VAP: Online Data Valuation and Pricing for Machine Learning Models in Mobile Health

Anran Xu, Zhenzhe Zheng, *Member, IEEE,* Qinya Li, Fan Wu, *Member, IEEE,* and Guihai Chen, *Fellow, IEEE*

Abstract—Mobile health (mHealth) applications, benefiting from mobile computing, have generated numerous mHealth data. However, they are dispersed across isolated devices, which hinders discovering insights underlying the aggregated data. Considering the online characteristics of mHealth, in this work, we present the first online data VAluation and Pricing mechanism, namely VAP, to incentive users to contribute mHealth data for machine learning (ML) tasks in mHealth systems. Under the Bayesian framework, we propose a new metric based on the concept of entropy to calculate data valuation during model training in an online manner. In proportion to the data valuation, we then determine payments as compensations for users to contribute their data. We formulate this pricing problem as a contextual multi-armed bandit with the goal of profit maximization and propose a new algorithm based on the characteristics of pricing. Furthermore, to tackle the budget constraint, we incorporate a two-stage multi-armed bandit with a knapsack method. We also extend VAP to advanced ML models by computing the entropy on the prediction space. Finally, we have evaluated VAP on two real-world mHealth data sets. Evaluation results show that VAP outperforms the state-of-the-art data valuation and pricing mechanisms in terms of computational complexity and extracted profit.

Index Terms—Data Valuation, Online Pricing, Mobile Health

1 Introduction

Mobile health (mHealth) technologies offer real-time monitoring for health status, facilitate rapid diagnosis of potential health issues, and provide remote healthcare services [1]. The recent developments towards intelligent mHealth systems, such as Apple Health [2], Google Fit [3], and Microsoft Health [4] are pieces of evidence of these trends [5]. Various machine learning (ML) models have been developed to extract information underlying mHealth data [5], [6], [7]. However, the obstacle to the wide adoption of ML in mHealth applications comes from *model uncertainty* [8], which would provide unreliable prediction and is unacceptable in health applications [9]. The uncertainty of the model parameters often comes from insufficient training data and can be eliminated by acquiring enough data [8]. In mHealth, reducing the model uncertainty and accurately predicting a phenotype depends upon using a large amount of data from many other individuals with similar or related diseases. One potential approach to eliminate this dilemma is to collect extensive mHealth data from users as training data, harnessing the wisdom of crowd [10]. Thus, the further development of mHealth should have the ability to incentive users of mHealth services to contribute their data into the system to support ML models' training.

The valuable mHealth data are dispersed across isolated devices and have not been exploited efficiently. Users are reluctant to voluntarily share their personal health data due to the potential incurred costs and privacy concerns [11]. Health information privacy is the right of individuals to control the access, use, or disclosure of their identifiable health data [12]. And people's agreement to share their data usually revolves around the value, which refers to the benefit that is accrued to the user or society due to the use of data [10]. Therefore, it is highly necessary to design an incentive mechanism to stimulate users to contribute their mHealth data. For incentive mechanism design in mHealth, we need to take the online characteristics of the data acquisition into account. First, the sensing data collected by mHealth can be obtained remotely in a streaming manner, which is often used for real-time predictive modeling [13]. Second, within the changing mHealth contexts, traditional static mHealth models may fail to respond with a correct prediction result. For example, people may carry out the same activity in a different manner or suffer from the same disease with various clinical symptoms [14]. Furthermore, population demographics, the prevalence of disease, and clinical practice may also evolve over time. This implies that predictions based on static data and models can become outdated and hence no longer be accurate [15]. Last, the users' participation in the data acquisition process is dynamic. For example, in disease detection, the symptoms appear at an unpredictable time. To address these dynamics, many variants of online learning and incremental learning models are proposed [14], [15], [16], [17]. With these methods, the mHealth models could update over time as new data is collected and adapt quickly to new contexts. Besides, compared to the method that works with the sample pool once and for all, an online manner will not increase the permutation complexity.

There are two critical components in designing an incentive mechanism: *data valuation and data pricing*. The data valuation scheme quantifies the contribution of data within the context of ML model training. Based on this data valuation metric, the data pricing mechanism determines the compensation to users for their contributed data. We next summarize two major challenges for data valuation and pricing arising from the online characteristics of the data acquisition process in mHealth.

The first challenge is to evaluate the contribution of newly arrived data in ML model training. The traditional data valuation schemes [18], [19], [20], [21], [22], built upon the concept

• *A. Xu, Z. Zheng, Q. Li, F. Wu, and G. Chen are with the Department of Computer Science and Engineering, Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China. (E-mail: {xuanran,zhengzhenzhe,qinyali}@sjtu.edu.cn, {fwu,gchen}@cs.sjtu.edu.cn).*

of Shapley value from cooperative game theory [23], are not suitable for such an online learning situation. In these methods, all the data are collected in advance for model training, and the data contribution is evaluated at the end of model training. In contrast, we need to measure the data valuation in an online manner, based on the currently collected data, instead of the complete training data set. However, it is difficult to infer the data valuation at the intermediate model training without the global knowledge of the whole data set. Moreover, considering the privacy concerns in mHealth, compared to submitting whole data, it is more proper that users only upload part of the data (such as the feature of the data sample but not the label of it) to evaluate the data and query the price. Therefore, the data valuation module should have the ability to estimate the data contribution based on such kind of incomplete data.

The second challenge is on designing profit-maximizing data pricing mechanisms within an asymmetric information environment. Some auction-based mechanisms have been proposed for data pricing [18], [24], [25]. However, the bidding model in the auction is unnecessarily complicated for data pricing, as users may often be reluctant to provide the minimum willing payment for their data or even do not know the exact value of this information. To this end, we turn to the posted pricing mechanism [26], where the service provider posts a public price, and the users only need to determine whether to accept the price and contribute the data. Nevertheless, the posted pricing mechanism introduces a heavy burden on the service provider. There is an information asymmetry over the minimum payment to data between the users and the service provider. Furthermore, users' arrival sequences are also unknown to the service provider. Without complete information about the payment to data, it is hard for the service provider to set an appropriate price. The optimization of profit maximization needs to take both the revenue extracted from data valuation and the expenditure for data acquisition into account. In addition, there may be budget constraints in the system, maximizing the profit within a limited budget inevitably doubles the difficulty in the design of data pricing mechanisms.

In this work, jointly considering the above challenges, we propose the first online data valuation and pricing mechanism for ML tasks in mHealth, namely VAP. We summarize our contributions as follows.

• Firstly, we introduce a novel metric for data valuation under the Bayesian perspective for Bayesian linear regression. This metric gauges the influence of data on the machine learning model training procedure. It is quantified by evaluating the entropy of the distributions over model parameters, permitting us to appraise data value in an online manner, eradicating the necessity for complete dataset collection. Furthermore, we enhance this data valuation metric from Bayesian linear regression to more intricate machine learning models by transitioning entropy computation from the parameter space to the prediction space.

• Secondly, we present an online data pricing mechanism that incorporates both data valuation and users' reserve values. We formulate the determination of payments as a contextual multi-armed bandit (MAB) problem, aiming to maximize profit and propose a novel method for data pricing within this framework. Additionally, when facing the challenge of budget constraints in more complex scenarios, we model it as a two-stage multi-armed bandit problem with a knapsack and devise a solution. In both cases, a dual process of exploration and exploitation

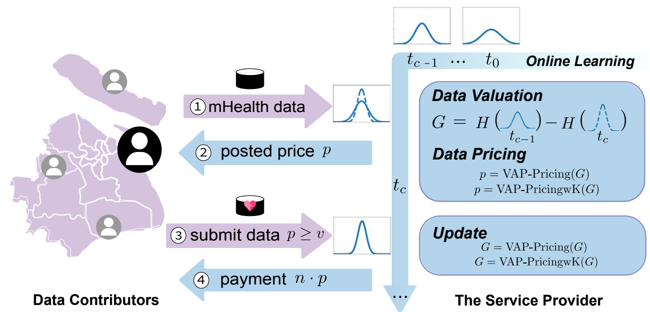


Fig. 1. Data acquisition process in a mHealth system.

is employed to pinpoint optimal data prices, informed by user responses across diverse price points. Appreciating the inherent monotonicity of pricing, we exploit an expanded user feedback base, thereby securing a more profitable endeavor.

• Finally, we assess VAP's performance utilizing two authentic mHealth datasets. The assessment results underline that our VAP holds supremacy over contemporaneous data valuation and pricing mechanisms for online Machine Learning tasks within mHealth frameworks, in terms of computational complexity and profit extraction.

A preliminary version of this work [27] was published in INFOCOM 2022, which only proposed an individual pricing method without a theoretical guarantee. In this work, we add necessary proofs and property comparisons with traditional methods for data valuation. As for online data pricing, we add the regret analysis of VAP-Pricing and substantially extend the data pricing problem to a new situation under a fixed limited budget, and propose a new algorithm, namely VAP-PricingwK. We also add some experiments to validate the newly proposed method.

The structure of this paper unfolds as follows: In Section 2, we present the system model and problem formulation. Subsequently, in Section 3, we propose an online data valuation metric based on the concept of entropy and delve into some of its characteristics. Moving forward, Section 4 is dedicated to the design of two distinct data pricing algorithms - one framed within the contextual Multi-Armed Bandit (MAB) schema, and the other following the Bandit with a Knapsack method, subject to a budget constraint. While in Section 5 we elaborate on incorporating VAP with more articulated machine learning models. The results of our performance evaluations take the stand in Section 6, followed by a review of related works in Section 7. Finally, we articulate the conclusion of our study in Section 8.

2 Preliminaries

We consider the data acquisition process for a mHealth system in an adaptive way, as shown in Fig. 1. There are two types of participants involved in a mHealth system: data contributors and a service provider. The service provider trains online ML models on the collected data from data contributors to provide healthcare services. Due to the limited amount of data and the fading freshness of historical data, the ML models' performance would decay over time. The service provider needs to acquire new mHealth data periodically to retrain the ML models. A specific data acquisition process is conducted as follows. At the time slot t, first, a data contributor arrives and queries the price of her data by submitting the training data $\boldsymbol{x}$ without the label y, where the feature $\boldsymbol{x}$ would help the service provider to evaluate the data valuation, and not releasing the label y would preserve the content of data before the data exchange. Second, the service provider evaluates the data based on its contribution to ML model training, calculated by the performance improvement between the current model and the updated model after the data is added. Based on the *data valuation*, the service provider posts the price determined by the *data pricing mechanism* to the data contributor as incentives. Third, if the data contributor is satisfied with the price, she would contribute the complete training data $(\boldsymbol{x}, y)$. Otherwise, she has no incentives to do so. Having received the data from multiple data contributors, the service provider would update the ML model, data valuation metric, and data pricing mechanism. Finally, the service provider gives the corresponding payment to the data contributor. We need to design an appropriate data valuation metric and a data pricing mechanism to quantify the performance improvement for model training and make a trade-off between the performance and data acquisition expenditure.

We present a system model to describe the above data acquisition process. Each data contributor owns a set of private mHealth training data, each of which is a pair of a feature and the corresponding label, denoted by $d = (\boldsymbol{x}, y)$. We use $G_{\boldsymbol{X}}(\boldsymbol{x})$ to denote the contribution of a new data sample $d = (\boldsymbol{x}, y)$ to the model training. We consider each data contributor has a *reserve value* v to her data set, which indicates the minimum unit willing price the data contributor would like to share her data. Similar to the previous work [24], [28], all data contributors' reserve values follow an independent and identical distribution with probability density function $f(v)$ over the range $[0, 1]$. Different from the classical Bayesian mechanism design [29], the probability density function is unknown to the service provider and needs to be learned from the interaction with data contributors. When one data contributor arrives at the online mHealth system, the service provider posts a unit price of p for purchasing each piece of data. If the data contributor accepts the offered price (*i.e.* $p \geq v$), she would upload her data and get the corresponding payment; otherwise ($0 \leq p < v$), she would leave without contributing her data. The goal of the data pricing mechanism is to determine the posted price p at each time slot to maximize the total profit, which will be defined in Section 4 later.

3 Data Valuation

3.1 A Simple Case: Bayesian Linear Regression

To illustrate the idea of data valuation, we first consider a basic model in ML, linear regression [30] under the Bayesian framework. In mHealth, linear regression models are widely used in heart rate monitoring [17], blood pressure monitoring [31], mental illness detection [32], etc. More specifically, we use the ridge regression model as an example in this subsection and extend the concept of data valuation to more complex models such as Gaussian process (GP) [33] and Bayesian neural networks [34] later.

