Privacy Leakage From Dynamic Prices: Trip Purpose Mining as an Example

Suiming Guo, Chao Chen, Zhetao Li, *Member, IEEE*, Chengwu Liao, Yaxiao Liu, Ke Xu, *Fellow, IEEE*, and Daqing Zhang, *Fellow, IEEE*

***Abstract*—Dynamic prices are used in many scenarios, e.g., flight ticketing, hotel room booking and ride-on-demand (RoD) service such as Uber and DiDi, and while they are beneficial for service providers, practitioners or users, they lead to the concern of privacy leakage – the possibility of learning user information from dynamic prices. In this paper, we aim to study this possibility and choose trip purpose mining in RoD service as an attack example, based on real-world large datasets. We discuss the criteria of choosing datasets – ubiquitous, collective and easily accessible – from the perspective of an attacker, and extract features describing trip information, spatio-temporal and dynamic prices context. The trip purpose mining problem is then solved as a multi-class classification problem and multiple binary-class problems. In the multi-class problem, we verify that dynamic prices information results in a 17.1% improvement in classification accuracy; in the binary-class problems, we quantify feature contributions and explain the different extents of privacy leakage in identifying different trip purposes. Our hope is that the study not only serves as a case study demonstrating the privacy leakage problem in RoD service, but also sheds light on such privacy problem in other services using dynamic prices and triggers more research efforts.**

***Index Terms*—Dynamic prices, privacy, trip purpose, urban transportation.**

I. Introduction

Dynamic prices, in various forms, have been widely applied in different scenarios. For example, prices of flight tickets may fluctuate based on supply, demand, and the time of buying a ticket; hotel room prices may be dependent on holiday seasons, room availability, and the time of creating an order. In ride-on-demand (RoD) services, such as Uber and DiDi, surge pricing, or dynamic pricing, mechanisms are used to determine the prices of each trip based on the supply and demand on the road [1], [2].

The introduction of dynamic prices brings benefits to service providers, practitioners, and users. The service provider could design pricing policies or algorithms for profit maximization. A user could choose an appropriate time to find an affordable price by observing how dynamic prices change over time; s/he could also take advantage of dynamic prices and get timely or better service if the user is in a hurry [3]. For practitioners, e.g., a driver in RoD service, it is possible to rank seeking locations based on dynamic prices so as to earn more [4], [5], [6].

Despite these benefits, dynamic prices lead to privacy leakage to a certain extent. Intuitively, business or family travellers may order hotel rooms or buy flight tickets at different times, due to different levels of price sensitiveness. In RoD service, one going to work in the morning rush hour may not care about dynamic prices because s/he could not be late; but a city traveller, even at the same time and location, may change travel plan or take a trip later to get a lower price. Hence, an order with a high dynamic price during morning rush hours has a high possibility belonging to a commuter, whereas an order with a lower dynamic price that happens later is more probable to be from a traveller. Therefore, it is possible that by observing the change of dynamic prices among orders, an attacker could violate user privacy and learn more information such as user identity, demographic characteristics, or trip purpose.

Motivated by the concern of privacy leakage, in this paper we aim to reveal the possibility of capturing privacy-related information from dynamic prices. Specifically, we choose trip purpose in RoD service as the target. Trip purpose is a vital user privacy because by mining and understanding it, an attacker is able to conduct some misbehaviors or even crimes, which are roughly divided as mild and severe attacks below:

- *Mild attacks* usually involve targeted advertisement or service/product recommendation and may not cause severe privacy violation or monetary loss. For example, if an attacker concludes that passengers heading towards specific regions during weekends are mostly going shopping, then it is possible to recommend products, coupons or shops to them. Our results of trip purpose mining should be applicable to this type of attacks.
- *Severe attacks* go further and usually violate the privacy of particular passengers(s). Examples include tracking and stalking – i.e., monitoring or following particular passenger(s) – with trip purpose, one could guess the commuting

Manuscript received 7 May 2023; revised 24 April 2024; accepted 28 May 2024. Date of publication 3 June 2024; date of current version 5 November 2024. This work was supported by Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515012094, in part by the National Natural Science Foundation of China under Grant 62002135, Grant 62322601, Grant 62425201, and Grant 62172066, and by the Excellent Youth Foundation of Chongqing under Grant CSTB2023NSCQJQX0025. Recommended for acceptance by S. K. Das. *(Corresponding author: Chao Chen.)*

Suiming Guo and Zhetao Li are with the College of Information Science and Technology, Jinan University, Guangzhou 510632, China (e-mail: guosuiming@email.jnu.edu.cn; liztchina@hotmail.com).

Chao Chen and Chengwu Liao are with Chongqing University, Chongqing 400044, China (e-mail: cschaochen@cqu.edu.cn; lcw@cqu.edu.cn).

Yaxiao Liu and Ke Xu are with Tsinghua University, Beijing 100084, China (e-mail: rootliu@gmail.com; xuke@mail.tsinghua.edu.cn).

Daqing Zhang is with Institut Mines-Telecom/Telecom SudPais, 91000 Paris, France (e-mail: daqing.zhang@telecom-sudparis.eu).

Digital Object Identifier 10.1109/TMC.2024.3408419

routes, the time of commuting, weekend destinations, family address, living habit, etc. Severe attacks may even cause monetary loss, e.g., a guess of family address may lead to stealing, robbing, etc. To conduct severe attacks, it maybe necessary to associate trip purpose with particular passengers, and thus techniques such as trajectory-user linking, which are not in the scope of this paper, are preferred.

Our study plays a role in two sides of privacy preservation. On one hand, we study such possibility, and inform attackers that dynamic prices indeed help them learn more information about trip purpose. On the other hand, the study on privacy leakage from dynamic prices is rare, and our work could thus alarm the privacy leakage problem in similar scenarios (e.g., RoD service, hotel room pricing, flight ticket pricing, etc.), triggering more efforts in designing proper privacy-preserving mechanisms.

From the perspective of an attacker, the methodology of trip purpose mining is different from previous work. A typical way of trip purpose mining usually involves a specific group of users, and by collecting their trip preferences, household information or demographic characteristics through travel survey or questionnaire, trip purposes of users belonging to this group could be inferred. For an attacker, such datasets or information is generally inaccessible, and it is also difficult to infer preferences based on data from only a small group of users. Instead, an attacker pays attention to *all* users, and attempts to mine trip purpose based on *ubiquitous* and *easily-accessible* datasets. Datasets fulfilling such requirements are either public data about, for example, city map, city planning, survey on a large population, or ubiquitous datasets such as trip order, GPS trajectories that are easily accessible and not targeted to individual or small groups of users. In our study, we choose trip order datasets from real RoD service with dynamic prices. We also crawl POI and public transportation data from online map service.

We aim to answer the following two questions:

- *Q1*: Do dynamic prices help an attacker in trip purpose mining?
- *Q2*: What features are quantitatively more important in trip purpose mining?

These two questions basically address the privacy leakage problem qualitatively and quantitatively, respectively. To answer **Q1**, we extract features from our datasets to describe trip information, and to augment the spatial, temporal and dynamic prices context of each trip. We then build an artificial neural network (ANN) model to fuse all these features to perform trip purpose mining. We not only show that it is possible to achieve a satisfactory and convincing accuracy with basic datasets and a simple algorithm, but also, more importantly, prove that considering features related to dynamic prices leads to a significant accuracy improvement. To answer **Q2**, we resort to a linear model to fuse these features, so that we could quantify and rank the contribution of different features in trip purpose mining. For the linear model, we also use feature crossing to compensate for the lack of non-linearity. To make sure that our linear model corresponding to **Q2** provides an high enough accuracy, the ANN model corresponding to **Q1** also serves as a baseline.

Our contributions are three-fold:

- To the best of our knowledge, we are the first to investigate the privacy leakage from dynamic prices based on real large datasets in RoD service. Though we choose trip purpose mining as an attack example, a lot of other services or scenarios involve dynamic prices in various forms. We hope our study could shed light on such privacy problem and trigger more research efforts.
- We design a simple but effective attack methodology for attackers to find out trip purpose of any user in RoD service, including:
  - We state clearly the datasets needed, and the possible ways to obtain them, from an attacker's perspective. It is thus feasible for an attacker to conduct trip purpose mining attack, and our discussions on privacy leakage from dynamic prices become convincing and tenable.
  - We not only confirm the existence of privacy leakage from dynamic prices, but also give quantitative evaluations of the impacts from different features on various trip purposes. Even the privacy leakage from dynamic prices is rarely studied, let alone the quantitative analysis. Our results are the first, as we know, to give the extent of privacy leakage for different trip purposes.
- We train our models and conduct extensive experiments based on real service datasets. On one hand, our datasets are from real service, and contain information such as trip purpose labels and dynamic prices. On the other hand, real service datasets guarantee that our results are tenable and could be used in practice.

The remainder of the paper is organized as follows. We review related work in Section II. In Section III we discuss the criteria of choosing datasets and present our datasets in details. Section IV shows the attack methodology for trip purpose mining. Evaluation results and discussions on relevant problems are presented in Section V. Finally, Section VI concludes the paper.

II. Related Work

RoD Service with Dynamic Prices. RoD service, also known as on-demand ride-hailing, ridesourcing, or transportation network company (TNC), is a relatively new transportation service compared to taxi. There are indeed subtle differences between these terminologies, but we do not go into details here. There are much fewer studies on this new service, many of which compare it with other public transportation services such as taxi from a number of perspectives. For example, [7], [8], [9] discuss the changes of market share of taxi, Uber or other public transportation services before and after Uber's entrance. [10] studies users' choices and preferences between ridesourcing and taxi. [11] claims that Uber can reduce passenger waiting time significantly. [12] studies the effects of TNC on traffic congestion. [13] compares Uber with taxi service spatio-temporally. [14], [15] pay attention to the market effects of Uber's entrance. [16] gives a comprehensive review of the studies on demand and pricing, supply and incentives, platform operations, competition, as well as impacts and regulations in ridesourcing. [17] also reviews the studies on the relationship between taxi (and public transit) and ridesourcing, demographics and spatial context, regulation, and etc.

Dynamic pricing is a key feature of RoD service that distinguishes it from taxi service, and there are studies on the design of pricing schemes, the effects of dynamic prices, as well as new applications built on dynamic prices. [18] studies the modelling and pricing in ride-sourcing. [19] designs a reward scheme integrated with dynamic pricing and studies passenger utility, driver income and platform revenue under such scheme. [20], [21], [22] focus on the effects in reducing passenger waiting time, balancing the supply and demand, and increasing driver revenue. [2] mines data from Uber and evaluates its surge pricing mechanism, by simulating app users on key locations. The authors in [1], [3], [4], [6], [23] study the demand, the effects of dynamic pricing, passengers' reaction to prices, dynamic price prediction, drivers' seeking strategies, and seeking route recommendation, based on real-world data. Other studies analyze RoD service from economic perspectives on topics such as supply response [24], supply elasticity [25], [26], customer retention [27] or consumer surplus [28].

Privacy Issues from Dynamic Prices. For traditional public transportation services such as taxi or bus, dynamic pricing mechanisms are not used so there is not any privacy issue generated from dynamic prices. For RoD service, there is currently not any study on this topic.

For smart meters used in recording electricity usage by power utilities, different forms of dynamic prices are used to manipulate the electricity usage. Privacy issues appear as people worry that the smart meter readings make it possible to infer electricity usage, guess family lifestyle, etc. [29] designs a privacy-preserving service outsourcing scheme when the power utility outsources the dynamic pricing determination to third parties. [30] proposes battery-based methods to flatten smart meter readings so that privacy could be preserved. [31] preserves privacy in real-time smart metering data based on the concept of differential privacy.

