# REVENUE OPTIMIZATION IN REAL-TIME BIDDING BASED ADVERTISING FOR MOBILE DEVICES


**SHALINDA ADIKARI**

*Bachelor of Science (First class) Honors in Information Technology*


**A THESIS SUBMITTED**

**FOR THE DEGREE OF DOCTOR OF PHILOSOPHY**
**DEPARTMENT OF INFORMATION SYSTEMS AND ANALYTICS**
**NATIONAL UNIVERSITY OF SINGAPORE**

**2017**

Supervisor:
Associate Professor Hahn Jungpil

Examiners:
Associate Professor Huang Ke Wei
Dr Phan Tuan Quang
Professor Syam Menon, University of Texas at Dallas

# DECLARATION

I do hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

Further, this thesis has not been submitted previously for a degree at any university.

**SHALINDA ADIKARI**

16th August 2017

# ACKNOWLEDGEMENTS

I take this opportunity to extend my wholehearted thanks to all those who have contributed to make this Ph.D. thesis a success. My wholesome gratitude goes to my parents for their blessings throughout my life and academic advancement. So I dedicate this thesis to my parents.

First and foremost, I offer my utmost gratitude to Associate Prof. Jungpil Hahn for being my advisor in the final year of my Ph.D. career. When I was troubled without having a supervisor in the middle of my Ph.D. he gave me the confidence to finish my thesis as being my supervisor. As the head of the department even though he was on a tight schedule, his timely feedbacks and expertise were fundamental support in flourishing this thesis. His valuable advice has considerably influenced in my academic experience. It was a great privilege and honor to have him as my supervisor, and I am very much thankful to him for being helpful always.

I cannot find words to express my gratitude to my former advisor and mentor Associate Professor Kaushik Dutta who has supported me throughout my Ph.D. career. He has spent his valuable time patiently with me without any limitations and he has molded me with his knowledge, skills and abilities. Without his personal attention, this research would not have been accomplished. He has motivated me to be the best that I could be. I have been very fortunate to have him as my Ph.D. advisor who believed in me and realized my potential ability.

Next, I would like to convey my heartful gratitude to Associate Prof. Anindya Datta. Even though he acted as my supervisor during the period that I was struggling to find a supervisor, he gave me fullest support on continuing my research with a free mind without facing any administrative matters. Also, he has given me valuable insights on identifying research questions for the dissertation. I would also like to deeply thank the Engineering Team of Mobilewalla. They never hesitate to provide me with the required data whenever I requested for it. No words to thank them for their patience, knowledge, and attitude.

Besides my advisors, I would like to express my sincere gratitude to the members of my thesis committee, including A/Prof. Tuan Phan, A/Prof. Huang Ke Wei, and external examiners, for their critical readings and constructive criticisms, which have been extremely helpful in refining and enriching my dissertation. I am greatly benefited from their instructive guidance, insightful criticism and raising timely queries.

I would also like to convey my thanks to all faculty members of School of Computing, who have helped me in numerous ways. I am very much grateful to the School of Computing for providing the financial support for my study. I also gratefully acknowledge the support that I received from the members of administrative staff in the School of Computing.

Last but not least, my thanks to my sister Dr. Prasadi and her family. Their kindness and support warmed my heart and gave me the courage to look forward and progress. I would also like to thank my friends and lab mates who helped me in my Ph.D. study in some way or other.

# TABLE OF CONTENTS

**Chapter 1. Introduction ................................................................ 1**

1.1 Background and Motivation ............................................................ 1

1.2 Challenges ............................................................ 9

1.3 Research Gaps ............................................................ 10

1.3.1 An Effective Bidding Strategy for DSP ............................................................ 10
1.3.2 Revenue Optimization for Publishers ............................................................ 11
1.3.3 Audience Selection in Campaign Optimization ............................................................ 12

1.4 Outline of the Dissertation ............................................................ 12

**Chapter 2. A New Approach to Real-time Bidding in Online Advertisements: Auto Pricing Strategy ............................................................ 15**

2.1 Introduction ............................................................ 15

2.2 Related Works ............................................................ 19

2.3 Solution Formulation ............................................................ 23

2.4 Auto Pricing Strategy ............................................................ 25

2.4.1 Budget Allocation Strategy ............................................................ 27
2.4.2 Bid Price Adjustment Strategy ............................................................ 28
2.4.3 App Selection Strategy ............................................................ 31
2.4.4 Dynamic Programming Model ............................................................ 32
2.4.5 Optimization Strategy ............................................................ 33

2.5 APS Algorithm ............................................................ 36

2.6 The Benchmark Approaches ............................................................ 40

2.6.1 The Exact Approach ............................................................ 40
2.6.2 The Greedy Approach ............................................................ 41
2.6.3 Optimal Real-time Bidding *(ORTB)* ............................................................ 42
2.6.4 Bidding Below Max eCPC *(Mcpc)* ............................................................ 43
2.6.5 Linear-form Bidding of pCTR (Lin) ............................................................ 43

2.7 Analysis and Results ............................................................ 44

2.7.1 Dataset ............................................................ 44
2.7.2 Comparison with Benchmark Approaches ............................................................ 45
2.7.3 Sensitivity Analysis of APS ............................................................ 51

2.8 Discussion ............................................................ 55

**Chapter 3. Publisher Return Optimization in Mobile App Advertising……………………………………………………………..58**

3.1 Introduction ............................................................ 58

3.1.1 Background ............................................................ 61
3.1.2 Methodology ............................................................ 62

3.2 Related Works ............................................................ 64

3.3 Problem Specification ............................................................ 68

3.4 Solution Formulation ............................................................ 71

3.4.1 Click Return Rate ............................................................ 73
3.4.2 Ad Network Effect ............................................................ 73

3.4.3 Advertisement Effect ... 76
3.4.4 Ad Network Performance Utility ... 80
3.5 The Benchmark Approaches ... 84
3.5.1 The Single Ad Network Approach ... 85
3.5.2 The First-Come-First-Serve (FCFS) Approach ... 86
3.5.3 The Exact Approach ... 87
3.6 Experiment and Dataset ... 87
3.7 Analysis and Results ... 91
3.7.1 Comparison with Benchmark Approaches ... 91
3.7.2 Sensitivity Analysis of APS ... 94
3.8 Discussion ... 96
**Chapter 4. Campaign Optimization in RTB Advertising ... 99**
4.1 Introduction ... 99
4.2 Related Works ... 101
4.3 Problem Specification ... 108
4.3.1 Design Artefact in the Business Process of Advertisement Bidding in DSP ... 108
4.3.2 Campaign Optimization ... 109
4.4 Solution Formulation ... 111
4.4.1 Feature Grouping Model ... 116
4.4.2 Rare Event Supervised Binary Classification Model ... 124
4.5 The Benchmark Approaches ... 128
4.5.1 Rare Event Logistic Regression ... 128
4.5.2 Feature Grouping based Approach ... 129
4.5.3 Random Forest ... 129
4.5.4 Support Vector Machine ... 130
4.6 Analysis and Results ... 131
4.6.1 Dataset ... 131
4.6.2 Parameter Significance ... 132
4.6.3 Evaluation Metrics ... 133
4.6.4 Comparison with Benchmark Approaches ... 135
4.6.5 Sensitivity Analyses ... 141
4.7 Discussion ... 144
**Chapter 5. Conclusion ... 148**
**References ... 152**
**Appendix A ... 165**

# LIST OF FIGURES

Figure 1. High-level communication diagram for an RTB Ecosystem ............4
Figure 2. Three steps of the auto pricing strategy..........................................27
Figure 3. TSPC vs. Target spend in different approaches ..............................46
Figure 4. The number of bid requests is selected by APS, ORTB, Lin, and Mcpc, to bid for different bid prices on when the target spend is (a) 1000, (b) 2000 and (c) 3000. ........................................................47
Figure 5. The number of bid requests selected by APS, the exact approach and the greedy approach to bid for different bid prices when the target spend is (a) 1000, (b) 2000 and (c) 3000.......................................47
Figure 6. Target spend vs. the percentage of the number of bid requests selected to bid (a) & (c) and target spend vs. the percentage of the number of conversions which occurred (b) & (d). ..........................50
Figure 7. (a) TSPC vs. Bid periods and (b) Conversions achieved vs. Bid periods in X, Y, and Z campaigns .................................................52
Figure 8. (a) TSPC vs. α value and (b) Conversions achieved vs. α value in X, Y, and Z campaigns......................................................................54
Figure 9. (a) TSPC vs. β value and (b) Conversions achieved vs. β value in X, Y, and Z campaigns......................................................................55
Figure 10. The ad delivery solutions of online advertising ecosystem...........61
Figure 11. High-level component architecture of the proposed solution.........70
Figure 12. Sample decision tree of advertisement attributes ..........................77
Figure 13. Sample decision tree with advertisement types and click behaviors .......................................................................................................77
Figure 14. The simulation process and a sample transaction at the app instance level ........................................................................................88
Figure 15. Comparisons of fill rate, CRR, and ARPC with benchmark approaches across various app instances........................................92
Figure 16. (a) Fill rate, (b) CRR and (c) ARPC across app periods ................92
Figure 17. Ad frequency across app instances................................................93
Figure 18. Number of ads delivered (%) to each ad type by each approach ...94
Figure 19. App period duration vs. CRR and ARPC......................................95

Figure 20. Number of distinct feature values of different features in each day ........................................................................................105
Figure 21. The flow diagram of the business process of advertisement bidding in DSP ..........................................................................................108
Figure 22. Number of bid requests and CTR change over consecutive periods in Device ISP ...............................................................................110
Figure 23. Communication flow of the proposed approach ..........................113
Figure 24. Plate notation for LDA with Dirichlet-distributed feature group-feature value distribution ..............................................................118
Figure 25. Initialization of Request-Feature matrix as Document-term matrix ........................................................................................119
Figure 26. (a) Bidding rate, (b) CTR, and (c) Click sensitivity in each week of the campaign ...............................................................................136
Figure 27. Bidding rate, CTR, and Click sensitivity based on long-term and short-term bid periods ...................................................................139
Figure 28. Bidding rate, CTR, and Click sensitivity based on different prediction thresholds of the Logistic regression ..........................141
Figure 29. (a) CTR, (b) Bidding rate and (c) Click sensitivity for different batch sizes; X=10000 ..................................................................142
Figure 30. (a) CTR and (b) Bidding rate at different iteration counts for different batch sizes; X=10000 ..................................................142
Figure 31. (a) CTR and (b) Bidding rate with top-k feature groups for different batch sizes; X=10000 ..................................................................144

# LIST OF TABLES


Table 1. Details of the bid request, bid response, win note and conversion note in the RTB process....8
Table 2. Notation for the model descriptions....26
Table 3. Notation for the exact approach and the greedy approach....40
Table 4. Data difference among three campaigns....44
Table 5. Comparison of the obtained conversions between APS and other approaches....46
Table 6. Model behavior at different bid periods....51
Table 7. Model behavior based on the number of bid requests of the applications which are not active during the previous period....53
Table 8. An example of ad network behaviors at the country level....70
Table 9. Notation for the model descriptions....71
Table 10. The ranges of the simulated data....89
Table 11. Descriptive statistics based on the ad networks....90
Table 12. ARPC with fill rate, ad frequency, network latency and CRR....95
Table 13. Notation for the model descriptions....115
Table 14. Examples of features, feature values, and feature groups....116
Table 15. Description of the features....131
Table 16. Descriptive statistics of the feature group based dataset....132
Table 17. MMR test statistics on Wilks' Lambda test....133
Table 18. Average number of new distinct feature values introduced in each day of the week....137
Table 19. Average and variance values of bidding rate, CTR, and Click sensitivity during the whole campaign....138
Table 20. CTR and eCTR performances for the whole campaign....140

# Chapter 1. Introduction

## 1.1 Background and Motivation

In the world of advertising, online/digital advertising is the source of "Schumpeter's gale", a process that is widespread across the advertising landscape. According to the Business Directory[1], online advertising is the use of the Internet as an advertising medium where promotional messages appear on a user's screen. Online advertising revenue in the United States totalled $49.5 billion during the year 2014 which is an increase of 16% from that of 2013 (LLP 2015). Similarly, according to the "Statista in 2016", global online advertising revenues are projected to increase even further to over USD$650 billion (Statista 2015a). Online advertising methods are, arguably, leading to significant reductions in transaction costs between advertisers and consumers (Bagwell 2007). Online advertising methods enable advertisers to deliver targeted information to consumers who are most likely to act on it. In advertising, advertisers play a buyer's role while publishers, who have space to advertise, play at the seller's role. In traditional approaches, the buyer and seller are linked sans focused target marketing strategies. Examples are broadcasting advertisements via television and radio, or publishing advertisements in newspapers or magazines. In contrast, online advertising

[1] http://www.businessdictionary.com/definition/online-advertising.html

approaches create this link based on their contextual or demographic attributes, such as in search-based advertising platforms (Evans 2008), where a search string is mapped with the most relevant advertisements. Hence, in online advertising, in order to match an advertising message to a consumer, various targeting techniques (demographic, behavioural, device, keyword, etc) are used. Based on the aforementioned targeting techniques, the most befitting ad space can be determined for the target audience of a particular advertisement which leads to the attainment of higher revenues for the advertiser. A key feature of online advertising is that publishers are remunerated for the aired content, and for their services which encompass receiving advertising messages (advertisers pay to send these messages). The delivery of online advertising exhibits rapid technological change where new economic structures emerge, and business relationships among the key players are changed.

Online advertising relies on a variety of techniques such as guaranteed contracts, search-based advertising, real-time bidding (RTB), etc. The onset of online advertising brought about the emergence of a novel mode of acquiring media. Initially, it was executed as guaranteed delivery contracts (or simply guaranteed contracts), where publishers pledge to advertisers to offer a requested number of impressions which is negotiated over manual Request for proposals (RFP)(Ghosh et al. 2009). In online advertising, the slot of an ad either in a web page or on a mobile application is called an “impression”, and almost all of the participants in intermediation between advertisers and consumers operate multi-sided platforms. Facebook, for example, operates a software platform (Evans 2009) that encourages developers to write applications that also enlist advertisers and consumers. In fact, online advertising is now carried out without any human intervention, through an automated process, commonly known as programmatic media buying (PMB) (Ebbert 2012). PMB refers to the process of automated media buying through platforms such as ad exchanges, agency trading desks, demand-side platforms (DSPs) and supply side platforms (SSPs).

Furthermore, as a result of PMB, the emergence of RTB has garnered a wealth of attention from both practitioners and researchers. Most publishers at present, attempt to sell the bulk of their premium inventory in a direct fashion and decide to sell off any excess inventory using RTB (Chen et al. 2014). In RTB, the buying and selling of ad impressions is performed through an auction bidding process that unfolds within milliseconds. According to the recently published Online Advertisers Survey Report (Econsultancy 2013), among 650 advertisers 62% see improved revenue as the primary advantage of RTB. In addition, the trading desk spent on RTB globally stands at 40%. Principally, RTB helps to reduce the time and labour costs of manual intervention. RTB has also significantly changed the online advertising concept, evolving from the traditional strategy of “media buying” to “target audience buying”, and is expected to become the standard business model for online advertising in the future (Yuan et al. 2014).

As shown in Figure 1, an RTB ecosystem has two sides; advertisers and publishers. Each side has its distinct processes in the bidding process (IAB 2014). Each step of the bidding process is demonstrated in Figure 1 (IAB 2014; Yahalom and Stopel 2011). In the RTB process, advertisers can target not only the context but also specific users which significantly improves their return on investment (ROI) (Kohno et al. 2005) through higher conversions. In online advertising, a conversion occurs when a user clicks an ad and proceeds to perform an action that is vital to the business (i.e., the client of the advertiser) such as making an online purchase or downloading a mobile app. Besides increased ROI, advertisers with smaller budgets are afforded the opportunity to access higher quality and more apposite impressions. As shown in Figure 1, the RTB ecosystem consists of three key components: the demand-side platform (DSP), the supply side platform (SSP) and the RTB exchange (or ad exchange). The high-level interaction flow of the RTB ecosystem can be expanded upon as follows.[2] Table 1 provides essential details with factual sample values that are included in the bid request, bid response, win note and conversion note (IAB 2014).

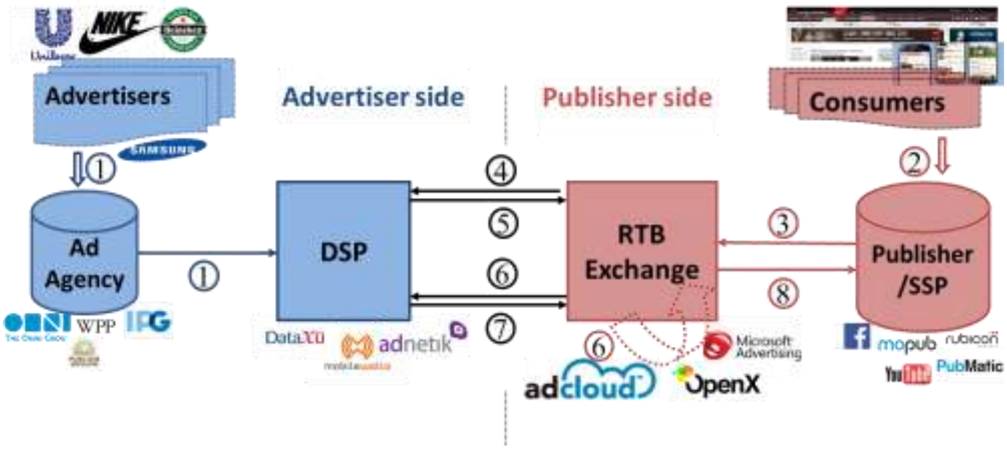


**Figure 1. High-level communication diagram for an RTB Ecosystem**

[2] In Figure 1. Step 1 occurs separately offline, whereas steps 2 through 8 occur in real-time as a continuous process.

*Step 1*. An advertiser requests a DSP to run an ad campaign for a particular product based on a predefined campaign budget, target audience, and campaign duration.

*Step 2*. When the user interaction begins, the mobile app or the web browser sends user preferences, context, location, and the mobile device/browser's information to the publisher to fill up the impressions in the mobile app or website.

*Step 3*. Subsequently, the publisher will check if there exists a contracted advertiser available for the mobile app or website based on their prior agreement and if it is still valid. The ad request is then sent to that particular advertiser. If the quota is unavailable or the contracted advertiser is uninterested in the new impression, the ad request is sent to the RTB exchange.

*Step 4*. The RTB exchange (or sell-side platform) will create a bid request for the incoming ad request and send it to all subscribed DSPs (See Table 1 for the content of a bid request and appendix A for the complete structure of a bid request ).

*Step 5*. The DSP will decide the bid price based on an anticipated ad campaign of a particular ad agency. All DSPs send their bid responses alongside the bid price for the relevant RTB exchange (See Table 1 for the content of a bid response).

*Step 6*. After a predefined and fixed time period has elapsed, the RTB exchange stops the auction and decides the bidder who has made the highest bid price as the winner through the second bid price auction

(Edelman et al. 2007). The RTB exchange then sends the win note to the winning DSP with the winning bid price (See Table 1 for the content of a win note).

*Step 7*. The DSP requests the ad from the ad agency and sends the ad response to the RTB exchange.

*Step 8*. The RTB exchange forwards the advertisement to the germane publisher and users are able to see the advertisement on their mobile application or web page.

As per the communication workflow of publishers and the ad agency, DSPs bid on behalf of ad agencies to buy impressions from applications whose audience is similar to the advertiser's target audience. DSPs are designed to support advertisers manage their ad campaigns and optimize their RTB activities (Zhang et al. 2014). An advertising agency is seen as any third party or in-house team, that works on behalf of companies, to broadcast their advertising demands. Therefore, DSPs support agencies by planning and executing ad campaigns, and analysing the best possible investments on bidding to improve ROI for advertisers. An ad exchange is responsible for defining the winning criteria and delivering the winning notification to the relevant advertisers through the DSP. SSPs are created with a central management console presenting various tools that serve publishers. For example, SSPs allow publishers to set a reserve price for a specific placement or even against a specific advertiser (Yuan et al. 2013). The ad network works as an aggregator that connects advertisers to publishers that want to host advertisements. When a particular publisher integrates an ad network into his mobile application, the ad network delivers the ads according to the ad requests that are produced at that application's app instance level. App instance level is defined as the client side of a mobile application where mobile user interactions occur. In this dissertation, we consider all publisher side ad delivery solutions (SSP, Ad exchanges, and Ad networks) to be *ad networks*.

Before delving into further detail, we define a few common terms which are important in comprehending the RTB ecosystem and are used throughout this dissertation. The *advertiser target* is a common term used to denote the advertisers' target spend, target audience and target number of conversions. The *target spend* is the total dollar value an advertiser can use in buying impressions during a particular campaign. The *target audience* of an application can be determined based on the characteristics of its users, such as age, income, ethnicity, languages, has children, gender, education, etc. The *campaign period* is the total duration of an ad campaign. *Conversions* are also called "actions" and they reflect how users interact with the advertisements, such as clicks, calls, SMS, views, etc. To further understand the aforementioned terms, let us consider an example. A DSP runs an ad campaign for a day promoting a Unilever Shampoo product named "Dove" towards a female audience. Here Unilever is the advertiser, and Dove is the product. In this instance, Unilever will decide the target spend as $1000, the campaign period as a day, target audience as 100% female and the target number of conversions as 2000 clicks.

| **Field** | **Description** | **Sample** |
|---|---|---|
| Bid request (Step 4) | | |
| id | Unique ID of the bid request, provided by the RTB exchange. | 9026174797775044599 |
| Timestamp | Time of the bid request initiated | 1402724400154 |
| imp | Describes the ad position or impression being auctioned. | "banner": {"topframe":1,"id":"1","w":320,"btype":[1,4],"battr":[3,8,9],"hmin":50,"api":[4,3,5],"wmin":300 |
| site/app | Whether the ad supported content is part of a website or mobile application. Also, it includes information such as identifier, name, Domain, publisher, content and keywords which describe the site/app | "id":"81134", "name":"AcacdemMedia Nail Manicure", "publisher":{"id":"194507","name":"AcademMedia"}, "domain": {"com.games4girls.NailManicure"} |
| Device | Information pertaining to the device including its hardware, platform, location, and carrier. | "make": "Apple", "model": "iPhone 6", "connectiontype": 3, "carrier": "VERIZON", "os":"iOS", "osv":"6.1" |
| Geo | Describes the current geographic location of the device (e.g., based on IP address or GPS), or it may describe the home geo of the user (e.g., based on registration data). | "zip":"10018","lon":-73.88476,"lat":40.73874,"city":"New York", "metro": 501, "region": NY |
| User | Describes the user details such as year of birth, gender and user interests | "gender":"M" |
| tmax | The maximum amount of time in milliseconds to submit a bid (e.g., generally the bidder has 120ms to submit a bid before the auction is complete) | 200 |
| Bid response (Step 5) | | |
| Id | Relevant bid ID which the response is mapped | 9026174797775044599 |
| Price | Bid value which is decided by the DSP | 1.50 |
| Currency | Type of the currency, which the bid is made | USD |
| nurl | Win notice the URL | http://inneractive.mobilewalla.com/inneractive/win/${AUCTION_ID}/${AUCTION_BID_ID}/${AUCTION_IMP_ID}/${AUCTION_SEAT_ID}/${AUCTION_AD_ID}/${AUCTION_PRICE}/${AUCTION_CURRENCY} |
| adid | An identifier that references the ad to be served if the bid wins and it is stated by the DSP. | inneractive-9026174797775044599 |
| Win note (Step 6) | | |
| Id | Same identifier as adid in the bid response | inneractive-9026174797775044599 |
| winPrice | Winning bid value which is decided by the RTB exchange | 0.66 |
| currency | Type of the win price currency | USD |
| Conversion note | | |
| impression_id | Same identifier as adid in the bid response | inneractive-9026174797775044599 |
| Timestamp | Time which is the conversion happened (the time user views the advertisement) | 1402724433677 |

**Table 1. Details of the bid request, bid response, win note and conversion note in the RTB process**

## 1.2 Challenges

The real-time nature and programmatic articulation in RTB has resulted many challenges such as a lack of practical solutions and limited exploration in the RTB ecosystem. In this dissertation, we delve into these two main challenges. The first challenge is limited data accessibility. Many researchers conducted their research using synthetic data instead of factual data and have assumed that their proposed solutions can access the information in its entirety in the RTB process. However, in the RTB process, the winning bid prices are not published to all DSPs and the RTB exchange does not send all bid requests to all DSPs (Adikari and Dutta 2015). Therefore, data limitation makes a global view of the RTB ecosystem unavailable to DSPs or any other stakeholder. And as a result, we cannot (and should not) design, develop, and evaluate a solution which considers at global view of the data. Any solution we design should be bound to at the local view of the data, i.e., from a particular DSP perspective only and should be running within the DSP system.

The second challenge is the dynamism in RTB and this is further convoluted due to the real-time nature of the RTB’s decision making. Most of the existing solutions on RTB are based on historical data on the number of bidders, impressions, bid values and winning prices (Chaitanya and Narahari 2012). However, if we consider the advertiser’s perspective, dynamism exists over the number of bid requests received from each application, the different types of active applications in a particular period, the rapid entrance of new audience types, the winning bid price of each application, the number of advertisers, their target spends, and their target goals (Adikari and Dutta 2015). The continuous entrance of various ad networks and their dynamic behaviors made the RTB process increasingly complex from the publisher's perspective. Also, understanding and determining the variations of mobile user behavior is very important from both advertiser and publisher perspectives as it can enrich the success of all RTB processes. Furthermore, when an ad campaign is launched not enough time exists to run a learning phase which can track down the dynamism entrenched in the bidding process. Consequently, to widely adopt the current RTB strategies, a set of open research problems need to be addressed. As new technologies and theorems are introduced into the RTB paradigm, some of the research gaps stem from their limitations. Next, we summarize the key research gaps that we addressed in this dissertation.

## 1.3 Research Gaps

### 1.3.1 An Effective Bidding Strategy for DSP

Many current bid price prediction strategies that are utilized in DSPs were developed to find a better solution to the campaign optimization problem. However, as elaborated upon previously, the problem becomes even more complex due to the dynamism that exists in the real-time bidding strategies, and certain data handling limitations in the DSP. For example, most existing works on real-time bidding make the naïve assumption that all bidders are aware of each other's valuation of the impressions, but in practice, DSPs receive the winning price of an impression only when they win it. As discussed previously, it is clear that none of the state of the art approaches on RTB completely admits of the dynamism which is inherent in the RTB process. These existing solutions do not support rapid real-time decision making and are unable to overcome data accessibility limitations. As the first study of the dissertation, we have addressed the challenges which had not been addressed in state of the art approaches and proposed the Auto Pricing Strategy (APS).

### 1.3.2 Revenue Optimization for Publishers

Most of the current studies, in both PMB and RTB, focus on either DSP or ad exchange perspectives. Only very few studies discuss the importance of revenue optimization from a publisher's or SSP's perspective. In the world of mobile apps, where 90% of apps are free[3], the predominant revenue generation option for mobile app publishers is advertising and to maximize the revenue obtained through advertisements, mobile app publishers need to select advertisements that provide higher user engagement (such as clicks). In order to do that, we need to develop a solution which optimizes revenue generation for the publishers based on ad network attributes, advertisement attributes, and user attributes. The ad network attributes are fill rate, ad network latency, ad frequency, ad recency, etc. The advertisement attributes are number of conversions, bid value, duration of the advertisement display, type of advertisement, size of the advertisement, etc. The mobile user attributes are user demographics, location, user interests, user's past behaviour etc. As the second study of this dissertation, we address this research gap with a comprehensive solution at the individual app level of the publisher's perspective.

[3] https://www.statista.com/topics/1002/mobile-app-usage/

### 1.3.3 Audience Selection in Campaign Optimization

Audience selection is an important aspect of campaign optimization in RTB for obtaining a higher return for the advertisers and the DSPs. To carry out the targeted advertising in a campaign, advertisers need to supply the preferred audience segments to the DSP. This becomes an interesting research challenge for data science researchers within the IS research community based on the conceptualization of new and innovative mechanisms for better audience targeting. The selection of audience segments must be based on the advertiser's target audience, the reliability of occurrence of the same type of audiences throughout a campaign and their expected number of clicks for the campaign. This has become even more challenging due to the sporadic nature of click behaviour and the abundance of feature values (audience types) which rapidly change over time. As the focus of the third study of this dissertation, we further elaborate on this research gap with more examples and present a fully- fledged solution that outperforms current ad campaign optimization strategies in audience selection.

## 1.4 Outline of the Dissertation

Chapter 1 explained the motivation behind the dissertation, provided an overview of the RTB framework, and outlined the specific research gaps to be answered in order to improve the effectiveness of the RTB ecosystem. The rest of the dissertation is organized as follows:

Chapter 2 elaborates on the first study of the dissertation. It comprehends a new bidding algorithm based on the key challenges discussed earlier in this chapter. The complete algorithm with the detail design artefact is given, and its performances are demonstrated relative to state of the art approaches.

Chapter 3 discusses the second study of the dissertation. It outlines the importance of mobile app publisher revenue optimization and provides an in-depth problem specification and solution formulation followed by a comprehensive evaluation of the proposed solution.

Chapter 4 focuses on the third study of the dissertation describing the audience selection of mobile app advertising with respect to campaign optimization in RTB. The problem is explained in detail while apprehending on prominent limitations of the related work. Design and development of the artefact is explained thoroughly, and the analyses are conducted compared with highly mobilized benchmark approaches.

Chapter 5 concludes the dissertation with a discussion on the impact of the studies conducted with an explanation of their theoretical and practical implications and an overview of the possible directions for future research.

