Engaging Drivers in Ride Hailing via Competition: A Case Study with Arena

Hao Cheng[†], Shuyue Wei[†], Lingyu Zhang[§], Zimu Zhou[‡], Yongxin Tong[†]
[†] *SKLSDE Lab, BDBC and IRI, Beihang University, Beijing, China*
[‡] *School of Information Systems, Singapore Management University, Singapore*
[§] *Shandong University and AI Labs at Didi Chuxing*
[†]{haocheng,weishuyue,yxtong}@buaa.edu.cn
[§]zhanglingyu@didiglobal.com, [‡]zimuzhou@smu.edu.sg

***Abstract*—Sustained work enthusiasms of drivers are crucial for the success of large-scale ride-hailing platforms. In this paper, we conduct the first-of-its-kind exploration to encourage active participation of drivers via competition. We design `Arena`, a competition where drivers compete for prizes via completing more trips. Through a pilot study covering over 2,600 participants, we uncover the easy-win problem, an overlooked and serious issue in competition design for real-world drivers. It refers to situations where one competitor does not show up during competition whereas the other easily wins. To solve the easy-win problem without impairing motivation of drivers, we devise a novel prediction-based matchmaking framework. On observing that no-shows are highly correlated to the online time of drivers during competition, we propose to identify potential no-shows by predicting drivers' online time and avoid matching potential no-show drivers with drivers that will show up so as to reduce easy-wins. We conduct large-scale experiments based on real competition data involving over 10,000 drivers. The results show that our prediction-based matchmaking scheme can effectively reduce the ratio of easy-wins.**

***Index Terms*—Spatial Crowdsourcing, Competition, Case Study**

I. Introduction

Ride-hailing services have gained unprecedented popularity due to its flexibility brought to the transportation industry. Large-scale ride-hailing companies such as Uber and DiDi have 3.9 million [1] and tens of millions [2] of drivers registered worldwide, respectively, where drivers can complete trips at will as long as their total driving time does not exceed the limit within a day. Such flexibility tends to result in relatively short working hours of drivers. According to some recent reports [3], more than half of UberX (the most popular product of Uber) drivers only drive for less than 15 hours a week, and the proportion of driving less than 35 hours a week is 83%. DiDi also experiences a similar situation, where 50.67% of drivers are online for less than 2 hours a day on average according to a report released by DiDi in 2017 [4]. The short service time of part-time drivers is likely to induce shortage in drivers especially during rush hours, an unwanted situation for both the ride-hailing platforms and the passengers.

To consistently provide sufficient ride-hailing services, the platforms need to encourage their drivers to complete more trips via various incentives. For instance, dynamic pricing mechanisms [5], [6] have been explored to dynamically balance demand and supply. Many non-monetary incentives have also been adopted. For example, Uber Pro [7] is a rewards program that gives more rewards to higher-status drivers. Drivers needs to drive more and give better service to earn a higher status.

Competitions have proven an effective incentive mechanism to boost players' performance [8], [9], [10], [11]. As the first-of-its-kind exploration, we introduce competitions into a ride-hailing platform and design `Arena`, a system that enables multi-round competitions among drivers where the driver who wins more rounds can get a bigger prize. Our vision of such a mechanism is threefold, *i.e.*, drivers will obtain higher income by completing more trips, passengers will find it more easily to be picked up when drivers are more willing to drive, and the platform will be better off with the increased satisfaction of both drivers and passengers. Through a pilot deployment of `Arena` in 3 cities where over 2,600 drivers participated in the competitions, we find that although competitions do motivate drivers, drivers who register for competitions are not guaranteed to attend each round as expected, leading their opponents to easily win some rounds. We call such a phenomenon the *easy-win* problem. Our pilot study shows an unexpectedly high easy-win ratio of 30% to 50%, which will severely impair the motivation of drivers and the effectiveness of competitions.

The identification of the easy-win problem reveals something important to take into consideration when applying competition-based incentive mechanism on ride-hailing platforms, or more generally, spatial crowdsourcing applications. To mitigate the easy-win problem without harming the motivation of drivers, we propose a novel prediction-based matchmaking scheme. The idea is to predict the online times of drivers in a future round, and only match drivers with similar predicted online times to avoid matching potential no-shows with drivers that will show up based on the predictions. We harness heterogeneous data and a Seq2Seq [12] model for accurate driver online time prediction. Experiments based on the data of two real-world competitions held in two different cities demonstrate the effectiveness of our proposed method.

The main contributions of this work are as follows:

- We design and deploy `Arena`, the first-of-its-kind competition for drivers of ride-hailing platforms. Through a

pilot study in 3 cities covering over 2,600 drivers, we observe the *easy-win* problem, a phenomenon largely overlooked in competition designs in other application domains yet may severely affect the effectiveness of competition designs for real-world drivers.
- We propose a novel prediction-based matchmaking scheme to mitigate the easy-win problem. Evaluations based on the data of two real-world competitions in two cities involving about 10,000 participants show that our proposed method can effectively reduce the ratio of easy-wins by up to over 20 percentage points compared with the baseline method.

The rest of this paper is organized as follows. We review related work in Sec. II, present the initial design of `Arena` and identify the easy-win problem through a large-scale pilot study in Sec. III, and introduce our solution to the easy-win problem in Sec. IV. In Sec. V we show the evaluation results and we conclude in Sec. VI.

II. Related Work

Our work belongs to one category of incentive mechanisms for crowdsourcing, and our solution relates to the research on user activity prediction. We briefly review the closely related studies below.

A. Competition-Based Incentive Mechanisms in Crowdsourcing

In the research field of crowdsourcing, incentive mechanism design attracts much interest because the participation and retention of workers directly affect the viability of the crowdsourcing market. Recently, gamification has been a popular means to enhance user engagement in an activity [13], [14]. Without designing a fully-fledged game, gamification achieves this by incorporating some game elements into the application [15]. Typical such elements include points, badges, levels and leaderboards [13]. Gamification has already been effectively employed in crowdsourcing to increase the motivation of the participants [16]. There are both *theoretical* and *experimental* studies on competitions in crowdsourcing.

