 Primitives

We first estimate primitives: types $\underline{\mathbf{x}}$, cost parameters $\boldsymbol{\omega}$ and C^{-1}, and the distribution of unobserved gaming ability V.

Types

We estimate types ($\underline{\mathbf{x}}$) and time period fixed effects ($\boldsymbol{\mu}$) from control periods, using the 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).

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

Costs

We parameterize the cost matrix as:

$$C_{iq} = \frac{1}{\gamma_{iq}} \cdot C$$

allowing heterogeneity by behavior, and by individual:

$$\gamma_{iq} = e^{-\boldsymbol{\omega}'\mathbf{z}_i} + v_q$$

Individual gaming ability can vary with characteristics $\mathbf{z}_i$ (observed in the training sample but not in implementation), and due to unobserved heterogeneity $v_q \sim V$ (where $\mathbb{E}v_q = 0$).

We estimate C and $\boldsymbol{\omega}$ using experimental variation in incentives and a general method of moments (GMM) loss function. We impose $\underline{\mathbf{x}}$ and $\boldsymbol{\mu}$. We limit the potential to overfit using two adjustments. First, we impose the constraint that behaviors move in the direction they are incentivized: $c_{jj} > 0$. Second, we penalize the ease of manipulation towards zero (cost towards infinity), which regularizes towards standard methods (such as OLS and LASSO). We use ridge penalization, allowing separate hyperparameters for diagonal and off-diagonal costs ($\boldsymbol{\lambda}^{costs} = \{\lambda^{costs}_{diagonal}, \lambda^{costs}_{offdiagonal}\}$). In our application we will penalize off-diagonal elements to zero because of noise in estimating them, and use three-fold cross validation to select $\lambda^{costs}_{diagonal}$.

We recover the distribution of unobserved gaming ability V by shrinking and then shuffling the gaming ability residuals. For more details, see Appendix A1.

3.2 Decision Rules

Given these primitives, a strategy-robust decision rule is given by:

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

for Q random draws of the cost matrix C_{iq}, where draws $v_q \sim V$ are treated as random effects. This loss includes any regularization term added to the decision rule itself $R_\lambda(\cdot)$; we set the regularization hyperparameter λ with cross validation in the

unmanipulated baseline sample, where we have more data.

4 Experiment

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' apps that are quickly transforming consumer credit in developing countries (Francis et al., 2017). In Kenya, at the time of our study, CGAP (2018) estimates that 27% of all adults had an outstanding 'digital credit' loan.

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.

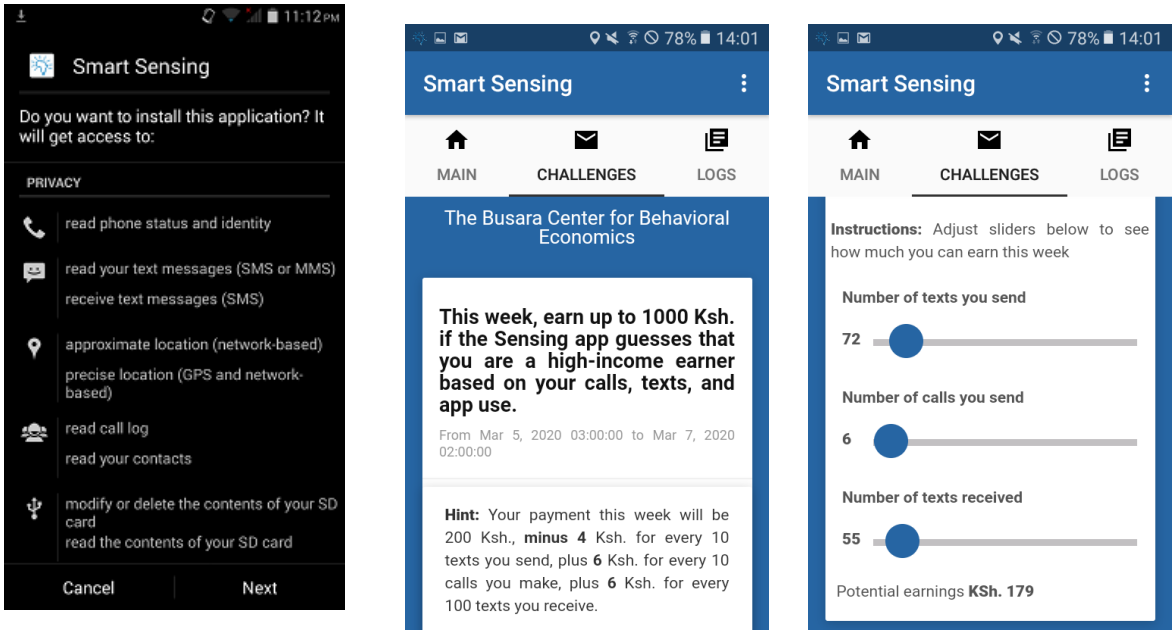
4.1 Experimental design and smartphone app

Our experiment is intended 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) – see Björkegren (2010); Björkegren and Grissen (2019). 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., intelligence, level of activity, education).[15]

[15] 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 with Hint (c) Earnings Calculator