Ridge regression can be explained under a Bayesian framework as a type of Bayesian linear regression [35], in which maximizing the parameter's posterior probability by the Bayesian formula is the same as minimizing the loss function in the traditional frequentist view. Without loss of generality, we assume the prior probability of the parameters in ridge regression satisfy Gaussian distribution, *i.e.* $P(\boldsymbol{\beta}) \sim \mathcal{N}\left(0, \ell^2 \mathrm{I}\right)$ with precision parameter (variance) ℓ^2. The training process of ridge regression is to use new data to obtain posterior parameter distribution. Thus, the Bayesian framework provides a new perspective to interpret the model training process: the change of posterior parameter distribution can represent the evolution of the model training process to some extent. To calculate this change, we first express the posterior probability of the model parameter $\boldsymbol{\beta}$ from the Bayesian theorem:

$$P(\boldsymbol{\beta}|\boldsymbol{Y}) = \frac{P(\boldsymbol{Y}|\boldsymbol{\beta})P(\boldsymbol{\beta})}{P(\boldsymbol{Y})} \propto P(\boldsymbol{Y}|\boldsymbol{\beta})P(\boldsymbol{\beta}), \tag{1}$$

where $\boldsymbol{Y}$ is the corresponding label of the data set $(\boldsymbol{X}, \boldsymbol{Y})$, and $P(\boldsymbol{Y}|\boldsymbol{\beta})$ is the generation probability of $\boldsymbol{Y}$ under the model parameter $\boldsymbol{\beta}$, and follows the Gaussian distribution. As the product of two Gaussian distributions $P(\boldsymbol{Y}|\boldsymbol{\beta})P(\boldsymbol{\beta})$ is still Gaussian, the posterior parameter distribution $P(\boldsymbol{\beta}|\boldsymbol{Y})$ follows a Gaussian distribution. We denote the corresponding mean as $\bar{\boldsymbol{\beta}}$, and the variance as Σ. In this posterior Gaussian distribution, the exponential power should be equal. So that

$$\left(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}\right)^{\Sigma^{-1}} \left(\boldsymbol{\beta} - \bar{\boldsymbol{\beta}}\right) = \frac{1}{\gamma^2}(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^\top(\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \frac{1}{\ell^2}\boldsymbol{\beta}^\top\boldsymbol{\beta}. \tag{2}$$

Deriving from Equation (2), by equal coefficients of the same order, we can get:

$$\begin{aligned} \boldsymbol{\beta}^\top \Sigma^{-1} \boldsymbol{\beta} &= \boldsymbol{\beta}^\top \left(\frac{\boldsymbol{X}^\top \boldsymbol{X}}{\gamma^2} + \frac{\mathbf{I}}{\ell^2} \right) \boldsymbol{\beta} \\ \Sigma^{-1} &= \frac{\boldsymbol{X}^\top \boldsymbol{X}}{\gamma^2} + \frac{I}{\ell^2}, \end{aligned} \tag{3}$$

$$\begin{aligned} -2\bar{\boldsymbol{\beta}}^\top \Sigma^{-1} \boldsymbol{\beta} &= -2\frac{\boldsymbol{\beta}^\top \boldsymbol{X}^\top \boldsymbol{Y}}{\gamma^2} \\ \bar{\boldsymbol{\beta}} &= \left(\boldsymbol{X}^\top \boldsymbol{X} + \frac{\ell^2}{\gamma^2}\mathbf{I} \right)^{-1} \boldsymbol{X}^\top \boldsymbol{Y}. \end{aligned} \tag{4}$$

Thus, $P(\boldsymbol{\beta}|\boldsymbol{Y})$ follows a Gaussian distribution with the mean and the variance of $\bar{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top \boldsymbol{X} + \frac{\ell^2}{\gamma^2}\mathbf{I}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$ and $\Sigma = \left(\frac{1}{\gamma^2}\boldsymbol{X}^\top \boldsymbol{X} + \frac{1}{\ell^2}\mathbf{I}\right)^{-1}$, respectively.

We regard the data's contribution as how much information the data samples provide to the model training process. We use the metric of differential entropy [36], a concept from information theory, to measure the information contained underlying the corresponding model. When a new data sample is added to the training set, the parameter distribution shrinks, implying the reduction of the model parameters' uncertainty. We quantify this uncertainty reduction as the differential entropy of the prior parameter distribution and the posterior parameter distribution. We further use the extent of this reduction to measure the contribution of a data sample to the model training. The differential entropy of the model parameter distribution (a Gaussian distribution) is defined as:

$$H(\boldsymbol{\beta}) = \frac{1}{2} \ln \left[(2\pi e)^n [\Sigma] \right], \tag{5}$$

which is only related to variance Σ of the current distribution. And for the parameters $\boldsymbol{\beta}$ of one model, the parameter distribution depends on the collected training data. Thus, we denote the differential entropy of parameter distribution $H(\boldsymbol{\beta}|\boldsymbol{Y})$ on the data set $(\boldsymbol{X}, \boldsymbol{Y})$ as $H(\boldsymbol{X})$, then the differential entropy of parameter distribution with the training data set $(\boldsymbol{X}, \boldsymbol{Y})$ can be calculated by:

$$\begin{aligned} H(\boldsymbol{X}) &= \frac{1}{2}\ln\left((2\pi e)^d \det\left(\Sigma_{\boldsymbol{X}}\right)\right) \\ &= \frac{d}{2}\ln 2\pi e + \frac{1}{2}\ln\det(\Sigma_{\boldsymbol{X}}), \end{aligned} \tag{6}$$

After adding new data sample $(\boldsymbol{x}, y)$, the differential entropy of the parameter distribution is updated to

$$\begin{aligned} H(\boldsymbol{X}+\boldsymbol{x}) &= \frac{1}{2}\ln\left((2\pi e)^d \det\left(\left(\Sigma_{\boldsymbol{X}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right)^{-1}\right)\right) \\ &= \frac{d}{2}\ln 2\pi e - \frac{1}{2}\ln\det\left(\Sigma_{\boldsymbol{X}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right). \end{aligned} \tag{7}$$

The posterior entropy reduction of the model parameter distribution is

$$\begin{aligned} G_{\boldsymbol{X}}(\boldsymbol{x}) &= H(\boldsymbol{X}) - H(\boldsymbol{X}+\boldsymbol{x}) \\ &= \frac{1}{2}\ln\frac{\det\left(\Sigma_{\boldsymbol{X}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right)}{\det(\Sigma_{\boldsymbol{X}}^{-1})} \\ &= \frac{1}{2}\ln\det\left(\mathbf{I} + \boldsymbol{x}\boldsymbol{x}^\top\Sigma_{\boldsymbol{X}}\right) \\ &= \frac{1}{2}\ln\left(1 + \boldsymbol{x}^\top\Sigma_{\boldsymbol{X}}\boldsymbol{x}\right). \end{aligned} \tag{8}$$

Definition 1. *VAP-Valuation: The data valuation of data sample $(\boldsymbol{x}, y)$ for the model with data set $(\boldsymbol{X}, \boldsymbol{Y})$ is measured by $G_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{2}\ln\left(1 + \boldsymbol{x}^\top\Sigma_{\boldsymbol{X}}\boldsymbol{x}\right)$.*

Thus, we can use $G_{\boldsymbol{X}}(\boldsymbol{x})$ to calculate the valuation of data $\boldsymbol{x}$, by measuring the marginal contribution that the data $\boldsymbol{x}$ will make to the model that already has been trained by the data $\boldsymbol{X}$.

Considering the importance of reducing uncertainty in mHealth, we emphasize the relationship between $G_{\boldsymbol{X}}(\boldsymbol{x})$ and traditional predictive uncertainty. In Bayesian linear regression, in prediction space, for a new set of features $\boldsymbol{x}$ to be predicted, the predictive distribution takes the form $P(y|\boldsymbol{x}, \boldsymbol{\beta}) = \mathcal{N}\left(\boldsymbol{x}|\boldsymbol{\beta}^\top\boldsymbol{x}, \sigma_N^2(\boldsymbol{x})\right)$. A classic conclusion is that the predictive uncertainty can be determined by the variance $\sigma_N^2(\boldsymbol{x})$ of the predictive distribution, which is given by

$$\sigma_N^2(\boldsymbol{x}) = \sigma^2 + \boldsymbol{x}^\top\Sigma_X\boldsymbol{x}. \tag{9}$$

The first term represents the inherent noise in the generation of data, whereas the second term can reflect the uncertainty associated with the parameter $\boldsymbol{\beta}$. We can notice that the determinants of a and b are the same, *i.e.*, $\boldsymbol{x}^\top\Sigma_X\boldsymbol{x}$. We will discuss the comparison between them in detail in Section 5. This important discovery will play an important role in our later extension of VAP-Valuation to more advanced models.

3.2 Properties of Data Valuation Metric

Compared with traditional data valuation methods in ML such as Shapley value [18], [19], [20], VAP-Valuation has the following characteristics:

Submodularity:

Definition 2. *([37]) Let Ω be a finite ground set and $f : 2^\Omega \to \mathbb{R}$. Then f is submodular if for all $\mathcal{S}, \mathcal{T} \subseteq \Omega$ with $\mathcal{S} \subseteq \mathcal{T}$ and every $x \in \Omega\backslash\mathcal{T}$,*

$$f(\mathcal{S}\cup\{x\}) - f(\mathcal{S}) \geq f(\mathcal{T}\cup\{x\}) - f(\mathcal{T})$$

For any data sets $\mathcal{S}, \mathcal{T}$ *s.t.* $\mathcal{S} \subseteq \mathcal{T}$ we define the set $\mathcal{U} = \mathcal{T} - \mathcal{S}$. We use $\boldsymbol{U}, \boldsymbol{T}, \boldsymbol{S}$ to denote the features of the data samples in the set of $\mathcal{U}$, $\mathcal{T}$ and $\mathcal{S}$:

$$\begin{aligned} \Delta &= G_{\boldsymbol{S}}(\boldsymbol{x}) - G_{\boldsymbol{T}}(\boldsymbol{x}) \\ &= \frac{1}{2}\ln\frac{\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right)\det\Sigma_{\boldsymbol{S}}}{\det\left(\Sigma_{\boldsymbol{T}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right)\det\Sigma_{\boldsymbol{T}}} \\ &= \frac{1}{2}\ln\frac{\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top\right)\det\Sigma_{\boldsymbol{S}}\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{U}^\top\boldsymbol{U}\right)}{\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{x}\boldsymbol{x}^\top\right)} \\ &= \frac{1}{2}\ln\frac{\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{x}\boldsymbol{x}^\top + \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{x}\boldsymbol{x}^\top\Sigma_{\boldsymbol{S}}\boldsymbol{U}^\top\boldsymbol{U}\right)}{\det\left(\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{x}\boldsymbol{x}^\top\right)} \\ &= \frac{1}{2}\ln\det\left(\mathbf{I} + \frac{\boldsymbol{x}\boldsymbol{x}^\top\Sigma_{\boldsymbol{S}}\boldsymbol{U}^\top\boldsymbol{U}}{\Sigma_{\boldsymbol{S}}^{-1} + \boldsymbol{U}^\top\boldsymbol{U} + \boldsymbol{x}\boldsymbol{x}^\top}\right) \\ &> 0. \end{aligned} \tag{10}$$

Thus, we can get $G_{\boldsymbol{S}}(\boldsymbol{x}) > G_{\boldsymbol{T}}(\boldsymbol{x})$, which means the data valuation metric $G_{\boldsymbol{X}}(\boldsymbol{x})$ we proposed is submodular. A more intuitive understanding is that the marginal contribution of the data diminishes with the size of the data training set, which means that for the same data sample, the earlier the data is submitted, the higher the contribution generated.

Additivity: For a collection of data sets submitted by a data contributor within a certain period, the total valuation of all the data (*i.e.*, the total entropy reduction of the model parameter distribution) is the sum of the individual valuation of each data set. It is independent of the internal order of the data sets. That is, the data valuation metric is a set function: Owning $(\boldsymbol{X}, \boldsymbol{Y})$, for any new data set $\mathcal{S}$, using $G(\mathcal{S})$ to denote the data valuation of data set $\mathcal{S}$, calculated by the features $\boldsymbol{S}$ of data in $\mathcal{S}$, it is a fixed value:

$$\begin{aligned} G(\mathcal{S}) = G_{\boldsymbol{X}}(\boldsymbol{S}) &= H(\boldsymbol{X}) - H(\boldsymbol{X}+\boldsymbol{S}) \\ &= \frac{1}{2}\ln\det\left(\Sigma_{\boldsymbol{X}}^{-1} + \boldsymbol{S}^\top\boldsymbol{S}\right)\det(\Sigma_{\boldsymbol{X}}) \\ &= \frac{1}{2}\ln\det\left(\mathbf{I} + \boldsymbol{S}^\top\boldsymbol{S}\Sigma_{\boldsymbol{X}}\right). \end{aligned} \tag{11}$$

More specifically, $G(\mathcal{S}) = \sum_{s_i\in\mathcal{S}} G_{\sum_{j=1}^{i-1}\boldsymbol{s}_j}(\boldsymbol{s}_i)$ regardless the position of s_i in $\mathcal{S}$, though the specific value of $G_{\sum_{j=1}^{i-1}\boldsymbol{s}_j}(\boldsymbol{s}_i)$ changes under different order of data sets. Moreover, the valuation of the entire dataset $\mathcal{I}$ is completely distributed among all data contributors, *i.e.* $G(\mathcal{I}) = \sum_{i\in\mathcal{I}} G(i)$, which is easily derived by the additivity.

Fairness: Two data samples that are identical in what they contribute to the model have the same valuation in an online manner. That is, for any data s and s' are equivalent in the sense that $G(\mathcal{S}\cup\{s\}) = G(\mathcal{S}\cup\{s'\}), \forall\mathcal{S} \subseteq \mathcal{I}\backslash\{s, s'\}$, then $G_{\mathcal{S}}(s_i) = G_{\mathcal{S}}(s_j)$. Meanwhile, data with zero marginal contribution to the model has zero valuation, *i.e.*, if $G(\mathcal{S}\cup\{s\}) = G(\mathcal{S})$, then $G_{\mathcal{S}}(\boldsymbol{S}) = 0$. Actually, because the variance of parameter distribution is non-negative, if data has zero valuation, it means the variance is zero, and then the Gaussian function becomes a Dirac delta function, in which $\boldsymbol{\beta}$ only has one possible value.

Label Anonymity: According to Definition 1, in VAP-Valuation, each data's valuation can be calculated only depending on the data features $\boldsymbol{x}$, without using data label y, which can preserve the content of the data before data exchange. On the other hand, VAP-Valuation can infer the contribution of $\boldsymbol{x}$ in real-time when the data is submitted, without waiting until the end of model training, which provides the possibility for subsequent online data pricing.

Comparison With Shapley Value Valuation (SVV): 1) SVV is not submodular, and due to its computational characteristic, whenever a new data sample is added, the SVV of all data samples will need to be recalculated. 2) SVV also has additivity. In addition to this, the sum of SVV of all data samples is equal to 1, which is not available in VAP-Valuation. 3)SVV is the only valuation method with strict fairness. While our method can satisfy online fairness according to the above. 4)SVV does not have label anonymity, and can only be calculated after the whole complete data samples are obtained.