Competitions are often studied theoretically in contest theory [17]. In general, a contest is modeled as an all-pay auction in a game-theoretic framework. Theoretical studies typically focus on analyzing the equilibria of contests [18], [19] and designing optimal prize allocation mechanism to maximize expected total effort or maximum effort [20], [21] under different contest models and assumptions. With the rise of crowdsourcing, many crowdsourcing scenarios where multiple workers compete for the reward of a task can be seen as contests. The design of crowdsourcing contests has been researched extensively [22], [23], [24]. Our work is different from these theoretical studies from the following two aspects. *(i)* In our work, the spatial crowdsourcing itself is not a contest since each trip is assigned to only one driver and drivers do not compete for a certain trip. Contests are used to motivate drivers as a supplement of the fare they earn from trips. *(ii)* Most theoretical results may not fit for real-world applications. For example, the optimal prize function in [24] is in a complicated non-closed-form expression and it is derived based on many assumptions. In contrast, we focus on adoption of competition design in real-world applications.

Among the experimental studies, Feyisetan *et al.* [8] find that game elements, including points and leaderboards, can improve the accuracy while reducing the cost in image labeling experiments. To explore the effects of reward distributions and information policies in competitions, Rokicki *et al.* [9] conduct large-scale experiments on Amazon Mechanical Turk to collect annotations for datasets. They find that the best design is to award several top users unequally and reveal close scores of other users to each one. Our work also focuses on practical problems that stem from real-world deployments. Particularly, we uncover the easy-win problem, which is largely overlooked but significantly affects the effectiveness of adopting competitions among real-world drivers.

B. User Activity Prediction

In customer relationship management (CRM), one important problem is predicting the future activity of customers/users. A related topic is churn prediction which aims to find out users who are likely to leave the supplier/platform in advance such that some actions can be taken to motivate them to stay [25], [26]. User activity prediction has been studied in different areas ranging from social networks [27] to telecommunication [28]. In social networks, the prediction can help the platform to deicde which users should be given new services, free e-gifts, etc. [27] in order to keep them active in the future. Zhu *et al.* [27] propose a method based on logistic regression, which incorporates personalization, dynamics and social influence, to predict whether a user will be active in the social network during the subsequent week. In telecommunication, the service providers can benefit from predicting user activity level because the service providers can adjust their management strategies to retain profit based on the prediction [24].

Our prediction method is not designed for driver churn analysis, but serves as an optimization component of the competition system. Most existing studies [24], [27] of user activity prediction only predict whether a user will be active or inactive in the future (*i.e.*, a classification problem), however, in our work, we predict the online time of drivers in the future (*i.e.*, a regression problem). In addition, we need to predict the online time of drivers in multiple hours (*i.e.*, multi-step prediction) while most existing studies [24], [27] only focus on one period in the future, *e.g.*, one day or one month (*i.e.*, single-step prediction). The prediction in this paper is more challenging due to both the finer granularity in time and the multi-step nature of the problem.

III. Measurements and Observations

In this section, we introduce `Arena`, a competition designed for drivers of ride-hailing platforms (Sec. III-A). Through measurements from real-world deployments, we show the problem of *easy-win*, a phenomenon largely overlooked yet crucial for the effectiveness of competition design for drivers (Sec. III-B).

A. `Arena`: Initial Design and Deployment

`Arena` is a competition organized among drivers of ride-hailing platforms. It serves as a gamification incentive for drivers to complete more trips, so as to improve the profit of both the drivers and the ride-hailing platform. Towards this goal, the competition content of `Arena` is aligned with the daily work of drivers, *i.e.*, drivers who complete more trips win. Fig. 1 illustrates the core workflow of `Arena`. We explain the details of each component as follows.

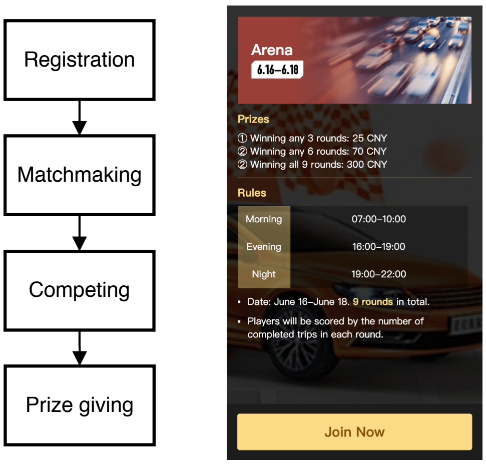


Fig. 1: `Arena` overview and a screenshot of an `Arena` competition held on drivers' mobile app.

- *Registration.* A competition is released to eligible drivers through drivers' mobile app (see the screenshot in Fig. 1). An eligible (a.k.a whitelist) driver has to register to join the competition. Registration begins several days before the competition to attract more drivers.
- *Matchmaking.* In each round, the platform needs to find an appropriate opponent for each driver. According to the theory of two-player standard all-pay contest [17], we should match drivers of similar ability to maximize their total effort. Hence in `Arena`, we perform matchmaking only among drivers who completed almost the same number of trips in the last 30 days before the competition.
- *Competing.* Each competition lasts for several days and consists of multiple rounds. Each round lasts for multiple hours, typically during peak hours. In each round, the matched drivers will complete trips from the ride-hailing platform. The platform will give each driver a score proportional to either the total number or total fare of the completed trips. Drivers are expected to compete for a higher score to win a round. A draw occurs when both drivers get no score.
- *Prize Giving.* After all rounds of a competition, a driver gets a prize according to the rounds of wins in the competition based on the prize allocation rule. For example, in the completion shown in Fig. 1, a driver will get 25 CNY for winning any 3 rounds, or 70 CNY for winning any 6 rounds, or 300 CNY for winning all the 9 rounds.

TABLE I: Impact of competitions on the number of completed trips.

City	Group	# drivers	Change	Growth
City A	Experimental	569	21.03%	27.14 pp
	Control	500	-6.11%	
City B	Experimental	1807	26.39%	23.09 pp
	Control	2000	3.30%	
City C	Experimental	278	30.64%	25.43 pp
	Control	300	5.21%	

As a pilot study, we deployed `Arena` in three Chinese cities of different scales in June, 2018, and organized 3 competitions consisting of 24 rounds involving over 2,600 drivers. The concrete parameters *e.g.*, number of rounds in a competition, duration of a round, prize allocation rule, etc. are heuristically set for the competitions in each city. Note that the focus of the pilot study is not an exhausted search on the optimal competition design that maximizes participation of drivers or profit of the platform, but the first-of-its-kind exploration of adopting real-world competitions as an incentive for drivers.

