Differentially Private Timeseries Forecasts for Networked Control

Po-han Li\textsuperscript{1}, Sandeep P. Chinchali\textsuperscript{1}, Ufuk Topcu\textsuperscript{2}

Abstract—We analyze a cost-minimization problem in which the controller relies on an imperfect timeseries forecast. Forecasting models generate imperfect forecasts because they use anonymization noise to protect input data privacy. However, this noise increases the control cost. We consider a scenario where the controller pays forecasting models incentives to reduce the noise and combines the forecasts into one. The controller then uses the forecast to make control decisions. Thus, forecasting models face a trade-off between accepting incentives and protecting privacy. We propose an approach to allocate economic incentives and minimize costs. We solve a biconvex optimization problem on linear quadratic regulators and compare our approach to a uniform incentive allocation scheme. The resulting solution reduces control costs by 2.5 and 2.7 times for the synthetic timeseries and the Uber demand forecast, respectively.

I. INTRODUCTION

Controllers relying on timeseries forecasts have applications in power grid operations [1], [2], cellular network traffic scheduling [3], and taxi fleet routing. For example, power grid operators use electricity demand forecasts to charge the batteries, and cellular network providers use city-wide mobility forecasts to allocate bandwidth among base stations. Ridesharing companies use customer demand forecasts to assign taxis to different queues in a city, which is the example we used in our experiments. In these examples, the controller relies on an accurate future timeseries forecasts to make control decisions and then minimizes its control cost. The control cost is a function of the forecasting error, which is the difference between the forecast and the actual timeseries.

We use the system model in Fig. 1 to characterize systems with a controller relying on a timeseries forecast. Multiple forecasting models transmit timeseries forecasts to the controller through a network. The controller can only receive forecasts from the forecasting models but cannot influence the timeseries. Fig. 1 shows the system model of the controller and forecasting models. The forecasts have different intrinsic errors, and the controller combines the forecasts to generate a more accurate one used for control decisions. However, forecasting models tend to add anonymization noise to the forecasts to protect data privacy, thus resulting in more inaccurate forecasts and additional control costs. We analyze a scenario where the controller is able to pay the forecasting models an economic incentive, such as money, to reduce their noise. The more incentive paid, the more accurate the forecasts, and the lower the control costs. The

\textsuperscript{1}Department of Electrical and Computer Engineering, The University of Texas at Austin. \textsuperscript{2}Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, \{pohanli, sandeepc, utopcu\}@utexas.edu
consensus algorithms [7], and cyber-physical systems [9]. Another work studies controllers with multiple sensors [10]. Our work also analyzes the trade-off between data privacy and performance among multiple sources of signals, but our controller interacts with sources using incentives. Thus, it results in another trade-off of forecasting models to receive incentives or protect privacy.

Other literature also studies settings that encourage less privacy noise [11], [12], [13], [14]. However, our work focuses on the setting of a networked controller that pays incentives to forecasting models, while they focus on mobile crowdsensing systems [11], [12], e-healthcare devices [13], and recommender mechanisms [14].

B. Insights and Contributions

Lee et al. study the appropriate range of differential privacy noise levels in a variety of applications [15], and their work gives us the insight that the noise levels do not need to be a priori fixed. In fact, forecasting models can choose their own noise levels from a range. Thus, the noise levels can be reduced by an economic incentive from the controller since it is a flexible choice. The more incentive the controller pays, the less noise, resulting in a more accurate forecast. The proper use of incentives to minimize costs is an emerging issue.

Based on the insights, our key contributions are threefold. First, we characterize the trade-off between privacy and incentives. Second, we formulate an incentive allocation problem, in which the controller optimizes the allocation of incentives and combines the forecasts to minimize its control cost. Lastly, we use the linear quadratic regulator as an example of this scenario and solve a biconvex optimization problem with guaranteed local optimality. We then numerically show that our method can significantly reduce the control cost on synthetic ARIMA timeseries and the Uber demand forecast.

II. PRELIMINARIES

A. ϵ-Local Differential Privacy

Let ε be a positive real number and A be a forecasting model with randomness in its output [16]. A takes a model’s private data as input and maps it to the range imA. The forecasting model A is said to provide ϵ-local differential privacy if for all pairs of private input data x and x′ ∈ X and all possible subsets S of imA:

\[ Pr\{A(x) ∈ S\} ≤ e^ε × Pr\{A(x′) ∈ S\}, \]

where the probability is due to the randomness of the forecasting model A, and X is the universal set of all inputs. The main difference between standard (global) differential privacy and local differential privacy is that the former takes all models’ private data pairs, and the latter takes a single model’s private data pairs. Here, ε is called the privacy budget. The larger the privacy budget ε, the less privacy preserved.