Privacy-preserving pricing schemes are also studied. In retail industry, for example, [32] proposes a theoretical pricing scheme based on differential privacy to prevent a third party agent from inferring personalized information and purchase decisions from price changes.

Trip Purpose Mining. Human mobility always shows regular patterns [33], [34], and by studying these patterns, it is reasonable and realistic to study social dynamics and personal travel behavior [35], [36], [37], [38], [39], [40]. Trip purpose mining is one of the many categories of study under travel behavior analysis.

There are a number of previous studies on trip purpose mining, inference or imputation. [41] gives a comprehensive survey on the related work on trip purpose imputation. Trip purpose mining studies could be roughly divided into two categories. The first category relies on surveys or questionnaires, and extracts trip preferences of a specific group of users to perform trip purpose mining. These studies are targeted to a small group of people and thus may have a higher accuracy, but they have the disadvantage that it is infeasible for an attacker to collect such data. For example, [42] uses online web-based diary and their dataset covers 260 participants; [43] is based on social media data and mail-based diary and studies the data from about 40,000 households; [44] chooses social network data and travel household survey; [45] uses smartphone-based travel surveys, etc.

The second category is based on large-scale datasets that do not contain individual preference information, such as GPS trajectories of taxis. These studies are generally less accurate due to the lack of passenger-specific data and a larger coverage of population, time periods, or locations. For example, [46] uses an unsupervised learning model to cluster trajectories of similar trip purpose; [47] uses a semi-supervised deep graph embedding framework; and [48] uses a probabilistic model. [49] uses a dual-attention graph embedding network to conduct trip purpose mining based on taxi GPS trajectories and text description of destinations.

Another relevant field of study related is raw mobility annotation. These studies usually first find out stays among a human mobility trace, and then associate each stay with a POI that people are most likely to visit during that stay. In other words, a mobility trace is annotated with POIs. For example, [50] considers basic POI attributes and POI-visit histories; [51] further uses Markov models to consider POI-visit histories; [52] fuses information from social media to conduct raw mobility annotation; and [53] fuses multiple context factors with a neural network. Raw mobility annotation is related to trip purpose mining because if important POI visits are determined, then it is possible to infer the corresponding trip purpose.

Compared to all the related work listed above, our work focuses on the existence of privacy leakage, instead of privacy-preserving mechanisms or pricing algorithms. We choose a concrete example, i.e., trip purpose, as the target of study, and use real service data to show the privacy leakage from dynamic prices. Furthermore, we also quantify feature contributions – what features about dynamic prices lead to privacy leakage and by how much – by using suitable machine learning models.

III. Datasets and Preprocessing

Datasets used by attackers should be different from the datasets used by previous work on trip purpose mining. Traditionally, a typical way involves gathering a group of people (e.g., students or volunteers), collecting their mobility data (such as trip orders or GPS trajectories), recording their trip purposes as ground truths, modelling mobility preferences of this small group of people by taking surveys or questionnaires, and finally building a model of trip purpose mining. Because of modelling mobility preferences specifically to a small group of people, the derived model is not suitable to a larger population.

Such datasets are also infeasible for an attacker. Firstly, an attacker is not able to gather a group of people and collect individual and private data. Secondly, an attacker pays attention to all passengers – mining the trip purpose of *any* passenger, given a trained model and the mobility data of this passenger, instead of mining the trip purpose of any passenger in a small group with specific preferences.

TABLE I
A SUMMARY OF OUR DATASETS AND PRE-PROCESSING

Dataset	Data entry	Pre-processing
Trip Order	$<\boldsymbol{p_o}, \boldsymbol{p_d}, \boldsymbol{desc_d}>$: the spatio-temporal point of origin and destination, and text description of destination, of a trip order.	$\boldsymbol{a}$: trip purpose label derived from $desc_d$ with a pre-trained NLP model.
Dynamic Prices	(event time, event location, estimated trip fare, price multiplier, user ID) of an EstimateFee event.	We calculate the average price multi--plier in each cell, each hour.
POI Data	$\boldsymbol{POI_o}$ and $\boldsymbol{POI_d}$: the 14-element vectors of POI counts around origin and destination of each order in the trip order dataset.	
Bus & Metro Distribution	$\boldsymbol{BM_o}$ and $\boldsymbol{BM_d}$: the 4-element vectors of the number of bus & metro lines and stations around origin and destination of each order in the trip order dataset.	

Based on the above discussions, datasets for an attacker should have the following characteristics:

- *Ubiquitous:* these datasets should be generated by ubiquitous devices or from ubiquitous behavior. For example, GPS trajectories or trip order datasets satisfy this requirement. On the contrary, surveys or questionnaires manually created by hands are not desirable.
- *Collective, not Individual:* these datasets should cover all passengers, instead of individuals or a small group of passengers. In other words, we treat all passengers as equals, and do not consider any personal preference. Hence, if two passengers get on and off cars at exactly the same locations and times, they should have the same trip purpose.
- *Easily accessible:* these datasets should be easily accessible for an attacker. For example, they should be open or public data, or could be approximated by open or public data, or could be collected by means of, say, crowdsourcing.

These characteristics not only facilitate trip purpose mining for an attacker, but also improve the applicability of our study, so that our results are meaningful and could be applied to similar services.

Below we introduce our datasets in details, and Table I summarizes our datasets and pre-processing.

A. *Trip Order Dataset*

The trip order dataset is widely used in mobility-related studies. This dataset describes the origin-destination information (i.e., OD pair) of each order in a RoD service. For each order, we use a spatio-temporal point $p_o = (lng_o, lat_o, t_o)$ to represent the origin of an order, with lng_o, lat_o and t_o as the longitude, latitude, and time that an order starts with. Similarly, the spatio-temporal point $p_d = (lng_d, lat_d, t_d)$ represents the destination of and order. In addition, for the order destination, there is a text description $desc_d$ that gives out the context of the destination. So $< p_o, p_d, desc_d >$ presents all relevant information of a trip order. We show an example of lng_d, lat_d and $desc_d$ below:

$$lng_d = 116.59318, lat_d = 40.07919,$$
$$desc_d = ''\text{Terminal 2 of Beijing International Airport}''. \quad (1)$$

The extra text description of destination, $desc_d$, comes from the mobile app in RoD service. In a RoD service such as Uber and DiDi, the passenger uses a mobile app to request for service: typing an origin and a destination in text, getting an estimate of the trip fare, and pressing a button to finally create an order [23]. The mobile app automatically transforms the origin and destination from text descriptions to longitudes and latitudes. This is different from the way of hailing a taxi, where no mobile app is involved, and the exact origin and destination locations, only in the form of longitude and latitude, are recorded by on-car GPS devices.

TABLE II
POI CATEGORIES AND THEIR CORRESPONDING TRIP PURPOSES ACCORDING TO THE CHECK-IN DATASET

Trip purpose #	POI category	Trip purpose
1	Recreation and Culture Facilities	Recreation
2	Outdoors and Sightseeing Places	Outdoors
3	Shop and Service Facilities	Shopping
4	Restaurant	Dining
5	School and Educational Facilities	Education
6	Transportation Facilities	Transportation
7	Apartment and Residence	Homing
8	Hospital and Clinic	Health
9	Office and Business Buildings	Working

The extra text description of destination enables us to generate a trip purpose label for each order, which is used as ground-truth in our model training and evaluation. This is impossible in studies on traditional taxi services based on GPS trajectories, in which ground-truth is usually obtained with the help of other auxiliary datasets such as travel diary, household survey, social network data, etc. As the auxiliary datasets may not cover the same locations or time periods as the GPS trajectories do, the ground-truth is inevitably inaccurate. By comparison, in our paper, the text description of destination is from the exactly same dataset, making it possible to extract more semantic information and generate the ground-truth more accurately.

Generating the ground-truth based on the text description of destination is not trivial. It should firstly be noted that this is auxiliary and does not belong to the attack methodology in Section IV. The trip purpose label is generated with the help from another location-based social network (LBSN) check-in dataset. This dataset is from Jiepang [54], and provides 510,000 text samples with information about POI descriptions and the corresponding POI categories in Beijing. In this check-in dataset, POIs in Beijing are divided into 9 categories, and each POI category is associated with one trip purpose, as shown in Table II. Specifically, the trip purpose label, denoted by a, is generated by a two-stage mapping operation:

- The first stage maps the text description to a POI category, e.g., from "Terminal 2 of Beijing International Airport" to "Transportation Facilities". This is done by using a pre-trained NLP model (i.e., ERNIE from PaddlePaddle, https://github.com/PaddlePaddle/ERNIE/tree/repro), fine-tuned with the above-mentioned check-in dataset. This model is used in [49] to perform similar trip purpose mapping operation, and is also used in other scenarios such as financial risk control, video recommendation and advertising [55]. Based on our trip order dataset, we manually transform the destination to POI categories of 1000 randomly chosen orders, and compare these POI categories with the NLP model's results. It is shown that the average mapping accuracy of this NLP model is around 99.3$\%$.
- The second stage maps a POI category to a trip purpose, e.g., from "Transportation Facilities" to "Transportation". This mapping could be accomplished by using Table II.

To sum up, each order in our trip order dataset contains a four element tuple $< p_o, p_d, desc_d, a >$, and the trip purpose label a is derived based on $desc_d$. In the tuple, only p_o and p_d are used in trip purpose mining, while a is used as the ground truth in evaluation. The trip order dataset is from Shenzhou UCar (https://bit.ly/2MG47xz), a major RoD service provider in China. The dataset contains 759,033 orders in Beijing during December, 2015.

B. Dynamic Prices Dataset

A typical and most widely used form of dynamic pricing in RoD service is the multiplicative form: the fare of a trip is the product of a *dynamic* price multiplier (based on the supply and demand) and a *fixed* normal price (based on trip time and distance) [1], [23].

The dynamic prices dataset we use is the event-log dataset from Shenzhou UCar. When one types the origin and destination on the mobile app, the app sends back all the information to the service provider, triggering an *EstimateFee* event. The service provider then replies a record containing the event time, event location, estimated trip fare, price multiplier, user ID, etc. Finally, the app displays the price multiplier and estimated trip fare. The dataset contains 3,646,357 entries in Beijing during December, 2015, and all are properly anonymized.

The dynamic price multiplier recorded in the event-log dataset is different from the multiplier associated with an order. One may use the mobile app to estimate the trip fare for multiple times before finally giving up or creating an order, because s/he is not satisfied with the current price and thus hesitates. Therefore, the number of entries in the event-log dataset is much higher than that in our trip order dataset, and the event-logs give a description of the changes of dynamic prices at a finer granularity. On the down side, it may become inaccurate if we need to associate a price multiplier with an order.

To associate a price multiplier with an order, we pre-process the dataset by calculating the average price multiplier. We first divide the map of Beijing into rectangular cells of 0.02 longitude by 0.02 latitude. The map of Beijing chosen in our paper is a rectangle ranging from 116.1 to 116.8 (east) in longitude and from 39.7 to 40.2 (north) in latitude. We thus have 875 cells in total. We could then calculate the average dynamic price multiplier based on all *EstimateFee* events taking place in each cell during every hour.