Table I summarizes the impact of competitions on the total number of completed trips. The experimental group are drivers registered in competitions. For comparison, we randomly select drivers from the same city at roughly the same scale as the control group for each competition. For each group, we count the number of completed trips during the competition as well as the number of completed trips in the same periods of the same days a week before the competition. For example, the competition in City A was held from June 7, 2018 to June 9, 2018 with 3 rounds, each lasting for 3 hours, every day. The periods for comparison are the same 9 hours from May 31, 2018 to June 2, 2018. The change in Table I is calculated as the difference in the number of completed trips during and before competitions for each group. The growth is calculated as the difference between the change of the experimental group and that of the control group. As shown in Table I, the growth between the experimental and the control groups are significant (from 23.09 to 27.14 percentage points). It suggests that competitions can encourage drivers to complete more trips, and thus potentially increase the profits of both the drivers and the ride-hailing platform.

B. The Easy-Win Problem

Although competition seems to motivate drivers to complete more trips (see Table I), a closer investigation on the participation behaviors of drivers reveals a high rate of *no-show* during competitions and correspondingly a high *easy-win ratio*, which may severely impair the motivation of drivers and decrease the profit of the platform.

1) Easy-Win: An easy-win occurs when one driver gets no score. In this case, the other driver can easily win that round. To quantify the severity of easy-win, we calculate the *easy-win* *ratio* of each competition held in our pilot study. It is the ratio between the number of easy-wins and total number of driver pairs in a single round or an entire competition. Table II shows the easy-win ratios of the competition held in City A (The other two competitions exhibit similar or even higher easy-win ratios). The winning status of a driver in a single round can be classified into three categories: easy-win, normal-win and draw. A normal-win occurs when both drivers get non-zero scores while a draw occurs only when both drivers get no score as mentioned above. In Table II, the others include normal-wins and draws.

From the table, we can observe that in most rounds, the easy-win ratio is higher than 40%. Furthermore, the competition covers workdays and weekends, as well as different periods of a day. The easy-win ratio remains high despite what day the competition is held, and seems irrelevant to the periods of a day. In summary, a high easy-win ratio tends to be ubiquitous in the competitions held by `Arena`.

2) Impact of Easy-Win: While easy-wins are not unique in competitions for drivers, a high easy-win ratio may weaken the motivating effect of competitions and decrease the profit of the platform.

To show the impact of easy-wins on drivers, we plot the distributions of winner's score in cases of easy-wins and normal-wins. Due to the limited space, we only show the results for City B (Fig. 2 left). The teal area represents the distribution of winners' scores in cases of normal-wins of the competition while the orange area represents the distribution in cases of easy-wins. We can observe that the scores of winners in cases of normal-wins tend to be higher. Using Welch's t-test, we find that such differences between the winner's scores in cases of easy-wins and normal-wins are statistically significant ($p < 0.0001$). The results indicate that when competing with a stronger opponent (normal-win), drivers incline to strive for a higher score. Therefore, one negative effect of a high easy-win ratio is that it discourages drivers to exert more effort for winning the competition.

To investigate the impact of easy-wins on the platform, we plot the distributions of the total score of each pair of drivers in the competition of City B (Fig. 2 right). In cases of easy-wins, the total score equals the score of the winner. Hence the distribution of the total score is the same as that of the winner's score. However, in cases of normal-wins, the distribution of the total score notably moves to the right, meaning that the drivers who fail also get non-negligible scores. The average total score in cases of easy-wins is 4.93, whereas that in cases of normal-wins is 10.74, more than twice the amount of the former. Note that the cost of each win for the platform is the same on average. Therefore, if the platform can turn all easy-wins to normal-wins, the platform may obtain more than twice the profit (due to more than twice the number of completed trips) with the same cost. In other words, the return on investment (ROI) for the platform will be higher. Thus the results demonstrate another negative effect of a high easy-win ratio: it decrease the ROIs of competitions for the platform.

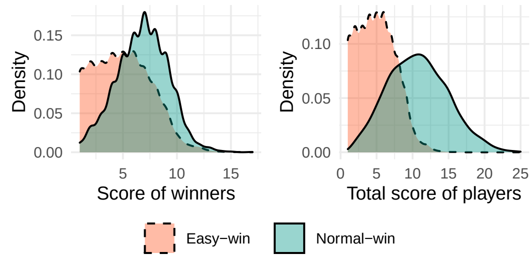


Fig. 2: Distributions of scores in cases of easy-wins and normal-wins of the competition in City B.

3) Countermeasures to Easy-Win: Easy-wins occur if one driver gets zero score. To reduce easy-wins, we analyze the behaviors of 0-score drivers and the underlying reasons. Most 0-score drivers were online for only several minutes during a 3-hour round and about 85% of 0-score drivers were never online in each round. Since no-shows comprise a large proportion of 0-score drivers, we may conclude that most easy-wins occur when drivers register for a certain competition but do not show up when the competition actually takes place. One way to mitigate easy-wins is to punish the no-show drivers. However, it may also harm the motivation of the drivers [29]. Alternatively, instead of changing competition rules (*i.e.*, punishing no-show drivers), we propose to reduce easy-wins by avoiding matching *potential* no-show drivers with drivers who will show up in *matchmaking*.

4) Summary of Observations: There is a high easy-win ratio (30% to 50%) in most competitions organized in `Arena`. A high easy-win ratio impairs both the motivation of drivers and the profit of the platform. To decrease easy-win ratio without discouraging drivers, we choose to mitigate easy-wins during *matchmaking*. Specifically, on observing that the driver's online time is an indicator to whether he/she will participate in the competition, we propose to reduce easy-wins by predicting the online time of drivers and avoiding matching potential no-shows with drivers that will show up, as will be elaborated on in the following.

IV. Prediction-based Matchmaking

In this section, we design a prediction-based matchmaking scheme for `Arena`, where the core enabler is to predict the duration of online time before matchmaking so as to avoid no-shows and thus mitigate easy-wins. We first show the feasibility of predicting driver online time (Sec. IV-A), then explore potential features for online time prediction (Sec. IV-B), as well as the prediction model (Sec. IV-C), and finally present our prediction-based matchmaking scheme (Sec. IV-D).