Smartphone app

The 'Smart Sensing' app we worked with Busara to build has two key features. First, it runs in the background to capture anonymized metadata on how individuals use their phones, such as when calls or texts are placed, which apps are installed and used, geolocation, battery usage, wifi connections, and when the screen was on. In total, we extract over $\bar{K} > 1,000$ behavioral indicators ("features"). Second, it delivers weekly "challenges" to users (see Figure 2). These challenges appear on the user's phone, and provide financial rewards based on the user's behavior. The challenges can be very simple ('You will receive 12 Ksh. for every incoming call you receive this week') or more complex ('Earn up to 1000 Ksh. if the Sensing app guesses you are a high-income earner'). Users are paid a base amount of 100 Ksh. for uploading data, plus any challenge winnings, directly via mobile money at the conclusion of each week.

Study population and recruitment

The subject population consists of Kenyans aged 18 years or older who own a smartphone and were able to travel to the Busara center in Nairobi. Participants were

recruited in person in public spaces in Nairobi, and were sequentially invited for an enrollment session at the Busara center. During enrollment, participants completed a survey, which captured ground truth characteristics that we later seek to infer based on phone usage behavior.

Prospective participants were asked to keep the Sensing App on their phones for about 16 weeks. During the informed consent process, participants were told the dimensions of behavior that would be recorded by the app, and were 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.

During onboarding, we discussed with participants strategies for altering different types of phone behavior, surfaced from prior focus groups (Musya and Kamau, 2018). We included this discussion to mimic what might be observed in the long run after individuals discover the easiest ways to manipulate these indicators.

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 the details of their challenge that week (see Figure 2).[16] Challenge incentives were valid until 1pm Tuesday. At the conclusion of the challenge, participants had 21 hours to ensure that their data was uploaded (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. Each week, participants were allocated one of several different challenges at random, which limited the possibility of learning from others.

Participants could attrite in two ways: by not uploading their data, or by not opting in to the challenge. These participants were sent text message reminders or

[16]To minimize the possibility of differential attrition, the pre-opt-in notification was the same for all participants regardless of their assigned challenge.

called by Busara staff, following an attrition protocol detailed in the Supplemental Appendix (Section 1.4). We include in our analysis only participant-weeks where the participant opted in and uploaded during the end-of-week upload window.

4.2 Baseline predictions and model estimation

Predicting user characteristics

During the first several weeks of the experiment, participants were observed but were not incentivized to change behavior.[17] Using data from these 'control' weeks, we find that baseline phone behaviors have a (weak) predictive relationship with participant characteristics. We focus on two primary characteristics ($\tilde{y}_i$): monthly income, and intelligence (above-median performance on Raven's matrices).[18] Results for these outcomes, based on OLS, are shown in Table 2: R^2 are approximately 0.03.[19] Because models with many coefficients can be difficult for participants to interpret, we will use LASSO penalization to restrict to three-variable decision rules.

We use these control weeks to estimate types $\underline{\mathbf{x}}$ for each participant using equation (6), with week fixed effects to absorb idiosyncratic weekly shocks.

Evidence that app-based challenges induce manipulation

During the main phase of the experiment, we randomized participants into groups that received incentives to change specific behaviors. The 'simple' challenges were of the form, 'We'll pay you β_j for each additional x_j you do', where behavior j and amount β_j are assigned randomly. For example, one challenge was, "You will receive 3 Ksh. for each text you send this week, up to Ksh. 250." We restricted consideration to behaviors j that had some predictive relationship to participant types at baseline (such as those shown in Table 2).[20] Payout levels β_j were drawn for each participant

[17] In these weeks, 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 50 for accepting this challenge.'

[18] In addition to monthly income and intelligence, we conducted experiments to predict whether a person is married, has advanced technology skills, has many friends, is below median age, or communicates a lot (several different measures of total phone activity). Pooled results for all characteristics are provided in the Supplemental Appendix (SA Table S1).

[19] These R^2's are fairly low, likely due to the fact that we have a small sample of relatively homogeneous users, observed for short time spans, and that our probes are relatively limited.

[20] Specifically, we run LASSO regressions for each characteristic to select models with three behavioral predictors. We included all selected behaviors and similar behaviors (correlates, different

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.

at random; most incentives were positive but some were negative (participants were incentivized to reduce behavior).[21]

Participants' behavior changed in response to these simple challenges. Table 3 presents a regression of each participant's weekly level of different behaviors (columns) on randomly assigned incentives to change them (rows). There are three takeaways. First, individuals manipulated the behaviors that were incentivized, as shown by the diagonal, which is positive and significant for most behaviors. Second, some behaviors were 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

measures of the same concept, or behaviors selected by LASSO if the original behavior was omitted). For example, if the original regression selects outgoing calls, we also include incoming calls. Note that by including only a subset of variables, our procedure implicitly assumes that omitted variables are costless to manipulate.