Using the average price multiplier in a particular cell during one hour may not be accurate enough, compared to using the price multiplier associated with an order, but it has the following advantages:

- The average price multiplier is easier to obtain for an attacker. The accurate price multiplier would be inaccessible without compromising service provider's database or collaborating with the service provider. The average price multiplier, by comparison, could be obtained or approximated by crowdsourcing or simply reading on the driver's app in RoD service [6]. Further discussions are provided in Section V-D.
- Using the average price multiplier avoids over-fitting. Under some circumstances the accurate price multiplier of an order may be an outlier, due to sudden unplanned events, bad weather, etc., and may lead to model over-fitting. Using the average price multiplier helps eliminate outliers.

C. POI Data

We use datasets about POIs to characterize the origin and destination locations of trip orders. We crawl the POI data from AMap service (one of the most popular on-line map service providers in China) using its API [56]. AMap categories each POI into 14 coarse categories: *car service, restaurant, shopping, sports & entertainment, hospital, hotel, scenic spot, business & residential building, government, education & culture, transportation facility, finance & insurance, business* and *lifestyle*. For a location given in a trip order, we count the number of POIs of each of these 14 categories within a 500-meter radius of the location, and use the resulting vector as our POI data. Hence, for each order in the trip order dataset, we obtain two 14-element vectors, denoted by POI_o and POI_d for the origin and destination respectively.

D. Bus & Metro Distribution Data

We use this dataset to describe the distribution of public transportation services around a location, and choose bus and metro as two examples. Such distribution helps to characterize the availability of public transportation nearby, which may have an impact on trip purpose mining.

The most accurate description of bus and metro distribution should be dynamic and in real-time, e.g., "the number of buses passing by during a particular time period", which could be obtained by, for example, examining the smart-card usage data or the GPS trajectories of bus and metro. However, this is hard for an attacker to obtain. Considering the fact that public transportation services usually have fixed time tables, and that most people decide whether to take public transportation based on the availability of bus & metro stations or lines nearby, instead of the availability of buses or trains nearby, we thus turn to public datasets, which are easily accessible for an attacker.

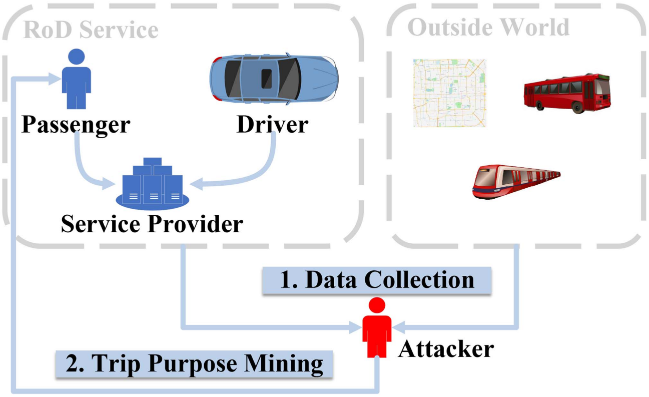

Fig. 1. Threat model: attacker's knowledge and goal.

We acquire our data also from AMap service, similar to our POI data. Specifically, for a location given in an order, we count the number of bus & metro lines and stations within a 500-meter radius of the location. Hence, for each order in the trip order dataset, we acquire two 4-element vectors, denoted by BM_o and BM_d for the origin and destination respectively.

IV. Attack Methodology

We introduce an example of attack methodology of trip purpose mining in RoD service. The models used in this example are simple but effective. The goal is to show that dynamic prices lead to privacy leakage: an attacker is able to obtain more accurate results with the help of dynamic prices.

A. Threat Model

As shown in Fig. 1, the RoD service consists of some *privacy-conscious passengers*, drivers and a service provider that is responsible for the matching between drivers and passengers. In our study, we focus on the privacy leakage on passengers, and hence we do not specify if drivers are privacy-conscious or if the service provider is honest or semi-honest here. The attacker sits outside the RoD service, and performs two tasks, namely, data collection and trip purpose mining. If the results of trip purpose are accurate enough, then it is a potential privacy threat to the privacy-conscious passengers.

Attacker's knowledge: the attacker's knowledge comes from the data collection from two data sources: RoD service, and the "outside world". From RoD service, the attacker tries its best to collect information about trip orders and dynamic prices. "Outside world" refers to the world outside the RoD service, and in our threat model the attacker could obtain public data including POI data and bus & metro distribution data.

Attacker's goal: the attacker's goal is to find out the trip purpose of *any* passenger, and this could be considered as a violation of passenger privacy. Based on the attacker's knowledge obtained from RoD service and outside world, the attacker performs trip purpose mining. This goal is well-motivated since, as mentioned in Section I, the attacker is able to perform many tasks such as intelligent advertising, recommending services to passengers, tracking or stalking certain passengers, or even causing monetary loss to passengers, as long as the trip purpose mining results are accurate.

B. Problem Formulation

The problem of trip purpose mining could be viewed as identifying the most likely trip purpose for every trip order based on all the information from the above datasets. Specifically, it is a classification problem, and if the accuracy of classification improves significantly with the information of dynamic prices, it is safe to claim that dynamic prices lead to privacy leakage and make it easier for an attacker to mine trip purpose. The output of the classification problem is one of the 9 trip purposes shown in Table II. We use A to represent the set of 9 trip purposes:

$$\begin{aligned} A = \{&''\text{Recreation}'', ''\text{Outdoors}'', ''\text{Shopping}'', ''\text{Dining}'', \\ &''\text{Education}'', ''\text{Transportation}'', ''\text{Homing}'', ''\text{Health}'', \\ &''\text{Working}''\}, \text{ or} \\ A = \{&A_1, A_2, A_3, \ldots, A_9\}. \end{aligned} \tag{2}$$

We have the following definition of the trip purpose mining problem:

Definition IV.1 (Trip Purpose Mining Problem). Given the following information about a trip order based on the four datasets:

- Trip's origin and destination $< p_o, p_d >$: the spatio-temporal point with $p_o = (lng_o, lat_o, t_o)$ and $p_d = (lng_d, lat_d, t_d)$, representing the time and location of origin and destination of a trip;
- The average dynamic price multiplier in any cell, any hour;
- POI_o and POI_d: the vector of POI counts around origin and destination of this order;
- BM_o and BM_d: the vector of the number of bus & metro lines and stations.

Extract an input feature vector, denoted by $\vec{X}$, based on the above information.

Predict $\hat{p}(y = \bar{a}|\vec{X}), \forall \bar{a} \in A$: the probability of a candidate trip purpose $\bar{a}$ being the actual trip purpose y of this trip order, with the input feature vector $\vec{X}$. The trip purpose with the largest probability is the output of the classification problem – the mined trip purpose of this order. □

To solve this problem, a model is trained to minimize the difference between the predicted probabilities $\hat{p}(y = \bar{a}|\vec{X}), \forall \bar{a} \in A$ and the ground truth, i.e., the trip purpose label a derived in Section III-A.

C. Overview

Fig. 2 illustrates the framework of our methodology. It consists of two stages, namely *feature extraction and context augmentation*, and *trip purpose mining attack*.

Feature Extraction and Context Augmentation. We already have some basic information from our four datasets, but they are still not enough to describe the context of each trip order. In this stage, we aggregate all the data, and augment the semantic

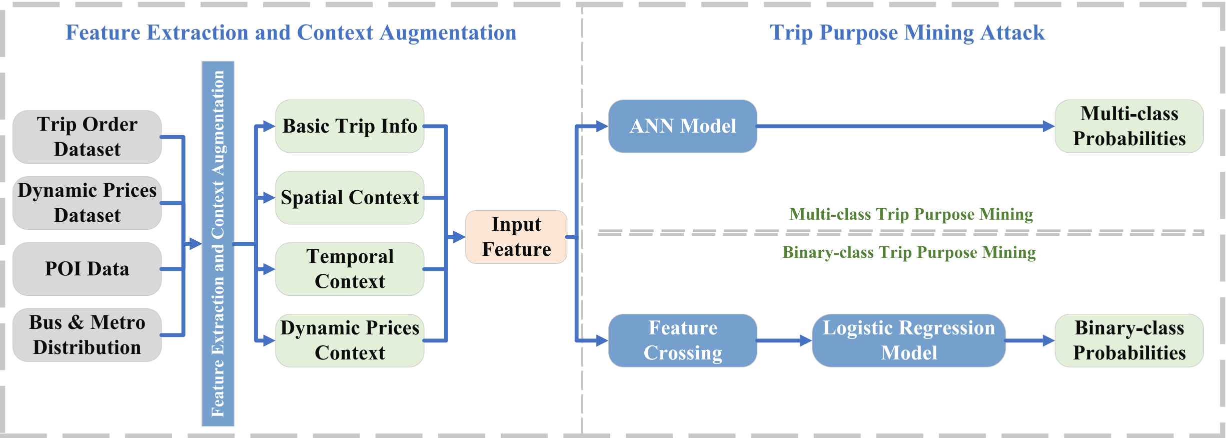
Feature Extraction and Context Augmentation
Trip Purpose Mining Attack
Trip Order Dataset
Dynamic Prices Dataset
POI Data
Bus & Metro Distribution
Feature Extraction and Context Augmentation
Basic Trip Info
Spatial Context
Temporal Context
Dynamic Prices Context
Input Feature
ANN Model
Multi-class Probabilities
Multi-class Trip Purpose Mining
Binary-class Trip Purpose Mining
Feature Crossing
Logistic Regression Model
Binary-class Probabilities

Fig. 2. The framework of our attack methodology.

meaning of each trip order's context, including the basic trip information, spatial, temporal, and dynamic prices context. These contexts are then aggregated to form an input feature vector, which is then fed into the models in trip purpose mining attack.

Trip Purpose Mining Attack. In this stage, we solve the trip purpose mining problem defined in Definition IV.1 in two different ways, namely, as a multi-class classification problem and a binary-class classification problem (by transforming the original multi-class problem using the one-vs.-all paradigm):

- In the multi-class classification problem, the input feature is fed into an artificial neural network (ANN) model, whose output is the probabilities being each of the 9 candidate trip purposes.
- In the binary-class classification problem, the output is the probabilities being one of the 9 candidate trip purposes or not. We choose to use a linear model, namely, logistic regression, in this problem. Before the input feature is fed into the logistic regression model, it is pre-processed using the feature crossing technique to compensate for the lack of non-linearity in the linear model.

There are two reasons of adopting two models in trip purpose mining attack:

- These two models answer **Q1** and **Q2** proposed in the Introduction, respectively. Specifically, the multi-class ANN model answers **Q1**: the goal is to prove that dynamic prices indeed help an attacker to conduct trip purpose mining attack with a higher accuracy. The binary-class linear model answers **Q2**, with the goal to further provide quantitative explanation of feature contribution such as the impacts of different features on the privacy leakage.
- The multi-class ANN model serves as a baseline to evaluate the binary-class linear model. We use the linear model to provide quantitative explanation, but there may be worries that a linear model is not accurate enough due to the lack of non-linearity. As is mentioned, the feature crossing technique is used to improve the linear model's accuracy, and to show its effectiveness, we compare the performance of our linear model with that of the multi-class ANN model.

D. *Feature Extraction and Context Augmentation*

Based on our datasets, we extract features to characterize each trip order, so that these features could be fed into classification models. There are generally two sorts of features: direct features are obtained directly from our datasets, whereas indirect features require some calculations from direct features.

According to the context a feature describes, we divide features into 4 categories: basic trip information, spatial context, temporal context, and dynamic prices context. The criteria of choosing and calculating features is that features should be able to reflect trip purpose from different perspectives.