A. Regularity of Driver Online Time

Our prediction-based matchmaking scheme relies on the assumption that drivers' online time exhibits regular patterns and thus tends to be predictable.

As an example, we randomly choose a driver who did not complete any trip during R1 and R2 on June 7, 2018 in the

TABLE II: Easy-win ratios of a competition held in City A.

Day Round	Day 1 (Thur., June 7, 2018) R1	 R2	 R3	Day 2 (Fri., June 8, 2018) R4	 R5	 R6	Day 3 (Sat., June 9, 2018) R7	 R8	 R9	Total
EW	57	75	80	99	112	115	90	111	120	859
Others	83	110	121	137	142	147	182	170	164	1256
EWR	0.4071	0.4054	0.3980	0.4195	0.4409	0.4389	0.3309	0.3950	0.4225	0.4061

EW: Easy-wins, EWR: Easy-win ratio.
Time of R1, R4, R7: 7:00-10:00. Time of R2, R5, R8: 16:00-19:00. Time of R3, R6, R9: 20:00-23:00.

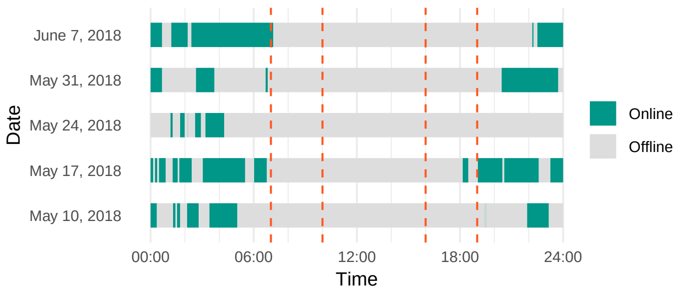


Fig. 3: Online time of a driver on 5 consecutive Thursdays. The dashed lines represent the periods of two rounds.

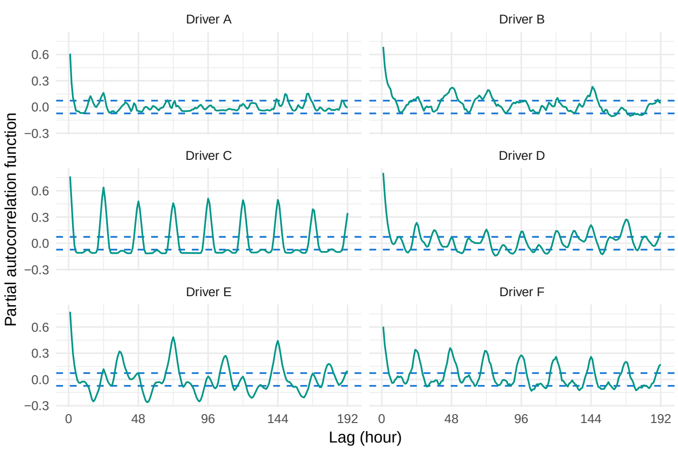


Fig. 4: Partial autocorrelation plots of six randomly sampled drivers' hourly online time in 30 days.

competition of City A shown in Table II. Then we plot the online time of this driver on that day and 4 past consecutive Thursdays, as shown in Fig. 3. The driver was almost always offline in these two rounds; thus he/she could not get any score in both rounds, allowing his opponents to enjoy an easy-win. However, if we check his/her online records in the past Thursdays, we can observe the regularity in the work schedules of this driver: he/she often works by night and rests by day. Using this driver's historical online time records, we may predict that the driver will not attend those two rounds and avoid potential easy-wins.

Such regularity of online time is common among drivers. Fig. 4 shows the partial autocorrelation plots of six randomly sampled drivers' hourly online time in 30 days. In these plots, the maximum lag is set as 192 hours (8 days) for clarity, and the dashed blue lines represent the 95% confidence interval beyond which the autocorrelations are statistically significantly different from zero. For most drivers, their online time has significant correlation with that in the past days. Therefore, it is feasible to predict drivers' online time in the future based on their online time records in the past. However, the patterns vary across drivers as shown in the figure. Hence predicting drivers' online time is non-trivial and additional features are necessary, as we will explain below.

B. Features for Driver Online Time Prediction

In addition to the historical online time of a driver, we further explore multiple explainable features that may impact a driver's online time. We mainly extract features from public data sources or drivers' basic information accessible by the ride-hailing platform. All the plots below are based on results from 10,000 randomly sampled drivers in City D, where no pilot study of `Arena` was launched.

1) Features from Public Datasets: Intuitively, time and weather affect the schedule of drivers [30]. We explore the following features from public time and weather datasets for driver online time prediction.

- *Day of the Week and Holiday.* Fig. 5 shows the daily average online time of sample drivers in 2017. The data points corresponding to weekends and holidays are marked in red. An obvious phenomenon is the weekly periodicity in the online time of drivers. In most weeks, the average online time stays at a relatively high level during workdays and drops at weekends. The average online time drops to the lowest level on Sunday for most weeks. Apart from weekly periodicity, the weekends and holiday effects can also be easily observed. Based on these observations, we can conclude that drivers' online times are affected by day of the week and whether it is a workday.
- *Weather.* Weather-related features such as temperature and rainfall are also likely to affect the work schedules of drivers. Due to the limit of data access, we only use the weather data and online record data from April, 2018 to July, 2018. To reduce the effects of other factors, we only analyze the online records of the sample drivers from 20:00 to 21:00 on all workdays. Fig. 6 shows the effect of temperature and rainfall intensity on the online time of drivers. As the temperature increases, the hourly average online time of the sample drivers gradually decreases,

which means drivers may be less willing to work when it is too hot. Compared with temperature, the increase of rainfall intensity does not affect the online time of drivers as much probably because the records of moderate rain and heavy rain are rare (only 3.57%). However, we can still observe that compared with non-rainy hours, the average online time of drivers drops when it rains. As a result, we use both temperature and rainfall intensity features in the prediction model.