# Chapter 2. A New Approach to Real-time Bidding in Online Advertisements: Auto Pricing Strategy

## 2.1 Introduction

With advances in digital advertising, making decisions using real-time data is now the norm. One such advance is autonomous and algorithm-driven real-time bidding (RTB), which completes a full transaction in milliseconds on pre-set parameters (IAB 2014). In online advertising, RTB helps publishers to sell impressions and advertisers to buy impressions via a programmatic instantaneous auction. RTB allows advertisers to launch their advertising campaigns via multiple ad-networks. This process enables advertisers to buy inventory of advertisement slots in a cost-effective manner and serve ads to the right person in the right context at the right time (Lee et al. 2013). According to Econsultancy's recently published "Online Advertisers Survey Report 2013," 62% of the 650 advertisers surveyed see improved performance as the main advantage of RTB. The survey also reported that the trading desk spend on RTB globally stands at 40% of global digital advertisement spend. Mainly, RTB helps to reduce the time and effort of manual intervention in running a digital advertisement campaign; it also facilitates better target marketing strategies in real-time.

According to the introduction chapter, DSP aids the agencies by planning and executing the ad campaigns and analyzing the best possible investments on bidding, to improve the ROI for advertisers (Yahalom and Stopel 2011, IAB 2014). The main goal of the DSP is to manage ad campaigns and optimize their real-time bidding activities (Zhang et al. 2014), such that the advertisers' constraints are satisfied with the optimal price to bid for each ad call while maximizing the ad campaign performance.

One of the most frequently cited and employed strategies for real-time ad allocations is Google's Adwords (www.google.com/adwords), which sacrifices the buyer's ability to control impression allocation. In Google Adwords, publishers specify a set of relevant keywords, and the agencies look for impressions that match their target keywords. However, Google has complete control over where the advertisement will be allocated; the advertising agency has no say. Recently a number of digital advertisement organizations (e.g., inMobi [Inmobi.com], Smaato [Smaato.com], Mopub [Mopub.com], OpenX [Openx.com]) have evolved, following a more open standard (Open RTB (IAB 2014)) for the advertisement, particularly in the context of the mobile ecosystem, in which Google holds only 47% of the total market share (Emarketer.com 2014). A number of algorithms, including plain greedy algorithms, are followed by DSPs. We discuss a number of such algorithms in detail in the related work section. All of them are based on the fundamental assumption that future impressions and bidding patterns will follow past behavior. However, through a real-life dataset, Adikari and Dutta

(2015) observed that in the case of the RTB system, it is difficult to predict a future pattern based on historical data. They demonstrated that forecasting the number of bid requests and the winning bid price based on static historical data has very low accuracy. Any approach that is solely based on static modeling and one-time optimization based on longitudinal historical data will produce suboptimal results. We have demonstrated that the effectiveness of historical data in determining future bids diminishes by 400% as we increase the length of historical data from the last 5 minutes to the last hour. Therefore, this study recommends *a novel approach where the bid price and the decision to bid on an impression are decided in real-time through constant adjustment of the decision process within very short time periods (5 minutes), based on a more recent (last 5 minutes) stream of impressions and their bidding results.* We have demonstrated our strategy on real mobile RTB campaign data. Through an extensive evaluation, we have demonstrated that our approach outperforms existing approaches with reasonable effectiveness. *Our approach improves the conversions (number of clicks) by 19% compared to a recent benchmark approach with the same campaign spending.*

Among the different types of ad platforms which practice RTB strategies, the main two streams are web and mobile. Almost all RTB systems facilitate both platforms in parallel without much differentiation. Consequently, in this study, we use the term “application” to address any mobile applications or websites that are incorporated with an ad platform. According to the RTB ecosystem (IAB 2014), the DSP needs to determine the best bid price which can win the impression. If the DSP is not interested in winning the impression, then the bid price is recorded as zero in the bid response. The key problem that we address in this research is *how to determine this bid price while achieving the advertiser target*. In relation to the above example, the DSP needs to decide the most appropriate bid prices for a selected set of bid requests which enable Unilever to achieve the maximum return from the campaign. The return is the number of actions (clicks) by the target audience at a given target spend.

We have made a few clear theoretical and practical contributions in this study. In terms of the theoretical contribution, first, we demonstrate the importance of defining a domain specific problem by considering its real-world challenges. Second, our solution depicts the importance of developing solutions using immediate past data for the dynamic and complex problems. In terms of the practical contribution, first, we explain the problem of bid price determination from the DSP perspective in an RTB system. Second, we present a dynamic programming (DP) based approach to adjust the bid price of applications. Third, we compare the proposed approach with existing approaches and demonstrate that the proposed approach has better performance in achieving a higher number of conversions for the same target spend.

The rest of the chapter is organized as follows. Section 2.2 provides an overview of the prior research on real-time bidding strategies. In Sections 2.3-2.5, we describe the DP approach in detail, including the solution formation and the proposed algorithm. In Section 2.6, the benchmark approaches are elaborated briefly. Section 2.7 provides a summary of the dataset which is used to test the proposed models and describes the analysis of the DP model, and its performance is compared to the benchmark approaches. Section 2.8 discusses the conclusion of the study.

## 2.2 Related Works

Buying impressions for and allocating them to the advertisers is the most critical process of DSP. In terms of selling advertisements, online advertising consists of two main dimensions, either "sold in spot" through an auction mechanism which is called RTB, or in advance via guaranteed contracts (Chen et al. 2014). The research conducted by Ghosh et al. (2009) to allocate impressions randomized to contractual online advertising campaigns can be easily applied to the RTB context. Ghosh et al. (2009) have developed an adaptive bidding agent for RTB exchanges to learn the distribution of bids and then bid according to that. However, this cannot be practically implemented in the DSP context because only partial data is accessible (King and Mercer 1985). Chen et al. (2014) and Chen (2017) developed mathematical models to allocate impressions between real-time auctions and guaranteed contracts; therefore the application of such a model only in an RTB context will not be effective. Our proposed strategy focuses only on the RTB context.

Some of the RTB-related studies looked at ad allocation in general either from advertiser or publisher perspectives. Rogers et al. (2007) have proposed a probabilistic model of behaviors of the users (advertisement viewers) and the advertisers. The authors tested their model through simulated comparisons with alternative allocation tools, with the objective of identifying the most effective bid value for each auction and maximizing the number of impressions in a given time frame. Hegeman et al. (2011) emphasize the key criteria for building a bidding strategy based on the historical value of the impression, the time or date of the impression, total allocated budget, available budget, the identity of the entity requesting, the predicted likelihood that the ad will be selected, the presence of social functionality, total number of impressions of the ad, the remaining number of impressions to be achieved, etc. Yuan et al. (2013) posit that temporal behaviors, the frequency, and recency of ad displays, would be nontrivial attributes when developing an RTB strategy.

In addition to the general discussions on bidding strategies, there are studies on aspects of the online RTB ecosystem, such as DSP, SSP, and RTB exchanges. Among them, (Chakraborty et al. 2010) have come up with a joint optimization framework through online algorithms and stochastic modeling to optimize the allocation and solicitations in RTB exchanges. Their solution is an online recurrent Bayesian decision framework with bandwidth type constraints. Conversely, many researchers attempted to find better strategies for the advertisers from the DSP perspective. For example, through the greedy approach and linear programming, one bidding strategy optimizes both the budget and the bid price and guarantees the convergence at a locally envy-free equilibrium (Chaitanya and Narahari 2012). Chaitanya and Narahari (2012) have theoretically shown that under linear programming primal-dual formulation, a real-time bidding algorithm can be implemented to adjust bid values according to target spend and fine-grained impression valuation. Also, Sun et al. (2016) have studied information revelation design and policies in terms of the ad exchange perspective. They discussed the One-call and Two-call approaches. Under the One-call approach the ad exchange makes one call to all bidders (DSPs) at the beginning of an auction and under the Two-call approach in addition to the One-call, there is a second call to the winning bidder (DSP) at the end of the auction to match the right advertiser for the impression.

In many proposed RTB strategies, optimization plays a key role. Zhang et al. (2014) proposed the optimal real-time bidding strategy based on the non-linear relationship between the optimal bid and the conversion rate. In the research on sponsored search auctions (Karande et al. 2013), in selecting the impressions, DSPs optimized the advertisers' ROI and quality of ads. Li and Guan (2014) predicted the bid value while acquiring an impression at the lowest cost. Their strategy predicted the win rate and the winning price based on a logistic regression model and then derived the bidding strategy from the resulting model. (Lee et al. 2013) emphasized the challenge of reaching multiple targeted users with a restricted budget. Therefore, they propose a logistic regression to select high-quality impressions and adjust the bid price based on the prior performance.

The above strategies were all based on historical data, on the number of bidders, impressions, bid values and winning prices (Chaitanya and Narahari 2012). However, in the RTB process from the DSP perspective, the system is dynamic. The parameters used in RTB decisions (such as the number of bid requests received from each application, the different types of active applications in a particular period, winning bid price of each application and the number of advertisers) are constantly changing. For example, in our dataset, we noted that some mobile applications are highly active on a particular day with a larger number of incoming bid requests, but on the next day, some of them did not even appear, and others had far fewer bid requests. For this reason, applying any predictive logic could be futile: (Adikari and

Dutta 2015) tried to forecast bid prices and the number of bid requests for each week of a particular month, based on past performance data from a DSP. According to their results, both the bid price and the bid request forecasts have very low accuracies (15 to 66% error in the case of the bid price and 25 to 79% error in the case of the bid request count) with a considerable difference between the weeks' results.

Perlich et al.(2012) used supervised learning algorithms, among other optimization related techniques, for the bid price prediction from the DSP perspective. But the result is not exclusively dependent on the supervised learning algorithm. Analyzing conversion rates, Yuan et al. (2013) indicate the need for an optimization algorithm which incorporates factors which earlier studies have not adequately addressed, such as the frequency and recency of the ad displays. Consequently, on behalf of advertisers, DSPs need to establish bid prices based on the combination of advertisers' target spend, target audience and behavior of the previous bid period, which includes winning rates, the number of bid requests and conversion rates.

Additionally, unlike past studies, under the challenges discussed in the introduction chapter, we consider some practical constraints of designing a bidding strategy for DSP, important for real-life applicability. Firstly, compared to other auction systems, in the RTB process, each DSP receives only its own winning bid prices, so it does not have similar data for the other DSPs. RTB exchanges also publish the winning bid average (WBA) for each application. However, the WBA is computed at a lower frequency (such as every 24 hours) and is based on a longer duration of data (such as a week). This data does not add much value other than aiding the current approach of

DSP bidding, bidding higher than the published WBA of the desired application. Secondly, the RTB exchange does not send all the bid requests to every DSP. The bid requests distribution is based on the agreement between the DSP and the RTB exchange. For these reasons, as given in challenge one, the global view of the RTB exchange is unavailable to DSPs. As a result, we cannot develop a solution which considers the global view of the data. Any solution we design should be bound to the local view of the data, i.e. from a particular DSP perspective only, and will be run within the DSP system. Furthermore, when an ad campaign is launched there is not enough time to run a learning phase which can track down the dynamism entrenched in the bidding process[4]. Because earlier research does not admit the dynamism which is embedded in the RTB process, the existing solutions do not bind with the rapid real-time decision making, whereas the dynamism can be implemented through the proposed approach. Lastly, this is one of the few studies on RTB that uses a real-life dataset, which is described later in the chapter.

## 2.3 Solution Formulation

Our main objective in the research is to find the applications to bid at any time (decision variable $\widehat{k_{jt}}$) so that we maximize the total number of conversions (number of clicks, $\widehat{C_{jt}}\widehat{V_{jt}}$) in the campaign, for the given campaign budget (*Budget)*. Thus we present the model of the problem as follows.

$$Maximize \sum_{t=0}^{T} \sum_{\forall j} \widehat{C_{jt}}\widehat{V_{jt}}\widehat{k_{jt}}$$

Subject to,

$$\sum_{t=0}^{T} \sum_{\forall j} \widehat{P_{jt}}\widehat{V_{jt}}\widehat{k_{jt}} \leq Budget$$

[4] This has been discussed under the second challenge in the introduction chapter.

$$\widehat{k_{jt}} \in \{0,1\} \quad \forall j, t$$

We use the following symbols in the above model.

| | |
|---|---|
| *Decision Variable:* | |
| $\widehat{k_{jt}}$ | 0-1 decision variable,1 if application *j* is selected to bid at time *t*, 0 otherwise |
| *Indices:* | |
| *j* | Index for applications |
| *t* | Current instant (time) |
| *Parameters:* | |
| *T* | Total campaign period |
| $\widehat{C_{jt}}$ | Conversion rate for application *j* at time *t* |
| $\widehat{V_{jt}}$ | Number of impressions won in application *j* at time *t* |
| *Budget* | Total campaign budget |
| $\widehat{P_{jt}}$ | Bid price for application *j* at time *t* |

The conversion rate, the number of impressions won, and the bid price for applications are time variant variables. However, these parameters are dependent on many external factors, such as how the RTB engine is distributing the app impressions to various DSPs, how the popularity of an app is changing, and how an app is being used by users. Also, conversion rates and the number of impressions won in an application depend on the bid price on each application and the bidding decisions. So it is impossible to find a mathematical representation of a stochastic distribution to fit these parameters. The distributions of these parameters are unknown. Nor is it not possible to predict the future distribution of these parameters to solve the above model. There exist additional complexities: (1) the campaign spending needs to be as uniformly distributed over the total campaign duration as possible and (2) the bid price needs to be determined by the DSP.

To solve the problem addressing the above complexities, we resort to a heuristic approach, auto pricing strategy.

## 2.4 Auto Pricing Strategy

In this section, we present a heuristic, auto pricing strategy (APS), to solve the above model. The APS helps the DSP to maximize the number of conversions by making the following decisions:

- Determine how to distribute the budget over the total campaign period
- Determine the bid price of each application
- Determine the number of impressions to bid for each application.

Before describing the approach, we present a list of notations which are used throughout the chapter in Table 2.

| | |
|---|---|
| *Indices* | |
| $j$ | index for applications $j = 1, \dots, J$ |
| $b$ | index for the selected applications for bidding $b = 1, \dots, B$ |
| $x$ | index for applications' target audience characteristics $x = 1, \dots, X$ |
| $d$ | index for advertiser's desired target audience characteristics $d = 1, \dots, D$ |
| $t$ | index for a bid period $t = 1, \dots, T$ |
| *Parameters* | |
| $S_t$ | remaining budget of an advertiser at bid period $t$ |
| $I_{j,t}$ | total number of impressions available for application $j$ at bid period $t$ |
| $P_{j,t}$ | bid price for application $j$ at bid period $t$ |
| $C_{j,t}$ | conversion rate for application $j$ at bid period $t$ |
| $Y_{x,j}$ | target audience options for characteristic $x$ on application $j$ |
| $Y'_d$ | advertiser's target audience option for characteristic $d$ |
| $J$ | total number of available applications to bid |
| $X$ | total number of target audience characteristics |
| $T$ | total number bid periods performed during an ad campaign period |
| $B$ | total number of applications which have been selected to bid |
| $A_j$ | accessible target audience for application $j$ |
| $W_{b,t}$ | target winning rate for application $j$ at bid period $t$ |
| $P_{b,t}$ | bid price for application $b$ at bid period $t$ |
| $S'_t$ | budget allocation for bid period $t$ |
| $M_t$ | moving average on total bid requests at bid period t |
| $MAR_t$ | moving average ratio at bid period t |
| $V_{b,t}$ | number of impressions won in application $b$ at bid period $t$ |

| | |
|---|---|
| $\mathbb{Z}^*$ | set of zero and all positive integers |
| *Decision variables* | |
| $K_{j,t}$ | number of impressions selected to bid from application $j$ at bid period $t$ |

**Table 2. Notation for the model descriptions**

To utilize the full campaign period properly and achieve the advertiser target, we divide the campaign period into equal multiple intervals which are called bid periods and denote as $t$ from this section onwards. Please note that the discrete time ($t$) in section 2.3 is being modified to represent bid period ($t$) in this and subsequent sections in the chapter. The duration of a bid period will be decided by the DSP.

As explained in Figure 2, when the campaign is running, based on the previous bid period's data, the proposed model will determine the target applications and the number of bid requests per app. This process is considered the app selection strategy. The application selection strategy depicts an autonomous and dynamic bidding strategy for the whole campaign and facilitates identifying the best possible applications. Such applications can optimize the advertiser target by maximizing the advertiser's utility value with respect to exogenous factors such as the accessible target audience, conversion rate, remaining target spend, and the bid price. The changes in these exogenous factors need to be predetermined by the DSPs. For example, to distribute and allocate the target spend for the next bid period and to determine the bid prices to reach the target winning rate in the next bid period, DSPs should predetermine the variations in target spend and bid prices respectively. Hence, we define two other supportive strategies, namely a budget allocation strategy and a bid price adjustment strategy.

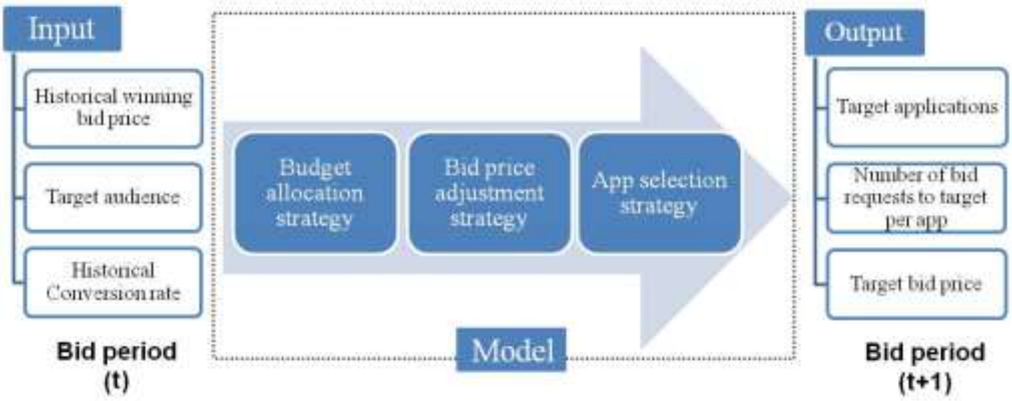


**Figure 2. Three steps of the auto pricing strategy**

### 2.4.1 Budget Allocation Strategy

When a new bid period is starting, we need to allocate the target spend for each remaining bid period, depending on the previous periods' total number of bid requests received and the remaining target spend. As real-time bidding takes place in a very rapidly changing environment, the number of bid requests which received by a DSP can fluctuate in adjacent bid periods capriciously. To enforce this dynamism, we calculate the moving average value of the total bid requests for each bid period until the last executed bid period.

$$M_t = \frac{\sum_t t \times \sum_j I_{j,t}}{\sum_t t} \tag{2.1}$$

According to the Eq. (2.1), we have considered the index of the bid period as the weight of each period's total bid request. As a result, the current period has the highest weight compared to the previous bid periods. Using the moving average value of the current period $(t)$ and the previous period$(t-1)$, we compute the Moving Average Ratio (*MAR*) for the next period as shown in Eq. (2.2). This will help to apply the recent changes of the received bid requests in the last bid period in the model with respect to the previous bid periods.

$$MAR_{t+1} = M_t / M_{t-1} \ , \forall t. \tag{2.2}$$

When we have *MAR* for the next period, we compute the budget for the next period as shown in Eq. (2.3),

$$S'_{t+1} = \begin{cases} \frac{S_t \times MAR_{t+1}}{T-t}, & MAR_{t+1} < T-t \\ S_t, & MAR_{t+1} \geq T-t \\ S_t, & t = T \end{cases}, \forall t. \tag{2.3}$$

According to the Eq. (2.3), we allocate the budget for the next period based on the number of remaining bid periods and the number of bid requests received in the past bid periods. Since we consider the moving average values of two consecutive bid periods to calculate *MAR* and there is no huge change in the total number of bid requests received by a DSP in a bid period, the *MAR* value will be most of the time less than the remaining number of bid periods. During our analysis, we have deduced that in most cases *MAR* resides between 0 and 2. Therefore, in any given bid period, after allocating the budget for the next period, the algorithm can preserve the target spend for the remaining bid periods.

### 2.4.2 Bid Price Adjustment Strategy

In RTB, the winning bid price has a dynamic behavior which varies frequently. Thus, the winning bid price which is predicted for a particular bid period might not be the optimal bid price in the following period. If we bid higher than the optimal bid price, we will pay more than required for the desired goal, so that we would not be able to achieve a higher number of conversions due to the restricted target spend; on the other hand, if we bid for a lower bid price, then also we would not win enough to achieve a higher number of conversions, due to limited duration of the campaign. As a result, in this step, we adjust the bid price with regard to the actual winning rate, which is defined in Eq. (2.4):

$$W_{b,t} = \frac{V_{b,t}}{K_{b,t}} \ , K_{bt} > 0, b \in J, \forall \text{b}. \tag{2.4}$$

In our model, the actual winning rate is calculated at the end of each bid period (i.e. merely the beginning of the next period), and it is used to adjust the bid price for the next period. Mainly, the bid price adjustment strategy is developed on the idea of maintaining a higher winning rate for a lower bid price. For instance, in a particular application, in a particular bid period, if the number of actual winning bid requests is equal to the number of expected winning bid requests, it is possible to reduce the bid price. Winning all bids in a bid period indicates that the bid price may have been higher than that required to win the bid, indicating paying more, and thus there is a possibility of reducing the bid price.

However, in many situations, the actual winning rate is lower than the expected winning rate. Hence, to increase the winning rate for the subsequent bid period, we increase the current period bid price with respect to the actual winning rate for the current period. Furthermore, if the actual winning rate for the current period is zero for a particular application, then we increase its bid price for the next period by a constant value multiplication.

The new target bid price for the next bid period is computed as Eq. (2.5),

$$P_{b,(t+1)} = \begin{cases} P_{b,t} \times \alpha, & W_{b,t} < 0.5, \ 1 \leq \alpha \leq 2 \\ \frac{P_{b,t}}{W_{b,t}}, & 1 > W_{b,t} \geq 0.5 \\ P_{b,t} \ \times \ \beta, & W_{b,t} = 1, \ 0.5 \leq \beta \leq 1 \end{cases}. \tag{2.5}$$

In Eq. (2.5), we have defined two constants, $\alpha$ and $\beta$, as thresholds to limit the scope of bid price adjustment. $\alpha$ is defined to adjust the bid price when the current bid period's winning rate is zero for a particular selected app. Its boundary is based on the concept of doubling the price to increase the winning probability (Perlich et al. 2012). $\beta$ is defined to adjust the bid price of the apps which won all the bid requests that were selected to bid in the current period. Based on the same concept followed in $\alpha$, we define the boundary of $\beta$ to see the winning rate when the bid price is half of the current bid period's bid price. Through these experiments (Section 6.2), we have identified the best possible values for the $\alpha$ and $\beta$ as 1.5 and 0.8, respectively. In practice, DSPs can decide these two values based on data from previous campaigns. Also if the DSP is running a campaign for a longer campaign period with shorter bid periods, then during the first few bid periods, it can experiment with $\alpha$ and $\beta$ values to discover acceptable values. Apart from the bid price adjustment on the selected applications, the system also should keep track of all the applications and their details which generated the bid requests in the current period. We apply the price adjustment strategy to all the targeted applications during the current bid period. Therefore, to condense the effect from applications which were not targeted in the current period, we have updated their total number of bid requests in the current period as 1. Even if the model selects such applications to bid in the next period, it can expect only one bid request to be bid. As a result, if such bid requests could not win in the bid period ($t+1$), then the model will adjust their bid prices for the next bid period ($t+2$) and it will increase the probability of winning for the next period. Eq.

(2.6) defines this constraint. This strategy has been experimentally evaluated and the results are explained in the section 2.7.

$$I_{j,t} = 1 \;\; j \notin B, \forall \text{j}. \tag{2.6}$$

### 2.4.3 App Selection Strategy

Here we do not know what the conversion rate and the number of impressions of an app would be in the future. Moreover as these parameters (conversion rate and the number of impressions) of an app are dependent on so many external factors (such as how the RTB engine distributes the app impressions to various DSPs, how the popularity of an app is changing, or how the users are interacting with an app), we decided it is not prudent to fit stochastic distributions to these parameters and develop an approach based on these distributions. Rather we followed a dynamic approach based on the simple rule that the "near past is the best predictor of immediate future." In this approach, we develop a mathematical optimization problem for app selection based on maximizing a utility function of the app in the problem. The input of the optimization problem is the performance and impression data of apps from the past time period (*t-1*); the output of the optimization problem is the list of selected apps to bid in the next bid period (*t*). Subsequently, based on newer performance and impression data, the optimization model is again run after the next bid period (*t*) to decide the list of selected apps for the following time period (*t+1*). This continues until all the campaign budget has been exhausted or the campaign duration has come to an end. Below we present the dynamic programming representation of the above approach.

### 2.4.4 Dynamic Programming Model

For our problem following the dynamic programming (DP) (Lew and Mauch 2006) approach, we define the contribution function *R (.)* of the DP model as in Eq. (2.7).

$$R\left(\Theta_t, \mathrm{K}^{\pi}_{\mathrm{j,t}}(\Theta_t)\right) = \sum_{\mathrm{j}} U_{j,(t+1)}\left(\Theta_t, \mathrm{K}^{\pi}_{\mathrm{j,t}}(\Theta_t)\right), \quad \forall \mathrm{j} \qquad (2.7)$$

$$\text{where } \Theta_t = (C_{jt}, V_{j,t}, P_{j,t})$$

In Eq. (2.7), a particular state is defined with $\Theta_t$ and its outcome depends on the previous bid period's exogenous factors $C_{j,t}, V_{j,t}$, and $P_{j,t}$. We define the contribution function *R(.)* as a reward using the utility function $U_{j,t}(.)$, which is derived in the next section. $K^{\pi}_{j,t}(\Theta_t)$ defines the policy of selecting the apps based on the utility value at the end of each period. Therefore, to maximize the number of conversions obtained for a lower target spend throughout all the bid periods (*T*) of the campaign, we maximize the contribution function as given in Eq. (2.8). The notation $\mathbb{E}^{\pi}$ in Eq. (2.8) defines the possibility that the exogenous factors (conversion rate, winning rate, and bid prices) depend on the outcome of the previous bid period.

$$\mathrm{M}aximize_{\mathrm{j,(0<t\leq T)}}\left(\mathbb{E}^{\pi} \sum_t \sum_{\mathrm{j}} U_{j,(t+1)}\left(\Theta_t, \mathrm{K}^{\pi}_{\mathrm{j,t}}(\Theta_t)\right)\right), \forall \mathrm{j} \qquad (2.8)$$

$$\mathbb{E}^{\pi}\text{:}\ \ \mathrm{K}^{\pi}_{\mathrm{j,t}}(S_t)_{\forall j} = Maximize \sum_{\mathrm{j}} U_{j,(t+1)}\left(\Theta_t, C_{j,t}, V_{j,t}, P_{j,t}\right), \forall \mathrm{j} \qquad (2.9)$$

The policy $K^{\pi}_{j,t}(\Theta_t)_{\forall j}$ in our problem is obtained by maximizing the utility function across all apps for the next bid period (*t+1*) based on the values of exogenous factors received from the outcome in the past bid period (*t*) (Eq. (2.9)). This is equivalent to the look-ahead policy as defined by Powell (2014) with one time period as the horizon.

Next, we describe the estimation of parameters for the utility function and the maximizing problem to define the policy of transition from one bid period ($t$) to the next bid period ($t+1$).

### 2.4.5 Optimization Strategy

In the RTB process, the advertiser is looking for higher returns, through a higher number of conversions for his investments. Therefore, the goal of the model is to increase the advertisers' ROI of buying impressions, by optimizing the advertisers' target in all the available listed applications at a certain bid period. In the process of establishing the model, we can define a utility value to demonstrate the effectiveness of achieving conversions with regard to the advertiser target. To define the utility value of placing an advertisement in an app, first, we define the accessible target audience.

#### 2.4.5.1 *Accessible Target Audience*

In most cases, publishers provide the target audience for their applications to the DSPs based on user characteristics such as the percentage of males using the app, the percentage of different age groups using the app, etc. With such information and the advertiser's preferences, the following process defines the strategy to identify the accessible target audience. For example, if an advertiser requests targeting of his advertising campaign towards female users, then the best impressions to be published for such advertisements are those which belong to an application with a target audience gender of 100% female, and the worst case scenario is an application with a 100% male target audience. The same would hold for an advertiser who requires targeting of his ad campaign for females who are under the age of 30. If there is a particular application with 60% female and 80% under the age of 30 as the target audience, then the accessible target audiences will be computed by the product of female and age group percentages (48%). Hence, the accessible target audience for a particular application is captured based on the product of relevant target audience characteristics. It is defined as Eq. (2.10),

$$\mathrm{A_j} = \prod_{\mathrm{x}} \mathrm{Y_{x,j}}, \;\; \mathrm{Y_{x,j}} \in Y'_d, x \in d, \forall \mathrm{j}. \qquad (2.10)$$

2.4.5.2 *Utility Function*

The utility value can be determined based on the return rate. By multiplying the conversion rate and accessible target audience, we can calculate how many conversions of the required target audience can be obtained for a particular bid value of an impression. Then, to identify the exposure or return per dollar, the return rate (*RR*) is estimated as Eq. (2.11),

$$RR_{jt+1} = C_{j,t}A_j / P_{j,t} \,, \quad \forall j, t. \qquad (2.11)$$

Since we have the return rate per impression, the value of an effective number of conversions per dollar can also be computed. This value reflects the utility value for an advertiser. Eq. (2.12), defines the utility value (*U*) for the advertiser on a particular application j.

$$U_{j,(t+1)} = \frac{C_{j,t}A_jK_{j,(t+1)}}{P_{j,t}} \,, \quad \forall j, t \,. \qquad (2.12)$$

2.4.5.3 *Optimization Policy*

In transitioning from state $\Theta_t$ to $\Theta_{t+1}$ (bid period (*t*) to (*t+1*)), we apply the policy of maximizing the aggregated utility value (Eq. (2.9)) for all the apps for bid period (*t+1*) based on the state at the end of bid period *t*. When maximizing the utility value we can increase the return rate and an effective number of conversions that can be attained for a lower bid price, or in simple terms, when maximizing the utility value we can access to a higher conversion rate, higher accessible target audience, lower bidding price and a higher number of selected impressions. However, when identifying the highest utility value among all the applications' utility values, we have to endure the following constraints of the advertiser.

