COMPGW02 Niren Patel Individual Report

Niren Patel
zctpatb@ucl.ac.uk

1. INTRODUCTION

Real-Time Bidding (RTB) has become important in the display advertising landscape. Improvements in computer capabilities and internet speeds now allow advertisers to make bids for individual impressions in real-time before a page is loaded; placing advertisements almost seamlessly onto a page. This paper explores and analyses the dataset provided by iPinYou and by implementing numerous non-linear bidding strategies to increase Key Performance Indicators (KPIs). Overall it was found that using a gradient boosting algorithm combined with an effective bidding strategy like that of ORTB 1 (Optimised Real Time Bidding 1), can increase the number of clicks vastly, in comparison to simple linear bidding strategies.

Keywords

RTB; ORTB; CTR; Predicted CTR

2. RELATED WORK

This paper looks at strategies proposed by Zhang et al. to increase the number of clicks achieved on the iPinYou dataset.[4] Downsampling methods from “Practical Lessons from Predicting Clicks on Ads at Facebook”, are used to reduce training time and reduce compute expenditure.[2] To get a thorough understanding of the dataset, it is recommended to read “Real-Time Benchmark Bidding with iPinYou Dataset.[3] For more models of approaching similar problems that rely on other functions other than predicted Click-Through Rate (pCTR), Han Cai et al. provides a thorough and clear example of how Reinforcement learning can be used in the problem.[1]

3. DATA EXPLORATION

This section analyses the data available and provides a reminder of the advertisers involved in the data provided by iPinYou and information about their respective bidding strategies.[3] The section continues further to explain the importance of user feedback and how it can influence important performance metrics such as Click-Through Rate (CTR) and Cost-per-Click (CPC) for an advertiser's bidding strategy. Further additional findings between KPIs are also discussed.

3.1 Basic Statistics

Table 1 gives a breakdown of the advertisers involved in Table 2 and Table 3. Tables 2 and 3 both show that CTR values generally don't exceed more than 0.1%. However, Advertiser 2997, which is in the industrial category of “Mobile e-commerce app install”, exceeds this and achieves a unusually high score of 0.435% on the training set and 0.421% on the validation set. There are likely to be a number of possible explanations related to why a mobile environment has a greater proportion of clicks, but here are the most likely. Firstly, advertisements for application installments are usually very difficult to close because the ‘close ad button’ or ‘x’ is usually too small, leading to a ‘fat-finger’ effect. Secondly, application installments are usually free, whereas most of the other advertisers are likely to focus on tangible items that can put off consumers. Another reason is likely to be availability; if an application is interesting, there is only a few taps of a finger separating the user from being at a stage of viewing the ad, to being at a stage where they are finally using the application. Conversely, all the other advertisers products are likely to require a lot more effort to purchase and may put off users. Cost-per Mille (CPM) for Advertiser 2997 is also lower than the rest of the advertisers, this is likely to mean that advertising on mobile devices is likely to be cheaper. The combination of having the highest CTR and lowest CPM (both by considerably less) leads to a CPC value that is considerably lower than the rest of the advertisers. It can be deduced that the mobile app install advertiser bids in a very targeted and inexpensive manner.

Table 1: Table showing Advertiser Fields

Advertiser ID	Industrial Category
1458	Chinese vertical e-commerce
2259	Milk Powder
2261	Telecom
2821	Footwear
2997	Mobile e-commerce app install
3358	Software
3386	International e-commerce
3427	Oil
3476	Tire

Another advertiser with above-average CTR scores, is Advertiser 1458 (Chinese vertical e-commerce) with a CTR of 0.078% on the training set and 0.079% on the validations set. this is likely to be due targeted advertising; since vertical e-commerce usually specialises in one, or very few products. Due to the vertical nature of the products, it is likely that advertisements are targeted at users that visit specific URLs that have similar content. Additionally, bids are probably more likely to be made for regions that have strong trading

Table 2: Dataset Statistics for Training Data

Adv	Imps	Clicks	Cost	CTR	CPM	CPC
1458	492,353	385	33,969	0.078%	68.99	88.23
2259	133,673	43	12,428	0.032%	92.97	289.03
2261	110,122	36	9,874	0.033%	89.66	274.27
2821	211,366	131	18,828	0.062%	89.08	143.73
2997	49,829	217	3,129	0.435%	62.80	14.42
3358	264,956	202	22,447	0.076%	84.72	111.12
3386	455,041	320	34,932	0.070%	76.77	109.16
3427	402,806	272	30,459	0.068%	75.62	111.98
3476	310,835	187	23,919	0.060%	76.95	127.91
Tot	2,430,981	1793	189,985	0.074%	78.15	105.96