1) Basic Trip Information: Basic trip information consists of two features: travel time T_{od} and travel distance D_{od}. Calculating T_{od} is straightforward, i.e., $T_{od} = t_d - t_o$, where t_d and t_o are from p_d and p_o in the trip order dataset. Travel distance D_{od} is approximated by using the straight line distance between the origin (lng_o, lat_o) and destination (lng_d, lat_d). The precise travel distance should be based on GPS trajectories of cars, but such dataset may be hard for an attacker to obtain, and we thus use the approximated straight line distance instead.

2) Spatial Context: The spatial context tries to describe location characteristics of the origin and destination, and it captures information from POI and bus & metro distribution data:

- BM_o and BM_d: these are direct features from bus & metro distribution data.
- POI counts vector POI_o and POI_d: these are direct features from POI data. We use $POI_{oi}(1 \leq i \leq 14)$ and $POI_{di}(1 \leq i \leq 14)$ to denote each element of POI_o and POI_d, respectively.
- POI uniqueness vector $Uniq_o$ and $Uniq_d$: POI counts vectors fail to take the uniqueness of different POI categories into consideration. For example, the reason why a certain POI category has a higher count around a location may be that this category is much more common in the whole city. We turn to the TF-IDF statistics to characterize the uniqueness of different POI categories. For the i-th category, we use N_i to denote the total number of POIs of the i-th category in the city; we use N to denote the total

number of POIs in the city. Then, the TF-IDF of POI counts of the i-th category could be written as:

$$Uniq_{oi} = POI_{oi} \cdot \log_2 \left(\frac{N}{1+N_i} \right),$$
$$Uniq_{di} = POI_{di} \cdot \log_2 \left(\frac{N}{1+N_i} \right). \quad (3)$$

And the POI uniqueness vector, denoted as $Uniq_o$ and $Uniq_d$ around the origin and destination, are lists of values $Uniq_{oi}$ and $Uniq_{di} (1 \leq i \leq 14)$, respectively. In this way, different categories of POIs are not weighted equally - the more common a category of POI is, the more its weight is diminished. [23] shows the effectiveness of using the TF-IDF statistics.

- POI distance vector $Dist_o$ and $Dist_d$: we use POI distance vector to measure the minimum distance between each POI category and the origin or destination, considering that passengers may always want to go to the closest POI. For the i-th category, we calculate the minimum distance between any POI of this category and the origin or destination, denoted by, min_d_{oi} or min_d_{di}, and define:

$$Dist_{oi} = -\log_2 \left(\frac{min\_d_{oi}}{500} \right),$$
$$Dist_{di} = -\log_2 \left(\frac{min\_d_{di}}{500} \right). \quad (4)$$

Then, the POI distance vectors $Dist_o$ and $Dist_d$ are lists of values $Dist_{oi}$ and $Dist_{di} (1 \leq i \leq 14)$.

3) Temporal Context: The temporal context tries to augment the semantic meaning of the time the order starts and ends. We extract the following features:

- Day-of-week of the start time DW_o: this feature describes the day of week (e.g., Monday, Tuesday,..., Saturday and Sunday). We choose to extract this feature because trip purposes differ between different days of week, e.g., Monday vs. Friday, weekdays vs. weekends, etc. We use one-hot encoding to represent DW_o as a 7-element vector, with only one element being 1 and all other elements being 0. For example, $DW_o = [1, 0, 0, 0, 0, 0, 0]$ means "Monday".
- Timeslot of the start time TS_o and end time TS_d: instead of using the hour-of-day feature to describe the start time and end time, we introduce the concept of timeslot and describe the start time and end time at a coarser granularity. We divide one day into 4 timeslots of equal length: timeslot-0 to -3 refers to [4am, 10am), [10am, 4pm), [4pm, 10pm) and [10pm, 4am), respectively. Roughly speaking, for weekdays, these timeslots correspond to morning rush hours, non-rush hours around noon, evening rush hours, and night hours; for weekends, the partition of timeslots still makes sense, as [4] suggests that human activity remains relatively high and stable during the day, and becomes lowered during the rest of the day. [4] also claims that using 4-timeslot division is already enough to capture the information in the start and end time. Another advantage of using timeslot is to reduce the dimensions of the input feature to avoid overfitting. Similar to DW_o, we also use one-hot encoding to represent TS_o and TS_d as 4-element vectors. For example, $TS_o = [0, 1, 0, 0]$ means the start time of the order falls in timeslot-1.

4) Dynamic Prices Context: As claimed by RoD service providers, they design dynamic pricing algorithms based on the real-time supply and demand condition on the road. If pricing algorithms are robust and working, then we could view dynamic prices as a perfect reflection of what is going on instantaneously. Such reflection may be implicit, as pricing algorithms are, in most cases, kept as secrets by service providers. We involve dynamic prices context, based on dynamic prices dataset, into our features, as an effort to study such implicit reflection.

We extract features based on the average price multiplier in every cell, every hour. Here we use "origin cell" and "destination cell" to represent the cell the origin and destination fall in. Also, we use h_o to denote the hour-of-day of t_o, and h_d to denote the hour-of-day of t_d. We involve the following features:

- The average dynamic price multiplier in the origin cell during the hour $h_o - 1$, h_o and $h_o + 1$, denoted as $DP_{o,-1}$, $DP_{o,0}$ and $DP_{o,+1}$, respectively. We also use a vector representation $DP_o = (DP_{o,-1}, DP_{o,0}, DP_{o,+1})$ for convenience.
- The average dynamic price multiplier in the destination cell during the hour $h_d - 1$, h_d and $h_d + 1$, denoted as $DP_{d,-1}$, $DP_{d,0}$ and $DP_{d,+1}$, respectively. We also use a vector representation $DP_d = (DP_{d,-1}, DP_{d,0}, DP_{d,+1})$ for convenience.

5) The Input Feature Vector: Before creating the input feature vector, we perform normalization on the above features. Normalization is necessary in models with multiple features, to make the values of each feature covering roughly the same range; otherwise, the models would become difficult to train or converge. Features in the form of one-hot encoded vector do not need normalization, as they have values of either 0 or 1. For all other features, we calculate the Z-score (i.e., the number of standard deviations from the mean, https://en.wikipedia.org/wiki/Standard_score) of any value of each feature to normalize.

For each order, we then gather all the features, as summarized in Table III, to form the input feature vector $\vec{X}$, which are then fed into the models in Section IV-E:

$$\vec{X} = [T_{od}, D_{od}, BM_o, BM_d, POI_o, POI_d, Uniq_o, Uniq_d, Dist_o, Dist_d, DW_o, TS_o, TS_d, DP_o, DP_d] \quad (5)$$

E. Trip Purpose Mining Attack

With the input feature vector ready, the next task for an attacker is to infer a trip purpose for this order. As mentioned previously, we view the trip purpose mining attack problem as a classification problem. We solve this problem as a multi-class and as a binary-class classification problem.

1) Multi-Class Trip Purpose Mining: The multi-class trip purpose mining problem is straightforward: identifying the best suited candidate trip purpose out of the 9 candidate trip purposes in (2). For each order, the ground truth y is the trip purpose label a derived in Section III-A, and could be written in the form of probabilities. If $y = a = A_j$, then the probabilities

TABLE III
A SUMMARY OF FEATURES USED IN TRIP PURPOSE MINING ATTACK

Context	Feature	Dim.
Basic Trip Info	T_{od}: travel time	1
	D_{od}: travel distance	1
Spatial Context	BM_o: the number of bus & metro lines and stations around order origin	4
	BM_d: the number of bus & metro lines and stations around order destination	4
	POI_o: POI counts vector around order origin	14
	POI_d: POI counts vector around order destination	14
	$Uniq_o$: POI uniqueness vector around order origin	14
	$Uniq_d$: POI uniqueness vector around order destination	14
	$Dist_o$: POI distance vector around order origin	14
	$Dist_d$: POI distance vector around order destination	14
Temporal Context	DW_o: one-hot encoded vector of the day-of-week of order start time	7
	TS_o: one-hot encoded vector of the timeslot of order start time	4
	TS_d: one-hot encoded vector of the timeslot of order end time	4
Dynamic Prices Context	$DP_{o,-1}$: average dynamic price multiplier in the origin cell during hour $h_o - 1$	1
	$DP_{o,0}$: average dynamic price multiplier in the origin cell during hour h_o	1
	$DP_{o,+1}$: average dynamic price multiplier in the origin cell during hour $h_o + 1$	1
	$DP_{d,-1}$: average dynamic price multiplier in the destination cell during hour $h_o - 1$	1
	$DP_{d,0}$: average dynamic price multiplier in the destination cell during hour h_o	1
	$DP_{d,+1}$: average dynamic price multiplier in the destination cell during hour $h_o + 1$	1

$p_i (1 \leq i \leq 9)$ that y equals A_i are:

$$p_i = p(y = A_i) = \begin{cases} 0, & \text{if } 1 \leq i \leq 9, i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{6}$$

For example, if the trip purpose label equals A_2, then $p_2 = 1$ and $p_i = 0$ for all other i in the range [1,9].

We adopt a simple artificial neural network (ANN) model to solve the classification problem. The feature vector $\vec{X}$ is fed into two hidden layers, producing output hidden states $\vec{H_1}$ and $\vec{H_2}$, respectively:

$$\vec{H_1} = ReLU(\vec{W_1} \cdot \vec{X} + \vec{b_1}), \tag{7}$$

$$\vec{H_2} = ReLU(\vec{W_2} \cdot \vec{H_1} + \vec{b_2}). \tag{8}$$

In this model, the activation functions of both layers are the ReLU function. $(\vec{W_1}, \vec{b_1})$ and $(\vec{W_2}, \vec{b_2})$ are the parameters to learn in both layers. We also add a drop-out regularizer after the first hidden layer to avoid over-fitting. Finally, the output of the second hidden layer, $\vec{H_2}$, is fed into a softmax layer with 9 units to produce the output probabilities of 9 candidate trip purposes. We use $\hat{p_i} = \hat{p}(y = A_i)$ to denote the output probabilities of the softmax layer, then the inferred trip purpose, denoted by $\hat{y}$, is the candidate trip purpose with the highest $\hat{p_i}$:

$$\hat{y} = arg \max_i \hat{p}(y = A_i). \tag{9}$$

The loss function of the ANN model is based on the categorical cross-entropy, which calculates the distance between the predicted probabilities $\hat{p_i}$ and ground truth probabilities p_i:

$$L_{ANN}(\omega) = -\Sigma_{i=1}^{|A|} p_i \log(\hat{p_i}) + \lambda_2 ||\omega||_2^2 \tag{10}$$

In (10), the first term is the categorical cross-entropy between p_i and $\hat{p_i}$, and the second term is a L2 regularizer. In the first term, $|A|$ is the number of candidate trip purposes and equals 9 in our study; in the second term, ω is the set of all learnable parameters in the model and λ_2 is the regularization rate.

2) Binary-Class Trip Purpose Mining: In binary-class trip purpose mining, we transform the multi-class classification problem into 9 binary-class problems using the one-vs.all paradigm. Instead of inferring the best suited candidate trip purpose, each binary-class problem answers the question "Does a trip order belong to one particular trip purpose or not?".