B. Laplace Mechanism

The Laplace mechanism is commonly used for obtaining ϵ-local differential privacy. It perturbs the output of functions with random noise to protect the original information of the input. We use the definition from [17], [18]. Given any arbitrary function A : X → R^d, the sensitivity of A under the L1 norm is defined as:

\[ \text{Sen}(A) = \max_{x,x′ ∈ X} ||A(x) - A(x′)||_1. \]

The Laplace mechanism adds Laplacian noise to the output of the function A to make it ϵ-local differentially private:

\[ A(x, ε) = A(x) + \text{Lap}\left(0, \frac{\text{Sen}(A)}{ε}\right)^d, \]

where Lap(μ, b)^d is a random vector with independent and identically distributed Laplace random variables in each component. Here, A is the original forecasting model without randomness, and A is the differentially private model. x and x′ is any possible pair of private data from a single model.

C. Input-Driven LQR

We model our controller as input-driven LQR motivated by [1], [4], which introduce it in terms of data compression and adversarial attacks. For clarity, we summarize the derivation here and refer readers to [1], [4] for details. x ∈ R^n and u ∈ R^m represent the state and action of the controller, and S ∈ R^p is the external time series. For time step t, the linear system dynamics are given by:

\[ x_{t+1} = Ax_t + Bu_t + CS_t, \]

where A ∈ R^{n×n}, B ∈ R^{n×m}, and C ∈ R^{n×p} are the parameters describing how the current state x_t, action u_t, and external timeseries S_t affect the next state x_{t+1}. The quadratic cost function is defined with positive-definite matrices Q and R:

\[ J^c(u; S, x_0) = \sum_{t=0}^{T} x_t^T Q x_t + \sum_{t=0}^{T-1} u_t^T R u_t, \quad Q > 0, R > 0. \]

The optimal actions are determined by the initial state and the external timeseries observed by the controller:

\[ u^*(x_0; S) = \arg \min_u J^c(u; S, x_0) = -K^{-1}(L_1 x_0 + L_2 S). \]

According to [1], [4],

\[ K = \text{Diag}(R, T) + \sum_{t=0}^{T-1} M_t^T Q M_t, \]

\[ L_1 = \sum_{t=0}^{T-1} M_t^T Q A_{t+1}^T, \quad L_2 = \sum_{t=0}^{T-1} M_t^T Q N_t, \]

where Diag(R, T) ∈ R^{mT×mT} is a block matrix placing T R matrices on the diagonal. Thus, we define actions generated by perfect timeseries as u^* and actions generated by observed, possibly noisy, timeseries as ˜u^*. The two
actions and their difference are given by:
\[
\begin{align*}
    \nu^* &= -K^{-1}(L_1x_0 + L_2S), \\
    \hat{\nu}^* &= -K^{-1}(L_1x_0 + L_2\hat{S}).
\end{align*}
\] (5)
\[
\hat{\nu}^* - \nu^* = -K^{-1}L_2(\hat{S} - S).
\] (6)

Eq. 6 shows the error between actions is linear with respect to the difference of timeseries, and the coefficient is determined by the parameters of the system dynamics, initial state \(x_0\), and the parameters of the cost function \(Q, R\).

Now, we define the increase in the control costs due to forecasting errors as the regret \(\Delta J^c\):
\[
\Delta J^c = J^c(\hat{\nu}^*; S, x_0) - J^c(\nu^*; S, x_0) = (\hat{\nu}^* - \nu^*)^\top K (\hat{\nu}^* - \nu^*) = (\hat{S} - S)^\top L_2^\top KL_2(\hat{S} - S).
\] (7)

Note that both \(\Psi\) and \(K \succ 0\). An intuitive way to explain Eq. 7 is that extra control cost is quadratic in the forecasting error \((\hat{S} - S)\) since the weight of the error \(\Psi\) is positive definite.

### III. System Model

Imagine that a controller of a ride-sharing operator must assign its taxis to \(n\) locations in a city to serve the customers, and the control decisions are made based on the taxi demand forecasts for different locations. At time \(t\), the state \(x_t \in \mathbb{R}^n\) represents the difference between the number of free taxis and the number of waiting passengers at \(n\) locations, so \(x_t > 0\) means there are idle taxis, and \(x_t < 0\) means there are passengers waiting for taxis in the queue. The action \(u_t \in \mathbb{R}^m\) represents the number of taxis sent from \(m\) locations to \(n\) queues. In our experiment, \(n = m\), but they can be different in general.