The first constraint, Eq. (2.13), the number of impressions chosen from a particular application, is limited to its total number of available impressions. As per the results of the earlier section, we consider the predicted value of the number of available impressions for the next bid period. Therefore, when selecting impressions to bid, the total number of impressions which can be selected is limited to the number of available impressions of that particular bid period. This should be true for all the listed applications.

$$\mathrm{K}_{\mathrm{j},(\mathrm{t}+1)} \leq \mathrm{I}_{\mathrm{j},\mathrm{t}}, \quad \forall j, t. \tag{2.13}$$

The next constraint, Eq. (2.14) is that, when selecting applications, spending for all the impressions should be less than or equal to the target spend for the period. The mathematical formulation for this constraint is defined as follows:

$$S'_{t+1} \geq \sum_j P_{j,(t+1)} K_{j,(t+1)}, \quad \forall t. \tag{2.14}$$

However, since impressions are selected only from the applications which have maximum utility value, for the remaining applications, the numb_r of selected impressions could be zero. Therefore, the constraint, Eq. (2.15) should hold,

$$K_{j,(t+1)} \in \mathbb{Z}^*, \quad \forall j, t. \tag{2.15}$$

Based on the above three constraints, the aggregated utility value is maximized to find the most suitable applications which could optimize the advertiser's expectations. We define Eq. (2.16) to maximize the sum of the utility value across all applications at a particular bid period ($t$) while bonding through the constraints.

$$Maximize \sum_j U_{j,(t+1)} \quad \rightarrow \mathrm{M}aximize \sum_j \frac{C_{j,t} A_j K_{j,(t+1)}}{P_{j,(t+1)}} \;, \forall t. \quad (2.16)$$

According to the value of $K_j$, we find the actual number of impressions to bid for each application. The applications whose impressions are not selected will not be considered for the bid price adjustment of the next bid period, but, as explained earlier, their number of bid requests will be changed.

In summary, *the budget allocation strategy* allocates the relevant amount of target spend for the bid period, *the bid price adjustment strategy* adjusts the bid prices, according to each app's current bid period's winning rate, and *the app selection strategy* selects the apps and calculates the number of impressions to bid for the next period.

## 2.5 APS Algorithm

The complete approach is presented in algorithm form in Algorithm 1.

**Algorithm 1: APS algorithm**

1. Initialize $\mathrm{S}_0$ , $T, distanceVector = 0, bidReqTotal = 0, \sum_{j,t=0} I_{j,t=0} = 0\,, S_1' = S_0/\mathrm{T}\,, M_0 = 1$
2. Source state: $t = 0$ which is the most recent historic campaign data, is used to compute the application set B and $\mathrm{K_b}$ in the initial state: $t = 1$
3. $for\ \forall t\,, until\ t\ \leq T\ or\ \mathrm{S}_t > 0\ \ do$
4. $\quad bidResCount_j = 0,\ clickCount_j = 0, \forall j$
5. $\quad for\ \forall bid\ requests,$ until the end of the bid period t do
6. $\quad\quad I_{j,t} = I_{j,t} + 1$
7. $\quad\quad$ if( $j\ \in B\ and\ S_t' > 0\ and\ \mathrm{K_{jt}}\ \geq bidResCount_j$)
8. $\quad\quad\quad$ send bid response with $P_{j,t}$
9. $\quad\quad\quad bidResCount_j = bidResCount_j + 1$
10. $\quad\quad\quad$ if (win note receives)
11. $\quad\quad\quad\quad S_t' = S_t' - winPrice$
12. $\quad\quad\quad\quad V_{j,t} = V_{j,t} + 1$

If (conversion note receives)
$clickCount_j = clickCount_j + 1$
$end\ if$
end if
end if
end for // the end of the bid period t
calculate $\sum_j I_{j,t}$
calculate $C_{j,t} = clickCount_j / V_{j,t}$
$S_t = S_{t-1} - S'_t$
$distanceVector = distanceVector + 1$
$bidReqTotal = bidReqTotal + \sum_j I_{j,t}$
$M_t = bidReqTotal / distanceVector$
Calculate $MAR_{t+1} = M_t / M_{t-1}$
if ((*t*=*T*) or ($MAR_{t+1} \geq T - t$))
$S'_{t+1} = S_t$
else if ($MAR_{t+1} < T - t$)
Calculate $S'_{t+1} = \frac{S_t \times MAR_{t+1}}{T-t}$
end if
$for\ j = 1\ until\ j \leq J\ do$
if ($j \in B$ and $K_{j,t} > 0$)
Calculate $W_{j,t} = \frac{V_{j,t}}{K_{j,t}}$
if ($W_{j,t} < 0.5$)
$P_{j,(t+1)} = P_{j,t} \times \alpha$
else if ($W_{j,t} < 1\ and\ W_{j,t} > 0.5$)
$P_{j,(t+1)} = P_{j,t} / W_{j,t}$
else
$P_{j,(t+1)} = P_{j,t} \times \beta$
end if
else
$I_{j,t} = 1$
end if
calculate $A_j$
end for
compute $Maximize\ \sum_j \frac{C_{j,t} A_j K_{j,(t+1)}}{P_{j,(t+1)}}$, w.r.t. Eq. (2.13), (2.14), and (2.15)
output $K_{j,(t+1)}$ where $j \in B$ ($selected\ apps\ for\ the\ next\ period$)
end for // the end of the campaign period

Algorithm 1 is established on the bid periods and each bid period will define a new state. By iterating through the bid periods, the algorithm strives to optimize the advertiser return (maximizing the total conversions as discussed in Section 3). Before starting the algorithm, at the initial step (algorithm 1: line 1), we initialize a set of values including the advertiser target spend $\mathrm{S}_0$. Then (algorithm 1: line 2) we initialize the sources state ($t = 0$) which is used to find the set of $B$ and $K_b$ in the initial state in the bid period one. From algorithm 1: lines 3 to 44 define the complete set of functions that the APS carried out in a particular bid period. In each bid period (algorithm 1: line 4) we define a set of constants to track the number of bid requests which are bid $bidResCount_j$ and the number of clicks which are obtained in $clickCount_j$ with respect to the each application. As depicted in algorithm 1: line 5, we consider each incoming bid request and track the relevant bid request count from each application (algorithm 1: line 6), until the bid period ends. According to algorithm 1: lines 7 to 18, from each incoming bid request, the bid requests which are relevant to the selected applications $B$ will be selected to bid. However, this is conditioned on either the target spend allocation for the bid period or the expected number of impressions to bid. We keep track of each responded bid request ($bidResCount_j$). When a winning notification is received, the winning price of the winning bid is subtracted from the target spend allocation for the bid period. Also, the number of winning bids $V_{j,t}$ is counted. Subsequently, if there is a click for the winning impression, then that is ($clickCount_j$) counted too. At the end of the bid period, but before running the proposed model, we calculate the total number of bid requests received (algorithm 1: line 19), the conversion rate for each application $C_{j,t}$ (algorithm 1: line 20), and the remaining target spend $\mathrm{S}_t$ (algorithm 1: line 21).

In algorithm 1, lines 22 to 26 discuss the budget allocation strategy. Based on the $distanceVector$ and $bidReqTotal$, the $M_t$ is calculated. Afterward Moving Average Ratio ($MAR$) is computed for the next bid period using the current period and the previous period $M_t$. Since we have both $S_t$ and $MAR_{t+1}$ we are able to calculate the target spend allocation for the next period (algorithm 1: lines 26 to 30). Again from algorithm 1: lines 31 to 45, we compute the bid price adjustment strategy. As shown in algorithm 1: line 31, we iterate through all the applications and calculate the winning rate for each bid application of the current period. Then, based on the winning rate, we adjust the bid price as depicted in algorithm 1: line 34 to 40. As explained in Eq. (2.6), in line 42, we have changed the total bid request count to 1, for an application which was not selected for the current period. At algorithm 1: line 44 we compute the accessible target audience of each application based on the advertiser target audience (see Eq. (2.10)). Algorithm 1: line 46 depicts the optimization strategy with regards to the constraints discussed in Eq. (2.13), (2.14) and (2.15). The output of the first iteration of algorithm 1 will be utilized as the input to the next bid period.

As algorithm 1 elaborated, in each iteration the model optimizes for a higher number of conversions for a lower target spend. It implies that the algorithm achieves the local optima at each bid period. As we have shown later in the analysis section (Section 6.4), even though each bid period is solved individually for the local optima, the final output of the model is not the same as the global optimal solution obtained by the exact approached described in Section 5.1. The solution obtained by the exact approach is impossible to achieve without knowing the future. The final solution is in close approximation to the exact optimal solution.

## 2.6 The Benchmark Approaches

In this section, we try to evaluate and understand the possible gains of the proposed bidding approach. For that, we propose two brute force approaches, namely, the greedy approach and the exact approach. Moreover, we have compared our approach with the three latest approaches, which are selected based on the attributes which are available in our dataset. There are other approaches as discussed in Related Works section; however, they require data items from the ad network (SSP) or ad exchanges, data which is not possible in a realistic DSP scenario. These approaches were satisfactorily implemented using our data and provided a complete empirical comparison to our proposed approach using factual real-time data.

| | |
|---|---|
| *Indices* | |
| $r$ | index for the bid requests $r = 1, \ldots, R$ |
| *Parameters* | |
| $S$ | total budget of an advertiser |
| $CON_r$ | conversion value for bid request r either 1 (converted) or 0 (not converted) |
| $R$ | total number of bid requests received |
| $WP_r$ | actual winning price for bid *b* |
| *Decision variables* | |
| $N$ | total number of conversions achieved |

**Table 3. Notation for the exact approach and the greedy approach**

### 2.6.1 The Exact Approach

The exact approach can be implemented only when all the future information about the RTB process (such as target audience, winning price, and conversion outcome) is given because the bidding is made only for the bids whose conversion outcomes are positive and have the lowest winning price and correct target audience. Therefore, in practicality, it is impossible to implement the exact approach. However, for benchmarking purposes, we can implement the exact approach based on the historical campaign data and compare it with our APS. Following the algorithm 2 model provides some insights into the exact approach. Here we provide a list of bid requests as an input and attain a number of target conversions and a list of selected bid requests as outputs. See Table 3 for the notation described.

**Algorithm 2: The Exact Algorithm**

Initialize $L_{in}$ and $L_{out}$ to empty, where $L_{in}$ and $L_{out}$ are lists
Add all $r$....$R$ into $L_{in}$ with ascending order of $WP_r$
While ($L_{in}$ $length > 0$)
    If ($S \geq 0$)
        If ($WP_r > 0$ $and$ $Y_{xr} \in Y'_d$ $and$ $x \in d$ $and$ $CON_r = 1$)
            $S = S - WP_r$
            $N = N + CON_r$
            Remove $r$ from $L_{in}$
            Add $r$ into $L_{out}$
        End if
    Else
        Exit While
    End if
End While
**Output** : $N$, $L_{out}$

### 2.6.2 The Greedy Approach

One of the simplest and most commonly practiced bidding approaches in many DSPs is the greedy approach. When the DSP receives bid requests, it will bid with the highest possible bid price for all the bid requests which have the expected target audience. As a result, it will achieve a certain number of conversions within a shorter period of time for the whole target spend. Even though the bid price is set at the highest value, the winning bid price is decided by the second price auction, i.e., the second highest bid price. For example, if the DSP is interested in a particular incoming bid request, then it can bid for \$1, which is a really high price, but the DSP may have to pay only \$0.1 as the winning bid price, attributable to the second price auction. This can be considered as a benchmark approach for comparison purposes. Here no intelligence is applied in determining the bid price of the application. The algorithmic description of the approach is as follows. See Table 3 for the notation described.

**Algorithm 3: The Greedy Algorithm**

```
Initialize z to a timestamp during the campaign.
Initialize Z as the end time of the campaign
While (z ≤ Z)
        If (S ≥ 0)
                If ( Y_xr ∈ Y'_d and x ∈ d)
                        S = S − WR_r
                        If (CON_r = 1)
                                N = N + CON_r
                        End if
                End if
        Else
                Exit While
        End if
End While
Output : N
```

Note that here the advertiser will end up paying more because he is not pricing the bid as per his requirement and not utilizing the full supply source. For example, if an app is expected to receive 100,000 bid requests in a day, and the advertiser requires winning only 1000 impressions, the greedy approach will select the first 1000, regardless of the price. But if the bid price can be predicted, then the system should select the cheapest 1000 bid requests.

### 2.6.3 Optimal Real-time Bidding *(ORTB)*

Zhang et al. (2014) demonstrated that the optimal bid has a non-linear relationship with the conversion rate. Using real-world data, they have shown that the winning rate consistently has an (approximately) concave shape. Therefore, compared to the higher valued impressions, the lower valued impressions are more cost-effective and have a higher winning rate. In accordance with this rationale, a DSP is able to bid on more impressions rather than focus on a small set of high valued impressions. To further simplify the proposed solution and to calculate the bid price using conversion rate, they have applied the Lagrangian function.

### 2.6.4 Bidding Below Max eCPC *(Mcpc)*

As per Lee et al. (2012), the conversion rate is dependent on the past performance data. They demonstrated that when the advertiser's goal of max eCPC is the upper bound of cost per click, the bid price on an impression can be obtained by multiplying max eCPC and pCTR. This has also been followed by Zhang et al. (2014) to compare their solution while calculating the max eCPC for each campaign by dividing its cost and achieving a number of conversions in the training data. To compare with our APS approach, we also followed the same strategy to predict the bid price and compute the number of conversions using our dataset.

### 2.6.5 Linear-form Bidding of pCTR (Lin)

As per Perlich et al. (2012), the bid value is linearly proportional to the pCTR. They estimated the pCTR by a supervised learning algorithm. The approach has been applied by Zhang et al. (2014) also for the performance comparison of their proposed approach.

## 2.7 Analysis and Results

### 2.7.1 Dataset

To develop a model and test it empirically, we have preserved data from an ongoing RTB process of a leading mobile DSP[5]. Compared to previous studies where the model is examined with synthetic data, the factual data gives a proper insight into the model. The dataset includes the data for three campaigns, each of 10 days' duration, run by the DSP in August 2014. It includes 6,317,443 bid requests which are spread across the month of August 2014. Table 4 depicts some details of these three campaigns. To determine whether the three campaigns' data differ significantly, we have computed the difference between related samples using Friedman's two-way analysis of variance, based on the number of bid requests for each campaign. Since the Friedman test is non-parametric and distribution free, it can seek differences between each campaign datasets (Friedman 1937; Mack and Skillings 1980). According to the result, at the significance level of 0.05 (the resulting significance is 0.005), we reject the null hypothesis of no difference between each campaign dataset. As a result, we can state that the data selected for each campaign differ and this supports our approach in different types of RTB campaigns.

| | Campaign X | Campaign Y | Campaign Z |
|---|---|---|---|
| Total number of bid requests | 2,209,864 | 2,113,487 | 1,994,092 |
| Total distinctive applications | 240 | 205 | 160 |
| Winning bid average for the whole campaign $ | 1.06 | 1.15 | 1.08 |

**Table 4. Data difference among three campaigns**

[5] https://www.mobilewalla.com/

As we discussed in the prior section, the problem has been formulated using mathematical modeling, and we have developed a DP algorithm (Bertsekas 1995) to test the model. Since the model is developed in such a way that each bid period's input is set based on the previous bid period's output, this can be implemented incrementally while adjusting the bid prices and target spend dynamically during the algorithm execution. The algorithm was coded using Java programming language, and the solution to the optimization problem is implemented using the existing free and open-source Java library called Java Optimization Modeler (JOM). It offers a full-fledged platform to model Java programs and solve optimization problems.

To evaluate the efficiency and effectiveness of the model, we defined the metric Target Spend per Conversion (*TSPC*), defined by Equation (2.7)

$$TSPC = \frac{\text{Target spend per campaign}}{\text{Total number of conversions}} \tag{2.7}$$

The TSPC allows us to normalize the number of conversions based on the dollar spend in a campaign. It is also a metric that has the same unit as the *cost per conversion (CPC)*, frequently used in digital advertising.

### 2.7.2 Comparison with Benchmark Approaches

We have implemented three benchmark approaches, ORTB, Mcpc, and Lin, and applied these approaches in running the three campaigns X, Y, and Z. In implementing the benchmark approaches, we have derived all model parameters following the methodologies given in the respective papers. Additionally, we have implemented the exact and greedy approach for comparison purposes. In this section, we compare the results of all five approaches to our proposed APS approach.

| | Campaign X | Campaign Y | Campaign Z | Average number of conversions per campaign |
|---|---|---|---|---|
| APS | 6621 | 6103 | 5789 | 6171 |
| ORTB | 5194 | 4807 | 4988 | 4996 |
| Mcpc | 2290 | 2167 | 1893 | 2117 |
| Lin | 3871 | 4184 | 3608 | 3888 |
| The greedy approach | 1165 | 870 | 967 | 1001 |
| The exact approach | 25431 | 18055 | 19464 | 20983 |

**Table 5. Comparison of the obtained conversions between APS and other approaches**

We have compared the outcome of the APS, and all the other benchmark approaches with respect to the number of conversions obtained with $1000 target spend for the three campaigns in Table 5. For APS the bid period is set for 5 minutes. As shown in Table 5, based on the average number of conversions obtained across three campaigns, *our APS approach performs approximately 19%, 66%, 37%, and 84% better than the ORTB, Mcpc, Lin and the greedy approaches respectively*. On the other hand, according to the exact approach (which performed 70% better than our approach), we can say that APS has considered 30% of bid requests out of the most cost-effective bid requests (because the exact approach always bid only for such bid requests) to gain its conversions.

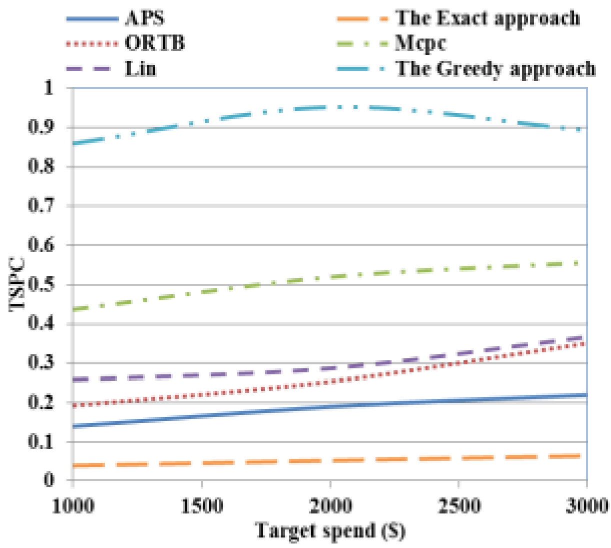


**Figure 3. TSPC vs. Target spend in different approaches**

Next, we plotted how the average TSPC values across three campaigns varied with respect to target spend in Figure 3. From Figure 3, we note that the TSPC values for APS are lower than ORTB, Mcpc, Lin and the greedy approach. Additionally, we also observe that, with the increased target spend, the difference between APS and other approaches is widened. Nevertheless, compared to the exact approach, APS has a slightly high TSPC, but the difference between the two approaches is less than 0.15. Therefore, we can conclude that the proposed *APS approach achieves a higher conversion rate with reduced spending for advertisers compared to other approaches*.

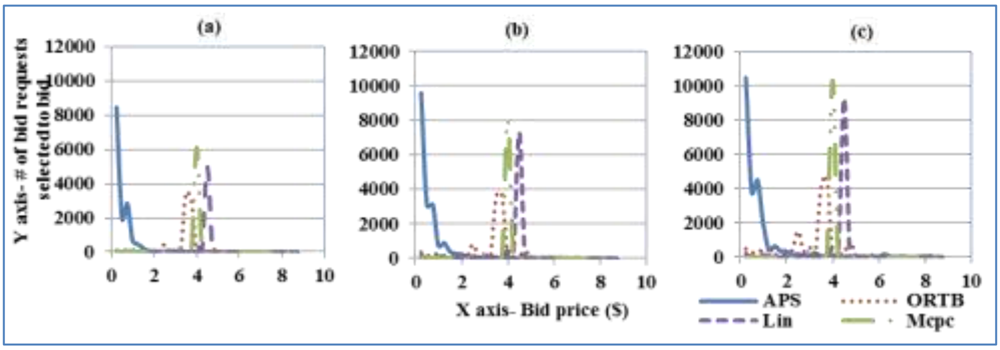


**Figure 4. The number of bid requests is selected by APS, ORTB, Lin, and Mcpc, to bid for different bid prices on when the target spend is (a) 1000, (b) 2000 and (c) 3000.**

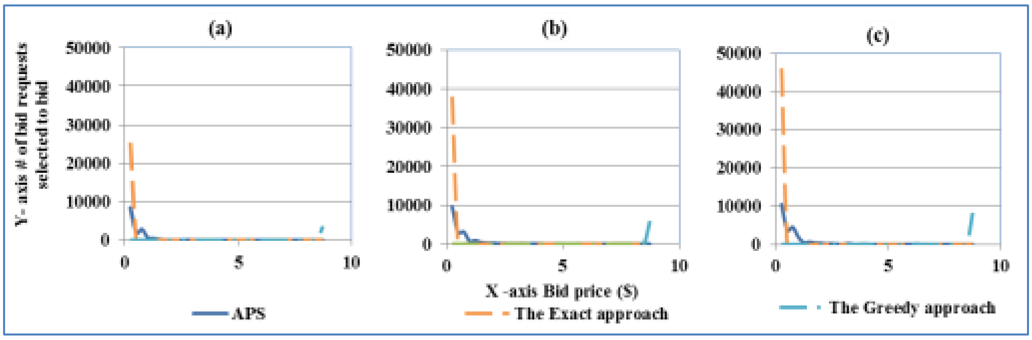


**Figure 5. The number of bid requests selected by APS, the exact approach and the greedy approach to bid for different bid prices when the target spend is (a) 1000, (b) 2000 and (c) 3000.**

Next, to better understand why the APS approach performs better than other approaches, we have compared the bidding behavior of each approach in

Figure 4 and Figure 5. In Figure 4 and Figure 5, the X-axis demonstrates the bid prices in $ and the Y-axis demonstrates the number of bid requests selected to bid by each approach. The graphs (a), (b) and (c) in both Figure 4 and Figure 5, illustrate the bidding behavior in campaign X when the target spend is $1000, $2000 and $3000 respectively. When the target spend increases, the number of bid requests selected to bid is increased. For example, in APS, it has increased from 14917 for $1000 target spend to 24125 for $3000 target spend. For ORTB the corresponding values have increased from 7827 to 15944. However, the distribution of bid prices has not changed significantly. The bid prices in ORTB range from $3.5-$3.75, for Mcpc the bid prices are around $4,  for Lin bids they are around $4.5, and for the exact approach, they are around $0.25. For APS, the average bid prices have increased from $0.25 for $1000 target spend to $1 for $3000 target spend. This demonstrates that, when the target spend is increased, APS is bidding at higher bid prices to achieve more conversions with higher winning bid prices. However, at a lower target spend, the bid price in the case of APS is decreased significantly to achieve better outcomes (higher conversions). This dynamic behavior cannot be seen in other approaches. Furthermore, as shown in Figure 4, it is interesting to see that for different target spends, APS always bids at a much lower bid price compared to other approaches. This can be further demonstrated by looking at Figure 5, where APS overlaps with the exact approach graph which always bids at the best bid prices for the advertiser. The main reason for this achievement is that the APS dynamically adopts the most recent bidding behavior by inspecting the most recent bid period and adjusts the bid price to enact the optimal bid price for the period. Therefore, combining the insights from Table 5, Figure 4 and Figure 5, we can conclude that *APS provides the improvisation to identify and bid the bid requests at a lower bid price but achieve higher conversion rates.*

We have further analyzed the bidding behavior with respect to apps that have a higher numbers of bid requests and apps that have lower average winning bid prices. For that, we followed two techniques, Type (1) and Type (2), and plotted the graphs in Figure 6.

Type (1): Sort the apps in descending order based on the number of bid requests received and select the top 50 apps that have higher numbers of bid requests. Then, based on these selected 50 apps,

- Figure 6 (a). compute the percentage of the bid requests selected to bid, out of the total number of bid requests selected to bid in each approach.
- Figure 6 (b). compute the percentage of conversions which occurred, out of the total number of conversions occurring in each approach.

Type (2): Sort the apps in ascending order based on the average winning bid price and select the top 50 apps that have lower average winning bid prices. Then, based on these selected 50 apps,

- Figure 6 (c). compute the percentage of the bid requests selected to bid, out of the total number of bid requests selected to bid in each approach.
- Figure 6 (d). compute the percentage of conversions which occurred, out of the total number of conversions occurring in each approach.

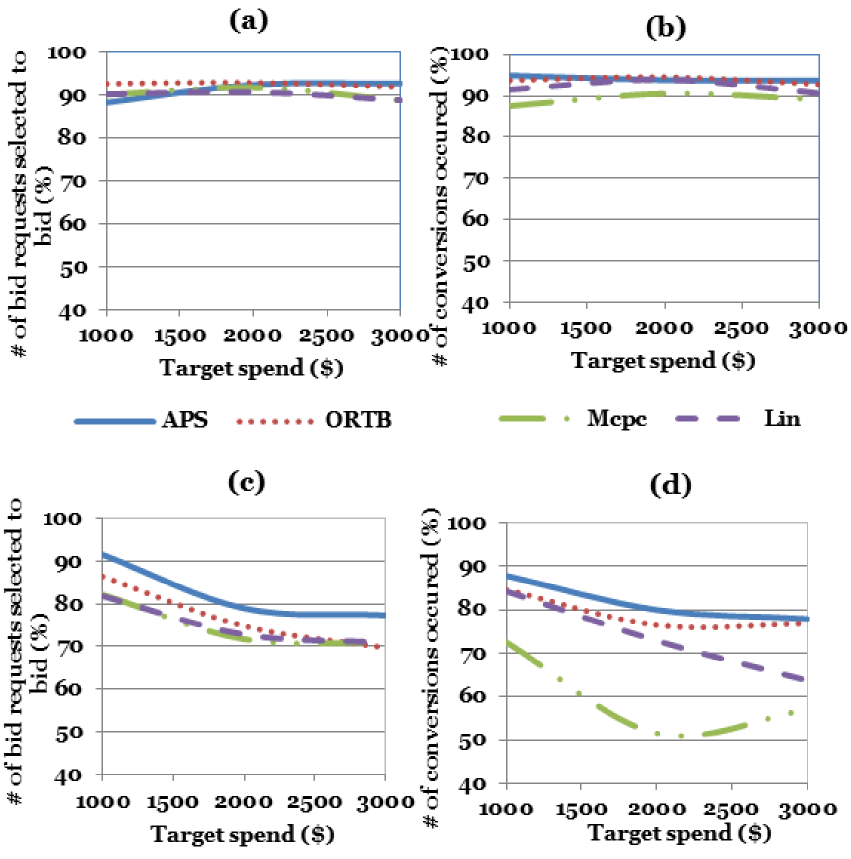


**Figure 6. Target spend vs. the percentage of the number of bid requests selected to bid (a) & (c) and target spend vs. the percentage of the number of conversions which occurred (b) & (d).**

As per Figure 6 (a) and (b), in the midst of all the related work approaches (ORTB, Lin, and Mcpc), there is no significant percentage difference in the total selection of bids and conversions occurring in Type (1) apps. However, when the apps are selected from Type (2) apps as shown in Figure 6 (c), APS bids for a higher number of bid requests than other approaches. According to Figure 6 (d), APS also achieves a higher number of conversions from the Type (2) apps. APS will specifically select the bid requests only from the apps which have a lower average winning bid price and a higher number of bid requests. Compared to the greedy approach, APS performs much better in both types of apps; because the greedy approach does not have any selection criteria for apps, it bids for bid requests based on the first-in-first-served (FIFS) basis. According to this, when the apps are selected to bid, APS gives higher priority to the apps with lower winning bid prices rather than the apps with a higher number of bid requests. In this process, the bidding constraints such as lower winning bid prices, higher conversion rates, and the relevant target audience are optimized meticulously to achieve the target goals. In summary, we can emphasize that *to achieve target goals* the *APS preferably selects higher quality apps compared to the apps with a higher quantity of bid requests.*

### 2.7.3 Sensitivity Analysis of APS

According to the metric formulated above, the objective of our analysis is to demonstrate that the proposed model can accomplish lower TSPC while achieving a higher number of conversions. According to Table 6, TSPC values for each target spend increase when the bid period is increased. We demonstrate this scenario in Figure 7. Comparing Figure 7, (a) and (b), we can determine that, when the bid period duration is small (this means a higher number of bid periods), a higher number of conversions can be achieved for a lower TSPC value.

| Campaign X | | 5 minutes | | 15 minutes | | 30 minutes | | 60 minutes | |
|---|---|---|---|---|---|---|---|---|---|
| | | TSPC | conversions | TSPC | conversions | TSPC | conversions | TSPC | conversions |
| Target spend | $ 1000 | 0.14 | 6621 | 0.21 | 4555 | 0.28 | 3879 | 0.46 | 1988 |
| | $ 2000 | 0.19 | 8961 | 0.23 | 6878 | 0.31 | 5006 | 0.51 | 2731 |
| | $ 3000 | 0.22 | 9877 | 0.27 | 7543 | 0.33 | 5766 | 0.54 | 3104 |

**Table 6. Model behavior at different bid periods**

However, when a particular bid period is considered, TSPC increases with respect to the target spend. That is because, when we allocate a higher budget for a campaign, the allocation for a particular period is also high. Since the model has a lower granularity to select applications, it will also bid for the applications with higher bid prices to maximize the number of conversions.