[21]Each individual's payment level for j was drawn from $\{-2r_j, -r_j, r_j, 2r_j, 4r_j, 8r_j\}$, for scalar r_j. We scaled the payout for each behavior so that the maximum payout could be achieved by someone reaching the 90th percentile of baseline behavior.

the number of texts sent (possibly because 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 (outgoing)	# Missed calls (incoming)	# 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	4.16	**0.709**	0.825	0.128	-0.002
	(2.196)*	(0.403)*	(0.597)	(0.252)	(0.995)
# Missed outgoing calls	-0.206	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	0.68	**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	0.522	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.

In Section 5.4, we evaluate other methods for measuring manipulation costs. In the Supplemental Appendix (section 2.1), we show that the quadratic cost assumption is a reasonable (if imperfect) approximation of how people respond to variable incentive amounts.

Estimation

We next use the data from these first parts of the experiment to estimate manipulation costs. We allow the distribution of manipulation cost $\mathbb{C}(z_i)$ to differ by whether a

person reports having high tech skills ($z_i \in \{0, 1\}$), and by an unobserved random effect v_q.[22] Table 4 summarizes these estimated costs for behaviors selected by our models (for all behaviors, see Appendix Table A1). With our sample size, we found that off-diagonal elements were noisily estimated, so we penalized them to zero ($\lambda^{costs}_{offdiagonal} \to \infty$); this results in a diagonal cost matrix C.

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.

4.3 Results: Naïve vs. Robust Decisions

The final and most important stage of the experiment compares decisions made by standard machine learning algorithms to those made by strategy-robust decision rules that account for the costs of manipulating behavior. The strategy-robust decision rules can be directly estimated with equation (3), using estimates of $\underline{\mathbf{x}}_i$ and $\mathbf{C}_i$ from prior stages of the experiment.[23]

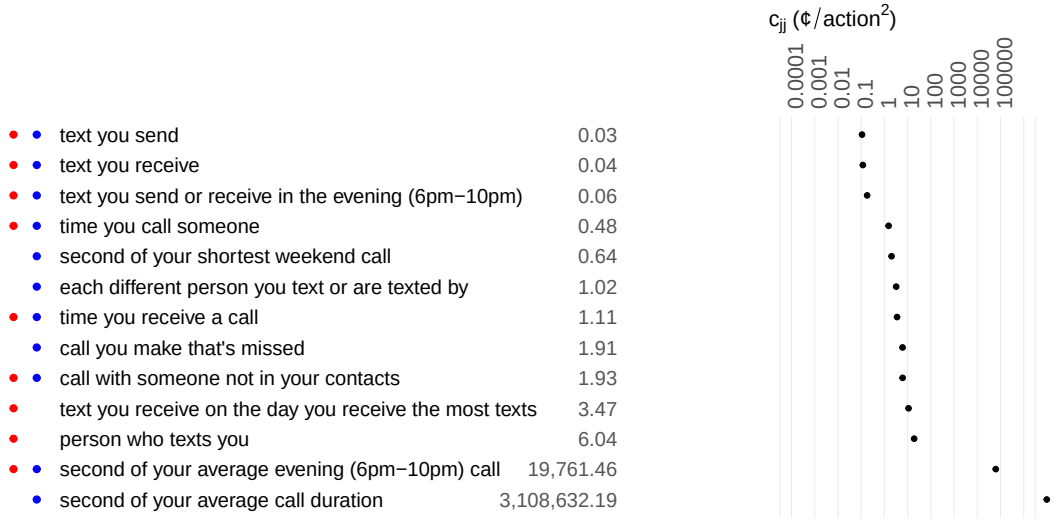
In this final stage, subjects received 'complex' challenges that rewarded them based on how the decision rule classified them. These challenges were designed to mimic real-world applications of machine learning, where people can receive a desirable benefit as a result of their classification, such as a loan (Björkegren and Grissen, 2020)

[22] Tech skills explained the most heterogeneity in preliminary analysis; Spence signaling will only be captured in this dimension of heterogeneity.

[23] For estimating decision rules we used $\underline{\mathbf{x}}_i$ equal to 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 Appendix (Section 2.2) 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 Appendix (Section 2.2) shows that our results are robust to their inclusion.

Table 4: Estimated Manipulation Costs

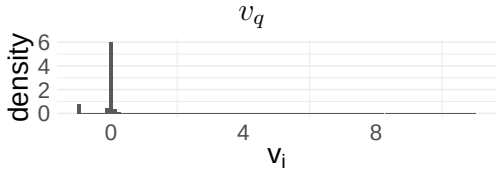
Heterogeneity by Behavior (**C** diagonal; subset of behaviors selected by models)



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.

or a cash transfer (Aiken et al., 2021). In our experiment, the challenges 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'; results from other complex challenges are provided in the Supplemental Appendix.[24]

Estimating Decision Rules

In order to keep decision rules simple and interpretable for our participants, we used LASSO regularization to restrict decision rules to have at most three predictors.[25] The distribution of unobserved gaming ability V is affected by a shrinkage parameter, which we calibrated based on performance on the first few weeks of decision rules (see Appendix A1).

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. Under the transparent treatment, users saw the coefficients of the decision rule, which revealed how much they would be rewarded for each behavior. In the transparent treatment, we also provided an interactive interface that showed participants how their payments would be calculated from different behaviors (see Figure 2c). Because the transparent treatment revealed information about potential decision rules, after a person had seen a transparent challenge for $\hat{y}$, we did not assign them to an opaque challenge for the same outcome.