Table 3: Dataset Statistics for Validation Data

Adv	Imp	Clicks	Cost	CTR	CPM	CPC
1458	62,353	49	4,295	0.079%	68.88	87.64
2259	16,715	2	1,569	0.012%	93.86	784.40
2261	13,550	3	1,215	0.022%	89.66	404.96
2821	26,503	23	2,395	0.087%	90.36	104.13
2997	6,176	26	389	0.421%	62.95	14.95
3358	32,939	23	2,794	0.070%	84.82	121.48
3386	56,665	28	4,351	0.049%	76.78	155.39
3427	50,183	37	3,777	0.074%	75.26	102.07
3476	38,841	11	2,994	0.028%	77.08	272.16
Tot	303,925	202	23,777	0.066%	78.23	117.71

ties with China; since it is a Chinese vertical e-commerce advertiser.

The worst performing advertiser in terms of all metrics is Advertiser 2259 (Milk Powder). This advertiser has a CTR of 0.032%, a CPM of 92.97 and an CPC of 289.03 on the training set. Even worse, the CPC on the validation data is 784.40 (the average is 117.71). The reasons for the poor performance is most likely the result of it being more of a "brick and mortar store" item. Users are less likely to click on an ad for an inexpensive item that is available in stores, especially when it is likely they already have the product at home or a variant of it.

3.2 Further Analysis on User Feedback and Bidding

For this section, further analysis and comparison is placed between advertiser 2259 (Milk Powder) and 2997 (Mobile e-commerce app install). These advertisers were chosen to dissect and get a better understanding of what features should be given importance when pre-processing data,these advertisers represent the best and worst performing advertisers from section 3.1. User-agent data was not split into browser and OS since there weren't many individual entries that corresponded to clicks.

Figure 1 shows what effects different features can have on CTR and the strategy of the advertiser (some of the graph y-axis' were logarithmic to account for magnitude of difference between the two advertisers). Overall, Advertiser 2997 performs better than 2259 across all categories, however this is more likely to be the cause of advertiser 2997 operating only on mobile user agents. From Figure 1 the following was observed:

(I) Advertiser 2259 only won impressions on Weekdays: 0, 1, 5 and 6, whereas Advertiser 2997 only won impressions Weekdays: 2,3,4,5. Both have consistent CTR rates on the days they won impressions for.

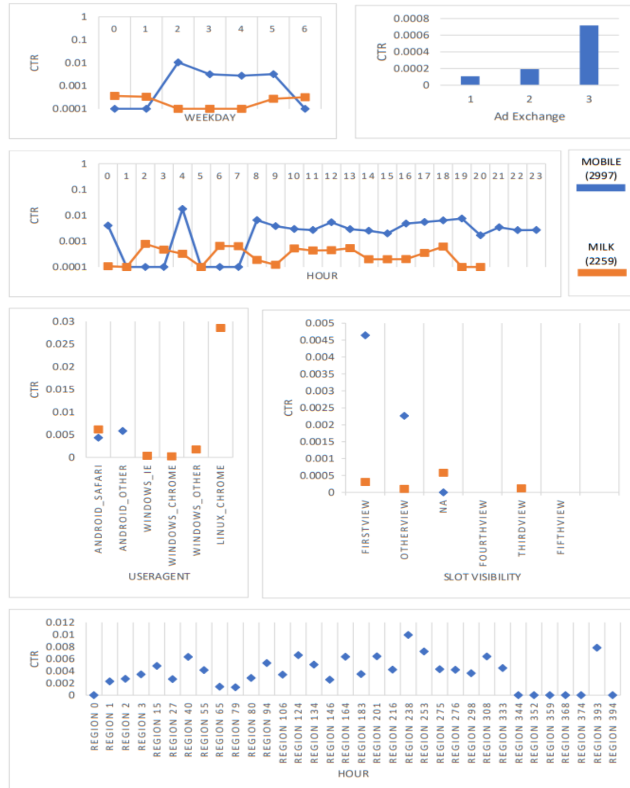

Figure 1: CTR performance across different categories between Advertiser 2259 and 2997.