As per the above scenario, we can demonstrate that, when the bid period is smaller, the model accuracy will be increased. Nevertheless, when the target spend is high, it will try to achieve more conversions from the applications with the higher bid prices. The insight from this analysis is that *when an advertising campaign is running in RTB, it needs to maintain the bid period at a minimum level*.

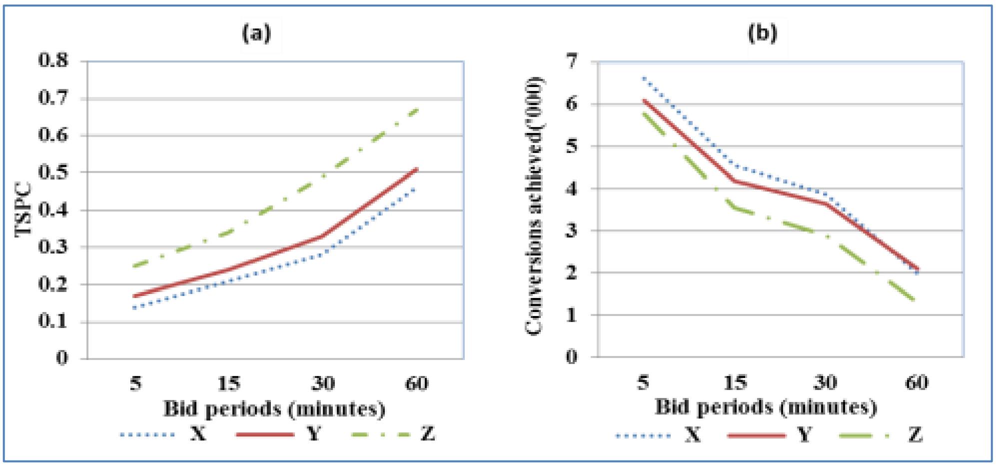


**Figure 7. (a) TSPC vs. Bid periods and (b) Conversions achieved vs. Bid periods in X, Y, and Z campaigns**

To validate the model with Eq. (2.6), we primarily look at how the model behavior would change based on the number of bid requests of the inactive applications, during the previous bid period. As a result, we test the model based on their last updated bid requests and updating them to single-bid requests and 100-bid requests. The analysis is performed for three campaigns when the bid period is 5 minutes. The results are listed in Table 7.

| | Based on last updated bid requests | | Based on single bid request | | Based on 100 bid requests | |
|---|---|---|---|---|---|---|
| | TSPC | # conversions | TSPC | # conversions | TSPC | # conversions |
| Campaign X | 0.18 | 4883 | 0.14 | 6621 | 0.19 | 4412 |
| Campaign Y | 0.21 | 4411 | 0.17 | 6103 | 0.28 | 3631 |
| Campaign Z | 0.27 | 3867 | 0.25 | 5789 | 0.32 | 3347 |

**Table 7. Model behavior based on the number of bid requests of the applications which are not active during the previous period**

As depicted in Table 7, we can get a better outcome when the number of bid requests adjusts to 1 for all the inactive applications during the previous period. This provides an added advantage: if the applications with a single bid request are selected by the optimization strategy, then based on their winning outcomes, the model can further adjust their bid prices and optimize the winning outcome for the next bid period. In parallel, when the bid price is increased by 1.5 times for an application which could not win any impressions during the previous period, the model provides two advantages. The first advantage is that its return rate will be reduced according to Eq. (2.11). This will increase the probability of bidding for another application with a higher return rate. Secondly, if it is still selected by the optimization strategy, then there is a high likelihood of winning its impressions due to the high bid price. The analysis provides insight in understanding two techniques which help to increase the model's performance; the first technique is *to assign the number of bid requests to 1 for all the inactive applications in the previous bid period* and the second technique is to *increase the bid price by 1.5 times for the unsuccessful applications in the previous period.*

In Figure 8 and 9, we illustrate how the performance of our model varies based on the two parameters $\alpha$ and $\beta$ in Eq. (2.5). We vary α between 1 and 2 in steps of 0.2. We vary $\beta$ between 0.5 and 1 in steps of 0.1. For three campaigns, we plot the TSPC values with $\alpha$ in Figure 8(a) and the number of conversions achieved with α in Figure 8(b). As shown in Figure 8, the TSPC and the number of conversions follow the opposite pattern (concave in the case of TSPC vs. convex in the case of number of conversions) when $\alpha$ is changed. The TSPC is the lowest at $\alpha$ = 1.5, at the same point at which the number of conversions is also the highest. This shows that when the winning rate is lower than 50%, the price to bid needs to be increased by 50%.

The similar behavior can be seen for $\beta$ in Figure 9. With the increase of $\beta$, the TSPC values follow a concave graph, whereas the number of conversions follows a convex graph. The TSPC value is the lowest at $\beta$ = 0.8, at which point the number of conversions is also the highest. Thus as per equation (2.5), when the winning rate is 1 (i.e. the DSP has won all its bids), the bid price needs to be reduced by 20%.

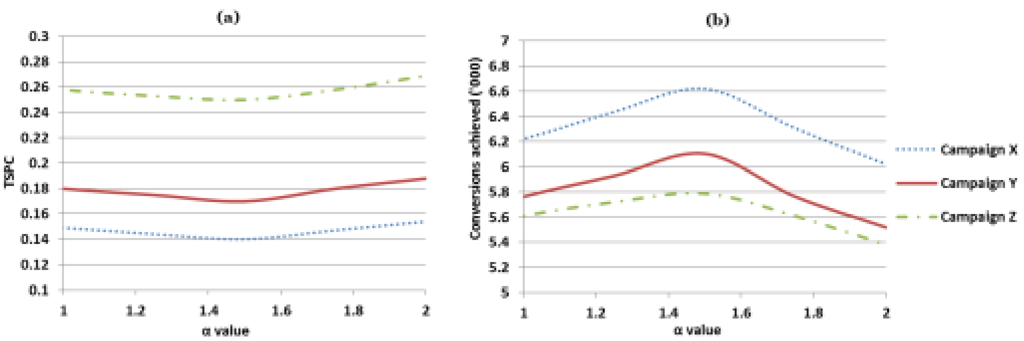


**Figure 8. (a) TSPC vs. α value and (b) Conversions achieved vs. α value in X, Y, and Z campaigns**

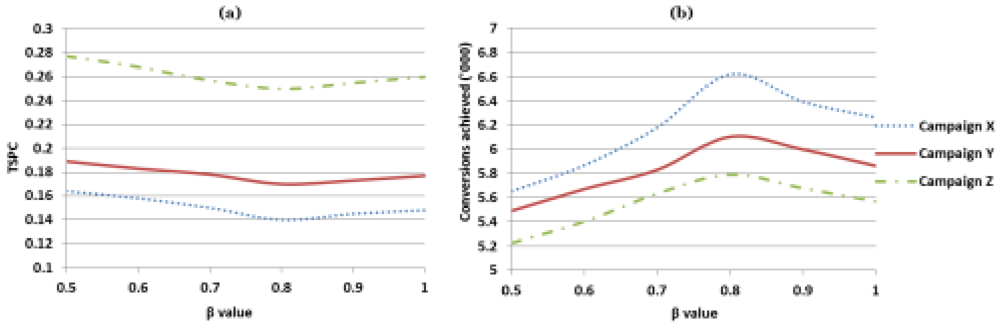


**Figure 9. (a) TSPC vs. β value and (b) Conversions achieved vs. β value in X, Y, and Z campaigns**

## 2.8 Discussion

Even though the APS approach outperforms the existing approaches, it has a few limitations as well. First, the APS approach is primarily developed following the characteristics of the target audience; it should be enhanced so that it can observe other vital characteristics like device type, type of network connection, location, mobile platform, etc. Second, the proposed solution does not emphasize any mechanism to increase the conversions from the mobile user perspective, where it only facilitates selection of the best applications which have higher conversion rates. Since this is one of the crucial issues that the online advertising is facing now (Lee et al. 2013), there is a need for more focused research on bridging this gap.

In conclusion, this study not only created a model which successfully attains a higher number of conversions for a lower target spend; it has also provided a theoretical lens which has proved empirically that using the DP approach would provide a simple but effective mechanism to achieve higher efficiency in a bidding process. The comparison with other approaches provides clear insight into the proposed approach's performance, and we have demonstrated how the parameters such as bid period and target spend can affect the performance of the approach. Additionally, due to the outstanding app selection strategy and cost-effective budget allocation and bid price adjustment strategies, APS outperforms to other benchmark approaches. To sum up, the proposed DP approach addresses the dynamism inherited in the RTB process with a novel and compelling solution, while embedding autonomous bidding decisions into the RTB in advertising.

# Chapter 3. Publisher Return Optimization in Mobile App Advertising

## 3.1 Introduction

With the explosive growth in the use of mobile phones and tablets, savvy advertisers have shifted much of their efforts and budgets to interactive devices, employing evolving technologies that can not only target highly defined consumer segments but also open a real-time bidding war for advertising display space. According to the "IAB 2013 Global Mobile Advertising Revenue Report," global mobile advertising revenue hit $19.3 billion in 2013, nearly doubling from the previous year (IAB 2013). One of the fastest growing segments within this mobile advertising market is mobile app advertising (MAA), in which ads appear on-screen while the user has the app open, like the commercial videos inserted between each game in the free "Words with Friends" app.

The MAA market, especially with free apps, is immense. Currently, among 3 million apps (Statista 2015b), 93% of apps are freely downloadable and installable (Gartner 2013). Flurry reports that consumers in the U.S. use smartphones and tablets for an average of  2 hours and 38 minutes per day and 80% of this time (2 hours and 7 minutes) is spent inside apps (Khalaf 2013).

Although there are multiple tactics for embedding revenue-generating options within an app itself, such as offering player "upgrades" in competitive video games, the preponderance of free apps means that the publishers often pursue other revenue sources: selling ad space. Because mobile devices usually carry the users' assigned identity and mobile advertising is able to deliver a strong platform for personalization based on that information, MAA has become the key source of marketing, and a significant source of revenue generation for mobile application publishers (Vallina-Rodriguez et al. 2012). Among the many forms of mobile advertising, MAA accounted for 47% of all forms of mobile advertising revenue in 2014 (LLP 2015).

Given the vast, global, fine-tuned targeting of market-segments in mobile advertising's real-time system of connecting ads with "landing pages" (the apps on which the ads show up), helping app publishers find and compete for advertising dollars has become a lucrative industry in itself. A complicated and almost instantaneous interfacing data system called real-time bidding (RTB) matches apps competing for the revenue with advertisers seeking the specific user-demographics each app is identified with. In RTB, the app publishers display their "prices" for ad space on the app and competing advertisers display their willingness to pay those prices. In a very small fraction of a second, the optimum match is selected, and the winner is the ad that appears on the user's screen.

According to Statista (2016), in 2018 mobile RTB advertising revenues will reach 6.8 billion USD, up from 100 million USD in 2013. RTB is becoming the hot ecosystem in the target based marketing in online advertising (Adikari and Dutta 2015). As a result, many ad networks (SSP, Ad exchanges, and Ad networks) connect with the RTB ecosystem via different types of customized solutions, increasing the complications of this growing field.

The global nature of mobile connectivity also complicates an app publisher's choice of an ad network. Mobile devices, especially mobile apps, offer an opportunity to reach a large (>800MM globally), diverse, global and engaged audience with new online media such as social networking, mobile games and text messengers that offer huge amounts of user-generated content. For example, an app for the game "Cricket" can be accessed by users both in India and the USA. However, the coverage of various ad networks in India and the USA will be different. Accordingly, nowadays, many app developers try to select and integrate multiple ad networks into their mobile apps (Forums.makingmoneywithandroid.com, 2013; Miracle, 2013; Stackoverflow.com, 2013; Viscuso, 2013).

But such solutions are at the app level; they are completely static and manually driven, not an optimal approach. A more efficient approach would be to select an ad network(s) for each app instance (i.e., each mobile application user), a selection which we would periodically check for continued best-fit, especially if the ad network shifts its ad campaigns. Ad networks count the number of "clicks" (number of times a user clicks on an ad) as one means of determining the revenue owed to the app publisher. Even networks providing ads with initially high click-through rates (CTR; in this study, we refer to click-return rates, or CRR: the number of times a user clicks on an ad) may eventually shift user preferences and provide lower revenues. Since the primary revenue stream for mobile apps is in-app advertising, we can increase the publisher revenue by selecting the most effective ad delivery solution(s).

### 3.1.1 Background

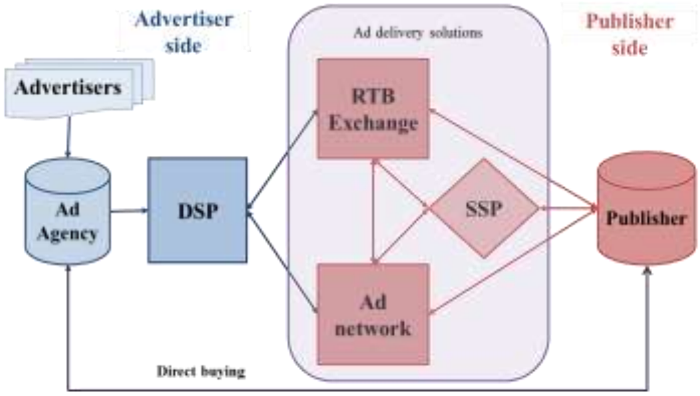


**Figure 10. The ad delivery solutions of online advertising ecosystem**

The real-time bidding exchange, as illustrated in Figure 10, connects advertisers with app publishers through two platforms (Adopsinsider.com 2013). Compared to the Figure 1, in Figure 10 we concentrate on the flow of the publisher's interactions, to publish their impressions to advertisers through Supply Side Platforms (SSP), ad exchanges, ad networks and through direct buying. SSPs provide a central management console which allows app publishers to define preferences such as setting a reserve price for a specific impression or even blocking ads from a specific advertiser (Yuan et al. 2013). According to the introductory chapter, an ad exchange is responsible for obtaining ad requests from publishers or any other third party services like SSP and deciding the winning criteria based on the auction type. The ad networks work as an ad inventory that connects advertisers to publishers that want to host advertisements. When a particular publisher integrates an ad network with his mobile application, the ad network will deliver the ads according to the ad requests that are produced at that application's app instance level.

Revenue models continue to evolve for defining how much the advertiser needs to pay the publisher. In most of the cases, publishers are paid either based on cost per thousand impressions or viewers (CPM) or cost per click (CPC) (Asdemir et al. 2012). In some studies, these costing models have been defined from the publishers' perspective as pay per view (PPV) and pay per click (PPC) respectively (Fjell 2008). However, CPM and PPV do not further track mobile users and do not assign a monetary value to consumer interactions with the advertisement. With them, therefore, it is difficult to see the outcome of the user behavior for different ad networks and advertisements types. To mitigate this, we propose a click-based approach called the Click Return Rate (CRR) which is similar to CTR but defined for a single user's click behavior, as an alternative revenue model (see section 4.1). Thus, in this study, we define *publisher's revenue* in terms of the *publisher's return* which is calculated based on the CRR. The CRR extends the PPC model by valuing each click of a particular user for a particular monetary value. Consequently, publishers need to increase the CRR to increase the returns.

### 3.1.2 Methodology

We use the framework proposed by Gregor and Hevner (2013) and Goes (2014), carrying out this study as design science research (von Alan et al. 2004). We interpret the main contextual design using the granularity theory (Henderson-Sellers and Gonzalez-Perez 2010), where we define app instance level as the fine-grained components of the app level. With that, we are able to optimize the publisher's return by determining the individual mobile user's characteristics in the app instance level.

This research falls into the *improvement* category which describes developing a new solution for a known problem (Gregor and Hevner 2013). We have applied the instance-based solution to solve the existing problem of maximizing the mobile app publisher's return. Therefore, the optimization at a higher granularity is a newer approach.

The primary design of this study is an artefact that determines the effects of both the ad network and the advertisement on the mobile app user's behavior and then uses that information to improve mobile app publisher revenue. Most of the existing solutions address this problem at the app level without including the mobile app user behaviors. In our solution we have selected the most relevant and effective attributes [6]of ad networks and advertisements (e.g. fill rate, network latency, and ad types), to determine user interactions (e.g. click, submitting a form and playing a game) on advertisements at the app instance level. We performed the analysis via a number of iterative empirical tests on a simulated dataset. In this manuscript, we present the outcome of this process and describe our proposed approach.

The rest of the chapter is organized as follows. The 3.2 section provides a detailed description of the related research on publisher level return optimization and mobile advertising. Section 3.3 provides the precise problem specifications. Section 3.4 details the solution. Section 3.5 provides details of the benchmark approaches and, following those, details of the simulation experiment and the synthetic dataset. Subsequent sections evaluate the

[6] When the number of advertisers in an ad network increases, it provides a better return to the publishers. Therefore, in terms of externalities, ad network size correlates with the ad network performance. But, yet, in practice at the granularity ad network performance only bounds to the variables that are discussed in this study.

proposed approach through a comparison with the benchmark approaches. The last section discusses the limitations and key contributions of the study.

## 3.2 Related Works

As given in chapter two, researchers have developed solutions to optimize the return for advertisers and intermediaries like ad exchanges, DSP, SSP, etc. (Yuan et al. 2012). Since this study focuses solely on publisher revenue optimization, under this section, we will discuss only studies relevant to that facet.

Existing research (Chen et al. 2014; Chickering and Heckerman 2003; Heavlin and Radovanovich 2012; Radovanović and Zeevi 2010; Shen et al. 2014) have developed publisher-return optimization solutions by either allocating impressions for different ad campaigns or mapping the impressions to advertiser’s target audience (i.e. representativeness)(McAfee et al. 2013). As a fact that, most of the existing optimization solutions are modeled at the publisher/app level. Many studies concentrate either on sponsored search auctions, in which search engines are considered to be the publishers (Agarwal et al. 2009; Jordan and Wellman 2010; Mohri and Medina 2016), or on display advertising, in which publisher revenue is maximized by allocating impressions on available ad inventories for guaranteed contracts (Amiri and Menon 2006) and across guaranteed contracts and spot (RTB) auctions (Ahmed and Kwon 2014; Ghosh et al. 2009). For example, Mohri and Medina (2016) defined the problem of selecting the reserve price for search auctions to optimize publisher revenue as a learning problem and presented a comprehensive theoretical and algorithmic analysis. Even though the study conducted by Ghosh et al. (2009) does not directly discuss publisher revenue maximization, it provides an approach to allocate the impression between guaranteed contracts and auctions via randomized bidding in a cost-effective manner. Ahmed and Kwon use (2014) contract-sizing optimization to maximize the publisher revenue. For guaranteed contracts, Pechyony et al. (2013) and Yang et al. (2010) provide a multi-objective solution to optimize both publisher revenue and advertiser objectives (i.e., representativeness and revenue). Similarly, Mungamuru and Garcia-Molina (2008) developed a revenue-sharing model for advertisers and publishers. Research on publisher return maximization assumes that the advertisements (ad inventory) that need to be placed in various ad slots is known a priori, whereas in a mobile RTB ad platform that is not the case. The placement of an ad in each and every ad slot is decided by RTB exchange instantaneously as and when the demand for an ad slot is created (i.e. a user is using an app with that ad slot in the Graphical User Interface (GUI)) based on then-available ads. Therefore, unlike browser-based display ads, in MAA, RTB is the primary ad delivery process, and we need a unique solution to address its idiosyncratic setup.

Other studies try to optimize ad allocation for more than one impression on a web page. Kwon (2011) has proposed an optimizing strategy to determine the number of impressions to display PPC advertisements on a web page. Another study optimizes publisher revenue through a variable-display frequency model which solves the NP-hard (Non-deterministic Polynomial-time hard) problem of banner advertisement scheduling (Deane and Agarwal 2012). For such scheduling problems, Menache et al. (2009) propose pricing strategies which maximize expected publisher revenue. The signaling scheme which is a randomized policy that specifies some signal to the bidders upon the choice they made is also used to maximize the publishers' revenue in a Bayesian setting (Emek et al. 2014). The signaling scheme can categorize impressions and provide the most relevant information (e.g. winning bid average, impression details and user details) to the advertiser while eliminating information asymmetry between the publisher and the advertiser (Emek et al. 2014). Roels and Fridgeirsdottir (2009) propose a dynamic optimization model to maximize the web publisher's online display advertising revenue. They dynamically select which advertising requests to accept and dynamically deliver the promised advertising impressions to viewers. One of the earliest studies, conducted by Fridgeirsdottir and Asadolahi (2007), modelled the advertising operation of the web publisher as a queuing system and computed the optimal price to charge per impression. They showed that when the number of impressions increases, the optimal price goes against the economies-of-scale intuition. In this study, they did not consider the advertisers' preferences (assumed all advertisers are homogeneous). As, in MAA, a particular app instance has only one impression to fulfill at a time, aforementioned studies cannot be applied to publisher returns optimization.

Behavioral studies on MAA can also determine certain optimization parameters. Hojjat et al. (2014) have incorporated a variety of interesting features like user-level diversity, the pacing of ads (how quickly each user is re-exposed to an ad), the reach (the number of unique individuals who should see an ad) and the ad frequency (the minimum/maximum number of times each user should see their ad) to achieve representativeness and higher CTR over a campaign period. Similarly, Yuan et al. (2013) have defined ad recency to determine how recently an advertisement has been displayed to a user.

Most of the prevailing solutions of publisher return maximization assume that publishers select advertisements either according to keywords or web page content. Yuan and Wang (2012) optimally select ads based on web page content and keywords by employing Partially Observable Markov Decision Processes (POMDPs) to maximize publisher revenue. Bilenko and Richardson (2011) describe a learning-driven client-side personalization approach for keyword advertising platforms which relies on storing user-specific information entirely within a browser cookie and predicting the user's future activity. In an MAA context, Nath et al. (2013) have proposed an advertisement recommendation system based on the mobile application's content. Despite the fact that there are mobile app recommendation engines which incorporate user behaviors in app recommendations (He et al. 2015; Natarajan et al. 2013; Shi and Ali 2012; Yan and Chen 2011), no study considers individual mobile user behaviors (with regards to the advertisements and the ad network attributes) for advertisement network selection (not particular advertisement selection) to maximize the publisher's return.

To the best of our knowledge, Lin and Hsu's 2003 work introducing the Analytic Hierarchy Process (AHP) is the only study that discusses ad network selection. Lin and Hsu identified seven key attributes (Internet media quality, business scale, advertising rates, ad management and delivery, creativity, integrated marketing planning, and service level) in selecting the appropriate ad network. Many of these parameters are subjective measures and cannot be applied in a dynamic setting like MAA for real-time ad network selection.

Even the objective measures, e.g. “number of members of allied website” and “monthly traffic of allied websites,” are static information that cannot be used in the dynamic scenario of MAA. Developing a real-time solution for ad network selection at the app instance level with such measures which can be used in practice is highly important.

## 3.3 Problem Specification

Today hundreds of mobile ad networks are continuously emerging globally, and many are also liquidating (Scocco 2014). This makes the ad network selection process very challenging for publishers. To increase their returns, publishers have to determine the most appropriate ad network(s), which perform(s) better for a particular mobile app user/app instance in a particular context. Selecting an ad network at the app instance level is more effective than selecting one globally at the publisher level. For example, assume that a publisher integrates two ad networks (not a single specific ad network like in the usual setup) such as AdMob (AdMob, 2015) and InMobi (InMobi, 2015) in his mobile application to display the advertisements to the app’s users. Now assume that AdMob’s ad campaigns better align with the current interest of a user; the mobile app user may make the most clicks on advertisements delivered through AdMob. In this situation, if all the advertisements for that particular user are obtained only from AdMob, the publisher can achieve a higher number of clicks. But the situation may be the reversed for another user—i.e., his interest matches with the type of campaigns being run by inMobi—and so for this user, all the ads should be delivered from inMobi. In such a scenario, the publisher will obtain a higher return (i.e. higher number of clicks) if instead of using a single ad network to deliver ads to all users, the publisher selects an ad network at the app instance level.

Currently, publishers do not have a way to track which ad network(s) perform(s) better for their application based on its users. Most of the time publishers follow certain criteria (Adspeed.com 2010) to select an ad network at the app level (but not at the individual app instance level), but the criteria are not invariably accurate for all publishers and their respective app users. The publisher or the app level criterion of selecting an ad network does not work well at the individual level. The problem becomes further challenging as the ad networks do not perform equally well every time, everywhere. Table 8 illustrates how ad networks behave in different countries based on the data retrieved from a mobile DSP about three ad networks in four different countries.

From Table 8, if we select only Ad network 1 we can achieve maximally 8478 clicks ($number\ of\ users\ \times fill\ rate\ \times CTR$), but Ad network 1 does not provide the highest number of clicks for each and every country. For example, for India and the UK, Ad network 3 provides the highest number of clicks, but for Singapore, Ad network 2 provides the highest. This is due to the dynamism entrenched in the MAA ecosystem which discussed under the challenges in the introduction chapter. In most cases, advertisers do not advertise in the same ad network consistently because they have many choices of ad networks. Also, an ad network itself has a certain set of constraints such as network latency, advertisement pacing, frequency, etc., which change the behavior of the ad network (i.e., fill rates, types of advertisements and CRR) from time to time and across locations (Vallina-Rodriguez et al. 2012). In addition to the ad network dimension, the advertisement dimension also plays a key role in determining the publisher’s return. (In the next section, we develop the model that maximizes publisher return).

| Country | Users | Ad network 1 | | | Ad network 2 | | | Ad network 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Fill rate % | CTR % | Clicks | Fill rate % | CTR % | Clicks | Fill rate % | CTR % | Clicks |
| India | 100,000 | 50 | 0.5 | 250 | 30 | 0.3 | 90 | 60 | 0.8 | 480 |
| USA | 300,000 | 70 | 2.5 | 5,250 | 50 | 2.0 | 3,000 | 50 | 1.5 | 2,250 |
| Singapore | 80,000 | 65 | 1.4 | 728 | 80 | 2.5 | 1,600 | 70 | 1.8 | 1,008 |
| UK | 150,000 | 75 | 2.0 | 2,250 | 60 | 1.5 | 1,350 | 80 | 2.0 | 2,400 |
| Total Clicks | | 8,478 | | | 6,040 | | | 6,138 | | |

**Table 8. An example of ad network behaviors at the country level**

Based on the above reasoning the component architecture of the proposed solution is illustrated in Figure 11. As illustrated in Figure 11, different app instances of an app publisher will connect with many different ad delivery networks. In the proposed solution, an app instance will run our proposed algorithm to switch from one ad network to another. Each app instance will store the behavior of the current ad network and the past performance of the previously used ad networks. (The details of the information stored in each app instance are described later in the chapter in Figure 14.) The proposed algorithm will periodically run in each app instance, based on this data, to select the ad network for the next time period. In the next section, we formally describe the data and the problem to solve with the proposed algorithm.

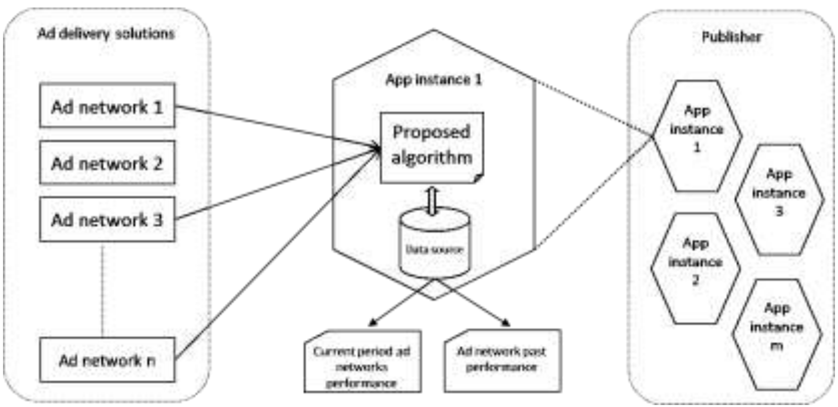


**Figure 11. High-level component architecture of the proposed solution**

## 3.4 Solution Formulation

Before probing into the details of the proposed model, Table 9 provides the notations for the different parameters that are derived from the different attributes of both ad network and advertisement dimensions.

| | |
|---|---|
| *Indices* | |
| $j$ | index for the received advertisements $j = 1, \ldots, j'$ (duplicate elements are also included in this set) |
| $k$ | index for ad networks $k = 1, \ldots, k'$ |
| $g$ | index for advertisement types $g = 1, \ldots, g'$ |
| $l$ | index for user locations $l$ |
| $t$ | index for an app period $t = 1, \ldots, t'$ |
| $i$ | index for an app instance $i = 1, \ldots, i'$ |
| $\bar{t}$ | index for an app period which has the most recent behavior of a particular ad network |
| $b$ | index for click behavior $b = click, noclick$ |
| *Parameters* | |
| $F_{ktl}$ | fill rate for ad network $k$ in app period $t$ in location $l$ |
| $L_{ktl}$ | average network latency for ad network $k$ in app period $t$ in location $l$ |
| $L'_{jktl}$ | network latency for advertisement $j$ on ad network $k$ in app period $t$ in location $l$ |
| $Q_{ktl}$ | average frequency for ad network $k$ in app period $t$ in location $l$ |
| $C_{ktl}$ | CRR for ad network $k$ in app period $t$ in location $l$ |
| $N_{ktl}$ | ad network effect for ad network $k$ in app period $t$ in location $l$ |
| $M_{ktl}$ | advertisement effect for ad network $k$ in app period $t$ in location $l$ |
| $A_{ktl}$ | total number of ad requests sent to ad network $k$ in app period $t$ in location $l$ |
| $J_{ktl}$ | total number of ad requests fulfilled by ad network $k$ in app period $t$ in location $l$ |
| $\tilde{J}_{ktl}$ | number of unique advertisements delivered by ad network $k$ in app period $t$ in location $l$ |
| $R_{ktl}$ | total number of ads clicked on ad network $k$ in app period $t$ in location $l$ |
| $D_{ktl}$ | average advertisement display duration for ad network $k$ in app period $t$ in location $l$ |
| $U_{ktl}$ | app usage duration with ad network $k$ during app period $t$ in location $l$ |
| $j'$ | total number of received advertisements |
| $k'$ | total number of ad networks |
| $g'$ | total number of advertisement types |
| $i'$ | total number of app instances |
| $G$ | set of combination of advertisement's attributes $g$ (e.g., banner ad x top x political) |
| $B$ | set of user click behaviors $b$ (ex: click, no click) |
| $P_g$ | probability of the number of clicks on advertisement type $g$ out of all the clicks on all the advertisement types $g'$ |
| $\tilde{P}_g$ | probability of the number of clicks on advertisement type $g$ out of all the received advertisements of advertisement type $g$ |
| *Decision variable* | |
| $W_{ktl}$ | Number of ad requests selected to send to ad network $k$ in app period $t$ in location $l$ |

**Table 9. Notation for the model descriptions**

In this study, we propose a solution to maximize the publisher's return by selecting the right ad delivery network(s) according to interests of a specific mobile app user. Therefore, the proposed model is formulated for each app instance.