Our system dynamics and control costs are the same as in Sec. II-C. The demand for taxis is a future timeseries \(S_t \in \mathbb{R}^p\), and the controller can only estimate it by forecasts \(\hat{S}_{t_{src}}\) from different sources \(i_{src} = 1, \ldots, n_{src}\). Here, each source \(i_{src}\) can be a firm, such as a cellular network operator, or simply a forecasting model which uses past \(w_{i_{src}}\) histories to forecast the next \(T\) steps of the future timeseries \(S\). The forecasts are generated by sensitive private data, thus the sources want to protect their data with privacy budgets \(\epsilon_{i_{src}}\) and \(\Sigma_{i_{src}}\) are the local sensitivity of the forecast, the privacy budget, and the variance of the forecasting error at source \(i_{src}\), respectively. \(D(0, \Sigma_{i_{src}})\) is any arbitrary distribution of random vectors with zero mean and variance \(\Sigma_{i_{src}}\). Here, we only assume the distributions of prediction errors to be zero mean distributions. It is a reasonable assumption since zero mean normal distributions are often used for errors estimations, and our assumption includes all zero mean normal distributions. Also, common regression models, such as linear regression models, have zero bias in their output estimations, thus resulting in zero mean distributions of errors.

In addition, we use the diminishing properties of logistic functions to capture the marginal effect of incentives \(\rho\) on privacy budgets \(\epsilon\). That is,
\[
\epsilon_{i_{src}}(\rho_{i_{src}}) = \frac{\alpha_{i_{src}}}{1 + e^{-\beta_{i_{src}}(\rho_{i_{src}} - \gamma_{i_{src}})}}, \quad \forall i_{src} = 1, \ldots, n_{src},
\] (10)
where \(\alpha_{i_{src}}\) is the maximum acceptable privacy budget, \(\beta_{i_{src}}\) is the increasing rate of privacy budget to incentive, and \(\gamma_{i_{src}}\) is the function’s center of symmetry. The logistic function ensures that \(\epsilon_{i_{src}}\) asymptotically approaches \(\alpha_{i_{src}}\) when \(\rho_{i_{src}}\) is infinitely large, so that the privacy is always preserved to a certain level with budget \(\epsilon_{i_{src}} < \alpha_{i_{src}}\).

### IV. Incentive Allocation for Differentially Private Forecasts

In this section, we use Eq. 7 to formulate the expected regret \(E[\Delta J^c]\) of the controller due to Laplacian noise and forecasting errors. We then formulate an incentive allocation problem minimizing the expected regret \(E[\Delta J^c]\). We denote full future control vectors in bold fonts. Specifically, \(\mathbf{u} = u_{0:T-1} \in \mathbb{R}^{mT}\), \(\mathbf{S} = S_{0:T-1} \in \mathbb{R}^{pT}\), and \(\mathbf{x} = x_{0:T} \in \mathbb{R}^{m(T+1)}\) for a finite time horizon \(T\).

The timeseries \(\hat{S}\) are forecasted by \(n_{src}\) sources, and we define \(\hat{S}_{t_{src}}\) as the timeseries sent to the controller from source \(i_{src}\) and \(\Delta \hat{S}\) as the overall forecasting error between the true timeseries \(S\) and the one observed by the controller \(\hat{S}\). Furthermore, forecasting error \(\Delta \hat{S}\) can be divided into differential private noise \(\Delta \hat{S}^{dp}\) and prediction error \(\Delta \hat{S}^{pc}\). The first one is caused by the fact that the differential privacy mechanism adds Laplacian noise to the original data, and the latter one is caused by the epistemic uncertainty of prediction. Hence, we define:
\[
\Delta \hat{S}_{t_{src}} := S + \Delta \hat{S}_{t_{src}}^{dp} + \Delta \hat{S}_{t_{src}}^{pc}, \quad \forall i_{src} = 1, \ldots, n_{src}.
\] (8)