4 Data Pricing

4.1 Profit Maximization Mechanism

In this section, we present a posted pricing mechanism, VAP-Pricing, to maximize the service provider's profit in an online manner. According to Definition 1, $G_{\boldsymbol{X}}(\boldsymbol{x})$ denotes the contribution that one piece of data $\boldsymbol{x}$ brings to the performance improvement of model training, from which the service provider can extract profit. As we have mentioned before, each data contributor has a reserve value of v. Only if the payment is higher than the reserve value would she upload her data and get the corresponding payment; otherwise, she would leave without contributing her data. Therefore, the profit that the service provider can obtain from one data contributor with a reserve value v is:

$$u(p,v)=\begin{cases}\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))-np & p\geq v,\\ 0 & 0\leq p<v.\end{cases} \tag{12}$$

where p is the unit price of each data and $\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))$ is the revenue extracted from the updated model after adding n data samples $\boldsymbol{x}$. We use $F(p)=\int_0^p f(v)dv$ to denote the probability that a data contributor accepts the data price p. Thus, given the distribution of the reserve value v and the price p, the expected profit extracted from n data samples can be written as:

$$\mathbb{E}[u(p,v)]=F_v(p)(\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))-np). \tag{13}$$

As we do not know the distribution of reserve value, we tackle the above profit optimization problem by leveraging the exploration and exploitation technique from bandit literature [38], in which a decision-maker (the service provider) takes actions to maximize his long-term rewards (profits), by balancing between exploration and exploitation. At each time slot $t\in\{1,2,\cdots,T\}$, a new data contributor with a value v_t arrives. The service provider chooses a posted price from the set of candidate prices $P\triangleq\left\{p_i \mid p_i=\frac{i}{K}, i=1,\cdots,K\right\}$, where we regard each price $p_i\in P$ as an arm as the traditional setting [39]. Then he observes the feedback from the data contributor and gets the corresponding profit according to Equation (12).

The classical method UCB1 algorithm [40] estimates the unknown expected reward (profit) of each arm by making a linear combination of previously observed rewards of the arm. However, in our problem, the reward distribution behind each candidate price (arm) is not fixed, which is also determined by the valuation of data provided by the data contributor. Thus, we cannot directly use UCB1 to solve the online pricing problem. We observe that we can regard the data valuation as a type of context associated with each arm, and thus the pricing problem can be formulated as a contextual bandit problem [41]. To solve it, we first rewrite the profit function as:

$$\begin{aligned}\mathbb{E}[u(p,v)]&=F_v(p)(\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))-np)\\&=\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))F_v(p)-npF_v(p)\\&=\left[\begin{array}{cc}\pi(G_{\boldsymbol{X}}(\boldsymbol{x})) & n\end{array}\right]\left[\begin{array}{c}F_v(p)\\-pF_v(p)\end{array}\right].\end{aligned} \tag{14}$$

At time slot t, we define $\Pi_t=(\pi(G_t),n_t)^\top$ as the features of the context, where G_t denotes the total contribution of the arriving data set, and n_t is the number of data samples. Then the expected reward of arm p_i can be expressed as:

$$\mu_{i,t}=\Pi_t^\top\omega_i^*, \tag{15}$$

where $\omega_i^*\triangleq(F_v(p_i),-p_iF_v(p_i))^\top$ represents the unknown coefficient vector. To post a reasonable price, that is, to select the best arm of each round, the service provider needs to estimate the expected rewards in Equation (15) of arms accurately. The service provider can obtain the value of Π_t based on the VAP-Valuation. Then we should learn ω_i of each arm, which can be explained as learning the reserve value distributions of data contributors implicitly. In this way, we can regard the features of the context as independent variables, and the expected reward is the dependent variable. With this, we can treat the observed context-reward pairs as training samples and train a regression model for each arm.

However, different from the traditional setting in the LinUCB [41] to solve the contextual MAB problem, in our problem, the feedback information from each choice of one arm (*i.e.* one possible posted price) can not only update the current arm but also be used as training inputs for other arms. That is when one data contributor rejects a specific price p_i, which means $0\leq p_i<v$, she would also reject the price p with $p<p_i$. Similarly, when one data contributor accepts the price p_i, which means $p_i\geq v$, she would also accept the price p with $p>p_i$. Thus, in this work, we define M_i as a design matrix of dimension $j_i^*\times 2$ at time slot t, whose rows correspond to $j_i^*=j_i+j_l+j_s$ training inputs, where j_i is the amount of training data with price p_i, j_l is the number of training data with $p>p_i$ and the data contributor rejects the price p, j_s is the number of data with price $p<p_i$ and the data contributor accepts the price p. And c_i is the rewards corresponding to these contexts. With this augmented training data set (M_i,c_i), we can have a better estimate of the coefficients by applying ridge regression:

$$\hat{\omega}_i=\left(M_i^\top M_i+I\right)^{-1}M_i^\top c_i, \tag{16}$$

where I is the 2×2 identity matrix.

Algorithm 1 gives a detailed description of the entire LinUCB algorithm for pricing, in which $A_i=M_i^\top M_i+I$ and $b_i=M_i^\top c_i$. For the input of the algorithm, α is a parameter to control the exploration scale, $G_{\boldsymbol{X}}(\boldsymbol{x})$ is the VAP-Valuation of $\boldsymbol{x}$, $\pi(\cdot)$ is the revenue function, K is the number of arms (candidate price), and T is the total time slots. At each time slot, there is a data contributor t querying the price by her data $\boldsymbol{x}_t$, and then we can observe the context (features of the current data) Π_t (Line 2). Then for all possible prices, we estimate the coefficients according to Equation (16) (Lines 3-6). It can be shown that with probability at least $1-\delta$:

$$\left|\Pi_t^\top\widehat{\omega}_i-\mathbb{E}[\mu_{i,t}]\right|\leq\alpha\sqrt{\Pi_t^\top A_i^{-1}\Pi_t} \tag{17}$$

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2023.3316145

Algorithm 1: VAP-Pricing

Input: $\alpha \in \mathbb{R}^{+}$, $G_{\boldsymbol{X}}(\boldsymbol{x})$, $\pi(\cdot)$, K, T

1 **for** $t = 1$ *to* T **do**
2   Observe the features of current data $\Pi_t = (\pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t)), n_t)^{\top}$;
3   **for** $i = 1$ *to* K **do**
4   **if** p_i *is new* **then**
5       $A_i \leftarrow I, b_i \leftarrow \mathbf{0}$;
6     $\widehat{\omega}_i \leftarrow A_i^{-1} b_i, \widehat{\mu}_{i,t} \leftarrow \Pi_t^{\top} \widehat{\omega}_i + \alpha \sqrt{\Pi_t^{\top} A_i^{-1} \Pi_t}$;
7   Choose the arm $I_t = \underset{i=1,\cdots,K}{\operatorname{argmax}}\ \mu_{i,t}$;
8   **Posted price** $p = \min(p_{I_t}, \lfloor \pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t))/n \rfloor)$;
9   Observe and record the response from the data contributor t;
10   **if** t *is satisfied with the price* ($p_{I_t} \geq v_t$) **then**
11   **for** $i = I_t$ *to* K **do**
12       $r_t = \pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t)) - n_t p_i$;
13       $A_i \leftarrow A_i + \Pi_t \Pi_t^{\top}, b_i \leftarrow b_i + r_t \Pi_t$;
14     $\boldsymbol{X} = \boldsymbol{X} + \boldsymbol{x}_t$;
15   **else**
16   **for** $i = 1$ *to* I_t **do**
17       $r_t = 0$;
18       $A_i \leftarrow A_i + \Pi_t \Pi_t^{\top}, b_i \leftarrow b_i + r_t \Pi_t$;
19   Update function $G_{\boldsymbol{X}}(\boldsymbol{x})$;

for any $\delta > 0$, where $\alpha_T = 1 + \sqrt{2 \ln \frac{1}{\delta} + 2 \ln \left(1 + \frac{t}{2}\right)}$. We will show this result in Section 4.3. The inequality gives a reasonably tight upper confidence bound for the expected reward of arm p_{I_t}, from which a UCB-type arm-selection strategy can be derived: at each time slot t, choose

$$I_t = \underset{i=1,\cdots,K}{\operatorname{argmax}} \left(\Pi_t^{\top} \widehat{\omega}_i + \alpha \sqrt{\Pi_t^{\top} A_i^{-1} \Pi_t} \right). \tag{18}$$

The criterion for arm selection can also be regarded as an additive trade-off between the reward estimation and model uncertainty reduction (Lines 7-8). After we post a price, we record the response from the data contributor. If the data contributor accepts the posted price, *i.e.*, $p_{I_t} \geq v_t$, we calculate the reward and update A_i as well as b_i for all price $p_i > p_{I_t}$. The data contributor would upload her data, and we add it to the data set (Lines 10-14). Otherwise, *i.e.*, $0 \leq p_{I_t} < v_t$, the data contributor would leave without contributing her data. The reward we get is 0, and we update A_i as well as b_i for all price $p_i < p_{I_t}$ (Lines 15-18).

Next, we introduce the design of revenue function π. By the additivity property of data valuation metric in Section 3.2, we can get $G(\mathcal{S} \cup \mathcal{T}) = G(\mathcal{S}) + G(\mathcal{T})$. To make the revenue function extend the additivity, *i.e.* $\pi(G(\mathcal{S} \cup \mathcal{T})) = \pi(G(\mathcal{S})) + \pi(G(\mathcal{T}))$, it is easy to prove that $\pi(\cdot)$ should be the linear function by Cauchy's functional equation [42] as

$$\pi(G(\mathcal{S}) + G(\mathcal{T})) = \pi(G(\mathcal{S})) + \pi(G(\mathcal{T})). \tag{19}$$

In this work, we set $\pi(G_{\boldsymbol{X}}(\boldsymbol{x})) = k \cdot G_{\boldsymbol{X}}(\boldsymbol{x}) - \epsilon$, where k can uniform the magnitude between the data valuation and the revenue. As we have mentioned, $k \cdot G_{\boldsymbol{X}}(\boldsymbol{x})$ guaranteed the additivity when converting data valuation $G_{\boldsymbol{X}}(\boldsymbol{x})$ to p. Besides, ϵ can be seen as the fee per operation, which can also control the trade-off between total entropy reduction (data valuation) and the total budget, which we will show in the evaluation part. By setting proper ϵ, the service provider can acquire data with different objectives. For example, a budget-limited service provider may have a limited budget who only wants to collect a smaller data set and can tolerate a slower data collection rate. On the other hand, a budget-sufficient service provider has more budget and wants to collect as much data as possible. A suitable $\pi(\cdot)$ can control the trade-off between the data collection scale and the total budget.

4.2 Properties of Data Pricing Mechanism

The data pricing mechanism we proposed in VAP-Pricing has the following characteristics:

Incentive for Data Contribution: VAP-Pricing motivates data contributors to submit data as early as possible because the data valuation function $G_{\boldsymbol{X}}(\boldsymbol{x})$ is submodular with respect to $\boldsymbol{X}$. Specifically, in VAP-Pricing, earlier data contributors will have a higher (marginal) data contribution and are more likely to get more profit, implying that we encourage data contributors to submit data as soon as possible in the online data collection process.

Robust to Strategic Behaviors: To guarantee the property of symmetry, Shapley value leaves the possibility for selfish data contributors to carry out strategic behaviors, such as copying data for extra benefits. There are some solutions to solve this issue, such as discounting the value of the same data [18], but it will break the property of fairness in Shapley value. However, VAP-Pricing can naturally decrease the similar data's valuation, as the later data will not impact the model too much due to the submodularity of VAP-Valuation. Meanwhile, the data with the same contribution will be given the same price at one specific time slot. Thus, VAP-Pricing guarantees fairness to some extent.

Moreover, this data pricing mechanism is also arbitrage-free when the VAP-Pricing is stable. Due to the additivity of VAP-Valuation, regardless of the data order in a data set, the sum of the data valuation for a dataset is the same, resulting in the identical posted price. Specifically, we consider the case when the pricing mechanism is stable, *i.e.* we have already got the accurate $F_v(p)$. Suppose the data contributor divides a data set $\boldsymbol{S}$ into several subsets $\boldsymbol{S}_i$, $i = 1, \cdots, n$, and submits each subset at different time slots. Then by the additivity property of VAP-Valuation and the definition of revenue function π, we can further get

$$\pi(G) = \pi \left(\sum_{i=1}^{n} G_i \right) = \sum_{i=1}^{n} \pi\left(G_i\right) + \epsilon n, \tag{20}$$

where we denote $G = G_{\boldsymbol{X}}(\boldsymbol{S})$ for the whole data set and $G_i = G_{\boldsymbol{X} + \sum_{j=1}^{i-1} \boldsymbol{S}_j}(\boldsymbol{S}_i)$ for each subset. We denote $p_{1,i}$ as the posted price for each separated data subset and p_2 as the posted price for the whole data set. Then,

$$\begin{aligned} p_{1,i} &= \arg\max_{p_{1,i}} F_v(p_{1,i}) \left(\pi\left(G_i\right) - p_{1,i}\right), \\ p_2 &= \arg\max_{p_2} F_v(p_2) \left(\pi\left(G\right) - p_2 n\right) \\ &= \arg\max_{p_2} F_v(p_2) \left(\sum_{i=1}^{n} \pi\left(G_i\right) - (p_2 - \epsilon) n \right). \end{aligned} \tag{21}$$

As $F_v(p)$ is already known, it is easy to prove that $p_{1,i} = p_2 - \epsilon = v$ when $p \geq v$, which means if the full dataset can be traded, data contributors can not get more payment by splitting the data set and submitting them separately. Intuitively, if the data contributor splits the data and submits them separately due to the influence of the operation fee ϵ in the mapping function, it results in a lower payment.