We use a linear model, i.e., logistic regression, to perform binary-class classification. The rationale behind solving binary-class problems and using a linear model is:

- Binary-class problems make it possible to learn the different extents of privacy leakage from dynamic prices for trips with different purposes. In other words, the accuracy improvement of trip purpose mining may differ across trips with different purposes. Intuitively, trips with "working" purpose may be insensitive to dynamic prices because people are in a hurry, whereas "homing" trips are more sensitive to dynamic prices, as people go home at different times and there is not a strict time limit. Hence, dynamic prices features should contribute more in trip purpose mining for trips with "homing" purpose, than trips with "working" purpose.
- Using a linear model to tackle the binary-class problems helps to quantify feature contributions. With the ANN model used in Section IV-E-1, it is possible to check whether the introduction of dynamic prices features improves accuracy, but it is hard to determine which features are more important and by how much. We use a linear model in binary-class problems, because of its interpretability.

Feature Crossing. A major drawback of linear models is the lack of non-linearity, and a linear model thus fails to characterize the correlation between features. To deal with this, a common and widely-used technique is feature crossing, i.e., multiplying two or more features. We use a very simple example to show feature crossing. Assuming that we have two basic features x_a and x_b, then a simple linear model could be expressed as $y = \omega_a x_a + \omega_b x_b + b$, where y is the predicted result, and ω_a, ω_b, b are the learnable parameters. Feature crossing tries to add a new feature $x_c = x_a x_b$, and the linear model then becomes $y = \omega'_a x_a + \omega'_b x_b + \omega'_c x_a x_b + b'$. This shows how feature crossing adds non-linear terms into a linear model.

We can cross virtually any two features, denoted by x_a and x_b without lost of generality, and in doing that, we may come across three situations:

- Both features are scalars: the result of feature crossing is also a scalar $x_c = x_a x_b$.
- x_a is a scalar, and $\vec{x_b}$ is a vector: the result is the product of scalar-vector multiplication, i.e., $\vec{x_c} = x_a \vec{x_b}$.
- Both features are vectors: if $\vec{x_a} = (x_{a1}, x_{a2}, \ldots, x_{ad_a})$ and $\vec{x_b} = (x_{b1}, x_{b2}, \ldots, x_{bd_b})$ have a dimension of d_a and d_b, respectively, then the resulting vector $\vec{x_c}$ has a dimension of $d_c = d_a d_b$, and could be written as $x_c = (x_{a1}x_{b1}, x_{a1}x_{b2}, \ldots, x_{a1}x_{bd_b}, \ldots, x_{ad_a}x_{b1}, x_{ad_a}x_{b2}, \ldots, x_{ad_a}x_{bd_b})$.

In our practice, we perform feature crossing between any two features in the input feature vector. In other words, we do not manually choose only a subset of features to cross, based on the following consideration. Firstly, crossing two features does not make the resulting feature's dimension prohibitively high, and we do not need to consider avoiding over-fitting even we cross every pair of features. Secondly, we could determine whether it is necessary to cross a particular pair of features by inspecting the corresponding weight in the trained linear model, and it is unnecessary to make decision in advance.

We do not perform feature crossing between more than two features for two reasons. Firstly, it is not intuitive enough to understand the correlation between more than two features. Secondly, feature crossing between more than two features dramatically increases the dimension of the resulting feature vector, and may lead to over-fitting.

The Logistic Regression Model. We denote the feature vector after feature crossing as $\vec{X_F}$, which is then fed into the logistic regression model to solve the binary-class trip purpose mining problem. In the logistic regression model, the output of the linear model passes a sigmoid activation function, and a probability $\hat{p}$ is generated. For the ground truth probability p, $p = 1$ if the ground truth is the particular trip purpose, and $p = 0$ otherwise. With the output probability $\hat{p}$, we also have a hyper-parameter, i.e., classification threshold p_{th}, and define the output of the classification problem $\hat{y}$ as:

$$\hat{y} = \begin{cases} 1, & \text{if } \hat{p} \geq p_{th}, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

The loss function is the binary cross-entropy, as a special case of categorical cross-entropy loss function:

$$L_{LR}(\omega) = -(p \cdot \log\hat{p} + (1-p) \cdot \log(1-\hat{p})) + \lambda_2 ||\omega||_2^2 \tag{12}$$

In (12) the first term is the binary cross-entropy between p and $\hat{p}$, and the second term is a L2 regularizer. In the second term, ω is the set of all learnable parameters in the logistic regression model. Note that the set ω and the rate λ_2 in (12) may not be the same as in (10).

V. Evaluation

In this section we evaluate the multi-class and the binary-class trip purpose mining model. In each model, we make different performance comparisons:

- For the multi-class model, we compare it with a state-of-the-art that shares a similar setting – i.e., based on large-scale datasets without individual preference information – to show the effectiveness of our model.
- For the binary-class model, we use our multi-class model as a baseline. As already mentioned in Section IV-C, we use feature-crossing to compensate for the lack of non-linearity in the linear model. So if the binary-class model achieves a similar accuracy to that of the multi-class model, it is then justified that the binary-class model gives convincing results and could be used to evaluate the different extents of privacy leakage under different trip purposes.

A. Experiment Setup

Evaluation metrics. To evaluate the performance of our ANN and logistic regression models, we adopt some common metrics, including *accuracy*, *precision*, *recall* and *AUC*. Specifically, accuracy is the ratio of the number of correctly classified samples to the total number of samples, and could be used in either multi-class or binary-class classification problem. The other three metrics are mainly used in binary-class problems. Precision measures the proportion of positive identifications that are actually correct; recall measures the proportion of actual positives that are identified correctly; and both precision and recall are dependent on the classification threshold hyper-parameter in binary classification. AUC, i.e., "Area Under the ROC Curve", is independent of classification threshold, and is usually used as an aggregate measure to reflect the performance of binary classifiers across all possible classification thresholds.

To use precision and recall in our multi-class problem, we also employ the *macro-average precision* and *macro-average recall*. In other words, we reduce the multi-class problem to 9 binary-class problems, and calculate the precision and recall for each binary-class problem, and then average the results.

Parameters settings. In the multi-class problem, we tune the ANN model, and our final model has 96 units in the first hidden layer, and 448 units in the second hidden layer, with the drop-out rate set to 0.1 in the drop-out regularization after the first hidden layer. The L2 regularization rate is 0.0001. In training the model, learning rate and batch size are set to 0.001 and 8,000, respectively, and the number of epochs is set to 300, as we observe that the model could converge in less than 300 epochs in most cases.

In the binary-class problem, the L2 regularization rate is also set to 0.0001. In training the logistic regression model, learning rate and batch size are set to 0.001 and 10,000, respectively. The number of epochs is set to 50, as the linear model converges much faster than the ANN model.

In training and evaluating both models, we randomly divide our trip order datasets into training, validation and test sets at a ratio of $7 : 1.5 : 1.5$. As we have 759,033 orders in our dataset,

TABLE IV
THE CONFUSION MATRIX OF OUR ANN MODEL IN MULTI-CLASS TRIP PURPOSE MINING

Classified result Ground truth	#1	#2	#3	#4	#5	#6	#7	#8	#9	Recall (%)
#1	**1777**	45	251	109	62	373	574	41	274	50.68
#2	86	**1115**	129	110	55	579	703	53	256	36.13
#3	73	46	**4624**	147	86	1102	1540	63	788	54.60
#4	101	78	373	**2255**	89	1108	1878	78	557	34.60
#5	40	34	125	55	**2673**	591	1401	129	478	48.37
#6	186	146	877	405	397	**28419**	4471	420	1956	76.24
#7	108	72	752	331	299	3954	**22712**	228	1162	76.68
#8	72	15	78	52	84	468	639	**3814**	271	69.43
#9	87	47	676	177	234	1962	1941	180	**9059**	63.07
Precision (%)	70.24	69.77	58.64	61.93	67.18	73.71	63.34	76.19	61.21	

this means that the training, validation and test sets have 531,323, 113,855, and 113,855 data samples.

B. Multi-Class Trip Purpose Mining

To inspect model performance for each candidate trip purposes, Table IV presents the confusion matrix of our ANN model. In Table IV, we use the trip purpose # (shown in Table II) to represent different trip purposes due to limited space. For example, trip purpose #1 corresponds to "Recreation". In this confusion matrix, each row shows how trips of a particular trip purpose in ground truths are identified, and each column shows how trips of different trip purposes in ground truths are identified as a particular trip purpose. Also, the number along the diagonal (in bold) means the number of samples of a particular trip purpose that are correctly identified. For example, the number at the intersection of #1 row and #1 column shows that 1,777 trips with the purpose "Recreation" in ground truths are correctly identified as "Recreation" trips; the number at the intersection of #3 row and #7 column shows that there are 1,540 trips with the purpose "Shopping" in ground truths are identified as "Homing" trips. It could be seen from Table IV that the accuracy of the ANN model is 67.15%.

Based on the confusion matrix, for each trip purpose, we also calculate the precision and recall, by reducing the multi-class problem to binary-class problems. For recall, the recall of a particular trip purpose is the number at the intersection of the diagonal and the corresponding row, divided by the sum of the row. For precision, it could be calculated by substituting column for row. For example, the recall of trip purpose #1 ("Recreation") is 1,777 divided by the sum of row #1, and the precision of trip purpose #1 is 1,777 divided by the sum of column #1. Furthermore, by taking averages among precisions (or recalls) of all trip purposes, we obtain the macro average precision (or macro average recall). The macro average precision and macro average recall of the ANN model is 66.91% and 56.65%.

To show the effects of dynamic prices, i.e., the improvement of model performance by considering dynamic prices, we exclude all the features belonging to dynamic prices context as shown in Table III and re-train the ANN model. Results show that, without features from dynamic prices context, the model achieves an accuracy of 57.34%.

We have the following observations based on the above results:

- *Our model achieves a satisfactory accuracy compared with state-of-the-art*. [49] studies trip purpose inference under similar settings – i.e., based on large-scale datasets without individual preference information, as is discussed in Sections I and II – and it adopts a dual-attention graph embedding network without dynamic prices. The accuracy of [49] is 64.57%, and it is thus clear that our model has an accuracy slightly higher than the state-of-the-art. The consideration of dynamic prices in our study is more effective than using a more complex model.
- *Our model in multi-class trip purpose mining is accurate enough for a privacy attacker*. The ANN model achieves an accuracy of 67.15% in a 9-class classification problem. This accuracy means that our model is generally applicable in real-life scenarios. We offer a more detailed discussion on model accuracy in Section V-D.
- *Our model has high precisions for almost all trip purposes*. We value precision more than recall, as a higher precision means that "if the model identifies a trip order to be of a particular trip purpose, the probability of being correct is high" – i.e., the attacker is more confident about the inferred trip purpose. The precisions range from 58.64% to 76.19%, with a macro average of 66.91%. In other words, the attacker could be, on average, 66.91% sure about the inferred trip purpose.
- *Features from dynamic prices indeed help an attacker to find out trip purpose*. Compared to the case without using dynamic prices features, using such features increases the model accuracy by 17.1%.
- *The ANN model is trained for all 9 candidate trip purposes, and there are large performance gaps between some trip purposes*. One possible reason is the class imbalance. For example, the numbers of trips with purpose #6 ("Transportation") and #7 ("Homing") are overwhelmingly high, and the model may thus tend to identify trips with other purposes to have a trip purpose #6 or #7, resulting in high recalls for these two trip purposes. This also motivates

TABLE V
THE OVERALL RESULTS FOR BINARY-CLASS TRIP PURPOSE MINING

Trip purpose	Accuracy (%)	Precision (%)	Classification Threshold	AUC
#6 Transportation	79.78	74.01	0.50	0.7533
#7 Homing	78.75	62.41	0.49	0.7736
#9 Working	90.21	60.29	0.43	0.7989

us to perform binary-class trip purpose mining to lower the impacts of class imbalance. Another possible reason is different mining difficulties, i.e., the difficulty of mining a trip purpose varies across different trip purposes.