We assume that all \(\Delta \hat{S}_{t_{src}}\) are independent, and that the controller knows the distributions of \(\Delta \hat{S}^{dp}\) and \(\Delta \hat{S}^{pc}\) but not the exact values. The distributions are given by:
\[
\begin{align*}
    \Delta \hat{S}_{t_{src}}^{dp} &\sim \text{Lap}(0, \frac{\text{Sen}(S_{t_{src}})}{\epsilon_{i_{src}}})^{pT}, \quad \forall i_{src} = 1, \ldots, n_{src}, \\
    \Delta \hat{S}_{t_{src}}^{pc} &\sim \mathcal{D}(0, \Sigma_{i_{src}}), \quad \forall i_{src} = 1, \ldots, n_{src},
\end{align*}
\] (9)
where \(\text{Lap}(\mu, \sigma)^{pT}\) is a random vector with independent and identically distributed Laplace random variable in each component, \(\text{Sen}(S_{t_{src}})\), \(\epsilon_{i_{src}}\), and \(\Sigma_{i_{src}}\) are the local sensitivity of the forecast, the privacy budget, and the variance of the forecasting error at source \(i_{src}\), respectively. \(\mathcal{D}(0, \Sigma_{i_{src}})\) is any arbitrary distribution of random vectors with zero mean and variance \(\Sigma_{i_{src}}\). Here, we only assume the distributions of prediction errors to be zero mean distributions. It is a reasonable assumption since zero mean normal distributions are often used for errors estimations, and our assumption includes all zero mean normal distributions. Also, common regression models, such as linear regression models, have zero bias in their output estimations, thus resulting in zero mean distributions of errors.
Since the error $\Delta \hat{S}$ is a random variable, by Eq. 7, the regret of the controller $\Delta J^c$ is also a random variable. Consequently, our goal is to allocate incentives to different sources in order to increase their privacy budget $\epsilon$ and minimize the expected regret $E[\Delta J^c]$. Note that in Eq. 7, the true control cost $J^c(u^*, S, x_0)$ is constant, so minimizing the regret is identical to minimizing the control cost. First, by Eq. 7, 8, 9, and 10, the expected regret can be simplified to:

$$E[(\hat{S}_{\text{src}} - S)^\top \Psi (\hat{S}_{\text{src}} - S)] = Tr(\Psi \Sigma_{\text{src}}) + 2 \left( \frac{1}{\alpha_{\text{src}}} \right)^2 Tr(\Psi \Sigma_{\text{est}}),$$

(11)

where $\Psi_n \in \mathbb{R}^{n \times n}$ is the identity matrix, and $Tr(\cdot)$ is the trace function. For simplicity, we denote

$$\sigma^2(\rho_{\text{src}}) = 2 \left( \frac{1}{\alpha_{\text{src}}} \right)^2.$$

Note that $\sigma^2(\rho_{\text{src}})$ is convex in $\rho_{\text{src}}$ since the square of an exponential function is convex. The controller can choose an element-wise linear combination of all $\hat{S}_{\text{src}}$ as the timeseries used for control $\hat{S}$:

$$\hat{S} = \sum_{i_{\text{src}} = 1}^{n_{\text{src}}} c_{i_{\text{src}}} \odot \hat{S}_{i_{\text{src}}}, \quad \text{s.t.} \sum_{i_{\text{src}} = 1}^{n_{\text{src}}} c_{i_{\text{src}}} = 1, \quad c_{i_{\text{src}}} \in \mathbb{R}^p_+, \forall i_{\text{src}} = 1, \ldots, n_{\text{src}},$$

(13)

where $\odot$ is the element-wise product, and $1$ is the vector of all-ones. Therefore, by Eq. 11, 12, and 13:

$$E[\Delta J^c] = \sum_{i_{\text{src}} = 1}^{n_{\text{src}}} c_{i_{\text{src}}}^\top \left[ A_{i_{\text{src}}}^\top + \sigma^2(\rho_{\text{src}}) \Sigma_{\text{est}} \right] c_{i_{\text{src}}},$$

(14)

By the Schur product theorem ([19, Theorem 7.5.3]), the element-wise product of two positive semidefinite matrices is also positive semidefinite, so $E[\Delta J^c]$ is convex in all $c_{i_{\text{src}}}$. Lastly, the controller allocates incentives to each source to minimize the expected regret $E[\Delta J^c]$:

$$\min_{c_1, \ldots, c_{n_{\text{src}}}} \sum_{i_{\text{src}} = 1}^{n_{\text{src}}} c_{i_{\text{src}}}^\top \left[ \Psi \otimes \left( \Sigma_{i_{\text{src}}} + \sigma^2(\rho_{\text{src}}) \Sigma_{\text{est}} \right) \right] c_{i_{\text{src}}}$$

(15a)

$$\text{s.t.} \sum_{i_{\text{src}} = 1}^{n_{\text{src}}} c_{i_{\text{src}}} = 1$$

(15b)

$$\sum_{i_{\text{src}} = 1}^{n_{\text{src}}} \rho_{i_{\text{src}}} = \rho$$

(15c)

$$\rho_{i_{\text{src}}} \geq 0, \forall i_{\text{src}} = 1, \ldots, n_{\text{src}},$$

(15d)

where $\rho$ is the total incentive that the controller can pay to all sources. Constraint 15b ensures coefficients of all sources sum to 1 (see Eq. 13), and constraint 15c ensures that the controller can only pay a total incentive of $\rho$ to all sources.