Table 5 provides suggestive evidence of how decision rule incentives affect behavior. The first panel simply indicates the estimated decision rule: high-income people make more outgoing calls, send fewer texts, and receive more texts. In the second panel,

[24] Full results are available at `https://dan.bjorkegren.com/manipulation-appendix-extra.pdf`. We map characteristics, $\tilde{y}$, into payouts, y, with transformation $y_i = \max\{0, \min\{1000, \tilde{y}_i \cdot \frac{1000}{\tilde{y}_j^{(90\%)}}\}\}$, where $\tilde{y}^{(90\%)}$ indicates the 90th percentile of raw outcome $\tilde{y}$.

[25] To do this, 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 used the same λ to penalize our strategy-robust decision rule, and also allowed it to select only among 3-variable models.

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. * $p < 0.1$.

we see that if we pay people to 'act like a high-income earner' without revealing 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 incentivized 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 incentivized participants to use 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). 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.

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 appendix.

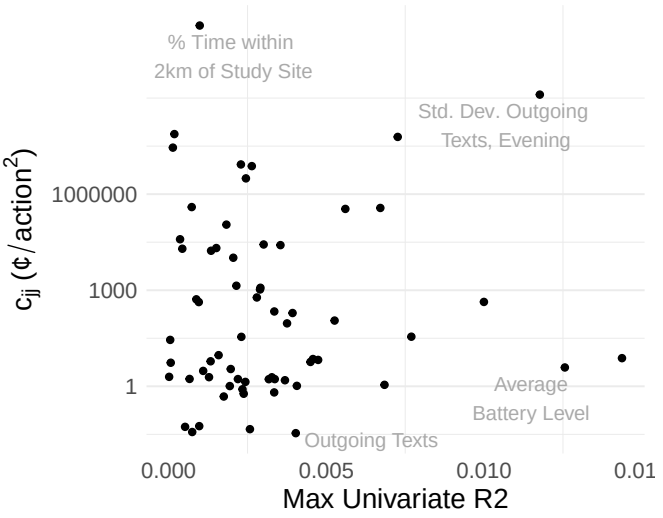
First, in Panel A, we observe important differences in the decision rules. LASSO places weight on the behaviors that were most correlated at baseline: outgoing calls, outgoing texts, and incoming texts. However, some of these behaviors, particularly text messaging, are easy to manipulate (as shown in the 'Costs' column). The strategy-robust decision rule both selects behaviors that are harder to manipulate (i.e., evening texts rather than incoming texts), and shrinks the importance of more easily manipulated behaviors (especially outgoing 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 of rows report the prediction error that was expected ex ante, based on behavior observed during the control weeks. The first row shows that when there is no manipulation, LASSO is 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 will differ from the expected prediction error if people respond differently than anticipated by our model. 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 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 SA Table 1).

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

Figure 3: Manipulation Costs vs. Baseline Predictive Power



Each dot is a feature. 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.

decision rule. In practical implementations, policymakers could adaptively tweak the level of robustness to match the level of manipulation.

5 Discussion

5.1 Contrast to standard estimators

Standard supervised machine learning estimators evaluate 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 manipulation costs. The figure compares the cost of manipulation (y-axis) to the baseline predictive power (x-axis) of several dozen features from our experiment. This spread in costs highlights how standard estimators may not be appropriate when people are actively manipulating 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 approach that is intuitive to economists 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 Appendix (section 4.3). 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.

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 Appendix, section 4.3). 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. 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 an equilibrium because they do not anticipate that agents will respond.

5.2 Application to poverty targeting

To help illustrate how the strategy-robust method can apply in other settings, Figure A2 sketches an application to poverty targeting. Here, a policymaker wishes to distribute cash transfers to individuals with a particular poverty status. The policymaker does not observe each individual's actual poverty status; instead, they predict the individual's poverty status based on observables (similar to a proxy means test). If individuals

were aware of this prediction function, they might manipulate their behavior to attain benefits (Banerjee et al., 2018). We consider proxies based on mobile phone usage, as in recent work by Aiken et al. (2021) and Blumenstock et al. (2015).

Figure A2 compares the Receiver Operating Characteristic (ROC) curves for poverty prediction models that are constructed using strategy-robust (dark green line) and naïve (light blue line) decision rules. The curves are simulated using data from the Kenyan smartphone app. We consider individuals with true below-median income to be 'poor' and above-median income to be 'non-poor.' We form a binary prediction of poverty status by taking thresholds of our income models (shown previously in Table 6). We assess performance on our experimental participants who were assigned a transparent income decision rule; to mitigate the effects of sampling variation, we present the performance of both models on those who were assigned the two different decision rules.

Our results suggest that the strategy-robust decision rule is more stable. We observe modest gains in performance from using the strategy-robust decision rule (AUC=0.632), relative to the naïve decision rule (AUC=0.597), on the sample of individuals assigned our strategy-robust rule (panel A). There is no difference in performance of the two rules in the sample assigned to the naive decision rule (panel B). These simulations should be taken only as proof of concept, but they are illustrative of the sort of benefits that might be achieved with strategy-robust decision rules in applied contexts.