Since we seek to determine the optimal ad network(s) which provide the maximum return to the publisher, we must check the performance of the existing ad networks over a certain elapsed time period. We define this time period as the *app period,* during which the selected ad network(s) is (are) are obtaining the advertisements. At the end of the app period, we run the proposed model again with the updated data of the current ad network performance and other ad networks' past performance data and select the ad network which can provide a better return to the publisher for the next app period. The app period could be 10 minutes, 30 minutes, hourly, daily or so on. Thus, app usage duration could be either less than (e.g., app period is hourly and app usage duration is only 20 minutes) or greater than (e.g., app period is only 10 minutes and app usage duration is 20 minutes) the app period. In the proposed approach we assume that in the current period, the behavior of both the user and the ad network is the most apposite behavior that we can predict for the next period. As demonstrated in Table 8, an ad network's performance varies across countries. Currently, most of the DSPs run their advertisement bidding algorithms at the country level (Adikari and Dutta 2015), so in this study, we similarly consider the location of each app instance at the country level, though further granularity about the location is possible. In this study, we consider the performance of an ad network in an app instance based on the app period and location combination. To model the performance of an ad network in an app instance, we first developed three metrics (CRR, ad network effect and advertisement effect). Next, we developed a math programming model to maximize the overall performance of advertisements in an app instance based on the selected ad networks.

### 3.4.1 Click Return Rate

Currently, in online advertising, the CTR is the key metric to measure click behaviors. It defines the number of users who have clicked on an ad rather than the number of users who have viewed the ad (Shankar and Hollinger 2007). Since CTR is defined for a particular advertisement, we need a unique measurement to capture a specific user's click return. Therefore, in this study, we define CRR as the likelihood of clicks return by a single user with respect to the total number of advertisements displayed to the user. In Table 9, we use the notation $C_{ktl}$ to define the CRR in the proposed context.

$$C_{ktl} = \frac{R_{ktl}}{J_{ktl}} \tag{3.1}$$

### 3.4.2 Ad Network Effect

To understand the behavior of the ad network at the app instance level, we look at its attributes and estimate the effect on the publisher's return (ad revenue). Below we define a few of these attributes and estimate the total ad network effect on the publisher's return. We have selected three key ad network attributes (fill rate, network latency, and ad frequency) and eliminated the other attributes such as ad pacing and ad recency, as they may correlate with the selected variables and bring the endogenous explanatory variables to the model.

#### 3.4.2.1 *Fill Rate*

The fill rate defines the likelihood that an ad from a particular ad network fills an ad request from the app. In Table 9, we have defined the fill rate with respect to a particular app period and a location. In practice, a particular ad network should have at least a 90% average fill rate (Bea 2013). However, at the app instance level, the individual fill rate can be much lower depending on the user's context such as country and time of the day. For example, a sample mobile game application has a higher fill rate for users in Singapore and much lower fill rate for users in India during the same time of the day. We define fill rate $F_{ktl}$ using Eq. (3.2):

$$F_{ktl} = \frac{J_{ktl}}{A_{ktl}} \qquad (3.2)$$

#### 3.4.2.2 *Network Latency*

In online advertising, network latency means the delay (in milliseconds) in loading an advertisement into an impression (Weide 2011). Just a small amount of latency such as 100 milliseconds can cost publishers and advertisers thousands of views, clicks, and conversions (Dyn.com 2013). A delivery delay might occur more frequently when the same advertisement is delivered to multiple publishers. The ad network's servers might overload as a result and delay the delivery of the requested advertisement. Since the network latency can vary, we estimate an average value of network latency as shown in Eq. (3.3).

$$L_{ktl} = \frac{\sum_{j=1}^{j'} L'_{jktl}}{A_{ktl}} \qquad (3.3)$$

#### 3.4.2.3 *Ad Frequency*

Ad frequency is used in online advertising to define how many times the same advertisement is displayed to a user (Hojjat et al. 2014; Yuan et al. 2013). In some circumstance, ad networks deliver the same advertisement repeatedly to the same user. When the pacing of an ad is high, then the ad network could have a higher frequency because the number of unique ads delivered decreases. Eq. (3.4) illustrates the calculation of ad frequency.

$$Q_{ktl} = \frac{J_{ktl}}{J'_{ktl}} \quad (3.4)$$

Using the above-discussed parameters, we can estimate the ad network's effect on the publisher's return. The publisher's earning potential is proportional to the number of clicks, i.e., CRR. When the fill rate is high, a higher number of advertisements will be received, and the CRR will increase. Therefore, the fill rate has a positive effect on the CRR, and thus the publisher's return. However, if the network latency is high, it will reduce the possibility that the user will click on the advertisement. Similarly, when the same advertisement is displayed to a single user repeatedly, the user's interest in that particular advertisement will decline, and, as a consequence, so will the publisher's return. Therefore, we maximize the fill rate and minimize the network latency and the ad frequency parsimoniously. Following the above discussion, we define the Eq. (3. 5) to calculate the ad network effect on the publisher's return.

$$N_{ktl} = \frac{F_{ktl}}{L_{ktl} \times Q_{ktl}} \ , L_{ktl}, Q_{ktl} > 0 \quad (3.5)$$

### 3.4.3 Advertisement Effect

Advertisement effect measures user interactions of advertisements. If an ad network can deliver a higher number of advertisements aligning closely with users' interests, the close fit will lead to a higher number of clicks and thus higher returns to the publishers (Cheng et al. 2012) through increased CRR. To measure the advertisement effect we can consider several attributes of an advertisement. Even though in this study we only consider three attributes (ad format, ad position, and ad genres types), the proposed model is structured so that other advertisement attributes—such as the ones proposed by Rodgers & Thorson (2000) and Lin & Chen (2009)—can be incorporated.

Advertisement attribute can take one of the several possible values. For example, an ad format can take values such as banners, video ads, interstitial, etc. Therefore, when we measure the captivation of a particular attribute, we look at which value of the advertisement attribute is preferred more by the user and how many advertisements with that attribute value are delivered by the ad networks. The likeability of a particular advertisement type is estimated based on the number of clicks a particular format receives compared to other formats. This becomes complex when there are multiple advertisement attributes to consider because the user captivation depends on a combination of multiple ad attributes. This can be modelled as a discrete decision tree using all the attributes for a particular ad network. Such a decision tree can be used to find out various advertisement attributes such as ad format, ad position and ad genres with their respective values (ad format: banner ad and interstitial ad), and click behaviors. Fig. 12 illustrates a sample decision tree for three attributes.

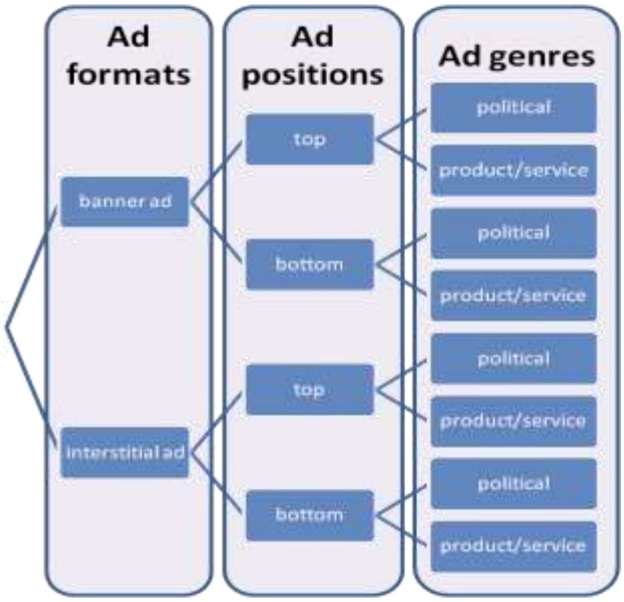


**Figure 12. Sample decision tree of advertisement attributes**

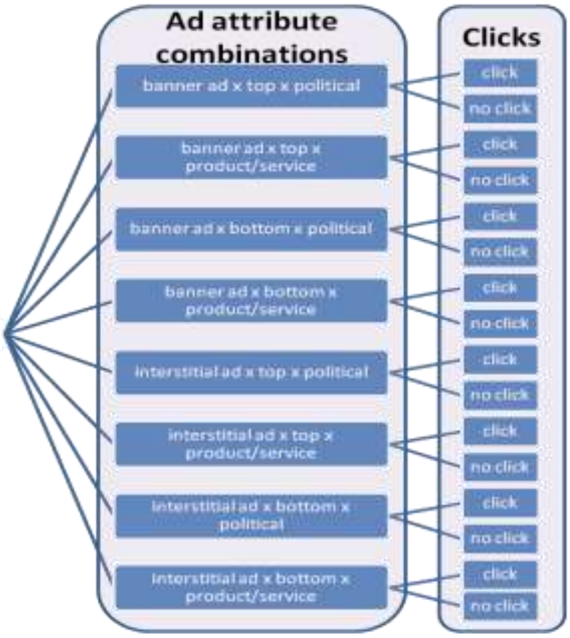


**Figure 13. Sample decision tree with advertisement types and click behaviors**

Even though in Figure 12 we illustrate only three attributes, a tree can be extended to any number of attributes that affect a user's advertisement clicks. Using the decision tree, we can evaluate how different advertisement types (i.e., combinations of advertisement attribute values) affect the user's clicks. For that, we extend the decision tree with the click behaviors and see how the clicks are distributed among different advertisement types. We append the number of clicks at the leaf level of the tree. For example, Figure 13 includes the relevant advertisement types and click behaviors. To understand what advertisements are delivered and how the clicks are distributed across different advertisement types, we compute what is called "conditional entropy." In information theory, the conditional entropy (or equivocation) is defined as a mechanism that quantifies the amount of information needed to describe the outcome of a random variable given that the value of another random variable is known (Kenett and Preis 2012). Similarly, given that the users' click behavior is known, we want to determine the preferred advertisement types. We compute the average specific conditional entropy using Kenett and Preis's $H(G_{ktl}|B_{ktlb=click})$ for user click behavior using Eq. (3.6):

$$H(G_{ktl}|B_{ktlb=click}) = -\sum_{g=1}^{g'} P_g\left(G_{ktlg}, B_{ktlb=click}\right) \log P_g\left(G_{ktlg}\middle|B_{ktlb=click}\right) \qquad (3.6)$$

Based on the click distribution among advertisement types we calculate the specific entropy. For example, consider the given sample of advertisement types in Figure 13. If the total number of clicks obtained from all the advertisement types is 1000 and the total number of clicks obtained by the first advertisement type (i.e. banner x top x political), is 10, then $P_g\left(G_{ktlg=(banner\ x\ top\ x\ political)}, B_{ktlb=click}\right) = 0.01$.

As in Eq. (3.6), we do not account for the total number of particular types of advertisements that are delivered and how many of them are not clicked; this turns out to be a drawback to explain the complete click behavior. Assume that in the earlier example, the ad network has sent 50 advertisements of the same advertisement type (banner x top x political). The user has not clicked 40 advertisements, but this absence of action is not reflected in the Eq. (3.6). To overcome this limitation, we provide the probability of a click for an ad type as a weight for each ad type in the specific entropy calculation. Thus we define, $\tilde{P}_g\left(B_{ktlg=(banner\ x\ top\ x\ political)b=click}\right) = 0.2$. The enhanced Eq. (3.7) of weighted specific entropy calculation is as follows.

$$H'(G_{ktl}|B_{ktlb=click}) = -\sum_{g=1}^{g'} \tilde{P}_g\left(B_{ktlgb=click}\right) P_g\left(G_{ktlg}, B_{ktlb=click}\right) \log P_g\left(G_{ktlg}\middle|B_{ktlb=click}\right) \quad (3.7)$$

According to our context, the entropy value has two boundaries related to two scenarios. In both scenarios, to calculate the boundary condition, we assume that all the delivered advertisements for each ad type are clicked.

*Case1:* the ad network delivers all the advertisements of an advertisement type which has the highest number of clicks. In Figure 13, if the user has the highest interest in the first advertisement type (i.e. banner x top x political), then the ad network will deliver only advertisements related to that advertisement type, and, following that, the user will click all the advertisements, i.e., $P_g\left(G_{ktlg=(banner\ x\ top\ x\ political)}, B_{ktlb=click}\right) = 1$. In this case, we get the weighted lowest entropy value, zero.

*Case 2:* the clicks are uniformly distributed among all of the advertisement types. Considering Figure 13, the ad network delivers the same number of advertisements for all 8 types of advertisements and the user will click all the advertisements, i.e., $P_g\left(G_{ktlg=(banner\ x\ top\ x\ political)}, B_{ktlb=click}\right) = {}^{1}\!/_{8}$.

In this case, according to Eq. (3.7) we get the weighted highest entropy value(s), $-log\frac{1}{g'}$.

We assume that when advertisers deliver more ads of types that interest the users, the CRR will improve and the publisher's returns will increase. We use the reciprocal of entropy value as a proxy for the effectiveness of ad networks' advertisement delivery. A lower entropy value reflects higher advertisement effectiveness. To overcome the zero boundary condition (as described in case 1 above), we transformed the entropy value by adding 1 to the average conditional entropy value before it is inverted. So we can estimate the advertisement effect using Eq. (3.8),

$$M_{ktl} = \frac{1}{H'(G_{ktl}|B_{ktlb=click})+1} \tag{3.8}$$

### 3.4.4 Ad Network Performance Utility

Based on the ad network effect, the advertisement effect and the CRR of an ad network, we define the ad network performance utility (*APU*) as

$$APU_{ktl} = N_{ktl} \times M_{ktl} \times C_{ktl} \ , \quad C_{ktl} > 0 \tag{3.9}$$

The *APU* is defined as a multiplicative function assuming that the three components of *APU* are independent of each other. Also, we can annotate each metric as a probability of the behavior in each effect. For an example, $C_{ktl}$ defines the probability of click returns by a user with respect to a particular ad network. Based on the Probability theorem, when two events are independent we can find the probability of occurrence of both events through the multiplication rule. According to the theorem, to get the overall behavior of the ad network effect, advertisement effect, and the user click return in *APU*, we multiply all the metrics.

When a user receives advertisements from an ad network with a higher *APU*, then there is a higher chance of a higher fill rate, lower ad frequency and ad latency, more advertisements which interest the user, and a higher CRR, all leading to a higher return for the publisher. Every time an ad network is selected we estimate the latest *APU* value of that ad network at the end of each app period with respect to its location. After the delineation of the most recent performance data of each ad network, we can designate as optimal the ad network which has the highest *APU* value and compute how many ad requests should be sent to that ad network in the next app period. In this process, to maximize the publisher's return, the ad network should be successful at receiving a higher number of clicks from a lower number of ad requests. Thus, we try to minimize the number of ad requests that are sent to the optimal ad network based on the following assumptions and constraints.

Assumption 1: *If user behavior remains unchanged* [7] *in the two consecutive periods, we need to display at least the same number of advertisements as in the current app period (t).*

*Constraint 1:* According to assumption 1, we cannot reduce the number of ad requests that are sent to an ad network than the number of advertisements displayed in the current period (*t*). As not all the ad requests will be fulfilled due to the fill rate of ad networks, we get the constraints in Eq. (3.10).

[7] Even though we assume that the user behaviour will be unchanged within a shorter period, for a longer period the user behaviour has a significant differentiation. To address this differentiation, we consider an incremental adaptation strategy which projects the previous period's actual user behaviours in the next period and so on.

$$W_{k(t+1)l} F_{k\bar{t}l} \geq \sum_{k=1}^{k'} J_{ktl} \tag{3.10}$$

Assumption 2: *If user behavior remains unchanged in two consecutive periods, in the next period we can display advertisements as the same period of app usage in the current period (t).*

*Constraint 2:* According to the assumption 2, the total time taken to display the advertisements, including the network latency is equal or lesser than the current period's total app usage (See Eq. (3.11)). Here we consider the network latency for both successful (advertisement loading time) and unsuccessful (waiting time) ad requests.

$$W_{k(t+1)l}(F_{k\bar{t}l} D_{k\bar{t}l} + L_{k\bar{t}l}) \leq \sum_{k=1}^{k'} U_{ktl} \tag{3.11}$$

*Constraint 3:* The number of ad requests selected to send to a particular ad network must always be a positive integer because the total number of ad requests could not have decimal values (See Eq. (3.12)).

$$W_{k(t+1)l} \in \mathbb{Z}^{*} \tag{3.12}$$

Assumption 3: *Due to the dynamism in MAA, an ad network will not perform in the same way throughout the app period.*

*Constraint 4:* As given in assumption 3, sometimes the selected ad networks may fail to satisfy the predicted fill rate within a particular app period. In such situations, we need to keep track of a set of alternative ad networks. The alternative ad networks should have the next best *APU* values. Due to computational and management complexities, we cannot keep track of a large number of alternative ad networks; it should be a limited number. Thus, we have used constraint 1 and 2 with constant α to create Eq. (3.13 and 3.14) constraints. The constant α needs to be heuristically determined for us to know how many alternative ad networks to keep track of; that is, if $\alpha = 1$ we keep track of only one alternative ad network, if $\alpha = 2$, we keep track of two alternative optimal ad networks, so on and so forth.

$$\sum_{k=1}^{k'} W_{k(t+1)l} F_{k\bar{t}l} \geq \alpha \sum_{k=1}^{k'} J_{ktl}\ ,\ \alpha \in \mathbb{Z}^*,\ 1 \leq \alpha \leq k' \quad (3.13)$$

$$\sum_{k=1}^{k'} W_{k(t+1)l} (F_{k\bar{t}l} D_{k\bar{t}l} + L_{k\bar{t}l}) \leq \alpha \sum_{k=1}^{k'} U_{ktl}\ ,\ \alpha \in \mathbb{Z}^*, 1 \leq \alpha \leq k' \quad (3.14)$$

These constraints will facilitate the adaptation of the model to the dynamic nature of the ad network behavior and user behavior. That is when an ad network fails to generate advertisements to fulfill ad requests in a particular app period, the model can decide when and which ad network select next to send ad requests.

Using the minimization function (See Eq. (3.15)) we select the optimal ad networks which will provide higher ad network effect, advertisement effect and CRR and a lower number of ad requests to be generated. The selected optimal ad networks will lead to a higher publisher return through the improved CRR at the app instance level.

$$Minimize\ \sum_k \frac{W_{k(t+1)l}}{N_{k\bar{t}l} \times M_{k\bar{t}l} \times C_{k\bar{t}l}} \quad (3.15)$$

The model is formulated in such a way that, based on the decision of the model, we can send the ad requests to the chosen ad network until the fill rate threshold is satisfied. However, when the fill rate threshold fails (i.e., the ad network has exhausted its advertisement set), we can select the next best optimal ad network from the model results and proceed with the ad request generation to that network.

In the next section, we describe the benchmark approaches (some of which are existing approaches on ad network selection by the publishers) to which our approach will be compared in subsequent sections.

## 3.5 The Benchmark Approaches

For two main reasons, it is impractical to compare the proposed approach with the prevailing approaches described in the academic literature (Deane and Agarwal 2012; Menache et al. 2009; Roels and Fridgeirsdottir 2009). First, in the RTB ecosystem, our proposed approach is applied at the individual app instance level, and those prevailing approaches are designed to apply at the publisher level. The parameters and attributes available at the publisher level are different from those available and considered in this research at the individual instance app level. Therefore, re-implementation of those models within the proposed context is not possible. Second, the traditional data mining techniques cannot be applied at the app instance level because sample sizes are smaller, observations are non-independent, and there exists the additional overhead of running such algorithms at the device level, where the battery power is precious.

Therefore, instead of comparing the proposed approach with some of the existing approaches which are orthogonal to the proposed approach, we evaluate the effectiveness of the proposed model with three approaches: using (1) the single ad network approach (which is the current industry norm in ad network selection for an app), (2) the First-Come-First-Serve (FCFS) approach (which can be a simple implementation of ad network selection at the individual app instance level) and (3) the exact approach (which cannot be implemented because of unavailability of future data, but can be used as a boundary condition for the best achievable situation).

### 3.5.1 The Single Ad Network Approach

The Single ad network approach has become the de facto norm in practice for selecting an ad network when a mobile application is developed. According to Ruiz et al. (2014), most of the mobile app publishers obtain ads from a single ad network. Publishers select the ad network that is expected to produce a higher revenue. As the pay per clicks (PPC) and pay per view (PPV) values are driven by the market, there is no significant difference in these values across different ad networks. Therefore, publishers will primarily consider the ad network which provides the advertisements that have the potential to offer a higher number of clicks (similar to the Ad network 1 in Table 8) or, in other words, that match with the interests of the users of the app.

For example, if a mobile app developer assumes that the ad network InMobi can provide a higher number of clicks than the other ad networks, he will integrate the InMobi platform in his mobile app. As a result, each app instance of his mobile application, every time and everywhere, will receive advertisements only from the InMobi ad network.

Motivated by this notion, we employ the single ad network algorithm to determine the total number of clicks that a publisher can obtain from an ad network across all the app instances within a single day. The selection process for the Single ad network is detailed in Algorithm 4. In the first stage (algorithm-steps 1 and 2) for a certain period, we send ad requests for all ad networks uniformly and compute the average CRR for each ad network. In the second stage (algorithm-step 3) we select the ad network which has the highest CRR. In the third stage (algorithm-steps 4 to 5), i.e., during execution, we send ad requests only to the selected ad network. In the last stage (algorithm-steps 6 to 13) we compute the total number of clicks obtained from the selected ad network.

**Algorithm 4: The Single Ad Network Algorithm**

Start
Step 1. Send ad requests for to all ad networks ($\forall k$) uniformly
Step 2. Compute and list the average $C_k, \ \forall k$
Step 3. Find $z$ using $argmax_k(list\ of\ C_k),$ where $z \ \in k$
Step 4. while (each ad request)
Step 5. send ad requests to $z$
Step 6. if (ad response true)
Step 7. display the advertisement
Step 8. if (click true)
Step 9. add a click to total clicks
Step 10. End if
Step 11. End if
Step 12. End while
Step 13. display total clicks $\forall l$
End

### 3.5.2 The First-Come-First-Serve (FCFS) Approach

According to our proposed approach, we have designed a simple FCFS approach to optimize the publisher return. In the FCFS approach (Algorithm 5), publishers will send any particular ad request to all the listed ad networks and accept the advertisement only from the first responding ad network while rejecting the rest of the responses. The drawback of this approach is twofold. First, the publisher does not generate revenue according to the number of clicks but rather according to the number of impressions" (i.e., the system counts impressions but not clicks, even though the clicks indicate a higher monetary value compare to the impressions) Those ad networks with lower network latencies will find a higher opportunity to display advertisements. Second, over the long term, this kind of approach is detrimental to the publishers, because ad networks may ban a publisher who continuously rejects their advertisements.

**Algorithm 5: The FCFS Algorithm**

Start
Step 1. while (each ad request)
Step 2. send ad requests to $\forall k$
Step 3. if (ad response true)
Step 4. display the advertisement
Step 5. reject other responses
Step 6. if (click true)
Step 7. add a click to total clicks
Step 8. End if
Step 9. End if
Step 10. End while
Step 11. display total clicks for $\forall l, i$
End

### 3.5.3 The Exact Approach

In the Exact approach, we calculate the total number of clicks when the ad networks' future performance is known before the fact (which is not possible in reality, but we use this as an upper-bound for the publisher's revenue). Here (Algorithm 6) in step 1 we calculate the total number of clicks obtained by each ad network for each app period, app instance and app location combination. In step 2, we find the ad networks that perform the best for each combination of app period, app instance and app location.

**Algorithm 6: The Exact Algorithm**

Start
Step 1. Calculate $TotCLK_{ktli} = C_{ktli} \times F_{ktli} \times A_{ktli}, \forall k, t, l, i$
Step 2. Find $z_{tli}$ using $arg\ max_k (TotCLK_{ktli}),\ where\ z_{tli} \in k$
End

## 3.6 Experiment and Dataset

With MAA, ad networks do not provide data at the app instance level. For example, Google AdMob (AdMob, 2015) provides only general reports with an overall value such as the total revenue earned and the number of clicks across all mobile app users for a specified duration. Therefore, to evaluate the proposed model we have implemented a Monte Carlo simulation (Gilks 2005). The high-level architecture of the simulation process and a sample transaction at the app instance level are depicted in Figure 14. According to Figure 14, first, when the user simulator launches an ad request, the user identification (app instance), location, and time of the transaction initiation are captured and stored in a database row (as shown in Figure 14). Second, the application simulator sends the ad request to the selected ad network simulator and stores the ad network ID. Third, if the selected ad network simulator delivers an advertisement, the advertisement ID, the type of the advertisement, the time taken to deliver the advertisement to the application simulator and the fill status are stored as the remaining columns of the previous database row. Lastly, if the user simulator clicks the advertisement, the click status will be stored as 1, otherwise 0.

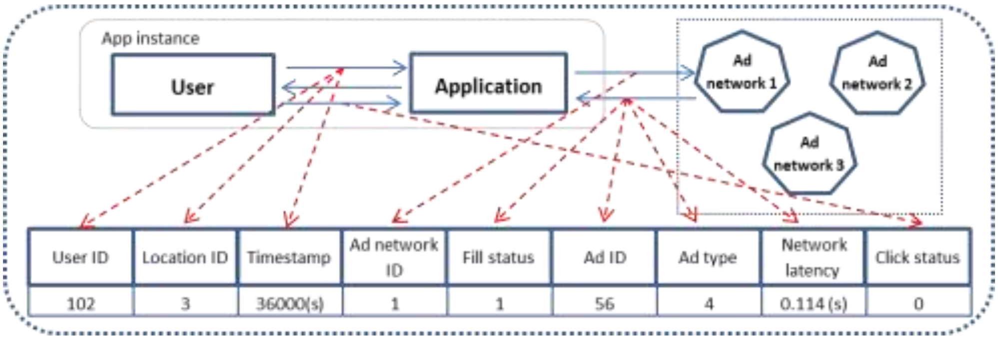


**Figure 14. The simulation process and a sample transaction at the app instance level**

The APU values are generated in-line with the actual value ranges. For example, advertisement display time ranges between 43 seconds and 50 seconds (Statista 2014) and ad network latency ranges between 100 milliseconds and 1 second (Adopsinsider.com 2013; Dyn.com 2013). In addition, the different user and network behaviors have been introduced to adapt the real world user and ad network behaviors into the simulation. The range of user behaviors is manipulated by changing the duration of the app usage, the CRR, captivation of advertisement types, etc. Also, the ad network behaviors are manipulated by changing the ad network latency, the fill rate, the CRR, the types of advertisements, and the ad frequency of each ad network. Together with, we must incorporate the constraints which consider the user and the ad network behaviors together. For example, when a particular ad network has a lower frequency, the relevant app instances should have a higher CRR; when the fill rate of an ad network is lower, the CRR is also lower; when the ad network latency is lower, the number of advertisements displayed to the user is higher; and the ad network which delivers more user-relevant ad types gets a higher CRR. Following these constraints, we have generated 15,936,000 ad requests and their respective user interactions. The range of each attribute in the dataset is given in Table 10.