Algorithm 1 Alternate Convex Search (ACS)

Input: Arbitrary feasible point $(c_1^0, \rho_1^0, \ldots, c_{n_{\text{src}}}^0, \rho_{n_{\text{src}}}^0)$, and convergence criterion $\eta$

Output: Local optimal solution $(c_1^*, \rho_1^*, \ldots, c_{n_{\text{src}}}^*, \rho_{n_{\text{src}}}^*)$ and the expected regret $E[\Delta J^c]^*$

1: $i \leftarrow 0$
2: $E[\Delta J^c]^0 \leftarrow$ evaluate Eq.15a on $(c_1^0, \rho_1^0, \ldots, c_{n_{\text{src}}}^0, \rho_{n_{\text{src}}}^0)$
3: $E[\Delta J^c]^{-1} \leftarrow 0$
4: while $E[\Delta J^c]^0 - E[\Delta J^c]^{-1} \geq \eta$ do
5: $\rho_{i_{\text{src}}}^{i+1} \leftarrow \arg \min_{\rho_{i_{\text{src}}}^{i_{\text{src}}}}$ Eq.15 for fixed $(c_1^i, \ldots, c_{n_{\text{src}}}^i)$
6: $c_{i_{\text{src}}}^{i+1} \leftarrow \arg \min_{c_{i_{\text{src}}}^{i_{\text{src}}}}$ Eq.15 for fixed $(\rho_1^{i+1}, \ldots, \rho_{n_{\text{src}}}^{i+1})$
7: $E[\Delta J^c]^{-1} \leftarrow$ evaluate Eq.15a on $(c_1^{i+1}, \rho_1^{i+1}, \ldots, c_{n_{\text{src}}}^{i+1}, \rho_{n_{\text{src}}}^{i+1})$
8: $i \leftarrow i + 1$
9: end while
10: return $(c_1^*, \rho_1^*, \ldots, c_{n_{\text{src}}}^*, \rho_{n_{\text{src}}}^*)$ and $E[\Delta J^c]^i$

The last constraint 15d ensures that all coefficients $c_{i_{\text{src}}}$ and incentives $\rho_{i_{\text{src}}}$ are nonnegative. Eq. 15a is convex in $c_{i_{\text{src}}}$ when $\rho_{i_{\text{src}}}$ is fixed and convex in $\rho_{i_{\text{src}}}$ when $c_{i_{\text{src}}}$ is fixed. This property is called biconvex, describing a function is convex in two sets of variables independently while the other set is fixed. However, biconvex functions are not convex in both sets of variables simultaneously. So far, no algorithm has been found to obtain global solutions of biconvex problems, but it is easy to obtain a local one. [20] proposed a heuristic – Alternate Convex Search (ACS) to obtain local solutions and showed that if a biconvex function is bounded below in the feasible set, ACS will converge ([20, Theorem 4.5]). It is exactly the case here since we know the expected regret is quadratic. Therefore, for all $\hat{S}$:

$$E[\Delta J^c] = E[(\hat{S} - S)^\top \Psi (\hat{S} - S)] \geq 0,$$

and the feasible set is a subset of all $\hat{S}$.

The inputs of Alg. 1 are a convergence criterion $\eta$ and any feasible point of incentive $\rho_{i_{\text{src}}}$ and coefficient $c_{i_{\text{src}}}$. The criterion $\eta$ determines when to terminate, and the feasible point gives the algorithm a point to start with. It returns a local solution of Eq. 15 with the corresponding value. In lines 1 to 3, the algorithm first calculates the initial expected regret. Then, the while loop at line 4 iteratively searches for a local optimal solution. It first optimizes incentives $\rho_{i_{\text{src}}}$ with fixed coefficients $c_{i_{\text{src}}}$ at line 5 and then does the opposite at line 6. Lastly, the algorithm calculates the new expected regret at line 7 and sees if the difference of the 2 newest expected regrets is larger than criterion $\eta$. In practice, we can modify it to be a running average over the last $M$ measurements. If it is, the algorithm loops again; otherwise, the algorithm terminates and returns the newest solution and the expected regret.

Limitations: Our system model assumes that the training set and the testing set of forecasting models have similar distributions so that the covariances of forecasting errors are similar. The deviation of the testing distribution from the training distribution is called concept drift [21], [22], [23].
In our case, we use the covariance of training set to estimate the one of testing set. Our following numerical results show that the training and testing covariances are similar, so ACS performs well. However, when the two covariances are very different, that is, if concept drift occurs, the performance of ACS will degrade.

V. EXPERIMENTS

We implemented ACS, the method described in Alg. 1, to show that our theoretical analysis can effectively reduce the regret $\Delta J^c$. We compared our method to a naive heuristic, which uniformly allocates coefficient $c_{src}$ and incentives $\rho_{src}$ among all sources. We denote this method as Uniform. For Uniform, $c_{src} = 1/\rho_{src}$ and $\rho_{src} = \rho/\rho_{src}$ for all sources $i_{src}$. The initial input of $c_{0src}$ and $\rho_{0src}$ in Alg. 1 are the same as Uniform for all sources $i_{src}$.