C. Binary-Class Trip Purpose Mining

We train logistic regression models to solve the binary-class trip purpose mining problem, created by reducing the multi-class trip purpose mining problem using the one-vs.-all paradigm. As mentioned previously, the goal is to find out and quantify the different extents of privacy leakage of dynamic prices regarding different trip purposes. Another goal comes from our observations in Section V-B: the model for multi-class trip purpose mining has large performance gaps between trip purposes due to class imbalance.

Due to limited space, in this section we show the results of binary-class problems regarding some of the 9 candidate trip purposes. We choose trip purpose #6 ("Transportation"), #7 ("Homing") and #9 ("Working") in this section. These trip purposes account for larger proportions among all trips, and correspond to trips going to transportation service facilities such as airport or train stations or commuting trips. We show the overall results on these trip purposes in Table V, including AUC, the classification threshold when we have the highest accuracy, the corresponding accuracy and precision. The comparison between Tables V and IV confirms the effectiveness of the logistic regression models with feature crossing:

- The logistic regression models in binary-class problems achieve precisions close to the precisions with the ANN model. The precisions of trip purpose #7 and #9 (i.e., 62.41% and 60.29%) are close to that with ANN model (i.e., 63.34% and 61.21%), and for trip purpose #6, the precision with the logistic regression model is even slightly higher (74.01% vs. 73.71%).
- The logistic regression models achieve much higher accuracies, i.e., 79.78%, 78.75% and 90.21% for the three chosen trip purposes. On one hand, it shows that the logistic regression models are accurate enough and applicable in practice. On the other hand, though this may be the result of class imbalance, the precisions already show the models' performance.

Besides having comparable performance with the ANN model, the logistic regression model also helps us quantify feature contributions. Knowledge of feature contribution could be used to answer questions such as "what features are important and by how much?". In a linear model, we could intuitively compare feature contributions by examining the corresponding weight of each feature. We examine the absolute value of the weight corresponding to each dimension of the input feature vector, and find out the feature or the crossed feature, as well as the context, it belongs to, and show them in Table VI. In Table VI, we only show the top-15 dimensions (i.e., approximately top 1% dimensions) for the three chosen trip purposes due to limited space. In the "*Feature*" column, a "×" symbol is used when the dimension in question belongs to a crossed feature; in the "*Context*" column, "B", "S", "T", or "D" represents "**b**asic trip information", "**s**patial context", "**t**emporal context" and "**d**ynamic prices context", respectively.

Regarding feature contribution for the three chosen trip purposes, we have the following observations:

- *Dynamic prices indeed lead to privacy leakage, but to different extents across different trip purposes*. Intuitively, among the top-15 dimensions, the number of dimensions that are related to dynamic prices context is 9 for "Transportation", 12 for "Homing" and only 1 for "Working". Besides, the weights of these features and dimensions fall in a broad range.
- *Feature crossing is indispensable in improving the performance of linear models*. The top-15 dimensions for all the chosen trip purposes are mostly (35 out of 45) from crossed features. Common crossings are between S and T, forming spatio-temporal features, or S (or T) and D, showing correlations between dynamic prices and spatial or temporal features.
- *The models achieve the biggest power only when information from dynamic prices, spatial context, and temporal context are all carefully extracted*. A lack of any one of the three information sources leads to performance degradation in trip purpose mining. This inspires us that designing proper privacy-preserving mechanisms for any one of the three sources could help to prevent privacy leakage from other information. For example, even with dynamic prices data, an attacker could not peek for accurate trip purposes without a trip's spatio-temporal information.
- *Trip purpose "Working" is special, as temporal context plays an overwhelming role*. 14 out of the top-15 dimensions are solely from temporal context, with the only other one being a crossed feature from dynamic prices and spatial context. Specifically, the day-of-week and timeslot of order start time are so discriminative that it is already enough to find out "Working" trips with them. In this case, the level of privacy leakage from dynamic prices is low.
- *For trip purpose "Transportation"*, the influence from temporal context features is reduced, but the timeslot of order start time is still important as it appears in 8 out of 15 top dimensions, showing that "Transportation" trips happen during particular timeslots. Secondly, travel time and distance are of high significance. This is unique and agrees with our everyday experience. Lastly, the level of privacy leakage from dynamic prices is much higher.
- *For trip purpose "Homing", the level of privacy leakage from dynamic prices may be the highest among the chosen trip purposes*. As we consider, "Homing" trips have distributions of destination and order time that are more random,

TABLE VI
THE TOP-15 FEATURES/DIMENSIONS RANKED BY WEIGHTS

	#6 Transportation			#7 Homing			#9 Working		
Rank	Weight	Feature	Context	Weight	Feature	Context	Weight	Feature	Context
1	0.499	$DP_d \times TS_o$	T, D	0.522	TS_d	T	0.588	$DW_o \times TS_o$	T, T
2	0.477	D_{od}	B	0.480	$DP_{d,+1}$	D	0.569	$DW_o \times TS_o$	T, T
3	0.471	$Uniq_d \times TS_o$	S, T	0.369	TS_d	T	0.503	TS_d	T
4	0.393	$DP_d \times Uniq_d$	S, D	0.363	$DP_d \times Uniq_d$	S, D	0.446	$DW_o \times TS_o$	T, T
5	0.386	$DP_d \times TS_o$	T, D	0.327	$DP_d \times TS_o$	T, D	0.406	$DW_o \times TS_o$	T, T
6	0.370	$Uniq_d \times TS_o$	S, T	0.322	$DP_d \times TS_o$	T, D	0.381	$DW_o \times TS_o$	T, T
7	0.364	$D_{od} \times TS_o$	B, T	0.318	$DP_{d,0}$	D	0.362	TS_d	T
8	0.353	$DP_{d,-1}$	D	0.308	$DP_d \times TS_o$	T, D	0.355	$DW_o \times TS_o$	T, T
9	0.349	$DP_d \times TS_o$	T, D	0.295	$Uniq_d \times TS_o$	S, T	0.343	DW_o	T
10	0.345	$DP_o \times Uniq_d$	S, D	0.272	$DP_d \times DW_o$	T, D	0.337	$DP_d \times Uniq_d$	S, D
11	0.332	$DP_d \times Uniq_d$	S, D	0.264	$DP_d \times DW_o$	T, D	0.325	$DW_o \times TS_o$	T, T
12	0.331	$T_{od} \times TS_o$	B, T	0.254	$DP_d \times TS_o$	T, D	0.310	$DW_o \times TS_o$	T, T
13	0.302	$DP_o \times Dist_d$	S, D	0.252	$DP_d \times DW_o$	T, D	0.306	$DW_o \times TS_o$	T, T
14	0.301	$DP_o \times Dist_d$	S, D	0.249	$DP_o \times Uniq_d$	S, D	0.299	$DW_o \times TS_o$	T, T
15	0.299	$Uniq_d \times TS_o$	S, T	0.244	$DP_d \times DW_o$	T, D	0.298	DW_o	T

and thus a closer coupling of features from dynamic prices and spatio-temporal context is necessary.

D. Discussions

We provide further discussions on the effects of dynamic prices, the performance of our models, data accessibility, the overfitting issue with feature crossing, the consideration of individual information, and the applicability of our study.

The effects of dynamic prices. Our evaluation results already confirm the privacy leakage from dynamic prices, as an attacker could use features from dynamic prices to improve the accuracy of models in trip purpose mining. We give some explanations on the effects of dynamic prices below.

As we have mentioned, the reason why dynamic prices help to find out trip purpose is that, passengers going for trips with a certain purpose always have a stable preference on dynamic prices. For example, people going for work ("Working") do not care about dynamic prices that much because they are in a hurry (this is also verified in [3]). By comparison, people going back home ("Homing") may accept a broader range of dynamic prices, as their destinations or order start time differ and they are probably not eager to go. In fact, this observation also holds for other services involving dynamic prices – users with a certain characteristic always have a stable preference for dynamic prices. It is thus possible to identify, for example, whether one is in a family trip or business trip based on the flight ticket price s/he buys.

Another reason is that, dynamic prices are actually an implicit description of trip semantics such as spatial or temporal contexts. Without dynamic prices, it is common to extract a lot of information from different sources to describe trip semantics by, for example, defining static or dynamic POI features to comprehensively model POI information, relying on other datasets such as check-in data from LBSN, trip surveys, household information, personal questionnaire, and etc. For a privacy attacker, all these information are hard to obtain, and even when they are available, sometimes the correlation between them is not accurate and realistic due to reasons such as time misalignment. Dynamic prices, on the other hand, are an integrated representation. Service providers always claim that their dynamic pricing algorithms are carefully designed to incorporate many factors such as the supply and demand condition; and these factors are, in turn, the result of the spatial and temporal contexts which otherwise we need to go to great lengths to describe. Therefore, if dynamic prices are determined in a reliable and sophisticated way as claimed by service providers, they could save attacker's efforts in describing trip semantics. It is true that we don't know how dynamic prices are related to trip semantics because pricing algorithms are mostly kept as secrets, but this could be approached by further studying dynamic pricing mechanisms, predicting dynamic prices based on various features, etc.

The performance of our models. Our ANN model in multi-class trip purpose mining achieves an accuracy and macro average precision of 67.15% and 66.91%, respectively. The logistic regression models used in binary-class trip purpose mining have similar performance. We claim that the performance of our models is enough for the problem, and we give a detailed discussion from the following four perspectives.

Firstly, the accuracy itself is not the main goal of our study; instead, the accuracy improvement brought by dynamic prices is, as this is a clear signal of the existence of privacy leakage from dynamic prices. The goal of our study is to confirm the existence of privacy leakage from dynamic prices, i.e., dynamic prices help to improve trip purpose mining results from the attacker's perspective. In other words, as long as the accuracy of a model is high enough so that the results are convincing, we value the improvement of accuracy with dynamic prices more than the absolute value of accuracy. Evaluation results show that, for the improvement of accuracy, the ANN model trained with dynamic price features has an accuracy 17.1% higher than the model without such features. Hence, the improvement of accuracy brought by dynamic prices show the existence of privacy leakage. For the absolute value of accuracy, we compare our model accuracy with state-of-the-arts later.

Secondly, we explain the reason why it is difficult for our work, as well as similar studies, to obtain a very high mining accuracy. Unlike other trip information such as trip destination, trip purpose is more implicit and harder to guess. As discussed in Section II, a large proportion of previous studies are based on travel surveys or questionnaires that indicate the characteristics of participants, and the number of participants is usually relatively small. Therefore, it is easier to obtain a high accuracy in these studies.

On the other hand, our work belongs to the category of studies that are based on large-scale datasets without personal preference information. Passengers and orders are spread across the whole city, and the number of orders reach hundreds of thousands during a whole month. The datasets used in our study also do not contain any description of the characteristics of particular passengers or group of passengers. As a result, it is much harder to obtain a higher accuracy in our work. We compare our results with that of [49], in which a dual attention graph embedding network is adopted to predict trip purpose based on large-scale datasets with an overall accuracy of 64.57%. Note that [49] uses a more complex model but does not consider the impacts of dynamic prices. The comparison shows that: (a) the absolute value of accuracy of our model is comparative to that of state-of-the-art; and (b) considering dynamic prices is effective in trip purpose mining and may be more important than adopting a more complex model.