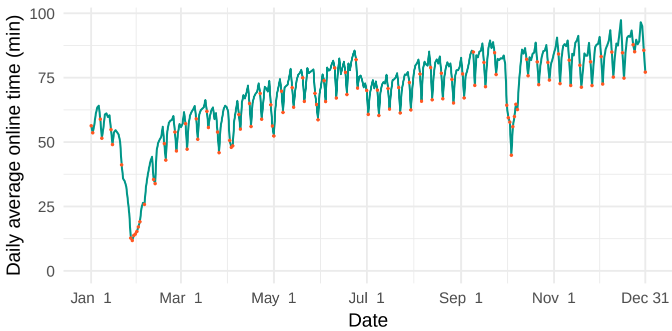


Fig. 5: Daily average online time of the sample drivers

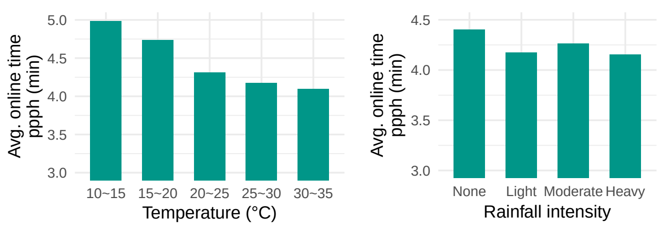
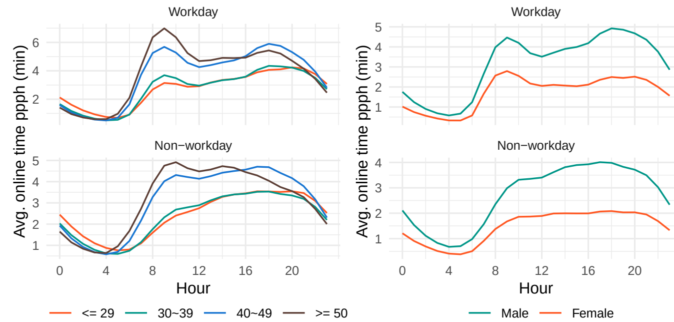


Fig. 6: Average online time per person per hour (ppph) of sample drivers under different weather conditions.

2) Features from Drivers' Basic Information: As is shown in Sec. IV-A, the daily schedules of drivers may vary from person to person. Hence it is necessary to account for driver-specific features when predicting their online time. Due to the limited access to driver information, we only explore two types of basic information as candidate features.

- *Age of driver.* Fig. 7a shows the average hourly online time of the sample drivers in different age groups in 2017. We divide the sample drivers into four age groups. Drivers older than 50 are the most active ones in the morning and the least active ones at night compared with other drivers. Similarly, drivers in their forties are also active in the morning, but they stay working till evening. In contrast, drivers younger than 40 are less active during the day while their average online time increases in the evening. At night, unsurprisingly, drivers in their twenties are the most energetic. The trends on workdays and non-workdays are similar except that the overall online time on workdays is longer.
- *Gender of driver.* Fig. 7b plots the effects of gender on the drivers' online time on workdays and non-workdays



(a) Effect of age.

(b) Effect of gender.

Fig. 7: Average online time per person per hour (ppph) of sample drivers of different driver-specific features.

in 2017. The difference between male drivers and female ones is apparent. The average online time of male drivers are consistently longer than female ones during a day no matter whether it is a workday. The trends on different days are similar except there is no clear peak in the morning on non-workdays.

To summarize, time-dependent features include history online time, time and weather information. More concretely, we use the online time records of a driver in the same hour on the same day during the last month and the average of them as the history features. Time features are the attributes of a specific time point itself, including hour, day of week, and whether the day is a workday. Weather features include the temperature and rainfall intensity in an hour. Driver-specific features are features that are related to a specific driver, *i.e.*, the gender and age range of the driver.

C. Prediction Model

We adopt Seq2Seq [12] model for driver online time prediction. Fig. 8 illustrates the structure of the Seq2Seq model. The Seq2Seq model consists of two parts, an encoder and a decoder. For both the encoder and the decoder, we use gated recurrent units (GRU) due to thier comparable performance to long-short term memory (LSTM) units [31] and their training efficiency as a result of the simpler structure.

The encoder first encodes the history sequence of a driver's online time in the past hours to a hidden state. At each time step, the input for the encoder is a vector consisting of the online time of the driver and other time-dependent features. The hidden state produced by the encoder is then passed to the decoder as its initial state. The decoder can recursively generate an output for an hour in the future if it is fed with an input. Its output is then concatenated with driver-specific features and fed to a fully connected layer together which will finally generate an estimated online time of the driver in the next hour. In our model, the first input to the decoder is a vector consisting of the last online time of the driver before the time we predict and time-dependent features. The output of the decoder will then be concatenated with time-dependent

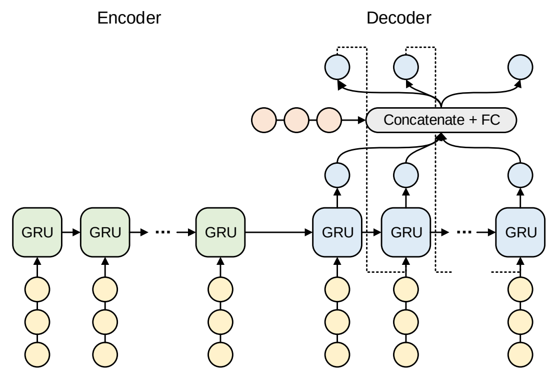


Fig. 8: An illustration of Seq2Seq model.

features of the next hour to be the next input to the decoder. The fully connected layer serves as a correction to the output of the decoder by incorporating driver-specific features. Given the number of time steps to predict, the final outputs of our model are the estimated online time of a given driver in the future hours.

D. Prediction-Based Matchmaking: Putting It Together

Once we make predictions about the online time of the registered drivers in the following hours of an upcoming round, we can use that to match them into pairs together with their ability information.

Algorithm 1 shows our prediction-based matchmaking algorithm. For an upcoming round, we first divide the registered drivers into groups according to their predicted total online time in the round and the predefined bin width. For example, if the round lasts for three hours and the bin width is 0.2 hours, then we will have at most 15 groups of drivers, *i.e.*, drivers that are predicted to be online for less than 0.2 hours, drivers that are predicted to be online for less than 0.4 hours but more than 0.2 hour, etc. In each group, we rank drivers according to their ability level (*e.g.*, the number of completed trips in the past 30 days in our experiments) and then iteratively pair two drivers with a similar ability level together.