We evaluated our methods on two timeseries datasets. The first one is a synthetic Autoregressive Integrated Moving Average (ARIMA) timeseries [24] with parameters $(p, d, q) = (0, 1, 1)$. The other data set is Uber Pickups in New York City [25], which describes the locations and time of Uber pickups in New York City. The control task of the Uber pickups is described in Sec. III. While we used Uber data to represent the demand of taxis. We used additional cost parameters $Q$ to penalize superfluous waiting queues or insufficient taxi supply at different locations, and used $R$ to represent the cost of sending taxis to different locations. We used the timeseries from April to July as the training set and August to September as the testing set for the forecasting models. We also discretized New York City into 4 regions and calculate their hourly pickup counts. See Fig 2 for the original spatial distribution of Uber pickups in New York City. We scaled all data to range(0,1) and then fit our forecasting models, so all sensitivities $\text{Sen}(S_{src})$ are 1. We used a timeseries forecasting package, tsai [26], to train neural network models for the forecasting task. We now describe the parameters of our control task. For ARIMA, $T = 10$, $x_0 = 1$, $n = m = p = 1$, $A = B = Q = R = 1$, and $C = -1$. For the Uber dataset, $T = 5$, $x_0 = 1$, $n = m = p = 4$, $A = B = Q = R = I_4$, and $C = -I_4$. See Table 1 for parameters of sources in both experiments.

We evaluate the performance of ACS and Uniform on two metrics: control regret $\Delta J^c$ and forecasting errors $\|S - \hat{S}\|_2$. Control regret $\Delta J^c$ is quadratic in forecasting errors $\|S - \hat{S}\|_2$ (Eq. 7), so they are positively correlated. We show the results of the ARIMA dataset in Fig. 3a and the results of the Uber dataset in Fig. 3b, respectively. For both datasets, we first show that the regret $\Delta J^c$ decreases as the total incentive $\rho$ increases. It is natural because all sources receive more incentive to lower their differential privacy noise levels. The control regret $\Delta J^c$ converges to a positive constant larger than 0 because the prediction errors are fixed and the differential privacy budget $\epsilon_{src}$ eventually converges to $\sigma_{src}$. The mean of coefficient $c$ is shown to emphasize how the controller weights the sources based on their prediction errors and the Laplace noise levels. When $\rho$ is small, all Laplace noise levels are similar since all sources start with similar $\epsilon$. The controller tends to weight source 1 more because its forecasts are more accurate. When $\rho$ increases, first the controller weights source 3 more because its $\beta_3$ is the largest, so its privacy budget $\epsilon_3$ increases rapidly to $\alpha_3$. Later, when $\rho$ is even larger, all sources’ $\epsilon$ approach $\alpha$, so the controller starts to weight source 3 less. Source 3 still has the largest weights since its maximum acceptable privacy budget $\alpha_3$ is the largest. The second row of Fig. 3a and Fig. 3b shows the true timeseries (black) and the forecasts by ACS (green) and Uniform (blue) when total incentive $\rho$ is 1 and 4. ACS is more accurate than Uniform, and when the total incentive is larger $\rho$, the forecasts are also more accurate. We show the distributions of forecasting errors under different total incentives $\rho$ in the right plot of the 2nd row to emphasize ACS effectively reduces forecasting errors and thus reduces the control regret.

Our results confirm that ACS can effectively reduce control regret due to imperfect forecasts, and it reduces the expected regrets by 2.5 and 2.7 times compared to the other benchmark, Uniform. Also, in Fig. 3, all coefficients $c$ are non-binary, meaning that the controller is combining forecasts from different sources. Intuitively, since all sources’ prediction errors and Laplacian noise are independent, combining them results in 0 covariance. Hence, it helps to minimize the regret.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we use the intrinsic prediction errors and privacy preferences of different sources of forecasts to formulate a timeseries forecast selection problem for LQR. We
use the diminishing properties of logistic functions to capture the marginal effect of incentives $\rho$ on privacy budgets $\epsilon$. We then obtain the local optimum of the proposed biconvex optimization problem by Alternative Convex Search (ACS). We use linear regression models to forecast ARIMA timeseries and state-of-the-art neural network forecasters to forecast real-world timeseries from Uber. Numerical results show that ACS reduces the expected regrets by 2.5 and 2.7 times compared to the other benchmark, Uniform.