<table>
<tr><td>Indices</td><td>Range</td><td>Attributes</td><td>Range</td></tr>
<tr><td>Ad networks</td><td>1 to 50</td><td>Ad types</td><td>8</td></tr>
<tr><td>Advertisements</td><td>1 to 100 for each Ad network</td><td>Fill status</td><td>0 or 1</td></tr>
<tr><td>Timestamp</td><td>0 to 86400</td><td>Ad display time (S)</td><td>43 to 50</td></tr>
<tr><td>App period durations (minutes)</td><td>5, 10, 30, 60</td><td>Latency (S)</td><td>0.2 to 1</td></tr>
<tr><td>App instances (Users)</td><td>1 to 100</td><td rowspan="2">Click status</td><td rowspan="2">0 or 1</td></tr>
<tr><td>Locations</td><td>1 to 5</td></tr>
</table>

**Table 10. The ranges of the simulated data**

To estimate the gain of the model, we define an evaluation metric based on the Fill rate and the CRR, called Number of Ad Requests per Click (ARPC):

$$ARPC = \frac{1}{Fill\ rate \times CRR} \tag{3.16}$$

A lower value of *ARPC* indicates that on average fewer ad impressions are required to obtain a single click. A decrease in the value of *ARPC* leads to an increase in the publisher's return. The descriptive statistics of the data for the fill rate, *CRR*, ad frequency, network latency and *ARPC* are given in Table 11. The simulation model is validated in two ways. First, based on the descriptive statistics (Table 11) and randomly selected samples, we test the conformity of the generated data with respect to the real world user and ad network behaviors received from a mobile advertisement company. Second, we applied the single network approach and checked whether the ad network performances converge with the real world performance.

| | Minimum | Maximum | Mean | Std. Deviation | Variance |
|---|---|---|---|---|---|
| Ad frequency (ad units) | 2.18 | 38.24 | 16.27 | 10.47 | 109.718 |
| Network latency (S) | 0.14 | 0.95 | 0.21 | 0.04 | 0.001 |
| Fill rate (%) | 18.05 | 72.99 | 45.91 | 19.47 | 379.147 |
| CRR (%) | 8.89 | 21.85 | 16.74 | 4.67 | 21.796 |
| ARPC (ad request units) | 6.41 | 36.80 | 18.19 | 10.61 | 112.62 |

**Table 11. Descriptive statistics based on the ad networks**

We use the free and open-source Java library called Java Optimization Modeler (JOM) to implement the proposed optimization model and the simulation; we coded the algorithms using the Java programming language. The simulation model is validated in two ways. Firstly, based on the descriptive statistics (Table 11) and randomly selected samples, we tested the conformity of the generated data in the real world user and ad network behaviors. Secondly, we applied the Single network approach and checked whether the ad network performance mimicked real-world performance. To reflect uniform weights in the model, all metric parameters were normalized before calculating the APU value.

## 3.7 Analysis and Results

### 3.7.1 Comparison with Benchmark Approaches

We further analyze the outcome of our approach compared to the other benchmark approaches. We run each experiment for 1 day (24 hours) with the 5-minute app period. Figure 15, demonstrates the comparisons of fill rate, CRR, and ARPC across all app instances across the benchmark approaches. We present the results across all app instances in the box plot in Figure 15, where the upper and lower limits correspond to minimum and maximum values, and the height of the box is proportional to the interquartile range (IQR). According to Figure 15, the FCFS approach has the highest fill rate and the proposed approach has a noticeably higher fill rate than the single ad network approach and the exact approach. This is because we consider the fill rate as one of the optimization parameters in the model, whereas the exact approach focuses solely on the number of clicks –. For CRR and ARPC measurements, the proposed approach has a slightly lower performance than the exact approach, but much higher performance than the FCFS and the single ad network approaches. Therefore, we can state that *the proposed approach is able to achieve a higher number of clicks using a lower number of ad requests compared to the common single network approach and the FCFS approach for each app instance.*

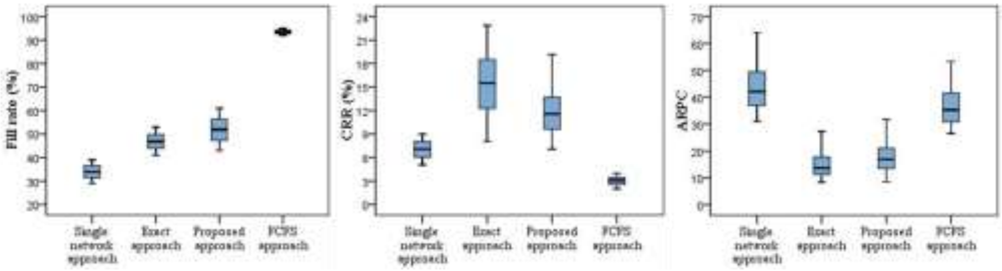


**Figure 15. Comparisons of fill rate, CRR, and ARPC with benchmark approaches across various app instances**

We also look at the model performance in each app period because the proposed approach focuses on improving the ARPC (or publisher's return) in each app period. Figure 16 illustrates in box plots the fill rate, CRR, and ARPC across overall app periods of 5-minute durations in a 24-hour window across all instances.

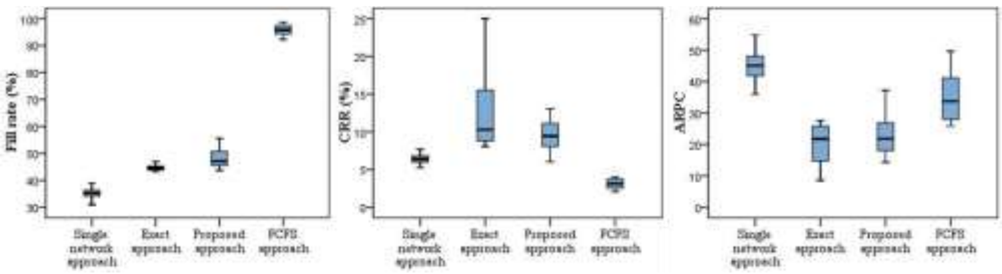


**Figure 16. (a) Fill rate, (b) CRR and (c) ARPC across app periods**

According to the results of Figure 16, we can see that the proposed approach can achieve a considerably higher fill rate and CRR for each app period because the model optimizes the performance of the displayed advertisements by selecting ad networks that best suited the requests in each app period. Therefore, *when the ad network is selected dynamically for each app instance based on the recent app period behavior, we can achieve a higher number of clicks compared to when advertisements are obtained from a single ad network for all the mobile app users all the time and everywhere.*

Additionally, we can see how other attributes in the model help to increase CRR. In particular, the proposed model optimizes the publisher's return based on the ad frequency and the network latency of the ad networks. Figure 17 shows the variation of ad frequency for each approach across app instances. As can be seen in Figure 17, the proposed model's ad frequency is very close to 1. The proposed approach selects the ad networks which have the highest number of unique advertisements, a circumstance which leads to higher CRR. Therefore, from both Figures 15 and 17, we can conclude that, *by optimizing the ad network attributes in the ad network selection process, CRR can improve*.

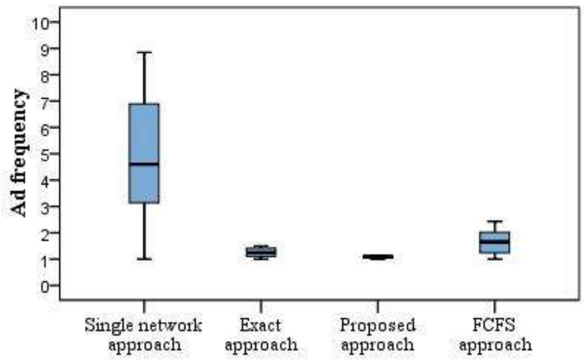


**Figure 17. Ad frequency across app instances**

The proposed model determines how clicks are distributed across different advertisement types and uses that distribution to determine the optimal ad network for the immediate app period. To increase user engagement with advertisements, the ad networks should deliver advertisements of a type that has a higher CRR for the app instance. So, in Figure 18, we consider the app instance A and illustrate the total number of ads deliveries for each ad type by each approach. According to the Figure 18, the proposed approach delivers most of the advertisements from ad type "three" while other approaches do not have such a sizeable difference between the numbers of ads of each ad type delivered. With further analysis, we were able to identify that among all ad types, ad type “3” has the highest CRR for the app instance A in the proposed model. So we can conclude that *for each app instance, the proposed model selects ad networks where the ad networks deliver advertisements that best matches with user interests.*

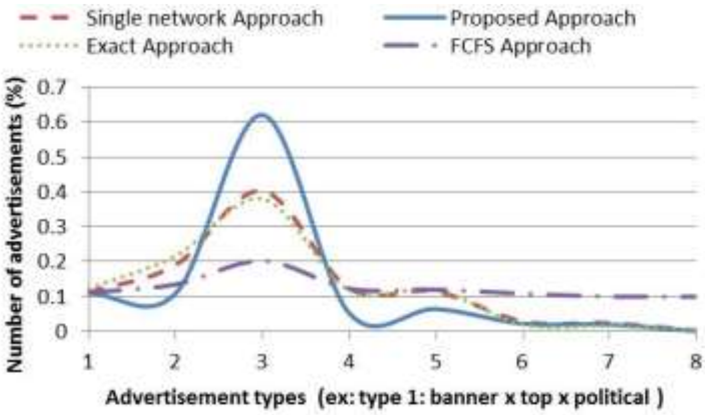


**Figure 18. Number of ads delivered (%) to each ad type by each approach**

### 3.7.2 Sensitivity Analysis of APS

In Table 12, we present the fill rate, ad frequency, network latency, CRR and ARPC for five random app instances as an example. In the experiment, the app period duration is 5 minutes; i.e., at every 5 minutes, the algorithm is run to select a new set of ad networks for the next 5- minute period. From Table 12, we can see that when the fill rate is higher and ad frequency and network latency are lower, the CRR is higher and the ARPC decreases. Thus, *when we select ad networks which have a higher fill rate and a lower ad frequency and network latency, we can achieve a higher number of clicks despite the smaller number of ad requests.*

| App instance | Fill rate (%) | Ad Frequency (ad units) | Network latency (S) | CRR (%) | ARPC (ad request units) |
|---|---|---|---|---|---|
| A | 44.34 | 1.01 | 0.27 | 10.36 | 21.77 |
| B | 47.85 | 1.1 | 0.36 | 7.98 | 26.19 |
| C | 42.16 | 1.14 | 0.31 | 8.04 | 29.5 |
| D | 54.73 | 1.01 | 0.28 | 13.88 | 13.16 |
| E | 61.01 | 1 | 0.21 | 17.24 | 9.51 |

**Table 12. ARPC with fill rate, ad frequency, network latency and CRR**

Using the same app instances considered in the above analysis, we analyze the model behavior with respect to the app period duration. As shown in Figure 19, when the app period duration is reduced, we can achieve a higher number of clicks for a smaller number of ad requests. This economy of requests is possible because the proposed system's adaptability suits the dynamic nature of the RTB ecosystem. Therefore, we can state that *the proposed model can achieve a higher performance level with a shorter app period duration*.

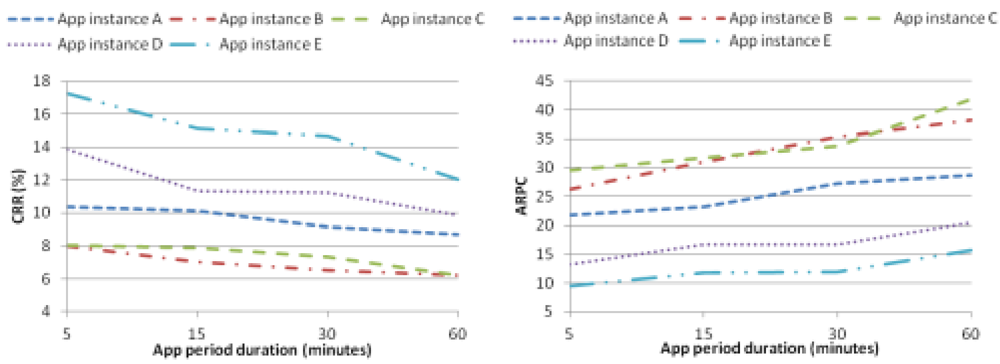


**Figure 19. App period duration vs. CRR and ARPC**

## 3.8 Discussion

The key contribution of this study is the development of a solution at the app instance level that selects the most suitable ad network for each mobile app user so that the solution improves the performance of advertisements and increase publishers' revenue via a higher return. With the simulated experiment, we have demonstrated that by using app instance level optimization can achieve higher publisher returns than by using solutions at the app level.

The proposed solution is the first aimed at publisher return optimization at the app instance level; there are many ways in which the solution can be improved. We have determined four limitations of the current model. The first limitation is that the proposed ad network utility function considers the three parameters equally, but in practice, these can have different weights. Due to data limitations, we did not, in this study, provide any heuristics to fine tune these parameter weights. The second limitation is a temporal issue of the ad network selection process. In each app period, if the selected ad network continuously performed better than the past performance of the other ad networks, we would not be able to track other ad networks' current performances (sometimes their new performance can be much better than the selected ad network's present performance). As a solution to this problem, we can send a fraction of the total ad requests (e.g., 10% of ad requests) to other ad networks and keep track of their performance rather than send all to selected ad networks (Adikari and Dutta 2015). The third limitation is an issue of determining which ad networks are included: in this study, we select the optimal ad network for a particular app instance from a predefined set of ad networks; the publisher will need to pre-determine the set of ad networks in consideration. The fourth limitation is that we did not include the monetary value of a click in the model. Even though there is no significant variation in the average pay [8]for clicks in different ad networks, the inclusion of pay per click in the APU may help to better understand the performance of the ad networks regarding the real publishers' revenue.

The proposed solution is not only a novel approach that improves the returns for the publisher, but it also provides a new way of thinking about addressing issues in the online advertising ecosystem, including preserving the privacy of mobile users. As Ji et al. (2014) noted, an individual can be de-anonymized using a little bit of information, but, in our solution, user behavior and the interactions are stored and processed only at the app instance level (i.e., within the mobile device). In principle, the proposed solution would be a win-win solution for both advertisers and publishers. As explained in the related work section, there is on-going research that seeks to optimize the RTB advertising ecosystem at three levels namely, at the DSP level, the ad exchange level, and the publisher level. Unlike the current study, none of those previous studies have considered app instance level solutions to improve their stakeholders' expectations. Therefore, this study introduces a novel approach of optimization for RTB in the mobile app advertising ecosystem.

[8] http://www.thalamus.co/global-data

# Chapter 4. Campaign Optimization in RTB Advertising

## 4.1 Introduction

RTB is now a billion-dollar business, and various sources predict that mobile RTB advertising revenues in the United States will reach USD $6.8B by 2018 up from 100 million in 2013 (Statista 2016). With RTB, DSPs need to select the bid requests to bid for in real-time based on target audience and on the group of the target audience with the highest likelihood of providing a higher return on the advertisement by providing more clicks. Therefore, in RTB, it is important to determine the most appropriate target audience to obtain a higher return on advertisements. The advertiser's target audience is based on a combination of characteristics (users' demographics, device, publisher, impression, location, etc.). For example, if an advertiser defines its target audience as females who have "iPhone 6" and play game applications and lives in California, then the DSP would bid for bid requests that are generated from a game application of an "iPhone 6" device belonging to a female user from California.

To achieve the advertiser's target return (number of clicks), the DSP needs to decide on the probability of receiving a click from the incoming bid requests and bid for the right bid requests from the advertiser's target audience and have a higher Click Through Rate (CTR). This is a significant step in RTB-based campaigns, called *campaign optimization* and it is a focal subtask of a bidding algorithm. The importance of effective *campaign optimization* has been discussed at length within the RTB industry (Programmatic 2016; PubMatic 2016) and commercial products (Datacratic 2016) have recently evolved to address this important step in running RTB-based advertisement campaigns. However, commercial products do not reveal campaign optimization techniques. In addition, there has been no academic discussion on approaches or algorithms for this important step of digital campaign management. Therefore, there is a need to conduct academic research by which the campaign optimization approach can be developed and made widely available to be adopted by numerous other DSPs (Gulyani 2016). In this study, we attempt to do exactly that – *develop an approach/algorithm for campaign optimization for DSPs in the RTB ecosystem*.

Through our novel campaign optimization solution, we introduce a few key theoretical contributions. Similar to the theoretical contributions of the study 1 and study 2, in this study also we discuss the importance of defining a problem in a more practical way by including the real world challenges. In addition, when the problem is dynamic in nature, we can separate its dynamism and address its challenges using a solution with two-stage processes. One process for the static components of the problem (similar to the supervised bidding phase to be discussed later in the chapter) and the other for the dynamic components of the problem (similar to the intelligent bidding phase to be discussed later in the chapter). The last theoretical contribution is that we actuate LDA as a new way of feature grouping in machine learning and bring an example for the literature to use LDA in a different context other than semantic document processing. All our practical contributions made through this study bring a new line of solutions to RTB advertising. We redefine the key problem in online advertising from the need for a higher CTR to solve the CTR vs Bid request dilemma. We develop an automated campaign optimization strategy to maximize the number of clicks. Our solution is able to address the dynamism in RTB advertising and retrain the classification model automatically and effectively based on new dynamic information.

The remainder of the chapter is organized as follows. Section 4.2 reviews state of the art approaches with a detailed description of their research gaps, and the section following that formally specifies the problem. The proposed solution is explained in detail afterward, after which the dataset is described, and the analyses and results presented. The chapter concludes with implications and discussion of limitations.

## 4.2 Related Works

In online advertising, the most simple and efficient way of measuring user attention to advertisements is user click behaviour. In the current RTB ecosystem, if DSPs do not achieve the expected number of clicks towards the advertisers from the target audience then the return on an advertisement will decrease, and advertiser retention will become susceptible to attrition. Therefore, having an explicit estimation of the probability that a bid request from a target audience will bring a click to the campaign is crucial. This has been studied in both guaranteed online advertising (Bharadwaj et al. 2012; Ghosh et al. 2009; McAfee et al. 2013; Shen et al. 2014) and sponsored search online advertising (Graepel et al. 2010; Liu et al. 2010; Liu and Chen 2006; Richardson et al. 2007; Zhu et al. 2010). Some studies related to display advertising have estimated the CTR as part of their bid price prediction algorithm to optimize the bid price (Chen et al. 2011; Chickering and Heckerman 2003; Pepelyshev et al. 2015). Some studies applied logistic regression (Lee et al. 2012; Tagami et al. 2013) and clustering algorithms (Geyik et al. 2015; Regelson and Fain 2006; Yan et al. 2009) to estimate CTR and select an audience. In the existing research, CTR is computed based on historical performance data with a pre-defined set of features and feature values.

For this line of research, behavioral targeting is considered as the fundamental school of thought on improving click-throughs. According to Chen and Stallaert (2014), due to the behavioral targeting, competition for an impression becomes relaxed and fewer advertisers target a particular user, therefore, advertisers need to pay less for impressions. In parallel, it will increase the probability of click through. In general consumers and the advertisers, both can benefit from behaviorally targeted advertising if the advertisements are more likely to be related to the consumers' interests (e.g., based on the consumers' purchase history) (Shen and Miguel Villas-Boas 2017). Some studies extend the development of behavioral targeting techniques from a simple user purchase history to marketing and operations such as customer acquisition, retention, service access quality, and capacity allocation (Afèche et al. 2017). Also, it has been determined that the probability of a click on an ad depends on two effects- the sojourn effect: the influence of the passage of time and the exposure effect: the influence of prior exposures of the ad to the user during that session (Sun et al. 2017).

Some publishers impose click-through-rate constraints on advertising platforms/firms (Mookerjee et al. 2016). Therefore targeting the right users with a higher click probability is important to retain such publishers in the ad campaign. So in their study, Mookerjee et al. (2016) applied logistic regression to determine the click probability associated with an impression. Then based on the click probability the decision is to show an ad and develop a decision model to decide which advertisement to pick. The impression details they observe are the visitor's search string, Internet browser, operating system, previous click data, the visitor information, the publisher's website and etc.

Using two large-scale field experiments and two lab experiments Bleier and Eisenbeiss (2015) demonstrate that behaviroal targeting increases click-throughs especially at an early information state of the purchase decision but its effectiveness is reduced as time passes since that last visit. This has been further confirmed by Ghose and Todri (2016) who find that advertising effects are amplified up to four times when consumers are targeted earlier in the purchase funnel path, and the longer the duration of an exposure to display advertising, the more likely the consumer is to engage in direct visits to the brands. On the other hand, Bleier and Eisenbeiss (2015) have discovered that behaviroal targeting increases click-throughs irrespective of whether banners appear on motive congruent or incongruent display websites. Further, in terms of view-through, ad effectiveness increases only on motive congruent websites, but decreases on incongruent websites.

Furthermore, due to the complex and dynamic behaviour of various stakeholders such as mobile/web users, publishers, advertisers, and other supportive systems (RTB exchanges, DSP, etc.), the programmatic advertisement ecosystem does not follow the same behaviour throughout a particular ad campaign, especially in the mobile environment. For example, when a campaign is originally targeted towards *Candy Crush* users on *iOS*, it may turn out that such users from *New York* are providing more clicks than the same groups of users in other regions. In such a dynamic mobile app ecosystem, having an exhaustive list of possible feature values for various features is also not possible. Table 14 describes the sample features and the feature value sets in our data.

Past research (Adikari and Dutta 2015) has demonstrated that the number of bid requests for each app/site in an RTB platform changes dynamically and does not follow any identifiable trend or pattern. This is due to the inherent dynamism in the app ecosystem[9], where the popularity of apps changes within short periods of time. Additionally, according to Lee et al. (2012) and Pepelyshev et al. (2015), characteristics of impressions drive CTRs. These make it difficult to predict CTR based solely on historical data. Furthermore, in a stream of bid requests, over time, new feature values appear and some of the old feature values may disappear. For example, due to a change in trends the bid requests from a game app may stop arriving from the state “Texas”, and start arriving from a new state “California”, because the game has lost its trendiness in “Texas”, but is gaining trendiness in “California” (such trend change is very common in the mobile app market). Thus, the feature “State”

[9] This has been discussed under the second challenge in the introduction chapter.

will have a new value “California” and the old value “Texas” will disappear from future bid requests. Using our dataset, we have computed the number of distinctive feature values for six features (App Name, Device ISP, Exchange, Inventory Type, Device Type and City) and of incoming bid requests in each period (a day of the month). The results are illustrated in Figure 20. As illustrated, the number of distinctive feature values of these features vary significantly over time. This issue of an undefined set of feature values exacerbates the campaign optimization problem.

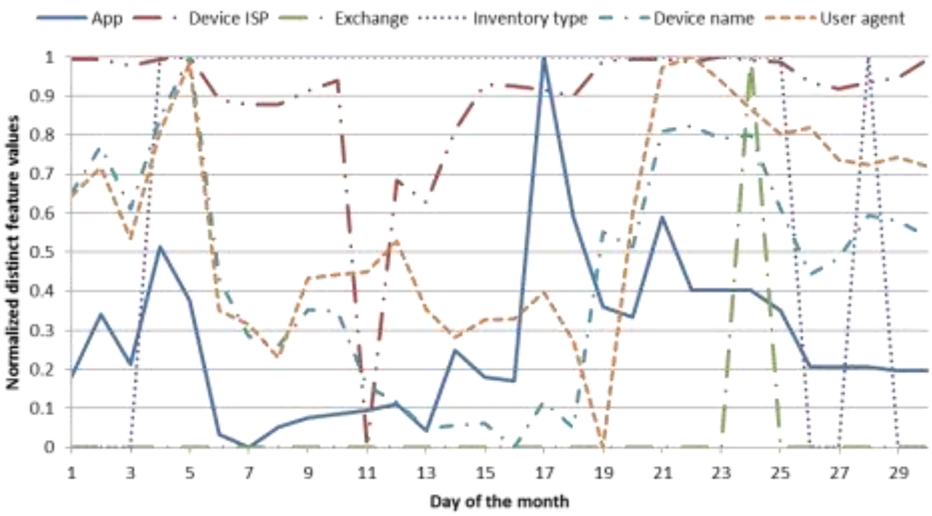


**Figure 20. Number of distinct feature values of different features in each day**

The situation at hand becomes even more challenging due to partial data availability at DSPs and to the low overall CTRs (Chaffey 2015). An RTB does not send all the bid requests to a DSP; it selectively distributes a subset of its bid requests across multiple sets of DSPs. If a DSP wins a bid, it knows the price it will pay for the impression, but if the DSP loses in the bidding at RTB, the DSP does not receive any information about the winning bid from the RTB.

Some existing research tried to work around this problem by introducing data hierarchies (Lee et al. 2012; Wang et al. 2010). Using data hierarchies, they proposed to consider higher level categories (such as a country instead of a state) when new feature values appear and when existing feature values disappear. However, there are a few issues with such a hierarchical approach. Firstly, the campaign optimization approach loses its fine-grained control over various feature values. Secondly, as many current studies have considered (Shan et al. 2016; Shen et al. 2015; Tagami et al. 2013; Zhang et al. 2016), in a mobile environment, we cannot uniquely determine the bid requests from a specific user. Thirdly, it is hard to determine the exact hierarchies. For example, we cannot determine whether gender is a user level or application level characteristics. Because, in most instances, DSPs get the gender characteristic in a bid request as an empty string. Therefore DSPs tend to derive it from the application characteristic, i.e. from the mobile application's demographic statistics, and not at the individual bid request level. According to Lee et al. (2012), it is hard to pre-define the data hierarchy. Consequently, determining data hierarchies (e.g. mobile application category [10] of a new mobile application) in real-time in online advertising is practically impossible. In summary, the implementation of exhaustive data hierarchies in practice is impractical and time-consuming.

To overcome the dynamic problem explained earlier, online incremental learning was proposed and their main focus was on solving the problem of concept drift (Bouchachia et al. 2007; Ditzler and Polikar 2013). The prediction models in the incremental algorithms do not retrain the classifier

[10] https://www.statista.com/statistics/270291/popular-categories-in-the-app-store/

instead continuously learn from the new data while in operation and extend the existing prediction model. However, in this way, the prediction model is associated with two risks (Sahel et al. 2007). One is that the model extension depends on the continuous use of potentially misclassified data. Second, the prediction models of the incremental algorithms may output an altered model when the same data is presented in a different order. Another stream of research tried to retrain the classification models using a fixed time frame (Widmer and Kubat 1996), using an adaptive time frame (Widmer and Kubat 1996) and density based training set (Salganicoff 1993). The intuition behind these approaches is to retrain the classification model in a timely manner and use all the available information so as to adapt it to the dynamic behavior. However, it is still tricky to decide at which point the model should be retrained, and given that the context is very dynamic, it would not always be effective to declare a fixed time frame (like 24 hours). Also, in many cases, the decision of retraining the classification model is taken manually or randomly. However such decisions will not be effective on every occasion. Therefore, there is a research gap to introduce an intelligent and automated mechanism to decide on the most appropriate time to retrain the classification model given that the context is overly dynamic.

In this study, we address the campaign optimization problem with such dynamism and data limitation in mind – dynamism in the number of incoming bid requests, the dynamism in the set of features and feature values of incoming bid requests, and the lack of information on non-bidded and non-won auctions.

## 4.3 Problem Specification

### 4.3.1 Design Artefact in the Business Process of Advertisement Bidding in DSP

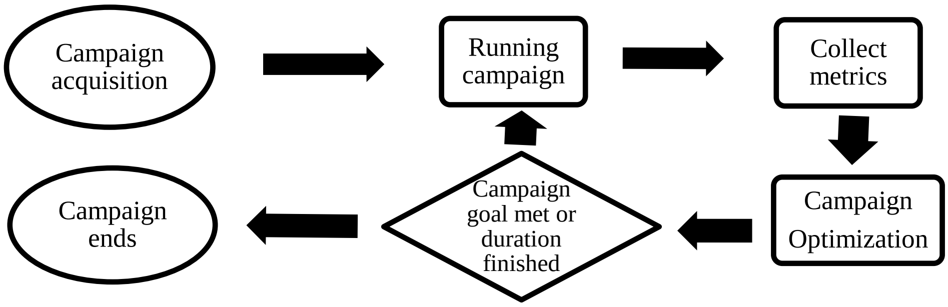


**Figure 21. The flow diagram of the business process of advertisement bidding in DSP**

The business aspect of advertisement bidding in a DSP is a complex process that involves multiple tasks and stages. Figure 21 illustrates the flow diagram of the business process. According to the Figure 21, initially, a DSP acquires a campaign from an advertiser; afterward it determines the target audience of the advertiser, campaign duration, and the number of clicks as the campaign goal. Next, the DSP runs the ad campaign to achieve the advertiser's goal. When the ad campaign is running the DSP attempts to decide and consume the bid requests that bring the highest return (a higher number of clicks from the advertiser's target audience) to the advertiser. To accomplish this task, the DSP computes metrics such as CTR, Winning bid average (WBA), and the number of bid requests consumed. Based on the collected metrics optimizes the bid request consumption process and re-runs the ad campaign till it either achieves its goal or till the campaign duration has expired. When the campaign goal is met, or the campaign duration is exhausted the campaign ends. The performance of the business process of the advertisement bidding depends solely on how effectively and efficiently the optimization process is carried out. Accordingly, this study focuses on the augmentation of the optimization stage of the business process of advertisement bidding. In the next sub-section, we provide an in-depth description of the "*Campaign Optimization*" step.

### 4.3.2 Campaign Optimization

Most of the existing studies (Lee et al. 2012; Wang et al. 2010) have modeled this problem with maximizing CTR as their primary objective function. This has been achieved by giving increased weight to the features which possess feature values associated with higher CTRs. However, this does not guarantee that the feature values associated with higher CTR will occur frequently. If these feature values do not occur a sufficient number of times in the incoming bid requests, the campaign may fail to reach its target goal of number of impressions and clicks.

To further understand this phenomenon, let us examine Figure 22, where we select the Internet Service Provider (ISP) as the feature. In Figure 22 we select the top five ISPs in the USA (Toptenreviews.com 2016) and demonstrate their CTRs and the number of bid requests for five consecutive days in our real bid request data. Even though Cox Communication and Cable One have higher CTRs, they do not have a higher number of bid requests. Conversely, Verizon Wireless has a higher number of bid requests but does not have a higher CTR compared to other ISPs. And though Time Warner Cable does not have either the highest CTR or the highest number of bid requests, its behaviour (both CTR and the frequency of occurrence of bid requests) is perpetual and possesses the capacity to bring a higher number of total clicks over the ad campaign duration.