Data Privacy Preserving for mHealth: By the Label Anonymity property of VAP-Valuation, the data contributor i can query the possible payment by x_i without uploading y_i. Then the label y_i is not involved in the data valuation and pricing processes, reducing the risk of privacy leakage. Thus, in the data collection process, the data contributors have the right to decide whether the data is used for model training under the VAP-Pricing framework. Suppose the data contributors are not satisfied with the current payment and choose not to contribute their whole data. In this case, they do not leak the whole information about their data (preserve the label y).

4.3 Regret Analysis

For stochastic linear bandits, a classic setting is a shared parameter with possibly infinite arms. In our problem, we follow the original version, considering fixed K arms and disjoint parameters, *i.e.*, the posted price set is fixed finite, and for each arm, coefficient vector $\omega_i^* \triangleq \left(F_v\left(p_i\right), -p_i F_v\left(p_i\right)\right)^\top$.

We define the regret of VAP-Pricing as:

$$\mathbf{R}_T = \sum_{t=1}^{T}\left(\Pi_t^\top \omega_{i^*}^* - \Pi_t^\top \omega_{i_t}^*\right), \tag{22}$$

where i^* is the optimal arm (price) and i_t is the arm taken at time slot t. The proof is divided into two steps, the first is that the regret of the shared-parameter setting will be $O(\sqrt{dT})$. In this setting, there is only a fixed unknown parameter ω^* for all arms, where $\omega^* \in \mathbb{R}^d$. After that, we show the regret of the disjoint-parameter setting under our problem setting will reach $O(\sqrt{dKT})$. Considering the shared-parameter setting, first, we make some assumptions, which are common in the traditional stochastic linear bandits problem.

Assumption 1. *We assume that the observed noise i.e., ($r_t - \Pi_t^\top \omega^*$) is independent standard Gaussian noise, where r_t is the reward and $\omega^* \in \mathbb{R}^d$ is an unknown but fix parameter.*

Assumption 2. *We assume that the contexts Π and the parameter ω^* are bounded. $\|\Pi\|_2 \le 1, \|\omega^*\|_2 \le 1$.*

Then the regret under the shared-parameter setting is:

$$R_T = \sum_{t=1}^{T}\left({\Pi_*}^\top \omega^* - \Pi_t^\top \omega^*\right), \tag{23}$$

where Π_* is the optimal action and Π_t is the action taken at time slot t. To note, it is different from the previous definition of Π. Here we reuse the symbol for simplicity. The actions here contain both the original context and the information of the arm selection, which will be introduced in detail in Equation (34) later. To complete the proof, we introduce the concept of confidence ellipsoid. The result shows that ω^* lies with high probability in an ellipsoid with center $\widehat{\omega}$ [43].

Lemma 1. *(Confidence Ellipsoid) Let $A_t = \lambda I + \sum_{\tau=1}^{t} \Pi\tau\Pi\tau^\top$, $\widehat{\omega}_t = A_t^{-1}\sum_{\tau=1}^{t} c_\tau \Pi\tau$, $\alpha_T = \sqrt{\lambda} + \sqrt{2\ln\frac{1}{\delta} + d\ln\left(1+\frac{t}{d\lambda}\right)}$, then with probability at least $1-\delta$, for all $t \ge 0$, ω^* lies in the set*

$$C_t = \left\{\omega \in \mathbb{R}^d : \|\omega - \widehat{\omega}_t\|_{A_t} \le \alpha_T\right\}. \tag{24}$$

Then we can get the regret of shared-parameter LinUCB.

Theorem 1. *With probability $1-\delta$, the regret R_T of shared-parameter LinUCB satisfies*

$$R_T \le \alpha_T \sqrt{8dT\ln\left(1+\frac{T}{\lambda d}\right)} = O(d\sqrt{T}), \tag{25}$$

where T is the total time slots, the hyperparameter $\alpha_T = \sqrt{\lambda} + \sqrt{2\ln\frac{1}{\delta} + d\ln\left(1+\frac{t}{d\lambda}\right)}$, d is dimension the unknown parameters, and λ is the coefficient of the identity matrix in the gram matrix.

Proof. By Cauchy-Schwarz and Lemma 1, we have

$$\left|(\omega^* - \widehat{\omega}_t)^\top \Pi\right| \le \|\omega^* - \widehat{\omega}_t\|_{A_t} \|\Pi\|_{A_t^{-1}} \le \alpha_T \|\Pi\|_{A_t^{-1}}. \tag{26}$$

Let $\widetilde{\omega}_t \in C_t$ be the parameter in the confidence set to make that $\widetilde{\omega}_t^\top \Pi_t = \max_{\omega \in C_t} \omega^\top \Pi$ Thus,

$$\begin{aligned} R_t &= {\Pi^*}^\top \omega^* - \Pi_t^\top \omega^* \\ &\le \Pi_t^\top\left(\widetilde{\omega} - \omega^*\right) \\ &= \Pi_t^\top\left(\widetilde{\omega} - \widehat{\omega}_t\right) + \Pi_t^\top\left(\widehat{\omega}_t - \omega^*\right) \\ &\le 2\alpha_T \|\Pi_{i_t}\|_{A_{t-1}^{-1}}. \end{aligned} \tag{27}$$

As the definition of α_T, we can get $\alpha_T \ge 1$. By the Assumption 3, we have $R_t \le 2\alpha_T \min\{1, \|\Pi_{i_t}\|_{A_{t-1}^{-1}}\}$. Then by the Cauchy-Schwarz inequality and $\min\{1,x\} \le 2\ln(1+x)$, we can bound the regret as

$$\begin{aligned} R_T &= \sqrt{T\sum_{t=1}^{T} R_t^2} \\ &\le \alpha_T \sqrt{4T\sum_{t=1}^{T}\min\left\{1, \|\Pi_{i_t}\|_{A_{t-1}^{-1}}^2\right\}} \\ &\le \alpha_T \sqrt{8T\sum_{t=1}^{T}\ln\left(1+\|\Pi_{i_t}\|_{A_{t-1}^{-1}}^2\right)} \\ &= \alpha_T \sqrt{8T\ln\prod_{t=1}^{T}\left(1+\|\Pi_{i_t}\|_{A_{t-1}^{-1}}^2\right)}. \end{aligned} \tag{28}$$

By the definition of A, we have

$$\begin{aligned} \det(A_T) &= \det(A_{T-1} + \Pi_{i_T}\Pi_{i_T}^\top) \\ &= \det\left(A_{T-1}^{\frac{1}{2}}\left(I + A_{T-1}^{-\frac{1}{2}}\Pi_{i_T}\Pi_{i_T}^\top A_{T-1}^{-\frac{1}{2}}\right)A_{T-1}^{\frac{1}{2}}\right) \\ &= \det\left(A_{T-1}\right)\det\left(I + A_{T-1}^{-\frac{1}{2}}\Pi_{i_I}\Pi_{i_T}^\top A_{T-1}^{-\frac{1}{2}}\right) \\ &= \det\left(A_{T-1}\right)\left(1+\|\Pi_{i_T}\|_{A_{T-1}^{-1}}^2\right) \\ &\cdots \\ &= \det\left(A_0\right)\prod_{t=1}^{T}\left(1+\|\Pi_{i_t}\|_{A_{t-1}^{-1}}^2\right). \end{aligned} \tag{29}$$

Then, by AM-GM inequality, we can get

$$\begin{aligned}\det(A_T) &\le \left(\frac{\operatorname{Trace}(A_T)}{d}\right)^d \\ &= \left(\frac{\lambda d + \sum_{t=1}^{T} \operatorname{Trace}\left(\Pi_{i_t}\Pi_{i_t}^\top\right)}{d}\right)^d \\ &= \left(\lambda + \frac{\sum_{t=1}^{T} \operatorname{Trace}\left(\Pi_{i_t}^\top\Pi_{i_t}\right)}{d}\right)^d \le \left(\lambda + \frac{T}{d}\right)^d,\end{aligned} \tag{30}$$

with $\det A_0 = \lambda^d$, we can further get

$$\frac{\det A_T}{\det A_0} \le \left(1 + \frac{T}{\lambda d}\right)^d. \tag{31}$$

Then, we can continue bounding the regret as

$$\begin{aligned}R_T &\le \alpha_T \sqrt{8T \ln \prod_{t=1}^{T}\left(1 + \|\Pi_{i_t}\|^2_{A_{t-1}^{-1}}\right)} \\ &= \alpha_T \sqrt{8T \ln \frac{\det A_t}{\det A_0}} \\ &= \alpha_T \sqrt{8Td \ln\left(1 + \frac{T}{\lambda d}\right)}.\end{aligned} \tag{32}$$

□

Till now, we obtain the regret of the shared-parameter setting with a single true ω^* assumed.

Theorem 2. *With probability $1-\delta$, the regret of disjoint-parameter VAP-Pricing satisfies*

$$\mathbf{R}_T = O(2K\sqrt{T}), \tag{33}$$

where T is the total time slots, K is the number of arms, and $\alpha_T = \sqrt{\lambda} + \sqrt{2\ln\frac{1}{\delta} + d\ln\left(1 + \frac{t}{d\lambda}\right)}$ in VAP-Pricing.

An intuitive understanding is that the regret of VAP-Pricing is related to the number of arms (prices) set K and the total time slots T. As a bigger arm size, the regret will increase due to a larger range of policies, and VAP-Pricing can get logarithmic accumulated regret. Also, as α is related to δ, the choice of α in VAP-Pricing will affect the probability of the regret guarantee.

Proof. In our problem, the parameter is disjoint over each arm so we have K separate parameters ω_i^* to estimate, one for each arm. Then it is obvious the regret of the disjoint-parameter situation is a factor of K worse than that of the shared-parameter LinUCB. Another explanation for this is that we can generate a new parameter Ω^*, to make

$$\Omega^* = \begin{bmatrix} \omega_1^* \\ \vdots \\ \omega_i^* \\ \vdots \\ \omega_K^* \end{bmatrix}, \quad \Pi_t = \left\{ \begin{bmatrix} 0 \\ \vdots \\ \Pi_t \\ \vdots \\ 0 \end{bmatrix} : i = 1, 2, \ldots, K \right\}, \tag{34}$$

where $\omega_i^* \in \mathbb{R}^d$. Thus by the definition of Ω^*, we have $\Omega^* \in \mathbb{R}^D, D = dK$. Then we can get under shared-parameter Ω^*, the regret is bounded by $O(D\sqrt{T}) = O(dK\sqrt{T})$. □

4.4 VAP-Pricing Under Fixed Limited Budget

In Section 4.1, we considered that the service provider's budget is not fixed, and it can be adjusted by ϵ. However, in this section, we consider another common situation in real-life situations, where the budget is fixed at the beginning. In this case, we cannot simply model it as an ordinary contextual multi-armed bandit problem as before. This is because we need to consider not only the revenue brought by the current arm each time we pull the arm but also the budget consumption at the same time to ensure that we can get the maximum revenue when the budget is depleted. Then a straightforward idea is to model it as a linear contextual bandit with backpacks problem, which is an extended version of VAP-Pricing above when considering fixed budgets. Linear contextual bandits with backpacks had been studied by previous work [44]. However, we can not apply the previous method to our model. In previous work, it was assumed that there is a fixed but unknown distribution D on the context. Whereas in our modeling, contexts ($\Pi_t = (\pi(G_t), n_t)^\top$, *i.e.*, data valuation and the number of data samples) do not follow a fixed distribution (the data valuation is diminishing marginal). The good news is that the part not known to the data provider and needs to be inferred by the VAP-Pricing is the reserve value distribution $F_v(p)$ of data providers. It is a deterministic distribution that does not vary over time. Thus, considering the budget limitation, we can remodel the problem to a static multi-armed bandit with a knapsack framework, instead of considering time-varying contexts (data valuations). In the first stage, we predict the reserve value distribution of the data contributors through a static multi-armed bandit, and then we combine the expected reserve value distribution and the current data valuation to compute the accurate posted price.

Next, we give a formal definition of this problem. The service provider is given access to d-dimensional of K arms (price) denoted as $a \in [K] := \{1, 2, \ldots, K\}$. Each time $t \in [T]$, the service provider pulls an arm a_t and observes the reward and consumption. We denote the unknown expected reward as r_t, and the corresponding resource consumption as c_t. We assume there is a fixed total budget $B \in \mathbb{R}_+$ on the consumption, And B is a hard constraint on resource consumption. The algorithm stops at the earliest time τ when B is exhausted.