V. Evaluation

This section presents the evaluations of our method based on the data of real-world competitions organized among drivers. We first introduce the experiment setup in Sec. V-A and then describe the results of our evaluations in Sec. V-B.

A. Experiment Setup

Our evaluation is based on the data of two competitions held in Cities D and A. Note that one competition was held in a different city from those in the pilot study. Table III summarizes the information of competitions. The competitions cover different days, including workdays (*e.g.*, Aug 6, 2018 to Aug 8, 2018) and weekends (*e.g.*, Dec 8, 2018 to Dec 9, 2018). In the competition of City A, drivers can register for rounds during different periods at will, and we report the number of unique participants here.

Algorithm 1: Prediction-based matchmaking algorithm

Data: Registered drivers for the next round, predicted total online time of each driver, ability level of each driver, bin width
Result: Matching M among drivers

```
Divide drivers into groups G_1, ..., G_n according to
  their predicted total online time and the bin width;
Initialize the matching M ← ∅;
Initialize the candidate set C ← ∅;
for i in 1, ..., n do
    D ← sorted drivers in G_i according to their ability levels;
    while D still has at least two drivers do
        Select two top-ranked drivers from D, add the
          pair of them to M, and remove them from D;
    if there is still one driver in D then
        Add the driver in D to C;
C ← sorted drivers in C according to their predicted
  total online time;
while C still has at least two drivers do
    Select two top-ranked drivers from C, add the pair
      of them to M, and remove them from C;
return M
```

TABLE III: Information of the real-world competitions where different matchmaking methods are tested.

City	Dates	Rounds	Participants
City D	Aug 6, 2018 to Aug 8, 2018	9	3686
City A	Dec 7, 2018 to Dec 9, 2018	9	5930

Based on existing data, we tested three matchmaking schemes and evaluated the easy-win ratios under those matchmaking schemes. Next we elaborate on how the experiments were conducted.

1) Compared Methods: We compare the performance of three matchmaking schemes.

- Baseline: It is a standard matchmaking algorithm without exploiting the prediction of drivers' online time. Specifically, it first divides registered drivers into two groups according to whether they are online when the algorithm is executed. Then in each group, drivers are matched in a similar way as Algorithm 1.
- RF: It is a prediction-based matchmaking scheme integrated with a simpler prediction model. RF adopts the prediction-based matchmaking framework in Algorithm 1, but uses random forest model to make predictions. The model is trained using the features identified in Sec. IV-B and drivers' online time records in the last 720 hours. During prediction, the later predictions are based on the former predictions and the weather features

are extracted from weather forecast data. The random forest model is implemented with scikit-learn [32], and the number of trees and maximum depth of the tree are set to 100 and 20, respectively. The bin width in Algorithm 1 is set to 0.2 hours.

- Seq2Seq (our method): It uses the Seq2Seq model introduced in Sec. IV-C to predict the online time of drivers in each hour of the future round and then follows Algorithm 1 to match drivers. The training data is the hourly online time records of all whitelist drivers in the last two months before the competition and other features described in Sec. IV-B. For each round, we first use the trained model to predict the online time of the registered drivers. The weather features used when predicting are extracted from weather forecast data. Then the matchmaking algorithm is executed based on the predicted total online time of drivers to match them into pairs. The bin width in Algorithm 1 is set to 0.2 hours.

2) Evaluation Metrics: We use easy-win ratio (EWR), *i.e.*, the ratio between the number of easy-wins and total number of driver pairs as defined in Sec. III, to assess the performance of different algorithms.

3) Experiment Procedure: For each competition, we evenly assign each driver a type and each algorithm only matches drivers of a specific type. Since Seq2Seq and RF are both prediction-based matchmaking schemes and Seq2Seq always outperforms RF in the competition of City D, we only evaluate Seq2Seq and Baseline in the competition of City A to test them on a larger body of participants. Since our experiments are based on existing data, we cannot measure the number of completed trips of drivers when the matching is changed; instead, we assume the number of completed trips of each driver does not change even when the matching is different.

B. Experiment Results

1) Evaluations of Easy-Win Ratios: The overall performance of each algorithm is shown in Fig. 9. In both competitions, our method, Seq2Seq, can effectively reduce the easy-win ratio compared with Baseline. The improvement of Seq2Seq over Baseline is the more notable in the competition held in City A where the easy-win ratio is almost halved. Another prediction-based matchmaking algorithm, RF, also achieves lower easy-win ratio than Baseline, which proves the effectiveness of our proposed prediction-based matchmaking scheme in reducing the easy-win ratio. However, Seq2Seq is even better than RF indicating the superiority of the Seq2Seq model. The easy-win ratios exhibit some variations across different competitions, probably because they are held in different time during different days.

Table IV shows the performance of each algorithm in the competition held in City D. Compared with the ubiquitous high easy-win ratio (about 40%) shown in Table II, the Baseline algorithm that considers whether the driver is online before each round begins can lower the easy-win ratio to some extent, especially in the evening rounds (R2, R5, R8) and night rounds (R3, R6, R9). However, in the morning

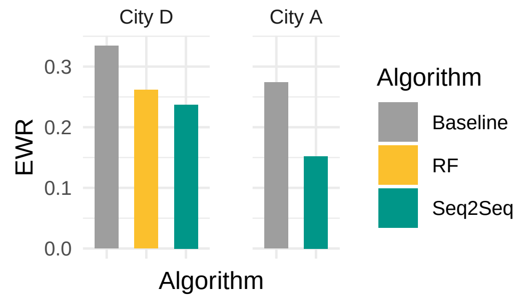


Fig. 9: Overall performance of compared algorithms.

rounds (R1, R4, R7), Baseline does not perform well, which might be because that the round begins too early and many drivers start working after the round begins. In other words, whether a registered driver is online before a round begins does not necessarily indicate whether he/she will actually show up in that round later. Compared with Baseline, both RF and Seq2Seq can further effectively reduce the easy-win ratios in all rounds, which proves the effectiveness of our proposed prediction-based matchmaking framework and the extracted features. Furthermore, Seq2Seq consistently outperforms RF in all nine rounds since Seq2Seq can make more accurate predictions about the online time of the registered drivers. In these rounds, Seq2Seq can reduce the easy-win ratio by up to 5.03 percentage points (R4) compared with RF and 22.08 percentage points (R7) compared with Baseline, which demonstrates the superiority of our method.