In this work, we assume that the private feature data are independent for each prediction period, so that multiple forecasts at different time steps do not cause privacy leakage. [30] proposed a way to allocate privacy budgets among different queries at different time steps to discard this assumption. However, the optimal allocation of privacy budgets $\epsilon$ among different time steps for control remains an open problem. We also use the Laplace mechanism to protect private feature data of forecasts. Nevertheless, the

Fig. 3: ACS outperforms Uniform in forecasting errors and control regrets Fig. 3a and Fig. 3b show the results of ARIMA and the Uber demand forecasts. (Row 1) The first 2 plots on the left in row 1 show the expected control regret (in curves) and actual distributions (in boxplots) of ACS (green) and Uniform (blue) versus total incentive of the controller $\rho$. The control regret $\Delta J^c$ decreases as the total incentive $\rho$ increases since incentives reduce the Laplacian noise levels. However, since the trade-off of privacy and incentive is a diminishing function (Eq. 10), the control regret converges asymptotically. The right plot shows the mean of coefficient $c$ of all sources versus total incentive $\rho$. When $\rho$ is small, the Laplace noise levels are similar among sources so the controller weights the most accurate source more. When $\rho$ is bigger, the effect of Laplace noise levels is more significant than prediction errors, so the controller tends to choose the source with the largest maximum privacy budget $\alpha$. Note that the lines in the boxes represent the medians of the distributions, while the curves of regrets are the expected value. (Row 2) The first 2 plots show the true timeseries (black) and the forecasts by ACS (green) and Uniform (blue) for 2 total incentives $\rho$, 1 and 4. ACS is more accurate than Uniform, which confirms why the control regret of ACS is better. Note that these 2 plots have different scales in the y-axis. Again, the larger $\rho$, the more accurate the forecasts. We show the distributions of forecasting errors under different total incentives $\rho$ in the right plot to emphasize our point. We only scaled the timeseries back to the real values when comparing to ground true timeseries; the other plots are calculated with timeseries scaled to range $(0, 1)$. The time series of Uber pickups are plotted in 48-hour increments, so one can clearly see the 2 cycles in the plots.

Fig. 3: ACS outperforms Uniform in forecasting errors and control regrets Fig. 3a and Fig. 3b show the results of ARIMA and the Uber demand forecasts. (Row 1) The first 2 plots on the left in row 1 show the expected control regret (in curves) and actual distributions (in boxplots) of ACS (green) and Uniform (blue) versus total incentive of the controller $\rho$. The control regret $\Delta J^c$ decreases as the total incentive $\rho$ increases since incentives reduce the Laplacian noise levels. However, since the trade-off of privacy and incentive is a diminishing function (Eq. 10), the control regret converges asymptotically. The right plot shows the mean of coefficient $c$ of all sources versus total incentive $\rho$. When $\rho$ is small, the Laplace noise levels are similar among sources so the controller weights the most accurate source more. When $\rho$ is bigger, the effect of Laplace noise levels is more significant than prediction errors, so the controller tends to choose the source with the largest maximum privacy budget $\alpha$. Note that the lines in the boxes represent the medians of the distributions, while the curves of regrets are the expected value. (Row 2) The first 2 plots show the true timeseries (black) and the forecasts by ACS (green) and Uniform (blue) for 2 total incentives $\rho$, 1 and 4. ACS is more accurate than Uniform, which confirms why the control regret of ACS is better. Note that these 2 plots have different scales in the y-axis. Again, the larger $\rho$, the more accurate the forecasts. We show the distributions of forecasting errors under different total incentives $\rho$ in the right plot to emphasize our point. We only scaled the timeseries back to the real values when comparing to ground true timeseries; the other plots are calculated with timeseries scaled to range $(0, 1)$. The time series of Uber pickups are plotted in 48-hour increments, so one can clearly see the 2 cycles in the plots.

(a) ARIMA timeseries dataset.

(b) Uber timeseries dataset.
information of forecasting models (e.g. parameters) is not protected. Hence, another open problem is how to protect information about the forecasting models’ architectures and parameters.

ACKNOWLEDGEMENT

This work was supported in part by the grants ARL ACC-APG-RTP W911NF1920333, ONR N00014-21-1-2502, ARO W911NF-20-1-0140, and National Science Foundation grant No. 2133481. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] J. Cheng, M. Pavone, S. Katti, S. Chinchali, and A. Tang, “Data sharing and compression for cooperative networked control,” Advances in Neural Information Processing Systems, vol. 34, pp. 5947–5958, 2021.

[2] P. Doni, B. Amos, and J. Z. Kolter, “Task-based end-to-end model learning in stochastic optimization,” in Advances in Neural Information Processing Systems, vol. 30, Curran Associates, Inc., 2017.

[3] S. Chinchali, P. Hu, T. Chu, M. Sharma, M. Bansal, R. Misra, M. Pavone, and S. Katti, “Cellular network traffic scheduling with deep reinforcement learning,” in Thirty-second AAAI conference on artificial intelligence, 2018.