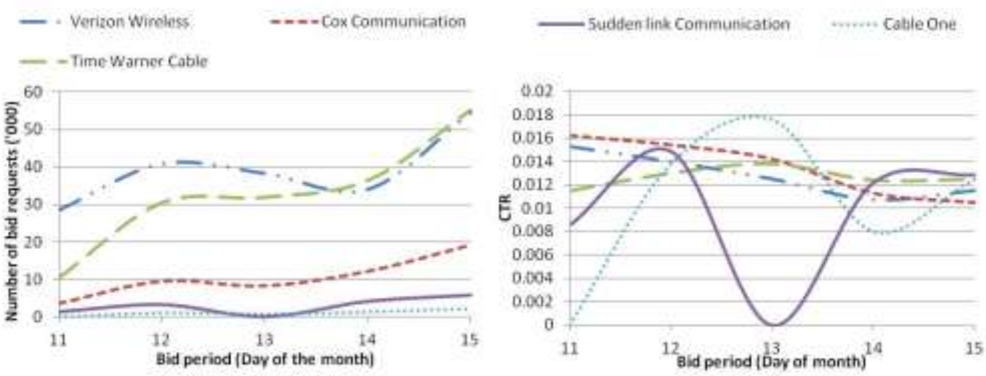


**Figure 22. Number of bid requests and CTR change over consecutive periods in Device ISP**

In this study, one of our objectives is to determine the feature values that frequently occur in correlation with many incoming bid requests and also are associated with higher CTRs. We refer to such feature values as *influential feature values*. If we maximize either CTR or the number of bid requests, we may be unable to achieve the expected number of clicks and increase advertiser return. We state this as the *dilemma of CTR vs. the number of bid requests* in RTB digital advertising. As explained in the previous section, due to the appearance of new feature values (for example, for feature "device model" – a new model may be added, for feature "app name" – a new app may have become popular), a static solution will not provide the desired outcome for this problem. Thus, we require a dynamic solution that can maximize CTR while meeting the requirement of a total number of clicks and determining the influential feature values over time.

In addition, advertiser retention is highly dependent on representativeness. Representativeness refers to the reach of *advertiser target audience* (Ghosh et al. 2009; Turner 2012). For example, when an advertiser's target audience consists of users of *Candy Crush* on *iOS* from *California*, the DSP should bid for bid requests which have similar feature values (characteristics). This ensures that the product is advertised towards the right market segment of the advertiser (Lee et al. 2012). In practice, DSPs observe click behaviors during ad campaigns and embrace more features *manually* to narrow the filtering options of the impressions (Lambert and Woods 2007). Since DSPs run many ad campaigns simultaneously, it is not possible to accurately and efficiently track such click behaviors manually from a large set of data. On the other hand, even if existing classification algorithms develop models by selecting sets of feature values that have higher CTRs, due to the abovementioned dynamism, it is naive to assume that the same sets of feature values will remain available in future bid requests and garner the expected number of clicks over the duration of the ad campaign.

Addressing the above issues, this study proposes a solution that can automate the campaign optimization process and dynamically optimize the relevant combination of influential feature values which can achieve a higher CTR in order to meet the campaign target objective (total number of target clicks or impressions) in running mobile advertisement campaigns.

## 4.4 Solution Formulation

Our solution is twofold. Firstly, we propose a supervised classification model to predict click behavior with a timely selected feature value set. Secondly, we introduce a feature grouping based bid request selection model to bid and determine trending patterns in the incoming bid requests. As shown in Figure 23 when combined, these models provide a framework to intelligently and automatically identify the incoming bid request patterns with new feature values and bid for the bid requests with a higher click probability. According to our context, we define the ***feature space*** based on the characteristics of the bid requests such as app name, state, platform, device ISP, etc. And each characteristic is considered a ***feature***. Referring to the example given earlier, {iOS, Candy Crush, New York} are the ***feature values,*** and these values belong to {Platform, App name, State} *features* respectively. This feature definition eases the need for data hierarchies to define characteristics of advertisements as discussed in past research (Lee et al. 2012).

As illustrated in Figure 23 when a new bid request is received, the proposed approach applies the incoming bid request to two paths - the Model training process and the Bidding process.

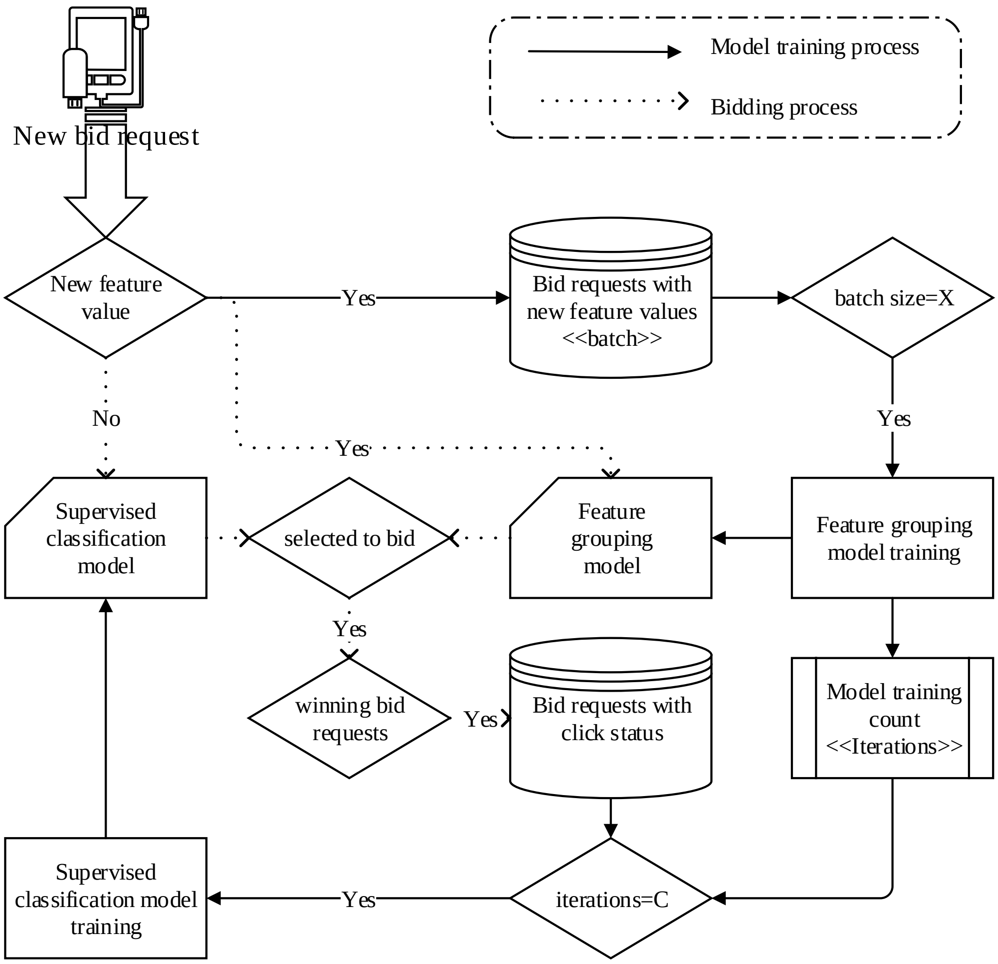


**Figure 23. Communication flow of the proposed approach**

Initially, the bid request will be checked for the presence of new feature values. For example, consider a new bid request consisting of {*iOS*, *Pokémon-go*, *New York*}. Here *Pokémon-go* is a new feature value under the *Mobile application* feature which had not been received by the DSP before. Therefore, the information in its entirety of the new bid request cannot be applied to the supervised classification model seeing as the model structure does not support the new feature value *Pokémon-go*. In our approach, such bid requests will be applied to a feature grouping model during the bidding process. The feature grouping model will decide whether or not to bid on that bid request. Aligned with the model training process; each bid request with a new feature value will be stored temporarily as a batch. When the batch size is equal to a predetermined value (X), the feature grouping model is retrained to find new patterns of feature values in recently received bid requests. Later in the chapter, we define these patterns as *feature groups* (e.g., a new feature group could be {*iOS, Pokémon-go, New York*}). The key intuition of this process is to bid for the bid requests with new feature values and determine their click behaviors which can be subsequently fed to retrain the supervised classification model. Through the feature grouping model we bid intelligently for a select set of bid requests with new patterns, and as a result, we name this phase as the *Intelligent bidding phase*. Since the next phase of the bidding process depends entirely on the supervised classification model, we call it the *Supervised bidding phase*.

In the supervised bidding phase, when a bid request does not consist of any new feature values it is applied directly to a supervised binary classification model to predict the click probability and to take the bidding decision. In the bidding process, every time a bid request is won (either using the supervised classification model or the feature grouping model), the bid request information will be stored alongside its click status. This stored information will be used to retrain the classification model after a predefined number of iterations (C) of the feature grouping model training. The intuition behind the use of a dataset which combines the output of the feature grouping model and the supervised classification model is to provide complete information (e.g., bid request characteristics and the click behaviors) of the receiving bid requests to the classification model with or without new feature values. As a result, the classification model can reconstruct its structure including the new feature values and bid for a higher number of bid requests while maintaining a higher CTR. The following sections explain the supervised classification model and the feature grouping model in detail. Before delving into the details of the models, a list of notations is declared in Table 13.

| | |
|---|---|
| *Indices* | |
| $g$ | index for feature group $g = 1, \ldots, G$ |
| $v$ | index for a feature values $v = 1, \ldots, V$ |
| $f$ | index for a features $f = 1, \ldots, F$ |
| $d$ | index for advertiser's desired target audience feature values $d = 1, \ldots, D$ |
| $t$ | index for a bid period $t = 1, \ldots, T$ |
| $c$ | index for an iteration count $c = 0, \ldots, C$ |
| $i$ | index for a bid request $i = 1, \ldots, I$ |
| *Parameters* | |
| $R_{gc}$ | CTR for feature group $g$ at iteration $c$ |
| $Q_{gc}$ | log value of total number of bid requests for feature group $g$ at iteration $c$ |
| $VS_{gc}$ | feature value significance for feature group $g$ at iteration $c$ |
| $GS_{gc}$ | temporal feature group significance for feature group $g$ at iteration $c$ |
| $N_{gc}$ | total number of feature values in feature group $g$ at iteration $c$ |
| $A_{gc}$ | total number of target audience feature values contained in feature group $g$ at iteration $c$ |
| $VQ_{vgc}$ | occurrence of feature value $v$ in feature group g at iteration $c$, $VQ_{vgt} \in \{0,1\}$ |
| $GQ_{gc}$ | occurrence of feature group g at iteration $c$, $GQ_{gt} \in \{0,1\}$ |
| $\mathcal{B}$ | click behavior $\mathcal{B} \in \{0,1\}$ |
| $G$ | total number of feature groups |
| $S$ | size of the target audience |
| $F$ | total number of features |
| $V$ | total number of feature values |
| $I$ | total number of bid requests |
| $T$ | total number bid periods performed during an ad campaign period |
| $C$ | total number iterations performed before retraining the classification model |
| $k$ | the number of feature groups selected to filter the bid requests |
| $\mathcal{v}_{gc}$ | feature value set of a feature group $g$ at iteration $c$ |
| $\mathcal{G}$ | set of feature groups |
| $\mathcal{F}$ | feature set |
| $\alpha$ | the parameter of the Dirichlet prior on the per-request feature group distributions |
| $\delta$ | the parameter of the Dirichlet prior on the per-feature group feature value distribution |
| $\theta_i$ | the feature group distribution for bid request $i$ |
| $\varphi_g$ | the feature value distribution for feature group $g$ |
| $Z_{iv}$ | the feature group for the $v^{th}$ feature value in bid request $i$ |
| $L_{iv}$ | the $v^{th}$ feature value in bid request $i$ |

**Table 13. Notation for the model descriptions**

### 4.4.1 Feature Grouping Model

The basic construct of a feature grouping model is a feature group. Based on similar click probability, we group the feature values as ***feature groups*** (a set of characteristics as mentioned earlier). For example, {*iOS*, *Candy Crush*, *New York*} can be a *feature group*. There are three feature values in this feature group, and they belong to three specific features (see Table 14). According to the definition, a feature group cannot have more than one feature value from the same feature. Therefore, the feature group size is the number of features/feature values in a group, and the maximum size of a feature group is equal to the total number of features considered in the algorithm.

| | |
|---|---|
| Feature List / Features ($\mathcal{F}$) | App, State, OS |
| Feature: Feature Values (v) | App: Candy Crush, Sudoku Free<br>State: New York, California<br>OS: iOS, Android |
| Feature Groups (g) | {Candy Crush, New York, iOS}<br>{Candy Crush, California, Android}<br>{Sudoko Free, California}<br>{New York, iOS} |
| Set of Feature Groups ($\mathcal{G}$) | {{Candy Crush, New York, iOS},<br>{Candy Crush, California, Android},<br>{Sudoko Free, California},<br>{New York, iOS}} |
| Total Number of Features (F) | 3 |
| Total Number of Feature Groups (G) | 4 |

**Table 14. Examples of features, feature values, and feature groups**

We define ***target audience*** with the unique feature values similar to those of the feature group. Therefore, the target audience can consist of only one feature value of a particular feature. In our example, {*iOS*, *Candy Crush*, *California*} can be a target audience. However, if the advertiser requested multiple feature values of a feature as the target audience, we need to define multiple target audiences. In our example, if the advertiser requests to have both *iOS* and *Android* as the platform, the target audiences will be {*iOS*, *Candy Crush*, *California*} and {*Android*, *Candy Crush*, *California*}. The ***size of a target audience*** (***S***) is defined as the total number of unique features in a *target audience*, and according to the given example, the size of the target audience is 3.

In alignment with the above definitions, we define a *temporal feature grouping* based adaptive strategy to find the most effective *feature values*, which delivers a higher return to the advertiser within the given *target audience*. We define a period called ***iteration*** (***c***) (the duration of the iteration depends on the time taken to reach the predetermined batch size with the new bid requests) to learn click behaviours and ascertain the bid requests which have a higher probability of a click. Therefore, the key assumption of this model is that the behaviour in iteration $c$ is the best input for estimating the behaviour in the iteration $c+1$. The learning is achieved by two steps, namely, *feature group discovery* and *feature group selection*.

#### 4.4.1.1 *Feature Group Discovery*

In this study, we propose *topic modeling* as the key technique for feature grouping. Topic modeling is a form of text mining that is used to identify patterns in a corpus (Blei et al. 2003) and defines a way of determining a recurring pattern of co-occurring terms. Following the same concept, we apply topic modeling to find a combination of feature values from the bid request data that reflects a new pattern. From various topic modeling approaches, Latent Dirichlet Allocation (LDA) (Blei et al. 2003) will be used to discover the feature groups containing different feature values as it provides the flexibility to decide the number of feature values (terms) and groups (topics). According to our problem definition, all the feature values of each feature are equivalent to terms and each bid request is equivalent to a document in LDA. Based on the plate notations given in Figure 24 we have described the dependencies among the different variables with respect to the standard LDA model representation for Dirichlet-distributed topic-term distribution. See Table 13 for the notation described.

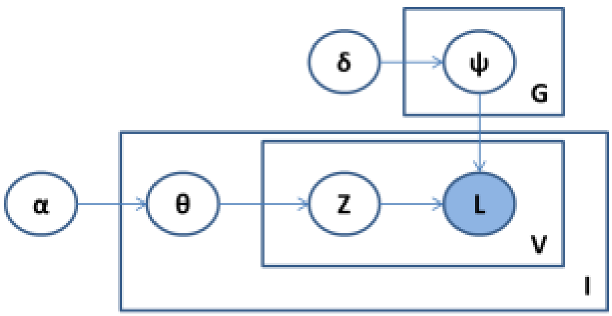


**Figure 24. Plate notation for LDA with Dirichlet-distributed feature group-feature value distribution**

In Figure 24, $L$ is considered an observable variable and the remaining variables are latent variables. $L$ is a specific feature value ($v$) in the bid request ($i$) which defines as $L_{iv}$ in Table 13. The Dirichlet priors use on feature group-feature value distribution can constrain the probability of the feature groups on a small set of feature values. The parameters of the Dirichlet-distributed feature group-feature value distribution are stored in $\varphi_1 \ldots \ldots \varphi_g \ldots \ldots \varphi_G$ F-dimensional vectors. F is the total number of features. The generative process for LDA on a corpus consisting of I number of bid requests each with the length of F, can be explained as follows. First select $\theta_i$ from the Dirichlet distribution with a symmetric parameter $\alpha$ ($\alpha < 1$), then select the $\varphi_g$ from the Dirichlet distribution with a symmetric parameter $\delta$. For each feature value in a bid request, choose a feature group $Z_{iv} \sim Multinomial(\theta_i)$ and feature value $L_{iv} \sim Multinomial(\varphi_{Z_{iv}})$ ( $Multinomial$ refers to the categorical distribution).

According to the discussion above, each bid request is considered a document and the features of each bid are considered terms in the topic modelling approach. The document-term matrix of topic modeling in our case is called the request-feature matrix (see Figure 25). As shown in Figure 25, different bid requests are composed of different feature values of each type of feature. On Request 1 of Figure 25, the features app name, state, and platform have the feature values *Candy Crush*, *New York* and *iOS*, respectively. In our request-feature matrix, these feature values have the value 1 and all remaining feature values have zeros. Therefore, in the request-feature matrix, the total number of ones that can exist for a bid request is equal to the total number of features in a bid request.

| Documents | Terms |
|---|---|
| Request 1 | Candy Crush, New York, iOS |
| Request 2 | Candy Crush, New York, Android |
| Request 3 | Sudoku Free, California, iOS |
| Request 4 | Sudoku Free, New York, iOS |
| Request n | Candy Crush, California, iOS |

| | Candy Crush | Sudoku Free | New York | California | iOS | Android |
|---|---|---|---|---|---|---|
| Request 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| Request 2 | 1 | 0 | 1 | 0 | 0 | 1 |
| Request 3 | 0 | 1 | 0 | 1 | 1 | 0 |
| Request 4 | 0 | 1 | 1 | 0 | 1 | 0 |
| Request n | 1 | 0 | 0 | 1 | 1 | 0 |

**Figure 25. Initialization of Request-Feature matrix as Document-term matrix**

Since LDA has the flexibility to decide the number of feature values in a feature group (similar to the number of words in a topic), we have constrained it to be between the number of features used to define the target audience of the campaign and the total number of features. That is, LDA will be applied repeatedly changing the number of feature values that should be in a feature group. However, the number of feature groups to be selected is decided heuristically. According to Wei and Croft (2006) to get a unique set of topics from LDA the number of topics parameter should be a value between 50 and 300. Algorithm 7 explains each step of the feature group discovery process.

**Algorithm 7: Feature Group Discovery**

1. Initialize the *F, S, numFG,* and $\mathcal{G}$ ={}
2. Select batch for iteration *c*
3. Determine distinct feature values *(v)* for each *f* in *F*
4. Initialize request feature matrix and assign *{0,1}*
5. For *numFV* between F and S
6. Apply LDA where the number of feature groups is *numFG* and the number of feature values in a feature group is *numFV*
7. Determine a feature group g out of selected group of feature values
8. add g to $\mathcal{G}$
9. End for
10. Output $\mathcal{G}$

In the Step 1 of the algorithm, we initialize the size of the advertiser target audience (*S*) and the total number of features (*F*). When the incoming bid request dataset fills the predetermined size of the batch (*X*), a new iteration will be started. At the beginning of each iteration, the feature grouping model will be retrained using the bid request data stored in a temporary storage (batch) during the previous iteration. So, in our algorithm selecting the batch when it reaches to the predetermined size (*X*) is Step 2 of the algorithm.

In the Step 3, we determine the distinct feature values for each feature in the batch under consideration. In Step 4, we initialize the request-feature matrix by creating columns for each feature value. As shown in Figure 25, if the feature "app name" has only two feature values, *Candy Crush*, and *Sudoku Free*, two columns are created with their associated names and assigned 0 and 1 based on their occurrence in each bid request. Once we have the complete matrix for the batch with all the feature values and the requests, LDA is applied in Step 5. We repeatedly apply LDA to get feature groups with different sizes (with a different number of feature values: *numFV*). Feature groups' sizes are varied between *F* and *S* Once we have the LDA model, in Step 6, a set of feature groups (*g*) is identified from the topics out of the LDA topic model results. Also, we set the number of feature groups parameter in LDA as 50, 100, 150, 200, 250 and 300 and checked the output with a unique set of feature groups (groups with the same set of feature values without considering the order). When the number of feature groups parameter is higher than 50, LDA delivers redundant feature groups but not new unique feature groups. Thus we consider numFG as 50 to train the LDA.The value of the number of topics parameter (numTopics) in the LDA application is decided heuristically and once the LDA model is applied in Step 6, in the Step 7 a set of feature groups (*g*) can be identified from the output of the LDA topic model results. As given in Steps 8 to 9, we combine all the selected feature groups (*numFG*) in each loop into the feature group set ($\mathcal{G}$) which we output at this stage. When a feature group contains a higher number of feature values, it is unlikely to occur frequently within the bid requests. For example in Figure 25, if we select a feature group with {*Candy Crush*, *New York*}, we will be able to select two bid requests. But if we select a feature group with {*Candy Crush*, *New York*, *iOS*}, we will only be able to select one bid request. Else, when a feature group has a smaller number of feature values, it is unlikely to have a high CTR. If we consider the same example and assume that bid requests with {*Candy Crush*, *New York*, *iOS*} will receive clicks, then for the feature group {*Candy Crush*, *New York*, *iOS*}, CTR becomes 1 however for the feature group {*Candy Crush*, *New York*} the CTR would be less than 1 (i.e., according to Figure 25, it is 0.5); because there can be bid requests which originate from other platforms such as *Android*. This is the essence of the CTR vs. bid request dilemma, and in the next phase, we define a ranking mechanism to select an optimal set of feature groups to solve this dilemma.

#### 4.4.1.2 *Feature Group Selection*

To solve the dilemma of CTR vs. bid requests, we need to determine the feature groups which have the highest probability of maximizing both the CTR and the number of bid requests. Therefore, the selection of a feature group should be anchored in a utility value. We define five metrics that aid to estimate the utility value for each feature group.

**CTR**: CTR is defined as the number of advertisements that are clicked at least once out of the total number of ads displayed. This is calculated for each feature group ($R_{gc}$). By selecting the feature group with a higher CTR, we can filter the bid requests with a higher click probability.

**Feature value significance**: Feature value significance aids to determine bid requests from an exact target audience, because we measure the significance of the feature values in a group based on two criteria. The first criterion is the number of feature values in the group ($A_{gc}$) that also exists in the target audience. This is considered as a ratio of the number of feature values in the group that exists in the target audience to the size of the feature group of the complete target audience ($A_{gc}/S$). For example, consider a campaign targeted towards {*Female*, *California*}. A feature group {*Female*, *California*, *iPhone*, *Verizon*} will have a ratio of 1, which indicates that the feature group is a representative subset of the target audience. The second criterion is the size of the feature group ($N_{gc}$). This is considered as a ratio to the total number of features (*F*), when the ratio $N_{gc}/F$ is less than 1, we can add more features to the group to target a subset of the audience and increase CTR. According to the two criteria, Eq. (4.1) depicts the feature value significance ($VS_{gc}$) of a feature group.

$$VS_{gc} = \frac{A_{gc}}{S} \times \frac{N_{gc}}{F} \tag{4.1}$$

**Feature group significance:** In addition to feature value significance, we determine the significance of a feature group compared to other feature groups in the same iteration and over the past *(C-1)* iterations. Therefore, by applying the feature group significance as a metric to select the feature groups, we can decide the bid requests that frequently occur with a set of common patterns. The *within-group significance* is measured by the average occurrence of the target feature group's feature values in other feature groups. The *between-group significance* is measured by how recently and frequently a particular feature group had occurred in the past iterations. To determine the recency of feature group occurrence, we use the iteration index as a weight. The current iteration index is marked as 1, the previous iteration is 2 and so on. We consider up to *C* iterations in the past. When the iteration index is low it gives a higher recency. By combining the *within* and *between* group significances, the feature group significance is formulated as Eq. (4.2).

$$GS_{gc} = \frac{\sum_v \sum_g VQ_{vgc}}{G \times N_{gc}} \times \frac{\sum_{c=1}^{C} (C-c) \times GQ_{gc}}{\sum_{c=1}^{C} c}, \; v \in \mathcal{v}_{gc} \tag{4.2}$$

4.4.1.3 *Utility Value and Campaign Optimization*

Using the parameters above, we define the *feature group performance* utility (*U*) as:

$$U_{g(c+1)} = \mathrm{R_{gc} Q_{gc} GS_{gc} VS_{gc}}, \quad \forall g, c \tag{4.3}$$

Based on this utility metric, we further define the objective function of the argument maximization (See Eq. (4.4)) to select the most effective feature groups:

$$arg\,max_g\ U_{g(c+1)} \rightarrow arg\,max_g\ (\mathrm{R_{gc}Q_{gc}GS_{gc}VS_{gc}}),\ \forall g \qquad (4.4)$$

We define the output of the argument maximization as a rank list of the feature groups, and this serves as a guide to filter the incoming bid requests. Seeing as we will receive many feature groups as the output of phase one, we will select the top *k* feature groups from the ranked list.

In a particular loop of the intelligent bidding phase, the feature grouping model will be retrained $C$ times. When the iteration count reaches the predefined total iteration count ($c$=$C$) the loop will be terminated. Afterwards, the next loop will be initiated ($c$=*0*). Therefore, at $c$=*0*, we have a feature grouping model which was trained at the previous loop's last iteration ($c$=$C$). Next, we discuss the supervised bidding phase.

### 4.4.2 Rare Event Supervised Binary Classification Model

The proposed supervised classification model in this study is based on Rare event logistic regression. Logistic regression has been applied in many studies of online advertising context as an effective binary classification model (Chakrabarti et al. 2008; Lee et al. 2012; Li et al. 2015; Richardson et al. 2007). In our problem space, we attempt to classify whether or not the incoming bid request has a higher click probability. Therefore based on the prediction threshold of the click probability we determine whether the bid request will be clicked on or not. Thus click status is the dependent variable (DV). All the other attributes of the bid requests are considered to be independent variables (IVs). Table 15 in section 4.6 lists such possible attributes (features) from our dataset. Following the DV and the IVs, the simplest logistic regression model for the problem can be defined as Eq.(4.5).

$$\Pr(\mathcal{B}_i = 1|x_i, \beta) = \left(\frac{1}{1+exp(-x_i\beta)}\right) \qquad (4.5)$$

In Eq.(4.5) $x_i$ defines a vector of $p$ IV values $x_i = (1, x_{i1}, \ldots\ldots, x_{ip})$ using a *(p+1)* coefficient vector $\beta = \beta_0, \beta_1, \ldots\ldots, \beta_p$. In the logistic regression model Maximum Likelihood (ML) estimate $\hat{\beta}_{ML}$ is computed by maximizing the log-likelihood function ($lnL(\beta|\mathcal{B})$) and its explicit form of the score function (see Eq. (4.6)) demonstrates that the proportion of observed events is equal to the average predicted probability.

$$\frac{\partial lnL(\beta|\mathcal{B})}{\partial\beta} \equiv U(\beta) = \sum_i^N (\mathcal{B}_i - \pi_i)\, x_i = 0 \quad where\; \pi_i = \frac{1}{1+exp(-x_i\beta)} \quad (4.6)$$

As mentioned earlier, the number of clicks received compared to the number of advertisements displayed to users is very low. According to Chaffey (2015), in display advertising, the industry average on CTR is approximately 0.06%, and it results in a dataset that is skewed towards zero clicks and is unbalanced. Similarly, compared to the total number of bidding requests selected to bid, the number of winning bid requests is also very low[11]. Therefore the ratio of the number of clicks received to the number of bid requests selected to bid is meager. Paul Allison explains this problem of a small number of cases in the rare of the two outcomes in Statistical Horizons[12]. According to Paul Allison, the problem is that ML estimation of the logistic model is known to suffer from small-sample bias (Some studies defined this problem as a class

---

[11] https://docs.openx.com/Content/publishers/reports_bidperformance_report.html
[12] https://statisticalhorizons.com/logistic-regression-for-rare-events

imbalance problem related to other classifiers such as Decision trees, Neural networks, and Support vector machines (Japkowicz and Stephen 2002)). The number of observations in the less frequent of the two outcomes will behold the degree of biasness (King and Zeng 2001). Therefore, it is inappropriate to apply the logistic regression in its standard form, and this necessitates a mechanism to compensate for the small sample bias. Therefore, in this study, we apply rare event logistic regression which is also called Firth logistic regression. The idea of this method is to penalize the likelihood function by utilizing the Jeffrey's invariant prior (Firth 1993). The Jeffrey's invariant prior (Jeffreys 1946) is given in Eq. (4.7).

$$|I(\beta)|^{1/2} = |X'WX|^{1/2} \tag{4.7}$$

In Eq. (4.7) $I(\beta)$ defines the Fisher information matrix, X defines the design matrix, and the diagonal matrix $W = diag\big(\pi_i(1-\pi_i)\big)$. By using the Jeffrey's invariant prior, Firth logistic regression (FL) counteracts the first-order term in the asymptotic bias of the ML estimates ($\hat{\beta}_{FL}$), while introducing a new score function. The new score function is given in Eq. (4.8).

$$\sum_{i=1}^{N}\left(\mathcal{B}_i - \pi_i + h_i\left(\frac{1}{2} - \pi_i\right)\right)x_i = 0 \tag{4.8}$$

In Eq.(4.8) $h_i$ is the diagonal elements of the standard hat matrix $H$ as per the penalized-likelihood.

$$H = W^{1/2}X(X'WX)^{-1}X'W^{1/2}$$

According to Firth (1993), compared to the standard ML estimates, when the Fisher matrix is maximized for $\pi_i = 0.5$, Firth penalization pushes the prediction towards 0.5. This overcomes the overestimated prediction due to the rare events.