Table V shows the performance of Baseline and Seq2Seq in the competition held in City A. It can be observed that the easy-win ratios of Baseline are higher in the night rounds (R3, R6, R9) compared with other rounds. A possible reason is that many drivers do not work till that late but Baseline mistakenly thinks they will show up in those rounds according to their online status in an earlier time. In contrast, our method does not suffer from the same problem, consistently reducing the easy-win ratios in all rounds compared with Baseline since our method does not solely take the online status of drivers before each round begins as an indicator of whether they will actually show up in the future round but instead predicts how long they will be online in the future and match drivers with similar online time together. The improvements of our method over Baseline are notable for all rounds. Particularly, our method reduces the easy-win ratio by up to 16.58 percentage points (R3), more than half of the easy-win ratio of Baseline in that round. The results demonstrate the effectiveness of our prediction-based matchmaking algorithm.

2) Effect of Prediction Accuracy: Since our proposed method relies on the predictions of drivers' online time, we further investigate the effect of prediction accuracy on the performance of our method. Fig. 10 shows the relationship between the prediction accuracy and the easy-win ratio of our method. In this figure, the x-axis represents the ratio of the mean absolute error (MAE) of the predicted drivers' total

TABLE IV: Performance evaluation on a competition held in City D from Aug 8, 2018 to Aug 10, 2018.

Algorithm	R1 (07:00-10:00)			R2 (17:00-20:00)			R3 (21:00-24:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	201	252	0.4437	150	340	0.3061	120	382	0.2390
RF	135	320	0.2967	144	347	0.2933	110	391	0.2196
Seq2Seq	123	331	**0.2709**	142	349	**0.2892**	94	407	**0.1876**
Algorithm	R4 (07:00-10:00)			R5 (17:00-20:00)			R6 (21:00-24:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	218	295	0.4250	177	363	0.3278	137	416	0.2477
RF	154	362	0.2984	148	392	0.2741	125	428	0.2260
Seq2Seq	128	388	**0.2481**	142	395	**0.2644**	108	445	**0.1953**
Algorithm	R7 (07:00-10:00)			R8 (17:00-20:00)			R9 (21:00-24:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	274	292	0.4841	193	417	0.3164	149	465	0.2427
RF	156	409	0.2761	168	441	0.2759	127	487	0.2068
Seq2Seq	149	417	**0.2633**	150	460	**0.2459**	113	502	**0.1837**

TABLE V: Performance evaluation on a competition held in City A from Nov 30, 2018 to Dec 2, 2018.

Algorithm	R1 (11:00-16:00)			R2 (16:00-19:00)			R3 (20:00-23:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	242	632	0.2769	180	723	0.1993	277	591	0.3191
Seq2Seq	346	1384	**0.2000**	193	1606	**0.1073**	263	1453	**0.1533**
Algorithm	R4 (11:00-16:00)			R5 (16:00-19:00)			R6 (20:00-23:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	260	641	0.2886	208	711	0.2263	279	603	0.3163
Seq2Seq	299	1484	**0.1677**	210	1623	**0.1146**	289	1462	**0.1650**
Algorithm	R7 (11:00-16:00)			R8 (16:00-19:00)			R9 (20:00-23:00)		
	EW	Others	EWR	EW	Others	EWR	EW	Others	EWR
Baseline	263	651	0.2877	205	723	0.2209	302	590	0.3386
Seq2Seq	301	1512	**0.1660**	189	1667	**0.1018**	351	1423	**0.1979**

online time in a round to the duration of the round, which can reflect the prediction accuracy. In both competitions, we can observe a strong correlation (the correlation coefficients are both greater than 0.75) between the prediction performance and the corresponding easy-win ratio. Based on this observation, we can conjecture that the more accurate the prediction model is, the lower easy-win ratio our proposed prediction-based matchmaking scheme may achieve. This not only testifies the rationality of our method, but also shows its potential to further reduce the easy-win ratio by improving the prediction model's accuracy.

3) Summary of Experiment Results:

- Our proposed prediction-based framework and the Seq2Seq model can almost consistently outperform Baseline and RF with a notable reduction in the easy-win ratio.
- Our proposed method is effective in different cities during different periods of time (from 07:00 to 24:00) on different days (workdays and weekends).
- There are signs that the more accurate the prediction model is, the lower easy-win ratio our proposed prediction-based matchmaking scheme can achieve.

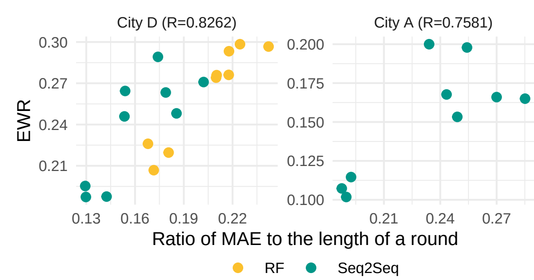


Fig. 10: Relationship between the prediction accuracy and EWR of our method. The correlation coefficients are annotated above each plot.

VI. Conclusion

In this paper, we conduct the first-of-its-kind case study to engage real-world drivers via competition. We design `Arena`, a competition where drivers compete for prizes by completing more trips. An initial deployment of `Arena` in 3 cities covering over 2,600 participants reveals the problem of easy-win, a critical bottleneck for the effectiveness of adopting competition-based incentive mechanisms to motivate drivers. We show that easy-wins are primarily caused by no-shows in a competition and demonstrate that easy-wins may weaken the motivating effect of competitions and decrease the profit of the platform. To solve the problem, we design a novel prediction-based matchmaking scheme based on the observation that no-shows are highly correlated to the online time of drivers. Our proposed method reduces the ratio of easy-wins by avoiding matching potential no-show drivers with drivers that will show up based on the predictions of their online time in the future. Experiments conducted on real competition data involving about 10,000 drivers demonstrate that our proposed prediction-based matchmaking scheme can reduce the easy-win ratio by up to over 20 percentage points compared with the baseline method. We envision this study will inspire practical guidelines for designing and deploying competition-based incentive mechanisms in real-world spatial crowdsourcing applications.