(II) Both advertisers won impressions for nearly every hour in the day, with a larger number during the daytime. However, Advertiser 2997 consistently had a higher click-through rate in comparison to 2259; peaking at 1.8%, whereas Advertiser 2259 peaked at 0.078%.

(III)Advertiser 2259 only won impressions in Region 216, achieving a CTR of 0.032%, indicating a more targeted bidding strategy in comparison to 2997. From the Region graph; showing only Advertiser 2997, it's clear that CTR is heavily affected by the region.Ranging from 0 to about 0.01 (1%). A 1% CTR rate is significant and it is clear this will heavily influence pCTR values.

(IV)The user agent was not separated by OS and browser since only a few labels had click values. CTR is generally higher in a mobile environment, and this might be the reason why Advertiser 2997 performs better than 2259 for CTR across most other features and labels.

(V) No data was available for Ad exchange for 2997. Advertiser 2259 has clear discrepancies between Ad exchanges, indicating the importance of Ad Exchange in CTR prediction.

(VI) FirstView slotvisibility performs better than other slotvisibility labels for both Advertisers.

Figure 2 shows CPC across different feature labels. The discrepancy in CPC is most likely to be the result of the different User environments (Mobile Vs. Desktop). In general, Advertiser 2259 has higher CPC values than 2997 across all features and labels. From Figure 2 the following was observed:

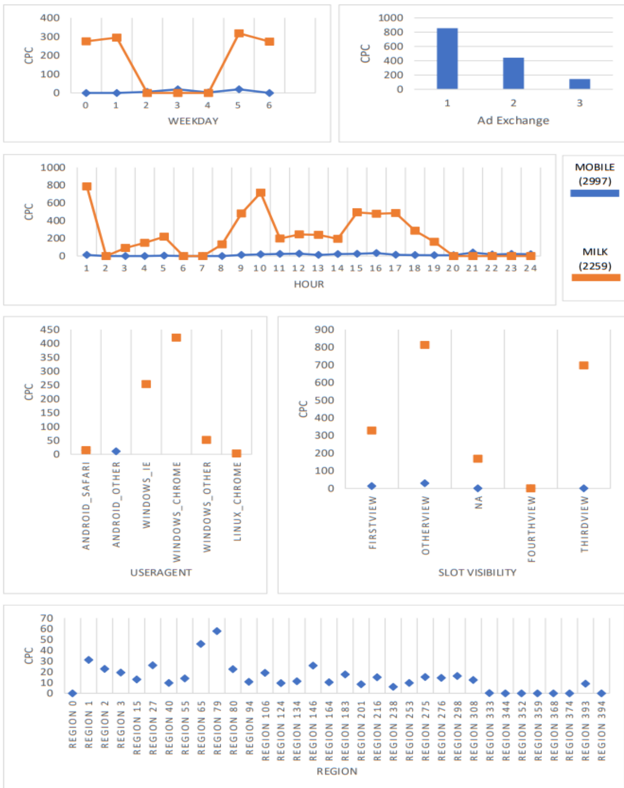

Figure 2: CPC performance across different categories between Advertiser 2259 and 2997.

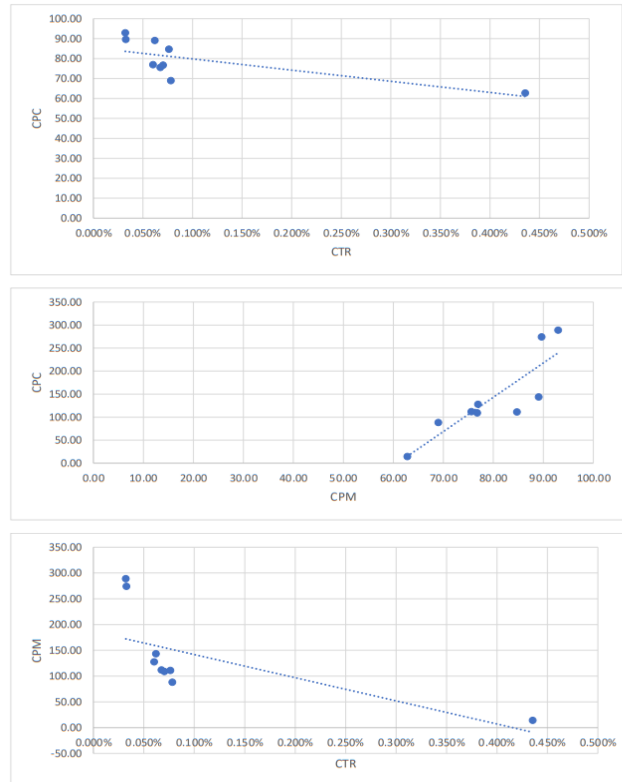

Figure 3: Graphs showing how CPC, CTR and CPM vary against each other

(I) Advertiser 2259 has CPC values ranging from 270 to 320, in comparison to 2997 where values range from 2-20 (difficult to see in graph because of y-axis scaling issues). Conforming to CPC data found in Table 2.