5.3 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 optimal opaque rule (under the assumption that opacity will prevent it from being manipulated) to the optimal strategy-robust transparent rule (factoring in equilibrium manipulation). Because the opaque rule also faces the threat of manipulation, this difference represents an upper bound of the true performance cost. Crucially, under the assumptions of our model, this quantity can be estimated without revealing the decision rule: it only requires the estimation of types and costs

(the first part of our experiment).[26]

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: $\$4.356 - \$4.224 = \$1.32$ (3%) for naive models, and negligible for our strategy-robust models.[27] The income and intelligence outcomes had a lower cost of transparency 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%.

5.4 Alternate methods to estimate manipulation costs

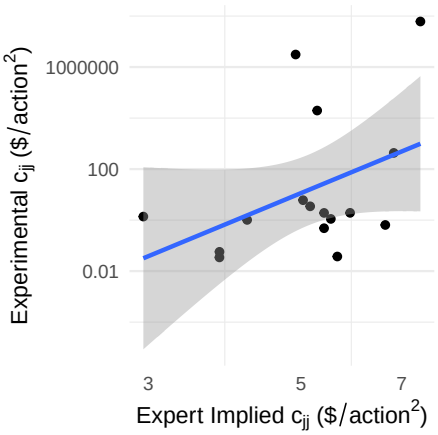
Estimating a strategy-robust decision rule requires beliefs about the costs of manipulating different behaviors. This paper demonstrates an experimental approach to eliciting those costs, but alternative approaches may be better suited to other settings:

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. 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 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

[26] 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.

[27] 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 vs. 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.

low-cost way to estimate manipulation costs. See Supplemental Appendix Section 3.

First principles/structural approach. In some cases, it may be possible to estimate the cost of underlying manipulations from market prices and first principles.[28] 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 messages.

5.5 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.

[28]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 noncontacts 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.

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.

5.6 Greenfield vs. brownfield implementations

Like our study, new applications of machine learning are typically trained in *greenfield* settings, using baseline data that was not incentivized, and not manipulated. Models trained in new settings can be acutely susceptible to manipulation, as baseline data does not expose evidence of manipulation. In greenfield settings, during training it is possible to infer individual types directly from baseline data (equation (6)), and then estimate costs with component-wise deviations. Our method can also be applied in *brownfield* settings, where a decision rule has already been implemented and baseline behavior is already manipulated. This can be done by experimenting locally around the implemented rule and adding moment conditions that invert observed behavior to estimate types and time period fixed effects (see Appendix A1).

5.7 Learning, and non-deterministic decision rules

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):

$$\mathbf{x}_i^*(\hat{y}(\cdot)) = \arg\max_{\mathbf{x}_i} \left[\mathbb{E}_{\tilde{\hat{y}}_i \sim G_i(\hat{y})} \left[v_i \left(\tilde{\hat{y}}_i(\mathbf{x}_i) \right) \right] - c(\mathbf{x}_i, \underline{\mathbf{x}}_i) \right] \tag{7}$$

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

We focus on a linear model with no risk aversion ($v_i(\hat{y}) = \hat{y}$), in which uncertainty would not affect expected behavior. However, if individuals are 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 (7). 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 may respond to approximations. For instance, even for uni-dimensional nonlinear functions like tax schedules, agents commonly respond to linear heuristics (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.[29]

Another possible consideration is social learning: while our model assumes manipulation costs are stationary, in some environments costs may vary over time and be influenced by social connections. Modeling those learning processes is an open area for future work; 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.[30] We also limited the number of times that an individual received a challenge targeting the same outcome, and never assigned an opaque challenge after a person had received a transparent challenge for the same outcome.

[29] If individuals differ in the heuristics they use, those differences could aid in prediction, just like heterogeneity in gaming ability γ_i in our linear model.

[30] 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.

6 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.

6.1 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 $\Delta_{ik} = \max_{x_{ik}}(\hat{y}([\mathbf{x}_{i,-k}^*, x_{ik}])) - \hat{y}([\mathbf{x}_{i,-k}^*, \underline{\mathrm{x}}_{ik}])$.

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:

$$R_1(k, s) = \{\mathbf{x} | x_{ik} \leq s\}$$

$$R_2(k, s) = \{\mathbf{x} | x_{ik} > s\}$$

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 a strategy-robust tree to a standard decision tree, using a simple example with two features. In this example, both features x_1 and x_2 appear to be predictive of the binary label. x_2 is costly to manipulate, but x_1 is manipulable at low cost and differentiates only a small part of the population.[31]

If we train a standard decision tree (Figure 5a), 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 tree is not stable: everyone will be classified as positive. In contrast, after the strategy-robust tree (Figure 5b) 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. Similar to the linear models, strategy-robustness in many cases attenuates the tree, reducing the number of feature interactions and thus the model variance. It similarly induces a commitment to not use information that is immediately informative, but would be manipulated in equilibrium. In Supplemental Appendix Figure S7, we provide an example of a tree where strategy-robustness does not attenuate but rather expresses different variation, using manipulation as a signal of type.