To solve this bandit with knapsack problem, we decomposite the budget into each slot. For submission with n pieces of data samples, we regard the strategy as the repeated n posted price. Thus, in each time slot, we only need to consider the unit reward and the unit cost of each price $p_i, i \in [K]$. First, we define the unit reward (profit) and cost for each arm. The reward (profit) that the service provider can obtain from one data contributor with a reserve value v is:

$$r(p_i, v) = \begin{cases} \pi(G_{\boldsymbol{X}}(\boldsymbol{x}))/n - p_i & p_i \ge v, \\ 0 & 0 \le p_i < v. \end{cases} \tag{35}$$

, where arm $i \in [K]$. And the unit cost that the service provider pays for each arm i with a reserve value v is:

$$c(p_i, v) = \begin{cases} p_i & p_i \ge v, \\ 0 & 0 \le p_i < v. \end{cases} \tag{36}$$

As we mentioned above, we transform them into functions related to $F_v(p_i)$,

$$\begin{aligned}\mathbb{E}[r(p_i, v)] &= F_v(p_i)(\pi(G_{\boldsymbol{X}}(\boldsymbol{x}))/n - p_i). \\ \mathbb{E}[c(p_i, v)] &= F_v(p_i)(p_i).\end{aligned} \tag{37}$$

Algorithm 2: VAP-PricingwK

Input: $\alpha \in \mathbb{R}^+$, $K, T, B, \epsilon, G_{\boldsymbol{X}}(\boldsymbol{x})$, $\pi(\cdot)$

1 **Initialization: for** $i = 1$ *to* K **do**
2 $\quad f(i) \leftarrow 0, m(i) \leftarrow 0$;
3 **for** $t = 1$ *to* T **do**
4 $\quad$ Exit if the budget B is exhausted;
5 $\quad$ **if** $t < K + 1$ **then**
6 $\quad\quad$ **Posted price** $p_{I_t} = p_t$;
7 $\quad$ **else**
8 $\quad\quad$ **for** $i = 1$ *to* K **do**
9 $\quad\quad\quad$ $F_v(p_i) = \frac{f(i)}{m(i)}$;
10 $\quad\quad\quad$ $\tilde{r}_t(i) = F_v(p_i)(\frac{\pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t))}{n_t} - p_i) + \alpha\sqrt{\frac{\ln t}{m(i)}}$
11 $\quad\quad\quad$ $\tilde{c}_t(i) = F_v(p_i)(p_i) - \alpha\sqrt{\frac{\ln t}{m(i)}}$
12 $\quad\quad$ solving $\begin{array}{ll} \max_{\boldsymbol{q}\in\Delta} & \tilde{\boldsymbol{r}}_t \cdot \boldsymbol{q} \\ \text{s.t.} & \tilde{\boldsymbol{c}}_t \cdot \boldsymbol{q} \leq (1-\epsilon)\frac{n_t B}{T} \end{array}$
13 $\quad\quad$ $I_t = \text{random}(\boldsymbol{q})$;
14 $\quad\quad$ **Posted price** $p_{I_t} = \min(p_{I_t}, \lfloor \pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t))/n_t \rfloor)$;
15 $\quad$ Observe and record the response from the data contributor t;
16 $\quad$ **if** t *is satisfied with the posted price* ($p \geq v_t$) **then**
17 $\quad\quad$ **for** $i = I_t$ *to* K **do**
18 $\quad\quad\quad$ $f(i) \leftarrow f(i) + 1$;
19 $\quad\quad\quad$ $m(i) \leftarrow m(i) + 1$;
20 $\quad\quad$ $\boldsymbol{X} = \boldsymbol{X} + \boldsymbol{x}_t$;
21 $\quad\quad$ Update function $G_{\boldsymbol{X}}(\boldsymbol{x})$;
22 $\quad$ **else**
23 $\quad\quad$ **for** $i = 1$ *to* I_t **do**
24 $\quad\quad\quad$ $m(i) \leftarrow m(i) + 1$;

Then in first stage, we calculate the $F_v(p_i)$:

$$F_v(p_i) = \frac{f(i)}{m(i)}, \tag{38}$$

where $f(i)$ is the number of times that data contributors accept p_i, *i.e.*, $p_i \geq v$ in $m(i)$. Then based on the estimation of $F_v(p_i)$, we can calculate the expected reward and cost. We give detailed data pricing with a knapsack algorithm, called VAP-PricingwK in Algorithm 2. For the input of the algorithm, α is a parameter to control the exploration scale, K is the number of arms (candidate price), T is the total time slots, B is the total Budget of the service provider, ϵ is the budget factor, $G_{\boldsymbol{X}}(\boldsymbol{x})$ is the VAP-Valuation of $\boldsymbol{x}$, and $\pi(\cdot)$ is the revenue function. We initialize the parameter $f(i)$ and $m(i)$ for each arm i (Lines 1-2), and for all possible prices, we pull each arm once (Lines 5-6). As for each time slot t, there is a data contributor t querying the price by her data $\boldsymbol{x}_t$. First, we estimate the reserve value distribution as Equation (38) based on the historical observation (Line 9). To estimate the reward $r_t(i)$ and cost $c_t(i)$ for each arm, inspired by the UCB-based algorithm for BwK [45], which is based on "optimism in the face of uncertainty", we make optimistic estimates of them:

$$\begin{aligned} \tilde{r}_t(i) &= F_v(p_i)(\frac{\pi(G_{\boldsymbol{X}}(\boldsymbol{x}_t))}{n_t} - p_i) + \alpha\sqrt{\frac{\ln t}{m(i)}}, \\ \tilde{c}_t(i) &= F_v(p_i)(p_i) - \alpha\sqrt{\frac{\ln t}{m(i)}}, \end{aligned} \tag{39}$$

where $\tilde{r}_t(i)$ is the upper confidence bound of $r_t(i)$ and $\tilde{c}_t(i)$ is the lower confidence bound of $c_t(i)$ (Lines 10-11). After getting $\tilde{r}_t(i)$ and $\tilde{c}_t(i)$, we solve the following linear programming to get the policy:

$$\begin{array}{ll} \max_{\boldsymbol{q}\in\Delta} & \tilde{\boldsymbol{r}}_t \cdot \boldsymbol{q} \\ \text{s.t.} & \tilde{\boldsymbol{c}}_t \cdot \boldsymbol{q} \leq (1-\epsilon)\frac{n_t B}{T} \end{array}, \tag{40}$$

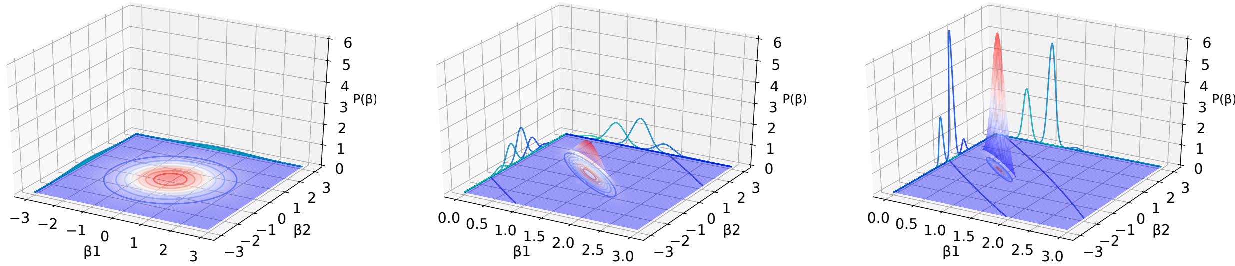
where $\boldsymbol{q}$ is the probability to pull each arm, and $\epsilon = \sqrt{\frac{\gamma d}{B}} + \log(T)\frac{\gamma d}{B}, \gamma = \log\left(\frac{Td}{\delta}\right)$. Then we select arm I_t randomly according to the probability in $\boldsymbol{q}$, and post the price (Lines 12-14). After posting a price, we record the response from the data contributor. If the data contributor accepts the posted price, *i.e.*, $p \geq v_t$, we update $f(i)$ as well as $m(i)$ for all arms $i > I_t$. The data contributor would upload her data, and we add it to the data set (Lines 15-21). Otherwise, *i.e.*, $0 \leq p < v_t$, the data contributor would leave without contributing her data. We only update $m(i)$ for all arms $i \leq I_t$ (Lines 23-24).

5 Extensions to General Models

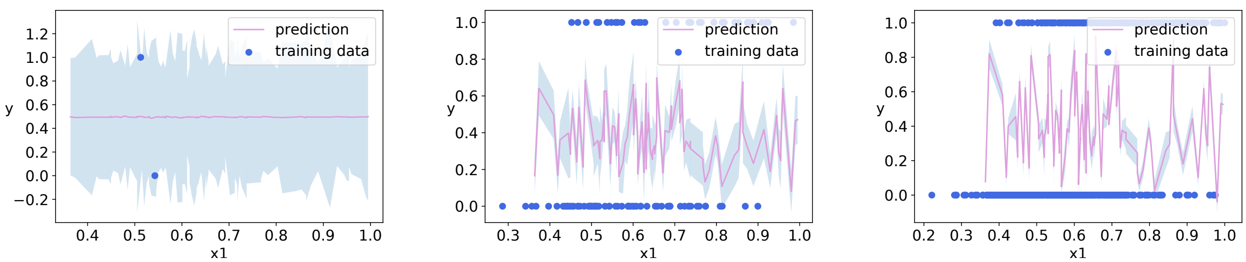
In this section, we extend VAP to advanced ML models. In Bayesian linear regression, we can easily calculate the posterior parameter distribution by a closed-form expression. However, in other advanced ML models, such as Bayesian neural network [34], parameter spaces are often high dimensional, and computing their entropy is usually intractable. Furthermore, for non-parametric processes, such as the Gaussian process [33], the parameter space is infinite-dimensional, which further increases the computational complexity.

To solve this problem, inspired by Equaition (9) and Equation (8), we transfer the objective from computing uncertainty in the parameter space to the prediction space, avoiding gridding parameter space (exponentially hard with dimensionality). Fig. 2 shows the comparison of parameter probability density distribution and prediction uncertainty. We can find that they have the same shrinking trend when adding more training data. The model's grasp of the parameter is getting higher, implying the model uncertainty and prediction uncertainty reduction. Thus, the data valuation we obtained can be regarded as a measure of uncertainty. The difference is that Equation (9) calculates the predictive distribution variance in the prediction task, the aim of which is to get the uncertainty in the current test data to evaluate the reliability of a prediction. However, Equation (8) calculates the entropy reduction of parameter $\boldsymbol{\beta}$ caused by adding new training data from the training data set. The goal is to get the model uncertainty changes caused by current training data to measure each data's contribution.

Thus, we can calculate entropy in low-dimensional output space using the idea of prediction uncertainty. For new data, $d = (\boldsymbol{x}, y)$, we calculate its contribution by regarding $\boldsymbol{x}$ as the features of the prediction task to calculate its prediction uncertainty. Specifically, for a representative non-parametric model, we write Gaussian process regression(GPR) as $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\varepsilon}$



(a) Change of parameter distribution. The amount of training data increases from left to right (2, 100, and 600). We only show two dimensions of parameter β_1 and β_2 for straightforward demonstration.



(b) Change of prediction uncertainty. The amount of training data increases from left to right (2, 100, and 600). x_1 is one of the features of training data, and y is the corresponding label. The pink line is the prediction of the current model, and the blue shaded area is the corresponding prediction uncertainty.

Fig. 2. Model Changes during data addition.

with the unknown function f follows a $\mathcal{N}(\mu, k)$ and $\boldsymbol{\varepsilon}$ follows a $\mathcal{N}(0, \gamma^2 I)$ [33]. Different from parameter $\boldsymbol{\beta}$ in range regression, there are no specific parameters in f. Thus, GPR is a non-parametric model. Consider the current purchased data set $D = \{d_i\}_{i=1}^n$ containing n data with $d_i = (\boldsymbol{x}_i, y_i)$, $[f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_n)]^\top \sim \mathcal{N}(\boldsymbol{\mu}, K)$, where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{K}$ is the $n \times n$ covariance matrix, $\boldsymbol{K}_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. To make a prediction of new data sample $\boldsymbol{x}$ by the current model, the predictive distribution is:

$$p(f(\boldsymbol{x}) \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{x}) = \mathcal{N}(\hat{\boldsymbol{\mu}}, \Sigma_{\boldsymbol{x}}), \tag{41}$$

where the predictive distribution variance is:

$$\begin{aligned} \Sigma_{\boldsymbol{x}} &= \boldsymbol{K}(\boldsymbol{x}, \boldsymbol{x}) \\ &\quad - \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x})^\top \left(\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) + \gamma^2 I\right)^{-1} \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{x}). \end{aligned} \tag{42}$$

Then, similar to the Equation (8), the valuation function in GPR can be set as

$$G_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{2}\ln(1 + \Sigma_{\boldsymbol{x}}). \tag{43}$$

Moreover, for the complex parametric model, neural network, similar to the Bayesian linear regression, we can put a prior distribution over its weights, such as a Gaussian prior distribution: $\boldsymbol{W} \sim \mathcal{N}(0, \gamma^2 I)$. Such a model is referred to as a Bayesian neural network (BNN) [34]. For each new data x, we can obtain the corresponding predictive distribution uncertainty using the BNN uncertainty [8]. Firstly, we optimize the parameters of the simple distribution instead of optimizing the original neural network's parameters in BNN, where the posterior $p(\boldsymbol{W}|\boldsymbol{X}, \boldsymbol{Y})$ is fitted with a simple distribution $q_{\boldsymbol{\theta}}^*(\boldsymbol{W})$, parameterized by $\boldsymbol{\theta}$. Then by the Dropout in BNN, which can be interpreted as a variational Bayesian approximation, epistemic uncertainty can be measured. For classification, the model prediction can be approximated using Monte Carlo integration as follows:

$$p(\mathbf{f}^{\boldsymbol{W}}(\boldsymbol{x}) = r \mid \boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{x}) \approx \frac{1}{T}\sum_{t=1}^{\top} \operatorname{softmax}\left(\mathbf{f}^{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x})\right), \tag{44}$$

with T sampled masked model weights $\widehat{\boldsymbol{W}}_t \sim q_{\boldsymbol{\theta}}^*(\boldsymbol{W})$, where $q_{\boldsymbol{\theta}}^*(\boldsymbol{W})$ is the Dropout distribution [8]. Then the valuation function can be calculated by:

$$G_{\boldsymbol{X}}(\boldsymbol{x}) = -\sum_{r=1}^{R} p_r \log p_r, \tag{45}$$

where R is the number of categories. For regression, the predictions are made by approximating the predictive mean:

$$\mathbb{E}(f^{\boldsymbol{W}}(\boldsymbol{x})) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x}). \tag{46}$$

The prediction uncertainty is captured by the predictive variance, which can be approximated as:

$$\operatorname{Var}(f^{\boldsymbol{W}}(\boldsymbol{x})) \approx \frac{1}{T}\sum_{t=1}^{T} f^{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x})^\top f^{\widehat{\boldsymbol{W}}_t}(\boldsymbol{x}) - \mathbb{E}^\top \mathbb{E}, \tag{47}$$

Similarly, the valuation function can be calculated by:

$$G_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{2}\ln(1 + \operatorname{Var}(f^{\boldsymbol{W}}(\boldsymbol{x}))). \tag{48}$$

Thus, we can extend the VAP for various online ML models as long as they can calculate prediction uncertainty, such as GPR and the model under the Bayesian framework. More intuitively, rather than collecting data for significantly reducing the parameter distribution's differential entropy, we marginally seek the data for which the model is most uncertain about the predictions.