(II) As expected, Hours where CPC is lower than normal corresponds to hours where CTR is higher than normal.

(III) Region can have a major affect on CPC, with some regions having a CPC six times greater than other regions for Advertiser 2997. Advertiser 2259 CPC values are only available for Region 216; indicating a product that is only sold regionally.

(IV)CPC is lower for Mobile environments as opposed to desktop environments. Suggesting it as one of the more important features of a dataset.

(V) CPC value for Ad Exchange 3 are almost a mirror image of the corresponding graph found in Figure 1.

(VI) Slotvisibility CPC ranges from 160 to 820 for Advertiser 2259 and between 10 and 30 for Advertiser2997 showing a huge discrepancy in CPC values across slotvisibility.

3.3 Additional non-trivial findings

For this section, CPC, CPM and CTR were plotted against each other to gain some insight into how they affect each other. Since CTR and CPM are both functions of CPC, it was expected that that graphs corresponding to CPC being a variable would have strong correlation, and this was the case. However, Advertiser 2997 was anomalous for CTR vs. CPC. For CPC vs. CPM, since the CPM values were generally in the same range, there was no data point that deviated away from the line of best fit.

However, when CPM was compared against CTR, it was not expected there would be a strong correlation. If there was a correlation, I assumed that a higher CTR value would have a higher CPM value, since they would most likely be in demand and higher bids would be placed. However Figure 3, which compares these two variables, shows otherwise. Once again Advertiser 2997 provided an anomolous result.

4. INDIVIDUAL BEST BIDDING STRATEGY

This section explores non-linear bidding strategies to be compared against numerous evaluation metrics. The results from these findings will then be used to create an improved bidding strategy for the group portion of this project.

4.1 Data Preprocessing

Due to there being 2,429,188 individual impressions in the training dataset, the data was downsampled to make data more manageable for the computing power available and reduce the problem of imbalanced classes. Only 0.073% of the original data had a click for an impression. By downsampling to a rate of 2.5% obtained from Xinran et al., without replacement, the sample was reduced to 62,523, with 2.87% of impressions containing clicks. [2] These values were recalibrated after the pCTR on validation set samples were created. This is explained at the end of Section 4.2 and Equation 1.

In addition to this, values from the 'useragent' column were separated and two new columns were formed, representing the Operating System (OS) and the browser, so the

Table 4: Confusion matrix of various CTR prediction models

Model	Confusion Matrix	F1 Score
Single Logistic Regression (Regularizer=0.001)	254,230 / 49,493 ; 63 / 139	0.91
XGBoost (Estimators=100, Max Depth=3, Learning Rate=0.1)	303,437 / 286 ; 130 / 72	≈ 1.00
Random Forest (Estimators=100, MaxDepth= None)	302,402 / 1321 ; 128 / 74	≈ 1.00

number of unique values could be reduced in the One Hot Encoding process. This was applied across both datasets. The following columns were also removed from both the training and validation datasets: 'bidid', 'userid', 'IP', 'url', 'urlid', 'slotid', 'domain', 'city', 'creative','payprice' and 'bidprice' because they contained far too many unique variables. The following categorical variables: 'weekday', 'hour', 'OS', 'browser', 'adexchange', 'slotwidth', 'slotheight', 'slotvisibility', 'slotformat', 'advertiser', 'region' and 'keypage' were 'One Hot Encoded' which led to 232 unique columns for both sets. These columns were then separated from the predictor variable and were read for the training process.

4.2 CTR Prediction Models

Three models were implemented to calculate each impression's pCTR: Logistic Regression, XGBoost and Random Forest. A python function was then created to tune each models key hyper-parameters using a manual grid-search method. Initially, large intervals were left between each hyper-parameter value being iterated over, but was then narrowed down to achieve the best results. Performance of the different models are shown in Table 4.