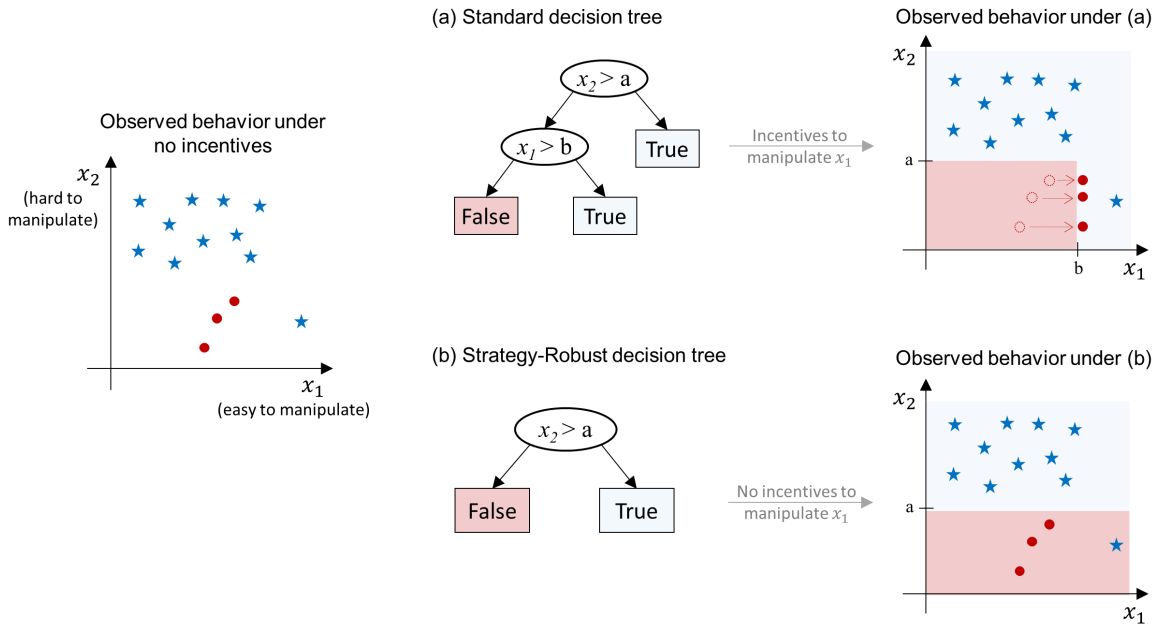
6.2 Discussion

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 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

[31] 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



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 ($c_{i1} \ll c_{i2}$). **(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.

that manipulated data is 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 when manipulation of such forms is taken into account.[32] 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.[33]

The shape of manipulation costs

Experiments can reveal the shape of manipulation costs, and thus suggest appropriate functional forms. In the Supplemental Appendix (section 2.1), we use our experimental data to analyze how behavior responds to random variation in financial 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).[34]

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

[33] 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.

[34] 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.

7 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.

Our discussion focuses 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 of embedding a model of behavior within a machine learning estimator may be relevant to a wide range of contexts where machine learning systems face a changing human environment. In this sense, it offers a machine learning interpretation of Lucas (1976), where algorithmic decisions change the context of the systems they model. In our setting, $\boldsymbol{\beta}$ 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.

Alatas, Vivi, Abhijit Banerjee, Rema Hanna, Benjamin A. Olken, Ririn Purnamasari, and Matthew Wai-Poi, "Self-Targeting: Evidence from a Field Experiment in Indonesia," *Journal of Political Economy*, March 2016, *124* (2), 371–427.

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.

Barnhart, Brent, "How to survive (and outsmart) the Instagram algorithm," June 2021.

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

Bharadwaj, Prashant, William Jack, and Tavneet Suri, "Fintech and Household Resilience to Shocks: Evidence from Digital Loans in Kenya," Working Paper 25604, National Bureau of Economic Research February 2019.

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

_ **and Darrell Grissen**, "Behavior Revealed in Mobile Phone Usage Predicts Credit Repayment," *The World Bank Economic Review*, 2019.

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

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.

Brailovskaya, Valentina, Pascaline Dupas, Jonathan Robinson, and Jonathan Robinson, "Digital Credit: Filling a hole, or digging a hole? Evidence from Malawi," Working Paper May 2021.

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.

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.

Hanna, Rema and Benjamin A. Olken, "Universal Basic Incomes versus Targeted Transfers: Anti-Poverty Programs in Developing Countries," *Journal of Economic Perspectives*, November 2018, *32* (4), 201–226.

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.

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

Hu, Lily, Nicole Immorlica, and Jennifer Wortman Vaughan, "The Disparate Effects of Strategic Manipulation," *Proceedings of the Conference on Fairness, Accountability, and Transparency - FAT* '19*, 2019, pp. 259–268. arXiv: 1808.08646.

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.

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.

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.

Niehaus, Paul, Antonia Atanassova, Marianne Bertrand, and Sendhil Mullainathan, "Targeting with Agents," *American Economic Journal: Economic Policy*, 2013, *5* (1), 206–238.