If there is a higher degree of uncertainty about the prediction of arriving data, we do not have enough data whose features are similar to its features, so we have less confidence in it. So when we add this data to our training data set, it will significantly reduce the model uncertainty in this data region. Thus, such data will contribute more to the model, leading to more entropy reduction of parameter distribution, and the service provider would like to post a higher price for it. In addition to online learning models, VAP can be used in some other domains to guide the data collection process. For example, in domains such as active learning [46] and Bayesian reinforcement learning [47], where the model should have the ability to identify the most valuable data for model training and add it to the training set.

6 Evaluation Results

In this section, we evaluate our VAP through extensive experiments on real-world human behavior indicators data, which can be involved in mHealth.

6.1 Evaluation Setup

We present the evaluation results based on two real-world human behavior data sets: 1) Human Activity Recognition (HAR) database [48], a data set built from the recordings of 30 data contributors performing daily living activities while carrying a waist-mounted smartphone with embedded inertial sensors. The obtained data set was randomly partitioned into two sets, where 70% of the volunteers were selected to generate the training data and 30% the test data. 2) Pima Indians Diabetes (PID) [49], a data set initially from the National Institute of Diabetes and Digestive and Kidney Diseases. The data set's objective is to diagnostically predict whether a patient has diabetes based on specific diagnostic measurements included in the data set.

6.2 Results of Data Valuation

6.2.1 VAP on Different Models and Tasks

We evaluate the performance of VAP-Valuation. Fig. 3 shows that VAP-Valuation is a proper model value evaluation metric leading to smaller model uncertainty and higher model accuracy. First, as for RC, in Fig. 3(a) and Fig. 3(d), the general trend in total entropy reduction and prediction accuracy boost is consistent, which means the goals of data collection and model optimization are consistent under VAP. Meanwhile, in Fig. 3(b) and Fig. 3(e), by observing that the model accuracy increases slowly with the decrease of VAP-Valuation and that the turning points of them are close (for about 20 in Fig. 3(a) and 500 in Fig. 3(c), we can conclude that the VAP is able to judge the proper scale of the data collection. That is to say, after collecting such an amount of data, the valuation of the new data is relatively small, and the accuracy of the model is relatively stabilized.

As for GPC and BNN, using the VAP-Valuation in Section 5, we value the data on the outcome space. As the PID is a smaller data set, we adopt the GPC model to it. Meanwhile, HAR is a more extensive data set, which is more suitable for training with the BNN model. In Fig. 3(c) and Fig. 3(f), we can get a similar result with the RC model. By adding a new data sample, the model uncertainty is smaller, leading each data's contribution to the model more negligible, and the model accuracy increases. Also, the turning points of them are close, for about 20 in Fig. 3(c) and 1000 in Fig. 3(f). Moreover, from all the results in these three models, we can notice that the contribution of each data point shows the characteristic of diminishing marginal, which is consistent with the properties we described in Section 3.2. Valuation on the outcome space (Fig. 3(c) and Fig. 3(f)) appears to fluctuate more than the valuation on the parameter space because there is only one parameter space, and its dimension is higher and the previous decline is more. While the predictive distribution for each data is a different distribution. We can also notice some prominent high points in VAP-Valuation. Such data points may be the data points of new distributions in the system that have not been acquired before. In practice, in addition to data points that may be of higher epistemic uncertainty and thus show high value to the model, it can also be some incorrect data due to the problems arising from equipment acquisition. Furthermore, the service provider can identify and distinguish between the two types of data based on specific tasks. For example, the service provider can distinguish whether data is from a hypertensive patient (150/100 mmHg) or is derived from an abnormal collection (500/100 mmHg).

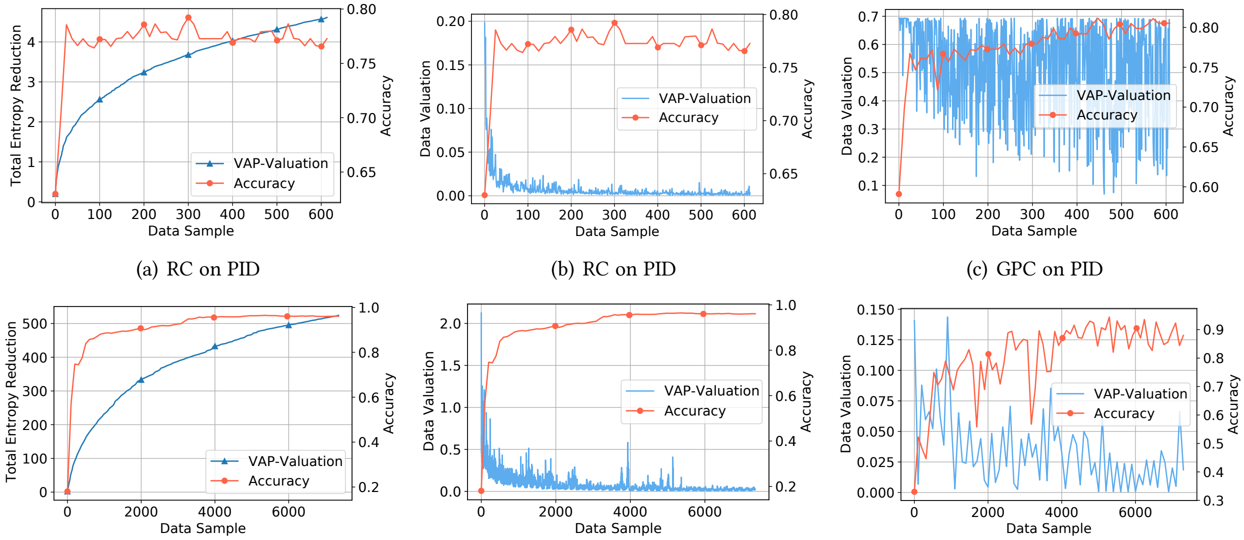
6.2.2 Performance of Different Data Valuation Metrics

We compare our method with other static data valuation metrics for machine learning, including TMC-Shapley [20], G-Shapley [20] and Random (one possible online metric) in Fig. 4. Compared with other methods, VAP-Valuation is more suitable for online learning for the following reasons. First, as Fig. 4(a) shows, the VAP-Valuation shows many excellent characteristics for data pricing and collection. It has a significant downward trend as the gradual increase of data over time considers the arrival order, which can incentive an earlier data submission. Besides, we can see that VAP-Valuation is always strictly positive, which provides convenience for data pricing.

Moreover, Shapley value and its variants are common practices in data valuation for the ML field, so here we emphasize why VAP outperforms Shapley in online learning tasks. Compared to the Shapley value, VAP-Valuation can perform online calculations without corresponding labels and testing data according to the inferrability of VAP-Valuation we mentioned in 3.2. Simultaneously, the computational complexity will increase significantly with the larger scale of the data set in static Shapely value. Although there are some approximate calculation methods such as TMC-Shapley [20], it still requires a lot of test data and high computational cost, which is impossible and inappropriate to achieve in a real-world mHealth system. G-Shapley, an approximation of TMC-Shapley, can be adapted to online learning. The marginal contribution in G-Shapley is the change of the model's performance. However, as shown in Fig. 4(a), we can find the G-Shapley does not achieve a good approximation of TMC-Shapley, because the calculation result can be affected by various factors, the size of the test set, learning rate, haphazard, etc. Finally, We can see that VAP-Valuation consistently outperforms the other two mechanisms as illustrated in Fig. 4(b), as it shows a better decrease over time than others as removing high-valuation data points. Thus, VAP-Valuation is more suitable for online learning tasks.

6.3 Results of Data Pricing

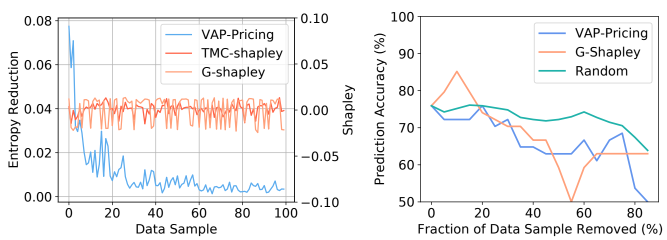
First, we compare the performance of different data pricing mechanisms under a regular situation: VAP-Pricing, Random, Half Fix, Half Valuation, LinUCB [41] and UCB1 [40]. In Random



(a) RC on PID (b) RC on PID (c) GPC on PID

(d) RC on HAR (e) RC on HAR (f) BNN on HAR

Fig. 3. VAP-Valuation on different models (Ridge classification (RC), Gaussian process classification (GPC)) and Tasks (HAR and PID Database).



(a) (b)

Fig. 4. Performance of different data valuation metrics. **(a)** Comparison of the valuation of the first 100 PID data; **(b)** The effect of removing high-valuation data points under different data valuation metrics.

pricing, the posted price p is uniformly distributed within $[0, 1]$. In Half Fix pricing, we set $p = 0.5$. And in Half Valuation pricing, we set $p = \min(0.5 \cdot G_{\boldsymbol{X}}(\boldsymbol{x}), 1)$. In all experiments, we set $\alpha = 1.2$, $\epsilon = 0$,$K = 10$, $T = 500$, and the reserve value is an approximately normal distribution within $[0, 1]$, where the mean is 0.5, and the variance is 0.01 unless otherwise noted. In Fig. 5, we can see that VAP-Pricing is always better than any other policies under different settings of reserve values of data contributors. Moreover, when the reserve value distribution is closer to the constant distribution, VAP-Pricing can get a higher profit. As for other mechanisms, we can see that Random is always the worst. The performance of Half Fix will be worse than contextual methods because it cannot capture the valuation information of the data samples. It can be considered as the optimal case of the traditional UCB1 method (also without considering the context), and Fig. 5 turns out that it is true. In addition, from the last figure in Fig. 5, it can be shown that Half Fix has a very high profit in the early stage. This is because when the reserve value is a constant $f(v) = 0.5$, the posted price $p = 0.5$ in each time slot will definitely be accepted by data contributors. The profit growth of Half Fix will be slow or even negative in the later period, also because the lack of data valuation results in the purchase of low-value data at high prices. In contrast, Half Valuation can always buy the data sample with a higher valuation by posting a high price, so it can always maintain a better growth trend. However, without the estimation of the reserve value will overbid, causing its total benefit to be damaged. Besides, the naive LinUCB method does not take into account the monotonicity of pricing and also leads to unsatisfactory profits.

Besides, we evaluate the performance of different ϵ. In Fig. 6, we can see that a bigger ϵ leads to a smaller budget and total entropy reduction while maintaining a high profit. Supposing that the service provider chooses a higher ϵ, correspondingly, he tends to use the limited budget to collect a smaller data set, this limited data set can effectively reduce the uncertainty of model predictions. On the contrary, if the service provider chooses a smaller ϵ, he wants to use more budget to collect more data. This adequate data set can further significantly reduce the uncertainty.

Comparing the price of different pricing policies in Fig. 7, we can see that the VAP-Pricing method can maintain the downward trend of valuation compared to Half Valuation, which is also fairer than other Random or Half Fix. Compared with other advanced bandit methods, *i.e.*, UCB1 and LinUCB, VAP-Pricing can better estimate the reserve value distribution of contributors, leading to faster convergence and a more reasonable price. It can monitor changes in data valuation and adjust the posted price promptly to maximize the profit.

In order to explain the effect of VAP-Pricing more intuitively, we also designed a set of experiments in a special case, that is when the reserve value $v = 0$. Fig. 8 shows that VAP-Pricing can converge quickly to the lowest price to extract more profit.

As for the performance of different data pricing mechanisms under a fixed limited budget, we compare the performances of VAP-Pricing, Random, Half Fix, Half Valuation, LinUCB, UCB1, and VAP-PricingwK in Fig. 9. VAP-PricingwK shows demonstrates exceptional budget control capability, consistently halting close to the predetermined timeframe ($T = 500$), contrasting with other methods that cease earlier due to budget exhaustion. Concurrently, the total profit of VAP-PricngwK is maximal and

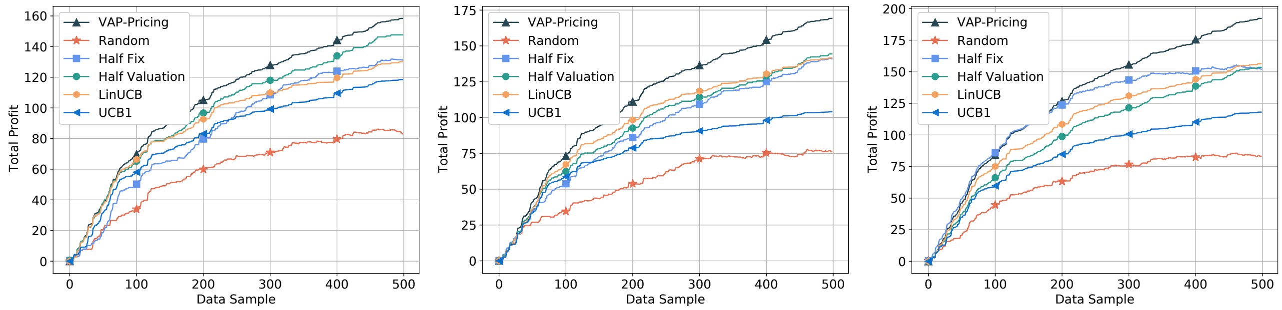


Fig. 5. Performance of different data pricing mechanisms under different reserve values' distribution, from left to right: $f_1(v)$: An approximately normal distribution within $[0, 1]$, where the mean is 0.5, and variance is 0.1; $f_2(v)$: An approximately normal distribution within $[0, 1]$, where the mean is 0.5, and variance is 0.01; A constant distribution as $f_3(v) = 0.5$.