These three models were chosen because they provide reliable results and have plenty of support. Logistic Regression, possibly the most simple and well known of the three, was chosen as a benchmark and was not expected to exceed the other two models in terms of performance. The other two models generally provide ample performance without being too complex.

From the table, the confusion matrix that accompanies each model is also shown as well as its F1 score. Taking the Random Forest as an example; The value 302,402 represents the model correctly predicting 0 for a click, 1321 represents the model incorrectly predicting 1 for a click, 128 represents the model incorrectly predicting 0 and 74 represents the model correctly predicting 1. A confusion matrix is used because it allows for clear comparison between different models. The F1 scores, assign a score from 0-1 based on values from the confusion matrix to provide further analysis of the performance; with 1 being the best and 0 being the worst. XGBoost and Random Forest performed so well, that the scores were approximately equal to 1.

From the table it can be seen that XGBoost has the best performance even though the number of clicks is less than the rest of the models; the other two models incorrectly predict clicks being 1 as a greater proportion. This is due to the boosting method prevalent in XGBoost, which prevents over-fitting by training weak learners. Random Forest also performs well in predicting clicks. Random Forest uses a technique called 'bagging' to increase model accuracy; where it combines the predictions from multiple machine learning algorithms to come at a result. The computational run-time is a lot longer in comparison to the other models and would increase exponentially with impressions and features.

Logistic Regression predicts the most clicks, but also predicts the most incorrect clicks. Logistic Regression is a very simple model and very fast to implement, however, overfitting is generally the problem with the model. Often, features for a Logistic Regression model should be limited to between 10-30 for a model this size. However, this also increases the likelihood of important information being lost. In this analysis, feature selection was not performed.

All of the models were exhaustively tested with the nonlinear bidding strategies to get an effective comparison of the pCTR prediction models. It is expected that a bidding strategy involving XGBoost is most likely to achieve the most number of clicks.

Before using these pCTR values onto bidding strategies, the values were re-calibrated using the following equation for each impression[2]:

$$\frac{p}{p + (1-p)/w} \tag{1}$$

where p is the model pCTRs calculated from the downsampling sample and w is the negative downsampling rate. By doing this, the mean went from being 1.9% for non-calibrated to 0.066% for calibrated; this corresponds to a mean similar to the original training set mean of 0.074%.

4.3 Bidding Strategies

Non-linear bidding strategies are discussed in this section to achieve the best results in comparison to the evaluation metrics. A total of nine variations were exhausted and hyper-parameters were once again tuned on the validation set to achieve the highest performing models. The results are explained in the Evaluation Metrics section.

4.3.1 ORTB 1

One of the most popular non-linear bidding strategies used includes an ORTB Strategy that is a function of the pCTR. Equation 2 shows how a bid is calculated using ORTB 1, where θ is the pCTR; with the hyper-parameters λ and c to be tuned[4]:

$$bid = \sqrt{\frac{c}{\lambda}\theta + c^2} - c \tag{2}$$

4.3.2 ORTB 2

Another variant of the ORTB used is shown below in equation 3[4]:

$$bid = c \cdot \left[\left(\frac{\theta + \sqrt{c^2\lambda^2 + \theta^2}}{c\lambda} \right)^{\frac{1}{3}} + \left(\frac{c\lambda}{\theta + \sqrt{c^2\lambda^2 + \theta^2}} \right)^{\frac{1}{3}} \right] \tag{3}$$

Once again the same hyper-parameters, λ and c were tuned and θ is the pCTR for that impression.

4.3.3 Novel Solution:

For a novel solution, the linear bidding strategy was taken and the term which divides pCTR by the Average CTR is

Table 5: Performance of the three pCTR models against different bidding strategies.

	ORTB 1	ORTB 2	Novel non-linear
Logistic Regression	126	112	95
XGBoost	161	149	105
Random Forest	153	133	101

squared. The hyper-parameter to be tuned is the base bid. Equation 4 shows the Bidding Strategy for this method:

$$bid = BaseBid * \left(\frac{\theta}{AvgCTR}\right)^2 \tag{4}$$

The variation was made on the linear strategy because it was assumed this strategy would 'penalise' low pCTR values and 'reward' higher pCTR values.