[4] P. Han Li, U. Topcu, and S. P. Chinchali, “Adversarial examples for model-based control: A sensitivity analysis,” 2022.

[5] C. Dwork, “Differential privacy,” in Automata, Languages and Programming (M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds.), (Berlin, Heidelberg), pp. 1–12, Springer Berlin Heidelberg, 2006.

[6] Y. Wang, Z. Huang, S. Mitra, and G. E. Dullerud, “Differential privacy in linear distributed control systems: Entropy minimizing mechanisms and performance tradeoffs,” IEEE Transactions on Control of Network Systems, vol. 4, no. 1, pp. 118–130, 2017.

[7] J. He and L. Cai, “Differential private noise adding mechanism: Basic conditions and its application,” in 2017 American Control Conference (ACC), pp. 1673–1678, 2017.

[8] J. Cheng, A. Tang, and S. Chinchali, “Task-aware privacy preservation for multi-dimensional data,” in International Conference on Machine Learning, pp. 3835–3851, PMLR, 2022.

[9] M. U. Hassan, M. Husain Rehmani, and J. Chen, “Differential privacy techniques for cyber physical systems: A survey,” arXiv e-prints, pp. arXiv–1812, 2018.

[10] V. Pacelli and A. Majumdar, “Robust control under uncertainty via bounded rationality and differential privacy,” in 2022 International Conference on Robotics and Automation (ICRA), pp. 3467–3474, 2022.

[11] Z. Wang, J. Li, J. Hu, J. Ren, Q. Wang, Z. Li, and Y. Li, “Towards privacy-driven truthful incentives for mobile crowdsensing under untrusted platform,” IEEE Transactions on Mobile Computing, vol. 22, no. 2, pp. 1198–1212, 2023.

[12] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, “Enabling privacy-preserving incentives for mobile crowd sensing systems,” in 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), pp. 344–353, 2016.

[13] W. Tang, J. Ren, K. Deng, and Y. Zhang, “Secure data aggregation of lightweight e-healthcare iot devices with fair incentives,” IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8714–8726, 2019.

[14] M. Kearns, M. Pai, A. Roth, and J. Ullman, “Mechanism design in large games: Incentives and privacy,” in Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, ITCS ’14, (New York, NY, USA), p. 403–410, Association for Computing Machinery, 2014.

[15] J. Lee and C. Clifton, “How much is enough? choosing e for differential privacy,” in Proceedings of the 14th International Conference on Information Security, ISC’11, (Berlin, Heidelberg), p. 325–340, Springer-Verlag, 2011.

[16] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, “Privacy at scale: Local differential privacy in practice,” in Proceedings of the 2018 International Conference on Management of Data, SIGMOD ’18, (New York, NY, USA), p. 1655–1658, Association for Computing Machinery, 2018.

[17] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” in Proceedings of the Third Conference on Theory of Cryptography, TCC’06, (Berlin, Heidelberg), p. 265–284, Springer-Verlag, 2006.

[18] C. Dwork and A. Roth, The Algorithmic Foundations of Differential Privacy, 2014.

[19] R. A. Horn and C. R. Johnson, Matrix Analysis. Cambridge University Press, 1990.

[20] J. Gorski, F. Pfeuffer, and K. Klamroth, “Biconvex sets and optimization with biconvex functions: A survey and extensions,” Mathematical Methods of Operations Research, vol. 66, pp. 373–407, 11 2007.

[21] I. Zhiobate, M. Pechenizkiy, and J. Gama, An Overview of Concept Drift Applications, pp. 91–114. Cham: Springer International Publishing, 2016.

[22] J. a. Gama, I. Zliobait unde defined, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” ACM Comput. Surv., vol. 46, mar 2014.

[23] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, “Learning under concept drift: A review,” IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 12, pp. 2346–2363, 2019.

[24] A. C. Harvey, ARIMA Models, pp. 22–24. London: Palgrave Macmillan UK, 1990.

[25] “Uber pickups in new york city.” https://www.kaggle.com/datasets/fivethirtyeight/uber-pickups-in-new-york-city, 2019. [Online; accessed 09-Sept-2021].

[26] I. Oguiza, “tsai - a state-of-the-art deep learning library for time series and sequential data.” Github, 2022.

[27] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A transformer-based framework for multivariate time series representation learning,” in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2114–2124, 2021.

[28] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, “Inceptiontime: Finding alexnet for time series classification,” Data Mining and Knowledge Discovery, 2020.

[29] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in NIPS Workshop on Deep Learning, 2014.

[30] K. Kenthapadi and T. T. L. Tran, “Pripearl,” Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Oct 2018.