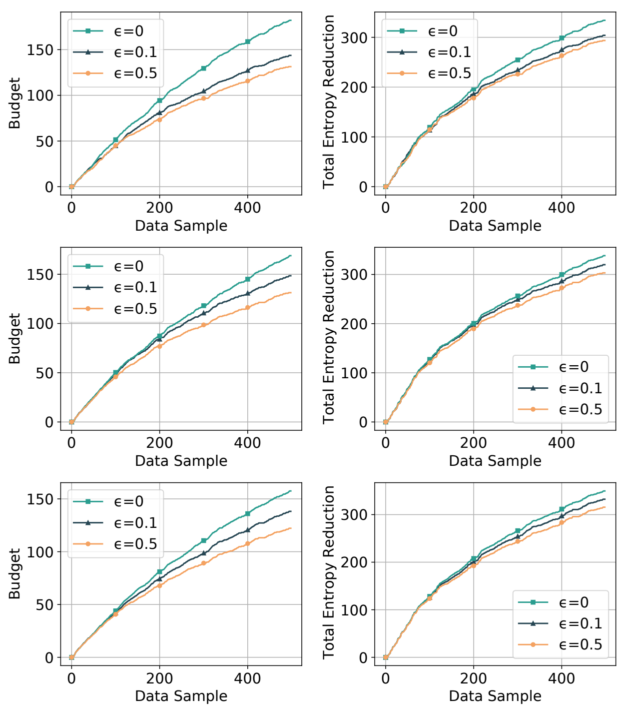


Fig. 6. Performance of Different ϵ (ϵ = 0, 0.1, 0.5) and different reserve values' distribution (From top to bottom are $f_1(v)$, $f_2(v)$, and $f_3(v)$) on budget and entropy reduction.

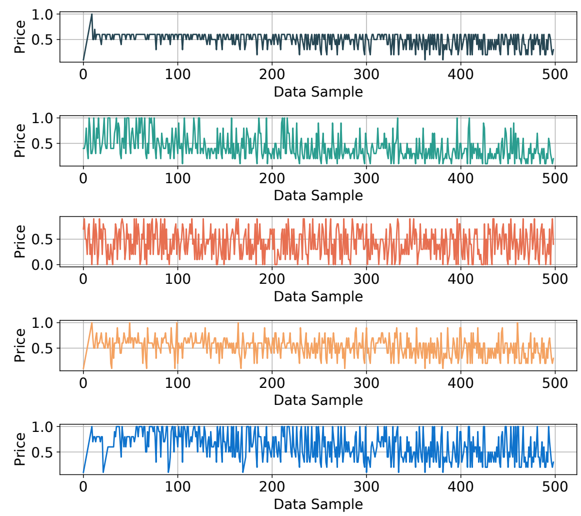


Fig. 7. Price Comparison of Different Pricing Mechanisms (From top to bottom are VAP-Pricing, Half Valuation, Random, LinUCB, and UCB1).

the UCB1 and random methods are the worst. In addition, it can be noticed that VAP-PricingwK does not grow as fast as some of the other algorithms in the early stages due to the need to control the budget and not to adopt a particularly aggressive exploration strategy, but since it retains a larger budget, it will have the opportunity to collect valuable data for higher profits in the later stages. It is also worth noting that although VAP-Pricing consumes the budget more rapidly, the performance is acceptable. This is ascribed to the incorporation of contextual information, which enhances learning speed. However, under a fixed budget, it is challenging to identify an appropriate budget control factor ϵ for VAP-Pricing in advance, resulting in a loss of final profit. Conversely, VAP-PricingwK's control under a fixed budget is automatic. Finally, we also find that when the variance of the reserve value v is smaller, the magnitude of the change is smaller, making it easier for VAP-PricingwK to estimate $F_v(p_i)$, thus obtaining larger profits.

7 Related Work

7.1 Mobile Health

The researchers develop multiple models by combining principled medical approaches with ML techniques in mHealth in a variety of domains, including diabetes [50], [51], [52], activity recognition [53], [54], depression treatment [55], [56], and blood pressure monitoring [31], [57]. Recently, researchers are making recent progress in COVID-19 [7], [58], [59], [60]. The design of the mobile device not only proposes a viable mHealth solution and drives the further development of mHealth, but also generates a large amount of mHealth data in the process. Based on such massive data, various machine learning models have been developed, especially some online learning models and incremental learning models are proposed [14], [15], [16], [17], in which the mHealth models would continuously update over time as more information is collected and made available. There are also many researchers who focus on the integration of Bayesian methods into mobile health [61], [62], [63]. However, these works are currently considering designs of hardware devices and ML models' improvements. Few of them consider the data

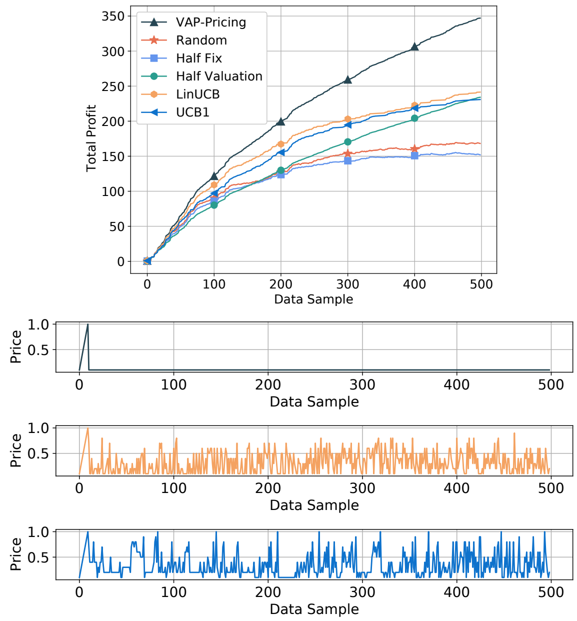


Fig. 8. Performance and Price when reserve value $f_4(v)$=0.

acquisition mechanism, neither data valuation, and data pricing mechanism. Barriers still exist in the journey of mHealth data from generation to use.

7.2 Data Valuation and Pricing for ML Tasks

Lately, Shapley value has been widely used in the data valuation and pricing problem for ML tasks. Agarwal *et al.* [18] design a market mechanism to price training data and match buyers to sellers based on Shapley value. Jia *et al.* introduce several additional approximation methods for efficient computation of Shapley values for training data [19]; subsequently, they provided an algorithm for the exact computation of Shapley values for the specific case of nearest-neighbour classifiers [22]. Meanwhile, Ghorbani *et al.* developed a truncated Monte Carlo sampling scheme (TMC-Shapley), demonstrating empirical effectiveness across various ML tasks [20]; subsequently, they proposed distributional Shapley, where the value of a point is defined in the context of an underlying data distribution [21]. However, these data valuation methods are not suitable for online ML tasks. Despite not being used for data valuation, ranking the importance of training data points has been used for understanding model behaviors, detecting data set errors, etc. Existing methods include using the influence function [64] for smooth parametric models, and a variant [65] for non-parametric ones. Ogawa *et al.* [66] proposed rules to identify and remove the least influential data to reduce the computation cost when training support vector machines (SVM). Kendall *et al.* measured the uncertainties in Bayesian deep learning for computer vision [67]. These approaches could potentially be used for valuing data.

7.3 Uncertainty in Machine Learning

The VAP-Valuation metric is closely related to the concept of epistemic uncertainty in machine Learning. In Bayesian modeling, there are two main types of uncertainty one can model. Aleatoric uncertainty comes from the noise when data is generated or collected, for example, sensor noise or motion noise. Aleatoric uncertainty cannot be reduced even if more data were to be collected. While epistemic uncertainty comes from the model's ignorance of the data when the collected data is not enough. This uncertainty can be explained away given enough data and is often referred to as model uncertainty. These two uncertainties were first studied and classified by Kiureghian and Ditlevsen [68]. And these two types of uncertainty have further been more specifically studied in bayesian deep learning for computer vision by Kendall and Gal [69]. Before that, Gal and Ghahramani proved that deep neural networks could be cast as performing approximate variational inference in a Bayesian setting [70] and extend it to arbitrary deep learning models [71]. Based on that, they model uncertainty with dropout NNs [72]. In previous work, uncertainty is the predictive distribution variance in the prediction task for the current test data to judge the credibility of a prediction. However, in VAP-Valuation, we calculate the posterior distribution entropy reduction of parameter or the predictive distribution variance of new data to measure each data's contribution.

7.4 Multi-armed Bandits

The multi-armed bandit (MAB) problem is a sequential decision-making model and widely studied by many works with different models and solutions, such as upper confidence bound [40], ϵ-greedy [73], and Thompson sampling [74], [75]. In the traditional setting of MAB, an arm can be represented by a scalar to infer the reward that is drawn from its distribution which is unknown to the player, while in the contextual bandit [41], [43], a context vector represents each arm, and $\tilde{O}(\sqrt{T})$ regret bounds can be achieved based on UCB. Moreover, as classical modeling, the linear reward model has been widely studied in contextual bandits [76], [77]. Considering knapsack constraints on various resources in the bandit framework, the bandit with knapsack (BwK) is first studied by Badanidiyuru *et al.* [78], who presented two algorithms and proved that the regret achieved by both algorithms is optimal up to polylogarithmic factors. Later, based on the optimal regret, Agrawal and Devanur further proposed alternative optimal algorithms under concave rewards convex knapsacks [45], and a linear contextual setting [44]. Many real-world problems can be modeled as various versions of bandit problems [79], [80], [81], because MAB represents an online learning paradigm that naturally captures the intrinsic exploration-exploitation tradeoff in sequential decision-making process.

8 Conclusion

In this work, we have introduced VAP, an innovative online data valuation and pricing mechanism designed specifically for ML tasks in the context of mobile health (mHealth). We value the data by measuring its contribution to the ML model under the Bayesian perspective, using the entropy of the distributions over model parameters. To address the profit maximization problem, we have developed an online posted price data pricing mechanism within a contextual multi-armed bandit framework, leveraging the data valuation metric provided by VAP. And further, for the limited budget situation, we have proposed VAP-PricingwK under a multi-armed bandit with a knapsack framework. Moreover, we have extended VAP from Bayesian linear regression to more complex ML models by computing the entropy from the parameter space to the prediction space.

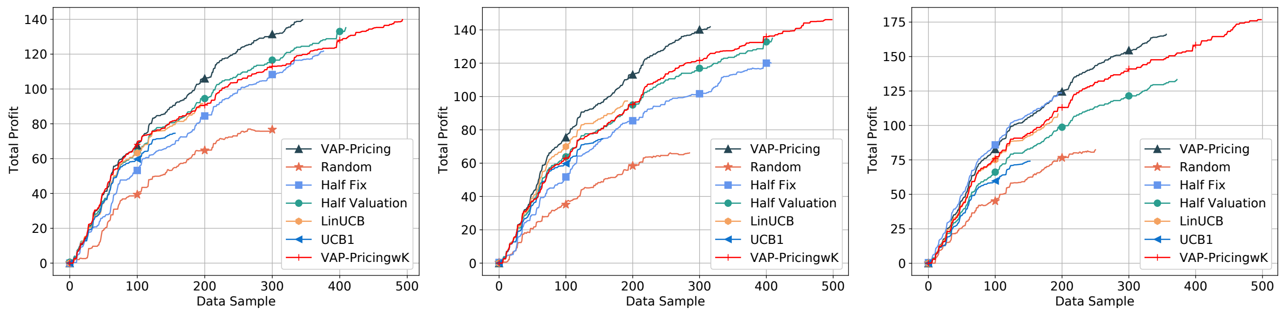


Fig. 9. Performance of different data pricing mechanisms under a fixed limited budget $B = 100$. Reserve values' distributions from left to right are $f_1(v)$, $f_2(v)$, and $f_3(v)$.

Through comprehensive evaluation, we have demonstrated that VAP outperforms existing online data valuation and pricing mechanisms. The results highlight the effectiveness and superiority of our approach in the mHealth domain.

References

[1] S. Kumar, W. Nilsen, M. Pavel, and M. B. Srivastava, "Mobile health: Revolutionizing healthcare through transdisciplinary research," *Computer*, vol. 46, no. 1, pp. 28–35, 2013.

[2] "Apple health," https://www.apple.com/ios/health/.

[3] "Google fit," https://www.google.com/fit/.

[4] "Microsoft health," https://www.microsoft.com/en-us/industry/health/microsoft-cloud-for-healthcare.

[5] R. S. Istepanian and T. Al-Anzi, "m-health 2.0: new perspectives on mobile health, machine learning and big data analytics," *Methods*, vol. 151, pp. 34–40, 2018.

[6] Q. Wu, X. Chen, Z. Zhou, and J. Zhang, "Fedhome: Cloud-edge based personalized federated learning for in-home health monitoring," *IEEE Transactions on Mobile Computing*, vol. 21, no. 8, pp. 2818–2832, 2020.

[7] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo, "Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data," in *KDD*, 2020, pp. 3474–3484.

[8] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, vol. 1, no. 3, 2016.

[9] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med Image Anal*, vol. 42, pp. 60 – 88, 2017.

[10] D. C. Mohr, M. Zhang, and S. M. Schueller, "Personal sensing: understanding mental health using ubiquitous sensors and machine learning," *Annu. Rev. Clin. Psychol.*, vol. 13, pp. 23–47, 2017.

[11] G. Xing, "Tackling the challenges of machine learning for mobile health systems," in *HealthDL*, 2020.

[12] S. Arora, J. Yttri, and W. Nilsen, "Privacy and security in mobile health (mhealth) research," *Alcohol research: current reviews*, vol. 36, no. 1, p. 143, 2014.