4.4 Evaluation Metrics

All three bidding strategies were tested against all three pCTR models for completeness and resulted in nine different results. Results showing the KPI (Number of Clicks) for each variation are shown in Table 5. For all three bidding strategies, XGBoost outperformed both the other pCTR models. As expected, Logistic Regression was the poorest performing of the three, but for the Novel non-linear solution, it performed just as well as the other two models. This suggests the bidding strategy can perform well even without an almost perfect pCTR model. All pCTR models achieved their best scores using the bidding strategy ORTB 1, with ORTB 2 being the second best performing. This data may seem conclusive in terms of which bidding strategy and pCTR model is the most effective. However, it is quite likely that for the final bidding strategy for the group, any one of these nine variants could perform favourably, given different pCTR models, tuning parameters and bidding strategies by group members.

The ORTB strategies were easy to fine tune, there was much deliberation with the Novel solution, the tuning had to be very precise, therefore it is likely that further analysis will be needed to find the optimal parameters across all bidding strategies and pCTR models in the group portion of the project. It may also be the case that the Novel solution may be adjusted to take on a cubic form or higher power to further exploit higher/lower pCTR values.

For further analysis, the bidding strategies with the best pCTR models were compared against each other. Since XGBoost had the highest performance for the KPI (number of clicks) for all the strategies, it was further analysed against the other key evaluation metrics. The results are shown in Table 6. As shown in the results, the combination of ORTB 1 and XGBoost has the best performance for both the clicks and CTR, the two most important evaluation metrics of the project scope. The CTR for the novel solution was also promising, suggesting it may be possible to refine this strategy in the final solution. The same applies for the ORTB 2 strategy.

Table 6: Further analysis of bidding strategies against the Evaluation Criteria.

	ORTB 1 + XGBoost	ORTB 2 + XGBoost	Novel sol + XGBoost
Clicks	161	149	105
Won Impressions	151,693	159,127	111,098
CTR	0.11%	0.09%	0.095%
Budget Spent	6249.978	6249.964	6249.989
Avg CPM	41.20	39.28	56.26
Avg CPC	38.82	41.95	59.52

5. CONCLUSIONS

From the data and analysis completed, it is clear that user feedback plays is important in determining whether an impression will be clicked on or not. It seems mobile environments are more favourable for high CTR rates. It is also apparent that high performance pCTR models will be needed to ensure bidding strategies are effective and perform favourably on the test set.

If more time was available for this assignment, I would have done more comprehensive Feature Selection for Logistic Regression. Furthermore, despite the prevalence of class imbalance in the training set, it would have been interesting to see how a model performed without downsampling the data or doing a larger rate of downsampling. However, due to the lack of computing power available, as well as lack of knowledge for using super computers on the cloud, I was unable to complete these tasks.

More pCTR models could have also been implemented to get a more exhaustive and comprehensive list, this was not done because of the lack of familiarity with them. Pay-price/Bid-price prediction could have also been used, and a Novel solution could have been created from that. However, the level of skill required for this would be out of my skill range and would require further time to learn.

I also wanted to try an Mcpc strategy from "Optimal Real-Time Bidding". [4] However, this was not suitable since it was not technically a non-linear strategy, there was also not enough eCPC data to carry this out and the goal of the function is different to what the KPI was.

For the assignment an independent approach was taken. All members attempted all sections of the group assignments including the basic strategies. All members also tested their individual strategies on the test set, to see who had the highest performing models. Both the other group members developed models that had better KPI results than mine and those were developed upon further, I also worked on some of the models developed by the other two to see if I could improve performance. For the actual report writing, Parham wrote the bulk of the content on a word processing software and I coded this onto LaTeXto ensure it met ACM proceeding standards. Sven and I also contributed small parts in different areas of the paper.

The Github to the repository of my individual work can be found from the following link:

url: https://github.com/Niren52/W.E.N.P

I used 'Google Colab' for this project, due to a damaged personal computer. So if you would like to run the files smoothly, I recommend using the files on Google Colab.

6. REFERENCES

[1] H. C. et al. Real-time bidding by reinforcement learning in display advertising. *WSDM*, 2017.

[2] X. H. et al. Practical lessons from predicting clicks on ads at facebook. *ADKDD*, 2014.
[3] J. W. X. S. Weinan Zhang, Shuai Yuan. Real-time bidding benchmarking with ipinyou dataset. *UCL Technical Report*, JULY 2014.
[4] S. Y. Weinan Zhang and J. Wang. Optimal real-time bidding for display advertising. *In Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1077–1086, 2014.