Acknowledgment

We are grateful to anonymous reviewers for their constructive comments. Yongxin Tong is the corresponding author of this paper. This work is partially supported by the National Key R&D Program of China under Grant 2018AAA0101100, National Science Foundation of China (NSFC) under Grant No. 61822201, 62076017 and U1811463.

References

[1] Uber. (2019) Company information. [Online]. Available: https://www.uber.com/en-PK/newsroom/company-info/

[2] J. S. Cobb. (2019) China's didi recruits colombian drivers ahead of bogota launch. [Online]. Available: https://www.reuters.com/article/us-didi-chuxing-colombia/chinas-didi-recruits-colombian-drivers-ahead-of-bogota-launch-idUSKCN1RY1NM

[3] J. V. Hall and A. B. Krueger, "An analysis of the labor market for uber's driver-partners in the united states," *ILR Review*, vol. 71, no. 3, pp. 705–732, 2018.

[4] D. Research, "The research report of the employment on didi in 2017 (in chinese)," Tech. Rep., 2017.

[5] S. Guo, C. Chen, J. Wang, Y. Liu, K. Xu, D. Zhang, and D. M. Chiu, "A simple but quantifiable approach to dynamic price prediction in ride-on-demand services leveraging multi-source urban data," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 112, 2018.

[6] Y. Tong, L. Wang, Z. Zhou, L. Chen, B. Du, and J. Ye, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *Proceedings of International Conference on Management of Data*. ACM, 2018, pp. 773–788.

[7] Uber. (2018) Driver rewards with uber pro. [Online]. Available: https://www.uber.com/us/en/drive/uber-pro/

[8] O. Feyisetan, E. Simperl, M. Van Kleek, and N. Shadbolt, "Improving paid microtasks through gamification and adaptive furtherance incentives," in *Proceedings of International Conference on World Wide Web*. ACM, 2015, pp. 333–343.

[9] M. Rokicki, S. Chelaru, S. Zerr, and S. Siersdorfer, "Competitive game designs for improving the cost effectiveness of crowdsourcing," in *Proceedings of International Conference on Information and Knowledge Management*. ACM, 2014, pp. 1469–1478.

[10] M. Rokicki, S. Zerr, and S. Siersdorfer, "Groupsourcing: Team competition designs for crowdsourcing," in *Proceedings of International Conference on World Wide Web*. ACM, 2015, pp. 906–915.

[11] T. Ye, W. Ai, L. Zhang, N. Luo, L. Zhang, J. Ye, and Q. Mei, "Predicting individual treatment effects of large-scale team competitions in a ride-sharing economy," in *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2020, pp. 2368–2377.

[12] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of Advances in Neural Information Processing Systems*. MIT Press, 2014, pp. 3104–3112.

[13] J. Hamari, J. Koivisto, H. Sarsa *et al.*, "Does gamification work?-a literature review of empirical studies on gamification," in *Proceedings of Hawaii International Conference on System Sciences*, 2014, pp. 3025–3034.

[14] L. Zhang, T. Song, Y. Tong, Z. Zhou, D. Li, W. Ai, L. Zhang, G. Wu, Y. Liu, and J. Ye, "Recommendation-based team formation for on-demand taxi-calling platforms," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, 2019, pp. 59–68.

[15] K. Huotari and J. Hamari, "Defining gamification: a service marketing perspective," in *Proceeding of International Academic MindTrek Conference*. ACM, 2012, pp. 17–22.

[16] B. Morschheuser, J. Hamari, J. Koivisto, and A. Maedche, "Gamified crowdsourcing: Conceptualization, literature review, and future agenda," *International Journal of Human-Computer Studies*, vol. 106, pp. 26–43, 2017.

[17] M. Vojnovic, *Contest Theory: Incentive Mechanisms and Ranking Methods*. Cambridge University Press, 2016.

[18] M. R. Baye, D. Kovenock, and C. G. De Vries, "The all-pay auction with complete information," *Economic Theory*, vol. 8, no. 2, pp. 291–305, 1996.

[19] J. D. Perez-Castrillo and T. Verdier, "A general analysis of rent-seeking games," *Public Choice*, vol. 73, no. 3, pp. 335–350, 1992.

[20] A. Glazer and R. Hassin, "Optimal contests," *Economic Inquiry*, vol. 26, no. 1, pp. 133–143, 1988.

[21] B. Moldovanu and A. Sela, "The optimal allocation of prizes in contests," *American Economic Review*, vol. 91, no. 3, pp. 542–558, 2001.

[22] N. Archak and A. Sundararajan, "Optimal design of crowdsourcing contests," in *Proceedings of International Conference on Information Systems*. AIS, 2009, p. 200.

[23] R. Cavallo and S. Jain, "Efficient crowdsourcing contests," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2012, pp. 677–686.

[24] T. Luo, S. S. Kanhere, H. Tan, F. Wu, and H. Wu, "Crowdsourcing with tullock contests: A new perspective," in *Proceedings of International Conference on Computer Communications*. IEEE, 2015, pp. 2515–2523.

[25] J.-H. Ahn, S.-P. Han, and Y.-S. Lee, "Customer churn analysis: Churn determinants and mediation effects of partial defection in the korean mobile telecommunications service industry," *Telecommunications Policy*, vol. 30, no. 10-11, pp. 552–568, 2006.

[26] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. A. Nanavati, and A. Joshi, "Social ties and their relevance to churn in mobile telecom networks," in *Proceedings of International conference on Extending Database Technology: Advances in Database Technology*, vol. 261. ACM, 2008, pp. 668–677.

[27] Y. Zhu, E. Zhong, S. J. Pan, X. Wang, M. Zhou, and Q. Yang, "Predicting user activity level in social networks," in *Proceedings of International Conference on Information and Knowledge Management*. ACM, 2013, pp. 159–168.

[28] J. Xiao, C. Chen, C. Hou, and X. Yuan, "Short-term user activity prediction with massive mobile broadband data," in *Proceedings of International Conference on Mobile Data Management*. IEEE, 2017, pp. 348–351.

[29] J. P. Thomas and Z. Wang, "Optimal punishment in contests with endogenous entry," *Journal of Economic Behavior & Organization*, vol. 91, pp. 34–50, 2013.

[30] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv, "The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms," in *Proceedings of SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 1653–1662.

[31] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.