[13] S. Kumar, W. J. Nilsen, A. Abernethy, A. Atienza, K. Patrick, M. Pavel, W. T. Riley, A. Shar, B. Spring, D. Spruijt-Metz *et al.*, "Mobile health technology evaluation: the mhealth evidence workshop," *Am J Prev Med*, vol. 45, no. 2, pp. 228–236, 2013.

[14] C. Hu, Y. Chen, L. Hu, and X. Peng, "A novel random forests based class incremental learning method for activity recognition," *Pattern Recognit.*, vol. 78, pp. 277–290, 2018.

[15] D. A. Jenkins, M. Sperrin, G. P. Martin, and N. Peek, "Dynamic models to predict health outcomes: current status and methodological challenges," *Diagn Progn Res*, vol. 2, no. 1, p. 23, 2018.

[16] K. Y. Ngiam and W. Khor, "Big data and machine learning algorithms for health-care delivery," *Lancet Oncol.*, vol. 20, no. 5, pp. e262–e273, 2019.

[17] S. Srinivasan, K. R. Srivatsa, I. V. R. Kumar, R. Bhargavi, and V. Vaidehi, "A regression based adaptive incremental algorithm for health abnormality prediction," in *ICRTIT*, 2013, pp. 690–695.

[18] A. Agarwal, M. Dahleh, and T. Sarkar, "A marketplace for data: An algorithmic solution," in *EC*, 2019, pp. 701–726.

[19] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. Hynes, N. M. Gürel, B. Li, C. Zhang, D. Song, and C. J. Spanos, "Towards efficient data valuation based on the shapley value," in *AISTATS*, 2019, pp. 1167–1176.

[20] A. Ghorbani and J. Y. Zou, "Data shapley: Equitable valuation of data for machine learning," in *ICML*, 2019, pp. 2242–2251.

[21] A. Ghorbani, M. P. Kim, and J. Zou, "A distributional framework for data valuation," in *ICML*, 2020, pp. 3535–3544.

[22] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gürel, B. Li, C. Zhang, C. J. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1610–1623, 2019.

[23] P. Dubey, "On the uniqueness of the shapley value," *Int. J. Game Theory*, vol. 4, no. 3, pp. 131–139, 1975.

[24] A. V. Goldberg, J. D. Hartline, and A. Wright, "Competitive auctions and digital goods," in *SODA*, 2001, pp. 735–744.

[25] S. Alaei, A. Malekian, and A. Srinivasan, "On random sampling auctions for digital goods," in *EC*, 2009, pp. 187–196.

[26] R. Kleinberg and T. Leighton, "The value of knowing a demand curve: bounds on regret for online posted-price auctions," in *FOCS*, 2003, pp. 594–605.

[27] A. Xu, Z. Zheng, F. Wu, and G. Chen, "Online data valuation and pricing for machine learning tasks in mobile health," in *INFOCOM*, 2022, pp. 850–859.

[28] R. Kleinberg and T. Leighton, "The value of knowing a demand curve: Bounds on regret for online posted-price auctions," in *FOCS*, 2003, pp. 594–605.

[29] J. D. Hartline and B. Lucier, "Bayesian algorithmic mechanism design," in *STOC*, 2010, pp. 301–310.

[30] S. A. Baldwin and M. J. Larson, "An introduction to using bayesian linear regression with clinical data," *Behaviour research and therapy*, vol. 98, pp. 58–75, 2017.

[31] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, "Cufless blood pressure estimation algorithms for continuous health-care monitoring," *IEEE. Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 859–869, 2017.

[32] W. Wang, S. Mirjafari, G. Harari, D. Ben-Zeev, R. Brian, T. Choudhury, M. Hauser, J. Kane, K. Masaba, S. Nepal *et al.*, "Social sensing: Assessing social functioning of patients living with schizophrenia using mobile phone sensing," in *CHI*, 2020, pp. 1–15.

[33] J. Q. Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *J. Mach. Learn. Res.*, vol. 6, pp. 1939–1959, 2005.

[34] I. Kononenko, "Bayesian neural networks," *Biological Cybernetics*, vol. 61, no. 5, pp. 361–370, 1989.

[35] G. C. McDonald, "Ridge regression," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 1, no. 1, pp. 93–100, 2009.

[36] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley, 2001.

[37] S. Fujishige, *Submodular functions and optimization*. Elsevier, 2005.

[38] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[39] L. Xu, C. Jiang, Y. Qian, Y. Zhao, J. Li, and Y. Ren, "Dynamic privacy pricing: A multi-armed bandit approach with time-variant rewards," *IEEE Trans. Inf. Forensics Secur.*, pp. 271–285, 2017.

[40] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, no. 2-3, pp. 235–256, 2002.

[41] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *WWW*, 2010, pp. 661–670.

[42] T. M. Rassias and T. M. Rassias, *Functional equations, inequalities, and applications*. Springer, 2003.

[43] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *NeurIPS*, 2011.

[44] S. Agrawal and N. Devanur, "Linear contextual bandits with knapsacks," in *NeurIPS*, 2016.

[45] S. Agrawal and N. R. Devanur, "Bandits with concave rewards and convex knapsacks," in *EC*, 2014, pp. 989–1006.

[46] D. Golovin, A. Krause, and D. Ray, "Near-optimal bayesian active learning with noisy observations," in *NeurIPS*, 2010, pp. 766–774.

[47] G. Chalkiadakis and C. Boutilier, "Bayesian reinforcement learning for coalition formation under uncertainty," in *AAMAS*, 2004, pp. 1090–1097.

[48] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.

[49] J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the adap learning algorithm to forecast the onset of diabetes mellitus," in *Proc Annu Symp Comput Appl Med Care*, 1988, p. 261.

[50] D. Preuveneers and Y. Berbers, "Mobile phones assisting with health self-care: a diabetes case study," in *HCI*, G. H. ter Hofte, I. Mulder, and B. E. R. de Ruyter, Eds., 2008.

[51] S. Kitsiou, G. Paré, M. Jaana, and B. Gerber, "Effectiveness of mhealth interventions for patients with diabetes: an overview of systematic reviews," *PloS one*, p. e0173160, 2017.

[52] M. J. Reading, E. M. Heitkemper, M. Lor, and L. Mamykina, "Exploring mhealth intervention designs to engage low-income, minority adults with type 2 diabetes in self-monitoring," in *AMIA*, 2018.

[53] U. Fareed, "Smartphone sensor fusion based activity recognition system for elderly healthcare," in *MobileHealth@MobiHoc*, 2015, pp. 29–34.

[54] S. Laskaridis, D. Spathis, and M. Almeida, "Federated mobile sensing for activity recognition," in *MobiCom*, 2021, pp. 858–859.

[55] R. Wang, W. Wang, A. DaSilva, J. F. Huckins, W. M. Kelley, T. F. Heatherton, and A. T. Campbell, "Tracking depression dynamics in college students using mobile phone and wearable sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 43:1–43:26, 2018.

[56] A. C. Myers, L. Chesebrough, R. Hu, M. R. Turchioe, J. Pathak, and R. M. Creber, "Evaluating commercially available mobile apps for depression self-management," in *AMIA*, 2020.

[57] N. Bui, N. Pham, J. J. Barnitz, Z. Zou, P. Nguyen, H. Truong, T. Kim, N. Farrow, A. Nguyen, J. Xiao, R. R. Deterding, T. N. Dinh, and T. Vu, "ebp: A wearable system for frequent and comfortable blood pressure monitoring from user's ear," in *MobiCom*, 2019, pp. 1–17.

[58] A. Asadzadeh and L. R. Kalankesh, "A scope of mobile health solutions in covid-19 pandemics," *Informatics in medicine unlocked*, p. 100558, 2021.

[59] A. Raposo, L. Marques, R. Correia, F. Melo, J. Valente, T. Pereira, L. B. Rosário, F. Froes, J. Sanches, and H. P. d. Silva, "e-covig: a novel mhealth system for remote monitoring of symptoms in covid-19," *Sensors*, p. 3397, 2021.

[60] C. P. Adans-Dester, S. Bamberg, F. P. Bertacchi, B. Caulfield, K. Chappie, D. Demarchi, M. K. Erb, J. Estrada, E. E. Fabara, M. Freni *et al.*, "Can mhealth technology help mitigate the effects of the covid-19 pandemic?" *IEEE Open J. Eng. Med. Biol.*, pp. 243–248, 2020.

[61] A. N. Repaka, S. D. Ravikanti, and R. G. Franklin, "Design and implementing heart disease prediction using naives bayesian," in *ICOEI*, 2019, pp. 292–297.

[62] A. Ragav and G. K. Gudur, "Bayesian active learning for wearable stress and affect detection," *arXiv preprint arXiv:2012.02702*, 2020.

[63] M. D. Koslovsky, E. T. Hébert, M. S. Businelle, and M. Vannucci, "A bayesian time-varying effect model for behavioral mhealth data," *Ann Appl Stat*, vol. 14, no. 4, p. 1878, 2020.

[64] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *ICML*, 2017, pp. 1885–1894.

[65] B. Sharchilev, Y. Ustinovskiy, P. Serdyukov, and M. Rijke, "Finding influential training samples for gradient boosted decision trees," in *ICML*, 2018, pp. 4577–4585.

[66] K. Ogawa, Y. Suzuki, and I. Takeuchi, "Safe screening of non-support vectors in pathwise svm computation," in *ICML*, 2013, pp. 1382–1390.

[67] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, 2017, pp. 5574–5584.

[68] A. Der Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?" *Structural safety*, vol. 31, no. 2, pp. 105–112, 2009.

[69] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *NeurIPS*, 2017, pp. 5574–5584.

[70] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Insights and applications," in *Deep Learning Workshop, ICML*, 2015.

[71] ——, "On modern deep learning and variational inference," in *Advances in Approximate Bayesian Inference workshop, NeurIPS*, 2015.

[72] ——, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016, pp. 1050–1059.

[73] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," *NeurIPS*, 2007.

[74] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *ICML*, 2013, pp. 127–135.

[75] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3-4, pp. 285–294, 1933.

[76] W. Chu, L. Li, L. Reyzin, and R. E. Schapire, "Contextual bandits with linear payoff functions," in *AISTATS*, 2011, pp. 208–214.

[77] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *ICML*, 2013, pp. 127–135.

[78] A. Badanidiyuru, R. Kleinberg, and A. Slivkins, "Bandits with knapsacks," in *FOCS*, 2013, pp. 207–216.

[79] H. Wang, Y. Yang, E. Wang, W. Liu, Y. Xu, and J. Wu, "Truthful user recruitment for cooperative crowdsensing task: A combinatorial multi-armed bandit approach," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4314–4331, 2023.

[80] C. Zeng, Q. Wang, S. Mokhtari, and T. Li, "Online context-aware recommendation with time varying multi-armed bandit," in *KDD*, 2016, pp. 2025–2034.

[81] F. Vannella, A. Proutière, Y. Jedra, and J. Jeong, "Learning optimal antenna tilt control policies: A contextual linear bandit approach," in *INFOCOM*, 2022, pp. 740–749.

Anran Xu is a Ph.D candidate in the Department of Computer Science and Engineering at Shanghai Jiao Tong University, P. R. China. She received her B.S. degree in Software Engineering from Shandong University, P.R.China, in 2019. Her research interests include algorithmic game theory, mobile computing, and computational advertising. For more information, please visit https://anran-xu.github.io/

Zhenzhe Zheng is an assistant professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received the B.E. in Software Engineering from Xidian University, in 2012, and the M.S. degree and the Ph.D. degree in Computer Science and Engineering from Shanghai Jiao Tong University, in 2015 and 2018, respectively. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a Post Doc Research Associate from 2018 to 2019. His research interests include game theory and mechanism design, networking and mobile computing, and online marketplaces. He is a recipient of the China Computer Federation (CCF) Excellent Doctoral Dissertation Award 2018, Google Ph.D. Fellowship 2015 and Microsoft Research Asia Ph.D. Fellowship 2015. He has served as the member of technical program committees of several academic conferences, such as MobiHoc, AAAI, MSN, IoTDI and etc. He is a member of the ACM, IEEE, and CCF. For more information, please visit https://zhengzhenzhe220.github.io/

Qinya Li received her B.S. degree in Computer Science and Engineering from Northeastern University, P.R.China in 2015, and the Ph.D. degree in Computer Science and Engineering from Shanghai Jiao Tong University in 2020. Her research interests include mobile computing, mobile crowdsourcing, and algorithmic game theory and its applications.

Fan Wu is a professor in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received his B.S. in Computer Science from Nanjing University in 2004, and Ph.D. in Computer Science and Engineering from the State University of New York at Buffalo in 2009. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a Post Doc Research Associate. His research interests include wireless networking and mobile computing, data management, algorithmic network economics, and privacy preservation. He has published more than 200 peer-reviewed papers in technical journals and conference proceedings. He is a recipient of the first class prize for Natural Science Award of China Ministry of Education, China National Fund for Distinguished Young Scientists, ACM China Rising Star Award, CCF-Tencent “Rhinoceros bird” Outstanding Award, and CCF-Intel Young Faculty Researcher Program Award. He has served as an associate editor of IEEE Transactions on Mobile Computing and ACM Transactions on Sensor Networks, an area editor of Elsevier Computer Networks, and as the member of technical program committees of more than 100 academic conferences. For more information, please visit http://www.cs.sjtu.edu.cn/~fwu/.

Guihai Chen earned his B.S. degree from Nanjing University in 1984, M.E. degree from Southeast University in 1987, and Ph.D. degree from the University of Hong Kong in 1997. He is a distinguished professor of Shanghai Jiao Tong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan in 1998, University of Queensland, Australia in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor networks, peer-to-peer computing, high-performance computer architecture and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, Wireless Networks, The Computer Journal, International Journal of Foundations of Computer Science, and Performance Evaluation, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS and ICDCS.