Thirdly, the performance is enough for possible application scenarios. For example, the attacker could be a third-party advertiser, using trip purposes to place suitable advertisements; or the attacker could be from a competing service provider, and wants to understand the spatio-temporal distribution of trip purposes so that s/he could take advantage of this and design optimal competition policies. Exceptions are those application scenarios in which the attacker needs to associate the trip purpose with one or more particular passenger(s), but such scenarios require the adoption of other techniques such as trajectory-user linking, and are thus out of the scope of our paper. We leave this as a future work.

Lastly, we also list some possible ways of performance improvement that could be considered in the future work:

- Using more datasets or more sophisticated algorithms: for datasets, one could choose those datasets describing passengers' trip patterns, demographic characteristics, order histories, etc. For example, the check-in data from LBSN services is able to describe trip pattern or POI popularity, and is thus used in many human mobility studies or smart urban services. But when choosing data, it is still necessary to inspect the feasibility of obtaining such data from the attacker's perspective. For algorithms, possible choices include random forest model, deep neural network, attention network, etc. Similarly, when choosing algorithms, one need to carefully inspect whether a particular algorithm helps to describe some features that are important to trip purpose results.
- Using pruning: some trip orders may not be helpful or necessary in model training. For example, some orders have origins or destinations that have no POIs around – i.e., they could be viewed as outliers; some orders have average dynamic price multipliers that are always 1 in the origin or destination cell – i.e., dynamic prices may have no effects. These orders should be carefully pre-processed before model training.
- Training different models for different spatio-temporal combinations: for example, different ANN models could be trained in central business district, around big residential communities, during morning rush hours, etc. There are many possible ways of dividing spatial regions, e.g., based on city plan, dynamic prices, the number of orders, the number of drivers passing by, statistics of similar services such as taxi, etc. There are also many possible ways of dividing temporal periods, e.g., using rush hours, using weekdays and weekends, relying on timetables of trains or buses nearby, etc.

Data accessibility. Our study uses four different datasets, including trip order data, dynamic prices data, POI data and bus & metro distribution data. The accessibility of these datasets is a prerequisite of not only the feasibility of the trip purpose mining attack, but also the applicability of our study, i.e., the possibility of applying our methodologies to other similar services or problems. Among these datasets, the trip order data (excluding the text description of destination), POI data, and bus & metro distribution data are either open, public datasets, or crawled from public services, and are thus easily accessible. In the following, we concentrate on the accessibility of the dynamic prices data, as well as the text description of destination.

Dynamic prices data. We clarify in Section III-B that we are using the hourly average dynamic price multiplier at the level of city cells, instead of the exact dynamic price multiplier associated with each order. The two reasons include avoiding outliers and making it easier for the attacker to obtain data.

There are indeed some possible ways to obtain the hourly average dynamic price multiplier. Firstly, price multiplier is not a privacy for drivers, so an attacker, or more generally, a third-party, could design proper incentive mechanisms to encourage drivers to share dynamic price multipliers of the orders they take, in a crowdsourcing fashion. Another feasible way is to deploy multiple mobile phones across carefully calculated and representative locations and register as drivers. The average dynamic price multiplier is usually displayed on drivers' mobile app to help them make seeking decisions, and in this way the attacker could collect average dynamic price multipliers across the city. Similar procedures have been carried out in [2] to study the distribution of dynamic price multipliers.

Text description of destination. It refers to the text input by passengers on the mobile app when specifying origins or destinations. In traditional taxi service, the origins and destinations are recorded by on-car GPS devices and there are only longitude and latitude records, with no semantic information. On the contrary, thanks to the development of intelligent transportation, nowadays there is a rapidly growing proportion of transportation services – not only RoD service (e.g., taxis) – adopting mobile apps for passengers. With mobile apps, semantic information such as the text description of destination are now recorded by service provider.

With such data already recorded, there are multiple possible ways to obtain such data:

- In some cases, the service provider may actively want to publish the data or share with some parties. An example is research collaboration.
- It is also possible for an attacker to obtain such data by, for example, compromising the servers of the service provider.
- Another possible approach is crowdsourcing. For example, one could design a mobile app and incentivize passengers to upload their trip information (including, certainly, the text description of destination). In this way, one may not be able to obtain the text description of destination of all orders, but could still obtain the data of a certain proportion of orders, depending on the incentive mechanism design, which is enough for model training.

Worries may still exist that none of the above ways work. In fact, the text description of destination is used to generate the trip purpose ground-truth (e.g., labelling the data), so if there are other possible ways to generate the ground-truth, obtaining the text description may not be a must. For example, there are already a large number of studies on trip purpose mining, imputation or inference (see Section II), and their algorithms and results could be used as ground-truth and thus help an attacker in model training. The fact that most previous studies are based on surveys or questionnaires instead of large-scale datasets does not render their results useless as ground-truth.

Another problem may arise that if an attacker could generate the trip purpose ground-truth from the text description of destination, why it is necessary to design all the learning models for trip purpose mining. We provide our thoughts from the following two perspectives.

Firstly, the data an attacker could obtain may be incomplete. Through the possible ways listed above, though it is possible to obtain data and generate trip purpose ground-truth, it is still somewhat difficult to collect a complete set of data. For example, in research collaboration a service provider usually publishes some selected data covering, say, a small part of the city, a short period of time, etc; in a compromising attack, it is even harder to get complete data; in crowdsourcing, it is also infeasible to incentivize everyone to upload their information. Therefore, the attacker could use the obtained incomplete data to train models, and use the learned models to infer trip purpose in a larger dataset, in which ground-truth is not available. To achieve this, the representativeness of the incomplete data should be guaranteed to some extent, so that the model learned on incomplete data could be applied to other scenarios.

Secondly, there are other application scenarios that require the generation of trip purposes and building a machine learning model simultaneously. For example, as a future work, we plan to study the privacy leakage problem across cities – if an attacker could obtain all the required data of city A and train a model for city A, is it possible that s/he transfers the model to city B and mine trip purpose there? What are the impacts from dynamic prices? A possible circumstance is that the attacker may not have enough data of city B so that the ground-truth in city B is not available.

The overfitting issue with feature crossing. It is a two-edged weapon that feature crossing generates high-dimensional features by multiplication. On one hand, it increases the non-linearity expressiveness and thus improves model accuracy; on the other hand, it may lead to overfitting due to the high-dimension. Our practices to avoid overfitting include:

- We do not cross more than two features. As is stated in Section IV-E-2, crossing more than two features not only makes the resulting feature non-intuitive and hard to interpret, but also dramatically increases the dimension of the feature vector.
- In crossing two features, we do not set any specific rule as to what features are to be crossed. Firstly, crossing all pairs of two features makes the dimension of the resulting feature vector to be 6,073, which is much smaller than the number of training samples. Secondly, no manual feature selection is done in advance, and we could then determine if a particular crossed feature is non-trivial by inspecting the corresponding weight. This may not be perfectly precise, but still gives constructive insights.
- We adopt the L2-regularization to control overfitting: weights of features are restricted to be small enough.

Besides these practices, there are also other possible techniques if necessary. For example, other types of regularization – such as L1-regularization, spatio-temporal regularization, etc., – could be used. L1-regularization puts special attention on feature selection; spatio-temporal regularization, also used in [23], captures the fact that the prediction target should not change much from one occasion to another if these occasions are close enough spatially or temporally. Another example is to perform feature crossing in a hit-and-trial fashion: e.g., crossing features from some particular contexts and then prograssive crossing more by inspecting the intermediate results.

The consideration of individual information. It is true that considering contextual or individual information would improve the accuracy of trip purpose mining. The most direct way of doing this is to extract as many contextual features from all kinds of datasets as possible before model training. Another possible way is to pre-train a model without contextual or individual information, and then use individual information of, say, different groups of people, to fine-tune the pre-trained model. Either way would possibly improve model accuracy and offer more in-depth explanations of trip purposes, but meanwhile the collection of more datasets is required – e.g., surveys, questionnaires, household information, demographics data, etc.

But in our study, it is infeasible to consider much individual information. As trip purpose mining is used as an example to validate the privacy leakage from dynamic prices, it is studied from an attacker's perspective. We therefore choose to use large-scale datasets without individual preference information. For an attacker, it is difficult to obtain the above-mentioned datasets describing contextual or individual information such as characteristics, behaviors, and preferences; instead, we pay special attention to the possibility to obtain datasets and provide relevant discussions.

The applicability of our study. We provide discussions from two sides: applying our methodologies to trip purpose mining with similar datasets, and applying our idea to similar problems or services.

Trip purpose mining with similar datasets. Our criteria of choosing datasets (i.e., “ubiquitous”, “collective” and “easily accessible”) make it possible to apply our methodologies to practice for an attacker. The above discussions about data accessibility explain the ways of collecting similar datasets: they should be either public, or could be approximated. And once datasets are ready, our methodologies could be applied.

Similar problems or services. The idea to study privacy leakage from dynamic prices is simple: if dynamic prices help to find out a privacy-related target easier or with a higher accuracy, then it is safe to claim the existence of privacy leakage. Such an idea could be easily applied to similar problems or services as long as there are relevant datasets to support the inference. For example, in flight ticket service, the target may be chosen as the type of tour, passenger's marital status, passenger's occupation, etc.; in hotel booking service, the target could be also the type of tour, or user's membership status, user's personal information such as age or income, etc.

VI. Conclusion

We focus on the privacy leakage from dynamic prices, which are used in various forms in different services or scenarios. Trip purpose is chosen as a specific example of privacy, and we aim to discover the existence of privacy leakage, and quantitatively measure feature contributions – i.e., which feature helps an attacker in trip purpose mining more and by how much. We tackle the trip purpose mining problem as a multi-class problem or multiple binary-class problems, based on the datasets that are ubiquitous, collective and easily accessible.

Based on the datasets, we extract features that describe the basic trip information, spatial, temporal and dynamic prices context. In the multi-class problem, we train an artificial neural network model, achieving an accuracy of 67.15% in trip purpose mining. The goal is to prove the existence of privacy leakage qualitatively. In the binary-class problems, we train logistic regression models with feature crossing and achieve similar classification performance. The goal is to quantitatively explain feature contributions. Our results also confirm that considering dynamic prices improve classification accuracy by 17.1%, and that information from spatial, temporal and dynamic prices context need to be carefully extracted and coupled together to achieve the biggest classification power.

We also show that dynamic prices lead to different extents of privacy leakage in identifying different trip purposes. Three representative trip purposes are chosen, i.e., "Transportation", "Homing" and "Working". For "Working" trips, the privacy leakage from dynamic prices is minimal, and temporal features such as day-of-week or order start time are the most important in identifying such trips. For "Transportation" trips, dynamic prices features show growing importance; travel time and distance become more significant, which is unique. For "Homing" trips, the level of privacy leakage is the highest, as the number of most influencing features that are relevant to dynamic prices is the largest.

References

[1] S. Guo, Y. Liu, K. Xu, and D. M. Chiu, "Understanding ride-on-demand service: Demand and dynamic pricing," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2017, pp. 509–514.

[2] L. Chen, A. Mislove, and C. Wilson, "Peeking beneath the hood of Uber," in *Proc. ACM Conf. Internet Meas. Conf.*, New York, NY, USA: ACM, 2015, pp. 495–508.