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.

Appendices

A1 Estimation Details

Moment Conditions

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

Incentives are orthogonal to idiosyncratic behavior shocks ($\mathbb{E}[\boldsymbol{\beta}_{itk}\epsilon_{itj}] = 0$). For each pair of behaviors jk (including $j = k$) this yields sample moment condition:

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\mathbb{T}_i}\beta_{itk}\left[x_{ijt} - \underline{\mathrm{x}}_{ij} - \mu_{jt} - e^{-\boldsymbol{\omega}'\mathbf{z}_i}\cdot[C^{-1}\boldsymbol{\beta}_{it}]_j\right] = 0 \tag{8}$$

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

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

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

where K_{it}^{eval} is the set of behaviors to be evaluated for i in period t.[35] 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, 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 $\underline{\mathrm{x}}$ and $\boldsymbol{\mu}$. Both are based on the condition $\mathbb{E}[\epsilon_{itk}] = 0$. For each individual i and behavior k, we have

[35] 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.

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

For each time period t and behavior k we have

$$\mu_{kt} = \frac{1}{|\{i | \mathbb{T}_i \ni t\}|} \sum_{i | \mathbb{T}_i \ni t} \left[x_{ikt} - \underline{\mathrm{x}}_{ik} - e^{-\boldsymbol{\omega}' \mathbf{z}_i} \cdot [C^{-1} \boldsymbol{\beta}_{it}]_k \right] \tag{11}$$

Identification still requires observing random variation along each behavior in the decision rule (and ensuing manipulation).[36]

Manipulation Cost Regularization

We add to our GMM loss function the regularization term:

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

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 (9), 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 $V = \{\max(\phi \cdot \tilde{v}_i, \underline{\mathrm{v}})\}_i$, where $\underline{\mathrm{v}}$ 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, $\underline{\mathrm{v}} = \min_i(\tilde{v}_i | \phi \cdot \tilde{v}_i \geq -\min_j(e^{-\boldsymbol{\omega}' \mathbf{z}_j}))$).

We calibrated ϕ to 1e-6; For details, see Supplemental Appendix 2.2.2.

[36] 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.

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 in equation (8) 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 (9) 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 (10) 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 (11), 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.

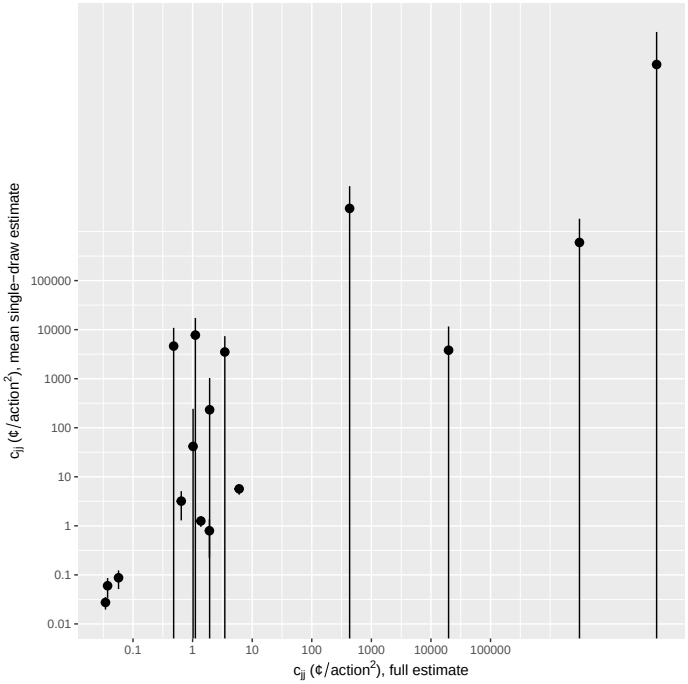
An estimate of each person's bliss behavior can then be obtained by undoing any predicted manipulation:

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

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

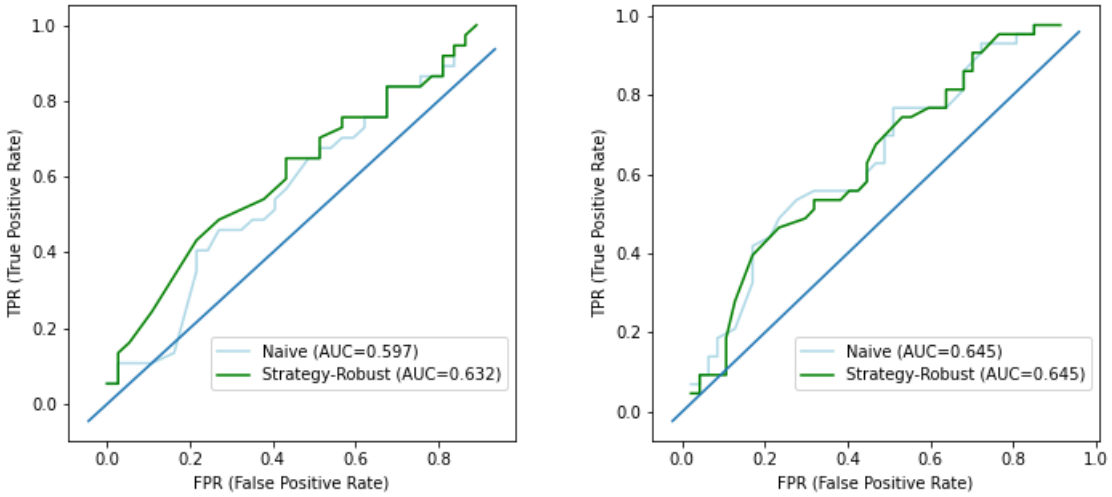
Figure A1: Manipulation Costs Estimated with only One Observation per Person



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.

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 A2: Application to Poverty Targeting

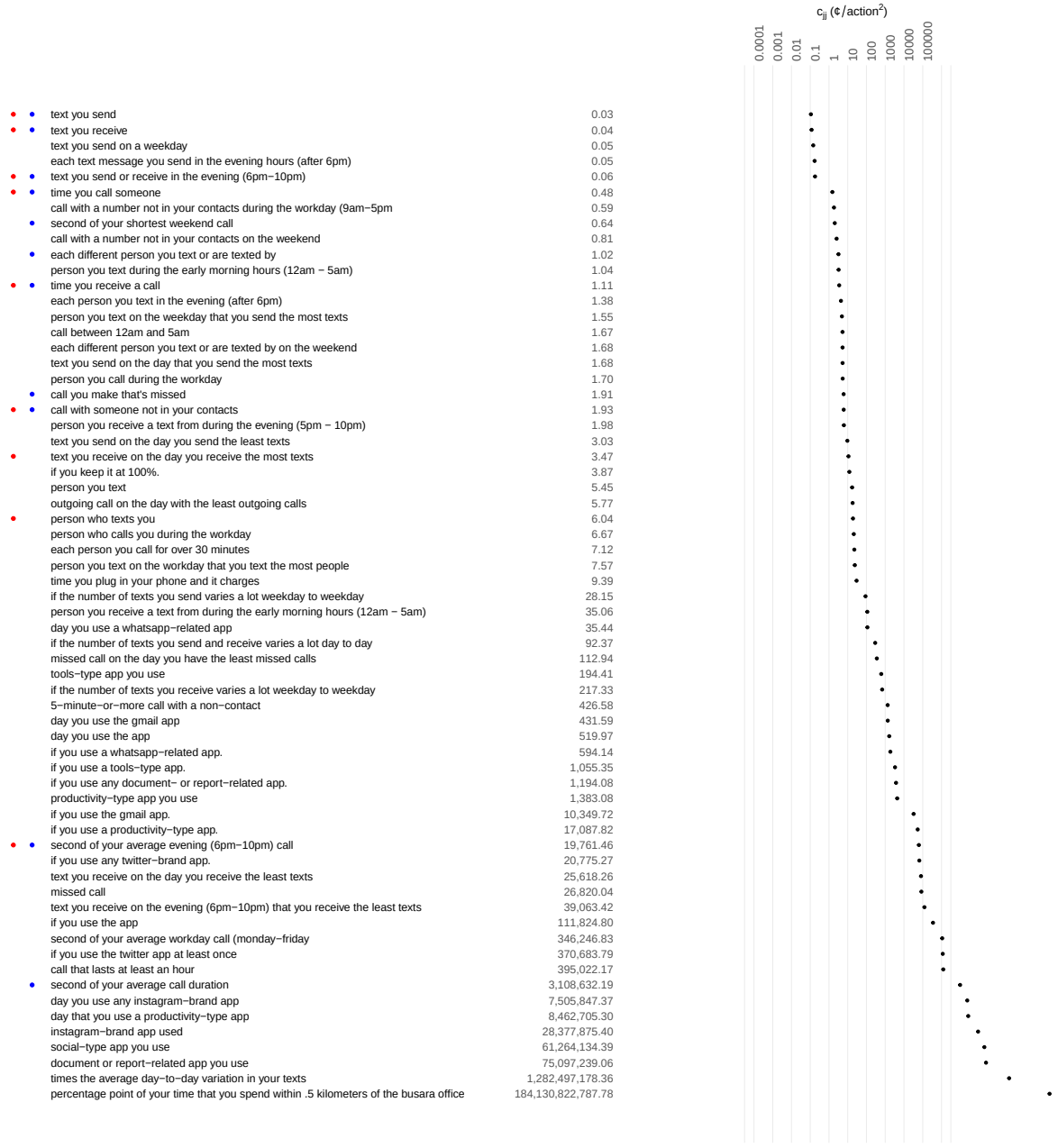


(a) Strategy-Robust sample (b) Naïve sample

Notes: Receiver Operating Characteristic (ROC) curves compare the performance of Strategy-Robust vs. Naïve machine learning rules, when the task is to predict poverty status (defined as those with below-median self-reported income). Simulations are evaluated on the sample of Kenyan smartphone users assigned to a transparent monthly income decision rule ($N = 164$). Panel A is estimated on the sample of individuals assigned Strategy-Robust decision rules; Panel B is estimated on the sample assigned Naïve rules.

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	5.11	5.07
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.