[3] S. Guo, C. Chen, Y. Liu, K. Xu, and D. M. Chiu, "Modelling passengers' reaction to dynamic prices in ride-on-demand services: A search for the best fare," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 1, no. 4, pp. 136:1–136:23, 2018.

[4] S. Guo et al., "ROD-Revenue: Seeking strategies analysis and revenue prediction in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2202–2220, Sep. 2020.

[5] S. Guo et al., "Seeking based on dynamic prices: Higher earnings and better strategies in ride-on-demand services," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 5, pp. 5527–5542, May 2023.

[6] S. Guo et al., "A force-directed approach to seeking route recommendation in ride-on-demand service using multi-source urban data," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1909–1926, Jun. 2022.

[7] Y. M. Nie, "How can the taxi industry survive the tide of ridesourcing? Evidence from Shenzhen, China," *Transp. Res. Part C: Emerg. Technol.*, vol. 79, pp. 242–256, 2017.

[8] S. Jiang, L. Chen, A. Mislove, and C. Wilson, "On ridesharing competition and accessibility: Evidence from uber, lyft, and taxi," in *Proc. World Wide Web Conf.*, 2018, pp. 863–872.

[9] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in san francisco," *Transport Policy*, vol. 45, pp. 168–178, 2016.

[10] Álvaro Aguilera-García, J. Gomez, G. Velázquez, and J. M. Vassallo, "Ridesourcing vs. traditional taxi services: Understanding users' choices and preferences in Spain," *Transp. Res. Part A: Policy Pract.*, vol. 155, pp. 161–178, 2022.

[11] A. Picchi, "Uber vs. taxi: Which is cheaper?," 2016. [Online]. Available: http://bit.ly/2DMgrMc

[12] G. D. Erhardt, S. Roy, D. Cooper, B. Sana, M. Chen, and J. Castiglione, "Do transportation network companies decrease or increase congestion?," *Sci. Adv.*, vol. 5, no. 5, 2019, Art. no. eaau2670.

[13] V. Salnikov, R. Lambiotte, A. Noulas, and C. Mascolo, "Openstreetcab: Exploiting taxi mobility patterns in New York City to reduce commuter costs," 2015, *arXiv:1503.03021*.

[14] J. D. Hall, C. Palsson, and J. Price, "Is Uber a substitute or complement for public transit?," 2017. [Online]. Available: https://bit.ly/2K6Vs7L

[15] T. Berger, C. Chen, and C. B. Frey, "Drivers of disruption? Estimating the uber effect," *Eur. Econ. Rev.*, vol. 110, pp. 197–210, 2018.

[16] H. Wang and H. Yang, "Ridesourcing systems: A framework and review," *Transp. Res. Part B: Methodological*, vol. 129, pp. 122–155, 2019.

[17] R. G. McKane and D. Hess, "The impact of ridesourcing on equity and sustainability in North American cities: A systematic review of the literature," *Cities*, vol. 133, 2023, Art. no. 104122.

[18] M. Nourinejad and M. Ramezani, "Ride-sourcing modeling and pricing in non-equilibrium two-sided markets," *Transp. Res. Part B: Methodological*, vol. 132, pp. 340–357, 2020.

[19] H. Yang, C. Shao, H. Wang, and J. Ye, "Integrated reward scheme and surge pricing in a ridesourcing market," *Transp. Res. Part B: Methodological*, vol. 134, pp. 126–142, 2020.

[20] J. Hall, C. Kendrick, and C. Nosko, "The effects of Uber's surge pricing: A case study," Oct. 2015. [Online]. Available: http://bit.ly/2kayk9O

[21] J. Gan, B. An, H. Wang, X. Sun, and Z. Shi, "Optimal pricing for improving efficiency of taxi systems," in *Proc. 22th Int. Joint Conf. Artif. Intell.*, 2013, pp. 2811–2818.

[22] L. Rayle, S. Shaheen, N. Chan, D. Dai, and R. Cervero, "App-based, on-demand ride services: Comparing taxi and ridesourcing trips and user characteristics in san francisco," 2014. [Online]. Available: https://bit.ly/35R28pl

[23] S. Guo et al., "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," *Proc. ACM Interact. Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 3, pp. 112:1–112:24, 2018.

[24] O. Besbes, F. Castro, and I. Lobel, "Surge pricing and its spatial supply response," *Manage. Sci.*, vol. 67, no. 3, pp. 1350–1367, 2021.

[25] M. K. Chen, "Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform," in *Proc. ACM Conf. Econ. Computation*, New York, NY, USA: ACM, 2016, pp. 455–455.

[26] W. Miao, Y. Deng, W. Wang, Y. Liu, and C. S. Tang, "The effects of surge pricing on driver behavior in the ride-sharing market: Evidence from a quasi-experiment," *J. Operations Manage.*, vol. 69, no. 5, pp. 794–822, 2023.

[27] K.-H. Huarng and T. H.-K. Yu, "The impact of surge pricing on customer retention," *J. Bus. Res.*, vol. 120, pp. 175–180, 2020.

[28] M. Xu Anupriya and P. Bansal, "Surge pricing and consumer surplus in the ride-hailing market: Evidence from China," *Travel Behav. Soc.*, vol. 33, 2023, Art. no. 100638.

[29] K. Xue et al., "PPSO: A privacy-preserving service outsourcing scheme for real-time pricing demand response in smart grid," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2486–2496, Apr. 2019.

[30] Z. Zhang, Z. Qin, L. Zhu, J. Weng, and K. Ren, "Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 619–626, Mar. 2017.

[31] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differentially private dynamic pricing for efficient demand response in smart grid," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.

[32] X. Chen, D. Simchi-Levi, and Y. Wang, "Privacy-preserving dynamic personalized pricing with demand learning," *Manage. Sci.*, vol. 68, no. 7, pp. 4878–4898, 2022.

[33] L. Alessandretti, U. Aslak, and S. Lehmann, "The scales of human mobility," *Nature*, vol. 587, no. 7834, pp. 402–407, 2020.

[34] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.

[35] P. S. Castro, D. Zhang, C. Chen, S. Li, and G. Pan, "From taxi GPS traces to social and community dynamics: A survey," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 17:1–17:34, 2013.

[36] Z. Fan, X. Song, R. Jiang, Q. Chen, and R. Shibasaki, "Decentralized attention-based personalized human mobility prediction," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 4, pp. 133:1–133:26, 2019.

[37] G. Wang et al., "sharedCharging: Data-driven shared charging for large-scale heterogeneous electric vehicle fleets," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 108:1–108:25, 2019.

[38] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 38:1–38:55, 2014.

[39] J. Li et al., "Drive2friends: Inferring social relationships from individual vehicle mobility data," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5116–5127, Jun. 2020.

[40] X. Han, L. Wang, and W. Fan, "Is hidden safe? Location protection against machine-learning prediction attacks in social networks," *MIS Quart.*, vol. 45, no. 2, pp. 821–858, 2021.

[41] M. H. Nguyen, J. Armoogum, J.-L. Madre, and C. Garcia, "Reviewing trip purpose imputation in GPS-based travel surveys," *J. Traffic Transp. Eng.*, vol. 7, no. 4, pp. 395–412, 2020.

[42] C. M. Krause and L. Zhang, "Short-term travel behavior prediction with GPS, land use, and point of interest data," *Transp. Res. Part B: Methodological*, vol. 123, pp. 349–361, 2019.

[43] Y. Cui, C. Meng, Q. He, and J. Gao, "Forecasting current and next trip purpose with social media data and Google places," *Transp. Res. Part C: Emerg. Technol.*, vol. 97, pp. 159–174, 2018.

[44] C. Meng, Y. Cui, Q. He, L. Su, and J. Gao, "Travel purpose inference with gps trajectories, pois, and geo-tagged social media data," in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 1319–1324.

[45] G. Xiao, Z. Juan, and C. Zhang, "Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization," *Transp. Res. Part C: Emerg. Technol.*, vol. 71, pp. 447–463, 2016.

[46] C. Chen, S. Jiao, S. Zhang, W. Lu, L. Feng, and Y. Wang, "Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 10, pp. 3292–3304, Oct. 2018.

[47] C. Liao et al., "Enriching large-scale trips with fine-grained travel purposes: A semi-supervised deep graph embedding framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 11, pp. 13228–13239, Nov. 2023.

[48] P. Wang, G. Liu, Y. Fu, Y. Zhou, and J. Li, "Spotting trip purposes from taxi trajectories: A general probabilistic model," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, pp. 29:1–29:26, 2018.

[49] C. Liao et al., "Wheels know why you travel: Predicting trip purpose via a dual-attention graph embedding network," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 6, no. 1, pp. 22:1–22:22, 2022.

[50] B. Shaw, J. Shea, S. Sinha, and A. Hogue, "Learning to rank for spatiotemporal search," in *Proc. 6th ACM Int. Conf. Web Search Data Mining*, 2013, pp. 717–726.

[51] F. Wu and Z. Li, "Where did you go: Personalized annotation of mobility records," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 589–598.

[52] T. Sakouhi and J. Akaichi, "Dynamic and multi-source semantic annotation of raw mobility data using geographic and social media data," *Pervasive Mobile Comput.*, vol. 71, 2021, Art. no. 101310.

[53] R. Hu, J. Zhou, X. Lu, H. Zhu, S. Ma, and H. Xiong, "NCF: A neural context fusion approach to raw mobility annotation," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 226–238, Jan. 2022.

[54] D. Lian, Y. Zhu, X. Xie, and E. Chen, "Analyzing location predictability on location-based social networks," in *Proc. 2014 Pacific-Asia Conf. Knowl. Discov. Data Mining*, 2014, pp. 102–113.

[55] PaddlePaddle, "The ERNIE open-source development kit," 2022. [Online]. Available: https://www.paddlepaddle.org.cn/paddle/ernie

[56] AMap, "API of AMap service," 2022. [Online]. Available: http://bit.ly/2n8YRbZ

Suiming Guo received the PhD degree from the Chinese University of Hong Kong. He is currently an associate professor with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include data mining, urban computing, pervasive computing, and smart cities studies.

Chao Chen received the PhD degree from UMPC (Paris 6) and Telecom SudParis. He is currently a full professor of computer science with Chongqing University, China. His research interests include pervasive computing, social network analysis, and mobile crowdsensing.

Zhetao Li (Member, IEEE) received the BEng degree from Xiangtan University, in 2002, the MEng degree from Beihang University, in 2005, and the PhD degree from Hunan University, in 2010. He is a professor with the College of Information Science and Technology, Jinan University. From 2013 to 2014, he was a postdoc in wireless network with Stony Brook University. He is a member of CCF.

Chengwu Liao is currently working toward the PhD degree with the College of Computer Science, Chongqing University, China. His research interests include mobile computing, spatiotemporal trajectory mining, urban data visualization, and intelligent transportation systems.

Yaxiao Liu received the PhD degree from Tsinghua University. He is currently a senior manager with AWS China. His research interests include AI based cloud architecture, spatio-temporal Big Data, stream computing, and smart cities.

Ke Xu (Fellow, IEEE) received the PhD degree from Tsinghua University. He is currently a full professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include next generation Internet, P2P systems, Internet of Things(IoT), network virtualization, and optimization.

Daqing Zhang (Fellow, IEEE) received the PhD degree from the University of Rome “La Sapienza” and University of L’Aquila. He is a full professor with Institut Mines-Telecom/Telecom SudPais. His research interests include large-scale data mining, urban computing, context-aware computing, and ambient assistive living.