Besides the class imbalance problem, the application of Firth penalization can provide a consistent and finite estimate of coefficients in the case of separation as well. Separation occurs in binary outcome models with the presence of one or more explanatory variables which perfectly predict the outcome of interest that is zeros, ones, or both. Two types of separations (complete separation and quasi-complete separation) exist except for type overlap, which is the usual case where we can apply the standard logistic regression and gain reliable estimates (Rainey 2016). Complete separation occurs when one or more explanatory variables perfectly predict both outcomes. Conversely, quasi-complete separation occurs when one or more variables predict either ones or zeros but not both. In the context of the problem at hand, quasi-complete separation can exist when clicks are attached to a particular feature value. For example, in a given period all the clicks that the DSP received could be received from users of a certain type of mobile application such as *Pokémon-go*. Therefore, the application of Firth logistic regression can resolve not only the small sample bias but also the separation issues.

To preserve the representativeness of the predicted clicks on the selected bid requests, we implement weighted logistic regression. With weighted logistic regression, we can weigh the important observations such that different observations will have different significances. The bid requests that include the target audience feature values should have a higher importance, and these should be weighted more heavily. Therefore, in our approach, an extra feature (new column added) called *target_aud* is computed as a weight based on the existence of relevant target audience feature values. If the target audience is users from *California* who play the mobile game *Candy Crush* on *iOS*, any bid request which has feature values {*Candy Crush, iOS*, *California*} will have the value 1 for the feature *target_aud.* Also, when the feature values existing in a bid request are *{Candy Crush, iOS}* the value for *target_aud* is 2/3.

## 4.5 The Benchmark Approaches

The complete algorithm of the most of the commercial approaches is not known. But the implementation of these approaches is based on some of the available basic classification techniques such as Logistic Regression, Support Vector Machine (SVM), Random Forest, etc. Therefore, to assess the performance of the proposed approach, we have selected the Logistic Regression, SVM and Random Forest benchmarking approaches for comparison. The Logistic Regression is the most viable binary classification model available. SVM is considered as one of the most robust and computationally efficient classifiers. Random forest, on the other hand, is the most appropriate classifier for the categorical data. Since our proposed approach is a joint approach of both Feature Grouping method and Logistic Regression, the performance of the Feature Group approach is independently measured for the comparison. In each of these approaches, the dependent variable is whether or not the user will click the receiving bid request.

### 4.5.1 Rare Event Logistic Regression

Similar to the proposed approach, rare event logistic regression is applied on a periodic basis. A bid period is defined as a period that the logistic model is applied to determine the bid. Thus at the end of each bid period (*t=1*), the classification model is retrained using the same bid period's data to predict the next bid period's (*t=2*) click behaviors of receiving bid requests. For the comparison purpose, we define a bid period as a day (24 hours). When a bid request at *t=2* has a new feature value which did not exist at the *t=1* period, that new feature value will be considered a missing value when the prediction model is applied to the bid request.

### 4.5.2 Feature Grouping based Approach

Instead of applying the combined algorithm of logistic regression and feature grouping as the proposed approach, in this benchmark approach, we only considered the feature grouping model. The feature grouping model is applied temporally, bid period wise. The discovery of the feature groups through the LDA and the selection of the feature groups with the highest utility value are the same as the proposed approach. The main difference is that this approach is executed without conditioning on the bid request with new feature values. For example, Adikari et al. (2016) implemented this approach (temporal feature grouping) [13]on bid requests collected for a 24 hour period and selected the top-5 feature groups in order to predict the following 24 hours' bid requests with high click probability.

### 4.5.3 Random Forest

Random forest computes a number of decision tree classifiers based on different subsamples of the dataset, then averages the results of all the classifiers to improve predictive accuracy and control overfitting. Since the bid request data is categorical unlike with logistic regression, the decision tree approach can handle new feature values during the prediction. Similar to the rare event logistic regression approach, we have applied the random forest on the daily (bid period) basis. Since our dataset is sparsed with clicks, we used

[13] https://www.youtube.com/watch?v=XKkGx6C2krg

the *class_weight* option in the *RandomForestClassifier* function in the Python *Scikit-learn* package (Pedregosa et al. 2011) in order to overcome the imbalance in the class variable. In addition, similar to the logistic regression approach, we utilized the *sample_weight* option in the *RandomForestClassifier* function to weigh the bid requests originating from the target audience. The maximum number of decision trees considered is 500.

### 4.5.4 Support Vector Machine

The SVM training was conducted with C-support vector classification (C-svc) which is a Quadratical Programming (QP) approach. C-svc is able to find the best possible hyperplane by measuring the margin between two classes using 2-norm of the normal vector (Zhang et al. 2013). According to Mercer's theorem (Cortes and Vapnik 1995), the kernel function K can be considered equal to a dot product in input space, and due to the nonlinearity of the features, SVM is able to create random decision functions in the input space of the kernel function. We applied Polynomial kernel (polydot) which produces the best performance for categorical data as it uses a combination of features from the input sample instead of determining their similarity independently. Also, we used *class.weight* option in R *libsvm* (Chang and Lin 2011) to address the class imbalance problem in our dataset. The bid requests with new feature values were not overlooked but were considered missing values when applying the prediction model.

## 4.6 Analysis and Results

### 4.6.1 Dataset

| Feature | Description | Number of feature values |
|---|---|---|
| exchange | Name of the RTB exchange, e.g., MOPUB | 6 |
| app | Name of a mobile app or a website, e.g., Sudoku Free | 178 |
| inventory_type | Type of the ad space, e.g., Mobile Web | 3 |
| connection_type | The method to connect to the Internet, e.g., Cellular | 3 |
| platform | Type of Operating System (OS), e.g., Android | 3 |
| user_agent | OS version eg: Android 5.1.1 | 89 |
| device_name | Type of the device, e.g., Galaxy S6 Edge | 658 |
| device_isp | Internet Service Provider eg: VerizonWireless | 1072 |
| state | USA State, e.g., Massachusetts | 51 |
| dma | Designated market area eg: 506 | 324 |
| is_click | Whether click or not, e.g., 1/0 | 1 |

**Table 15. Description of the features**

We were granted access to a dataset of an ad campaign from a leading mobile DSP organization. The dataset contains 72 million bid requests belonging to a complete ad campaign, which was executed between the 17$^{th}$ of November and 18$^{th}$ of December in 2015. From the complete dataset, we identified 16 features. To determine their significance, we performed the *Multiple Correspondence Analyses* (MCA) seeing as, with an extension of correspondence analysis (CA), MCA can analyse the pattern of relationships between several categorical variables (Greenacre and Blasius 2006). Via a standard correspondence analysis on an indicator matrix (i.e., a matrix whose entries are 0 or 1), MCA can be obtained. Based on the results of MCA, we removed city and device model features as they loaded to all dimensions with more than 70% correlation; if the algorithm combines highly correlated feature values into the feature groups, then the prediction results could be less accurate and more uncertain (Neter et al. 1996). The descriptions of all selected features are given in Table 15.

### 4.6.2 Parameter Significance

The utility value is given in Eq.(4.3) defined such that it is capable of selecting the feature groups which garner a higher CTR and a higher number of bid requests in the following iteration. Therefore, we tested the significance of each of the four metrics for both the predicted CTR and the predicted number of bid requests. We considered the CTR and log value of the number of bid requests at *(c+1)* as dependent variables (DV) and the remaining parameters (CTR, log value of the number of bid requests, feature value significance, and feature group significance) for iteration $c$ as independent variables (IV). In this experiment, the iteration ($c$) duration is considered 24 hours. We computed feature group-based data on all the aforementioned variables (both IVs and DVs). The dataset consists of 9060 tuples and six parameters, and their descriptive statistics are given in Table 16.

| | $\boldsymbol{Q_{gc}}$ | $\mathbf{R_{gc}}$ | $\mathbf{GS_{gc}}$ | $\mathbf{FS_{gc}}$ | $\boldsymbol{Q_{gc+1}}$ | $\mathbf{R_{gc+1}}$ |
|---|---|---|---|---|---|---|
| Std. Error of Mean | 0.013 | 0.002 | 0.002 | 0.003 | 0.014 | 0.001 |
| Mean | 2.633 | 0.043 | 0.189 | 0.436 | 2.479 | 0.015 |
| Variance | 1.611 | 0.053 | 0.024 | 0.064 | 1.780 | 0.003 |
| Minimum | 0.000 | 0.000 | 0.001 | 0.000 | 0.000 | 0.000 |
| Maximum | 4.947 | 1.000 | 0.799 | 1.000 | 4.959 | 1.000 |

**Table 16. Descriptive statistics of the feature group based dataset**

Based on the above-mentioned dataset in Table 16, the significance test is performed on Multivariate Multiple Regression analysis (MMR) (Rencher 2003). Through MMR, we select the most significant parameters among $\mathrm{R_{gc}}$, $Q_{gc}$, $\mathrm{GS_{gc}}$, and $\mathrm{VS_{gt}}$ to define the utility value to select the effective feature groups which discussed in section 4.4.1.3. The regression model is given by Eq. (4.9).

$$[R_{gc+1}\ \ Q_{gc+1}] = [R_{gc}\ \ Q_{gc}\ \ GS_{gc}\ \ VS_{gc}\ ] \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \\ \beta_{31} & \beta_{32} \\ \beta_{41} & \beta_{42} \end{bmatrix} + [\varepsilon_1\ \varepsilon_2] \qquad (4.9)$$

The MMR technique provides test statistics when there are multiple DVs and IVs. The most common method of estimating the significance on MMR is, firstly, applying the linear model and computing the estimators for each DV, then applying the Multivariate analysis of variance (MANOVA) (Rencher 2003) on the joint probability distribution of the two estimators. We have considered the test statistics of MANOVA on Pillai's test, Wilks' test, Lawley-Hotelling test, and Roy's test (Rencher 2003). According to the MMR analysis, Pillai's test, Wilks' test, Lawley-Hotelling test, and Roy's test, demonstrate similar significance for the DVs jointly (the next bid period's CTR and the log value of the number of bid requests) on each of the IV, thus, we only depicted the Wilks' Lambda test statistics in Table 17. While the feature value significance($\mathrm{VS_{gc}}$), the feature group significance ($\mathrm{GS_{gc}}$) and the log value of the number of bid requests ($Q_{gc}$) are positively significant at the 99.9% confidence level, CTR is positively significant at the 95% confidence level.

| | $\mathbf{R_{gc}}$ | $\mathbf{GS_{gc}}$ | $\mathbf{VS_{gc}}$ | $\boldsymbol{Q_{gc}}$ |
|---|---|---|---|---|
| Pr(>F) | 1.52e-03** | 1.66e-06*** | 4.22e-06*** | 2.22e-16*** |
| Signifiance codes: '***' 0.001 '**' 0.05 | | | | |

**Table 17. MMR test statistics on Wilks' Lambda test**

### 4.6.3 Evaluation Metrics

The key evaluation metrics for this study are *Bidding rate*, *Click sensitivity*, *CTR* and *Effective CTR (eCTR).*

***Bidding rate*:** Bidding rate is defined as the total number of bid requests selected to bid from the total number of received bid requests. The selected bid requests to bid include both actual click oriented bid requests (true positives), and non-clicked oriented bid requests (false positive). Bidding rate can provide a measure of the algorithm's capacity to achieve the campaign's expected number of clicks. A lower bidding rate may indicate a lesser capacity to achieve the expected click count. Also, a higher bidding rate may indicate potential exhaustion of the target spend before the end of the campaign duration while failing to achieve the expected number of clicks. The Bidding rate can be defined as follows:

$$Bidding\ rate = \frac{number\ of\ selected\ i}{N}$$

***Click sensitivity:*** Click sensitivity is similar to recall in data mining which can be defined as the total actual-click oriented bid requests selected to bid (true positives) out of the total actual clicks. Click sensitivity determines the efficiency of the algorithm in terms of correctly predicting the bid requests with clicks.

$$Click\ sensitivity = \frac{number\ of\ selected\ i}{N}, \mathcal{B}\ \in\ \{1\}$$

**CTR**: CTR is defined as the number of advertisements that are clicked at least once out of the total number of ads displayed and is considered as the most commonly used evaluation metric to assess the performance of the click prediction in the online advertising.

***eCTR***: The *effective CTR* metric is defined as the CTR solely for the ads that are clicked on by the target audience, which is of principal interest to advertisers. We consider eCTR and Click sensitivity as the foremost viable benchmark for the evaluation.

The complete algorithm of the proposed approach is developed using the R programming language. For the implementation of the rare event logistic regression, we use the R package, Logistf (Ploner et al. 2006) which supports both Firth penalized logistic regression and the weighted observations.

The results are computed on a daily basis for each evaluation metric and demonstrated on four independent weekly instances for the purposes of cross-validation.

### 4.6.4 Comparison with Benchmark Approaches

As given in section 4.5, we compared the proposed approach with four other benchmark approaches. We analysed the performance of each approach based on the four types of evaluation criteria, namely *CTR, eCTR, Bidding rate* and *Click sensitivity*. In the Feature grouping method, we considered only the top 5 feature groups and the prediction threshold for the logistic regression, Random forest, and SVM is tuned to 10%. To maintain the consistency of the comparison, we considered the top 5 feature groups and batch size X=10000 during the intelligent bidding phase and 10% prediction threshold during the supervised bidding phase. The iteration count to retrain the logistic regression model was set at 10.

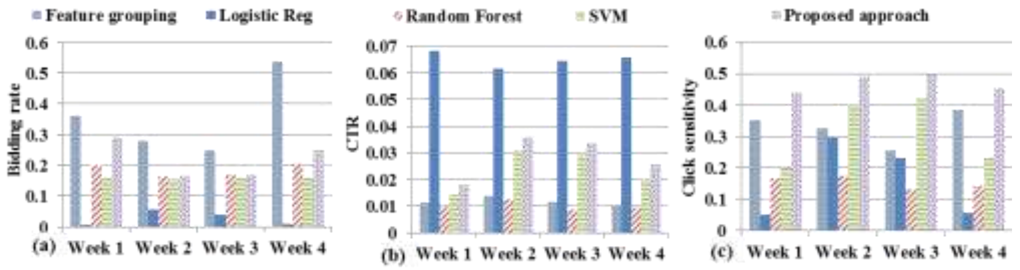


**Figure 26. (a) Bidding rate, (b) CTR, and (c) Click sensitivity in each week of the campaign**

Figure 26 demonstrates how each approach performs regarding the CTR, Bidding rate and Click sensitivity. As given in Figure 26(a), the Feature grouping approach has the highest bidding rate, and Logistic regression has the lowest bidding rate. Compared to Random Forest and SVM, the proposed approach achieved a higher bidding rate. This is because the feature grouping component of the proposed approach has the primary effect on the Bidding rate. Click sensitivity considers the actual performance of each approach in terms of the number of clicks obtained. Similar to the Bidding rate, the CTR of each approach was compared. As illustrated in Figure 26(b) Logistic regression has the highest CTR compared to all other approaches including traditional classification approaches (SVM and Random Forest). This is due to the regularization technique implemented in the Logistic regression and is specialized for the binary classification (King and Zeng 2001). Feature grouping method has the lowest CTR throughout the campaign because its utility function does not only focus on the CTR of a feature group. Even though the feature grouping approach is a component of the proposed approach's bidding process, due to the effect of the Logistic regression, the proposed approach is able to gain a higher CTR compared to the other approaches save for the standalone Rare event logistic regression approach. As illustrated in Figure 26(c) the proposed approach outperformed all others and contributed the highest number of clicks to the campaign. We conclude that even though Logistic regression has a higher CTR, it would not be able to bring a higher number of clicks to the campaign (which is the ultimate goal of the DSP). *Since the proposed approach is a combination of both Logistic regression and the Feature grouping method, it inherits the characteristics of both these methods and is able to bring a higher number of clicks to the campaign while maintaining a higher CTR as well as a higher Bidding rate*.

As illustrated in Figure 26, we can observe considerable variation in results of all the approaches across weeks. Thus, we analyzed our dataset and as given in Table 18, we discovered that there is a substantial difference in the number of new feature values introduced to the DSP during each day of the campaign. According to the Table 18, Week 1 and Week 4 received bid requests with many new feature values; while Week 2 and Week 3 received bid requests with fewer new feature values.

| | Week 1 | Week 2 | Week 3 | Week 4 |
|---|---|---|---|---|
| No. New Features | 25 | 16 | 13 | 28 |

**Table 18. Average number of new distinct feature values introduced in each day of the week**

To investigate the effects of the new feature values on the performance of each approach, we computed the mean and the variance of each evaluation metric for the whole campaign (see Table 19).

| | Bidding rate | | CTR | | Click sensitivity | |
|---|---|---|---|---|---|---|
| | Average | Variance | Average | Variance | Average | Variance |
| Feature grouping | 0.3587 | 0.0166 | 0.0120 | 0.0000 | 0.3295 | 0.0022 |
| Logistic Regression | 0.0303 | 0.0006 | 0.0652 | 0.0000 | 0.1583 | 0.0153 |
| Random Forest | 0.1868 | 0.0005 | 0.0104 | 0.0000 | 0.1537 | 0.0004 |
| SVM | 0.1624 | 0.0000 | 0.0239 | 0.0001 | 0.3130 | 0.0129 |
| Proposed approach | 0.2203 | 0.0039 | 0.0285 | 0.0001 | 0.4706 | 0.0008 |

**Table 19. Average and variance values of bidding rate, CTR, and Click sensitivity during the whole campaign**

According to Table 19, compared to the benchmark approaches, the proposed approach has a higher variance for both CTR and Bidding rate. However, regarding Click sensitivity, the proposed approach has the lowest amongst all approaches except for the Random Forest approach. In addition, the proposed approach has the highest click sensitivity of all. Therefore, we can conclude that, *even though due to the new feature values the proposed approach has a higher variation in Bidding rate and CTR, it is consistent in bringing the expected number of target clicks to the advertiser throughout the ad campaign.*

Next, we analyse the performance of the benchmark approaches compared to our proposed approach when given a longer period of historical information. We compared the performances of three key benchmarks approaches Logistic regression, SVM, and Random Forest, by training their classification models for a longer period (longer-term). We trained the models using three weeks of bid request data and then the last week of the campaign was used as the testing bid period to evaluate performances. In Figure 27, shorter-term is defined with a daily bid period similar to Figure 26. According to Figure 27, even though the CTR of Logistic regression and SVM increased, the bidding rate did not. Then again, Random Forest does not display much deviation of results in the long-term training period for all three metrics relative to the short-term training periods. Overall Click sensitivity of the three key benchmark approaches is far behind the proposed approach's performance. Therefore, this *analysis reveals that increasing the training bid period cannot increase the bidding rate and achieve the expected number of clicks for the ad campaign, as it does not adopt the dynamism in the RTB context.*

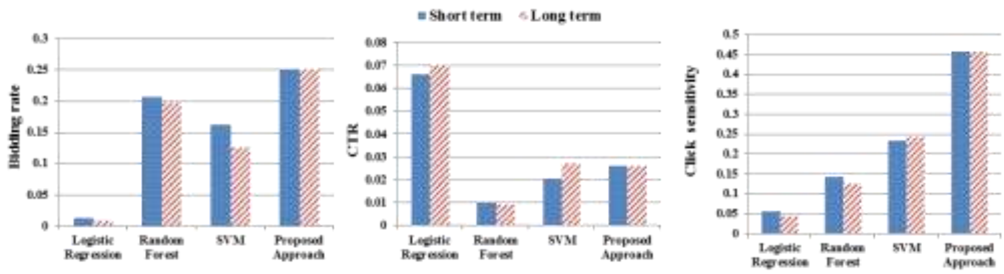


**Figure 27. Bidding rate, CTR, and Click sensitivity based on long-term and short-term bid periods**

For each approach, we computed the eCTR and compared it with the CTR. Table 20 provides the average eCTR and average CTR for the whole campaign under each approach and tables the difference between average CTR and average eCTR as a percentage of the average CTR. According to Table 20, the Feature grouping approach has no difference between CTR and eCTR, because, in the feature grouping approach, all bid requests selected to bid are from the target audience only. Other approaches, including Logistic Regression, Random Forest, and SVM demonstrate more than 25% of the decline in CTR when considering clicks from the target audience. That is, even though the DSP bids for many bid requests to meet the overall campaign goal of a number of clicks, the clicks do not originate from the target audience. However, the proposed approach achieves a smaller difference, and this is mainly due to the significant effect that the intelligent bidding phase has where the feature grouping model selects only the bid requests to bid if they are from the right target audience. Thus, as per the eCTR, the proposed approach has outperformed the current bidding processes via the two-phase bidding process, the intelligent bidding phase, and the supervised bidding phase. In conclusion, we can state that *the proposed approach can achieve a higher number of clicks from the target audience compared to the existing CTR based approaches of campaign optimization.*

| | CTR | eCTR | Difference (%) |
|---|---|---|---|
| Feature grouping | 0.012 | 0.012 | 0 |
| Logistic Regression | 0.0652 | 0.0488 | 25.15 |
| Random Forest | 0.0104 | 0.0074 | 28.85 |
| SVM | 0.024 | 0.0173 | 27.92 |
| Proposed Approach | 0.0285 | 0.0246 | 13.68 |

**Table 20. CTR and eCTR performances for the whole campaign**

In the final comparison, we look at how the Logistic Regression performance changes with its prediction threshold. The reason behind this analysis is that relaxing the prediction threshold also we can achieve a higher number of clicks. Four prediction thresholds (30%, 10%, 5%, and 1%) were considered alongside with the daily bid periods. Results were computed for the final week of the campaign (see Figure 28). The prediction threshold of the proposed approach's Logistic regression model was tuned to 10%.

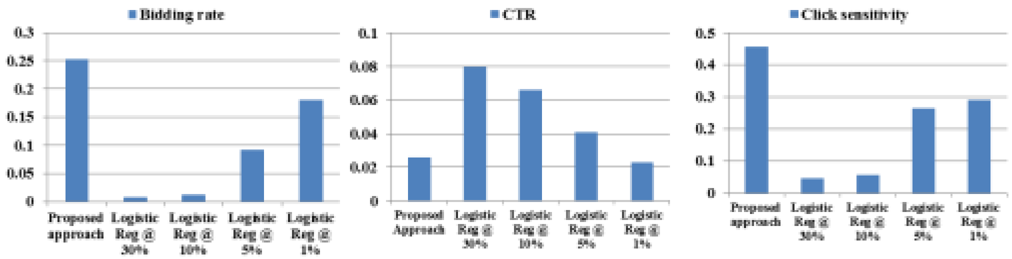


**Figure 28. Bidding rate, CTR, and Click sensitivity based on different prediction thresholds of the Logistic regression**

Even though we relax the prediction threshold of the logistic regression from 30% to 1%, the proposed approach still performed much better in terms of Bidding rate and Click sensitivity. The insight of this analysis is that *simply relaxing the prediction threshold of the classification model; we cannot resolve the dilemma of CTR vs. a number of bid requests.*

### 4.6.5 Sensitivity Analyses

Initially, the proposed approach is evaluated with batches of varying sizes. The batch size (X) is the key to select the feature groups in intelligent bidding. Since the dataset has a smaller number of bid requests, we limited the sizes of the batches. The smallest size of the batch we considered is 10000 (X) bid requests, and the experiment was conducted with the iteration count (C) as 10 and the number of selected feature groups as 5. The results for five different batch sizes are illustrated in Figure 29. According to the Figure 29 when a batch size is smaller the CTR is slightly higher. This occurs because the classification model can be retrained a greater number of times when the batch size is smaller. Similarly, the bidding rate is also much higher when the batch size is smaller. A higher bidding rate occurs with a smaller batch size because smaller batch sizes can quickly determine the new patterns of new feature values and adapt the intelligent bidding accordingly. As a result of a higher CTR and a higher bidding rate, when the batch size is smaller, click sensitivity also higher. Therefore, we can conclude that *when the batch size is smaller, the proposed approach will work considerably well and bring more clicks to the campaign.*

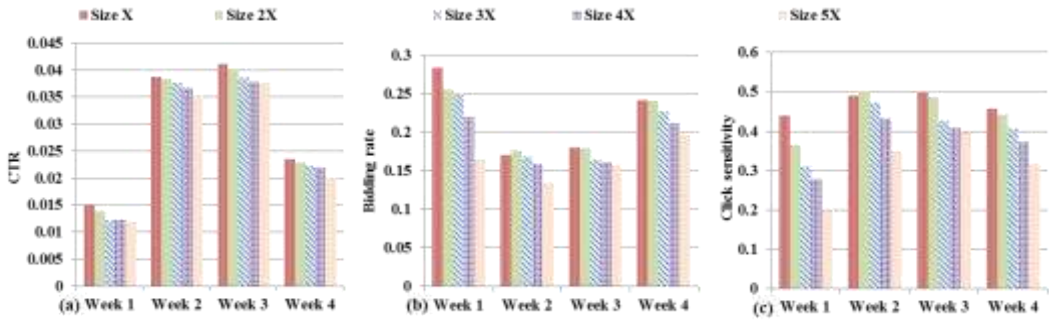


**Figure 29. (a) CTR, (b) Bidding rate and (c) Click sensitivity for different batch sizes; X=10000**

Next, we analysed the behaviour of the proposed approach with a different number of iterations. As illustrated in Figure 30, the proposed approach was executed with different iteration counts at different batch sizes. The experiment was conducted in the first week of the campaign with the number of selected feature groups as 5.

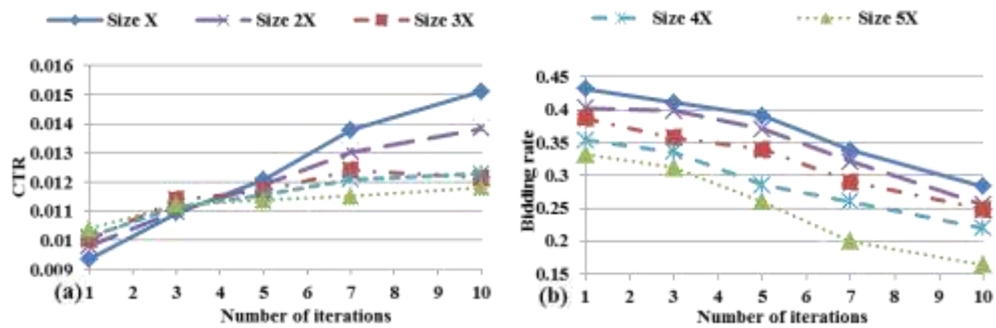


**Figure 30. (a) CTR and (b) Bidding rate at different iteration counts for different batch sizes; X=10000**

According to the Figure 30, when the iteration count is increased, CTR increases, and Bidding rate decreases. As depicted in Figure 30 when the batch size is larger, both CTR and Bidding rate are lower. However, during the initial iterations (until C=5), when the batch size is larger CTR is also large. This is due to the fact that, when a batch size is small and the proposed approach runs only a few iterations, the feature grouping model cannot consistently determine feature group patterns with new feature values. As a result, the training dataset of the classification model will have less information about the new feature values which leads to a lower CTR. Supported by the analyses on Figure 29 and Figure 30; we can conclude that *with a smaller batch size and a higher iteration count, the proposed approach can perform productively*.

The sensitivity analysis was conducted to test the performance of the proposed approach on a different number of feature groups selected to bid. Figure 31 illustrates the variation of CTR and Bidding rate in the top-*k* value of the feature group selection with respect to the different batch sizes. When the k is increased (*k*=10) for all batches, CTR decreases and Bidding rate increases. This is because when k increases feature groups which have a lower utility value and more common feature values are selected. Therefore, we can conclude that *by selecting the feature groups with the highest utility value we can achieve a higher CTR.*

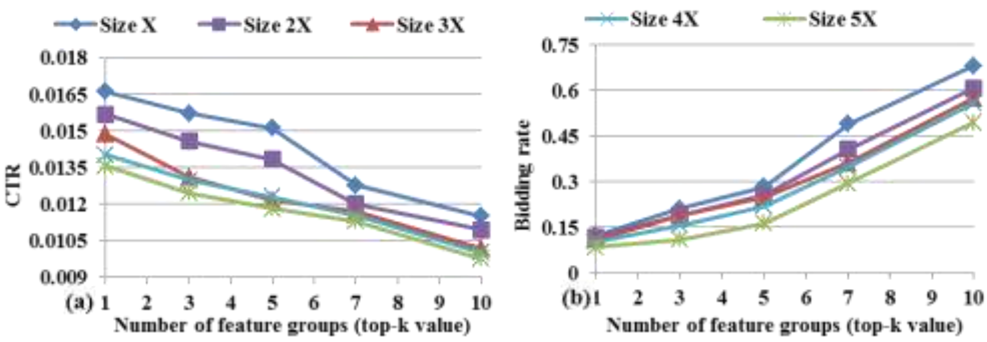


**Figure 31. (a) CTR and (b) Bidding rate with top-k feature groups for different batch sizes; X=10000**

## 4.7 Discussion

Campaign optimization is a major step in a bidding algorithm of the programmatic advertisement system. In campaign optimization, the DSP selects the appropriate audience to bid for against the incoming bid request traffic. Contrary to the existing CTR based manual approach of campaign optimization, this research proposes an automated campaign optimization strategy that balances the maximization of CTR with the maximization of the number of clicks during the whole campaign duration. The proposed approach consists of two bidding phases: intelligent bidding and supervised bidding. In the intelligent bidding phase, first, the feature groups are computed using the incoming bid requests data. The topic modelling approach has been applied here to identify the cluster (group) in the feature sets that produces higher CTR. Topic modelling approach is traditionally applied in textual document; the application of topic modelling is a novel application that is unique to this paper. The clustering techniques work better when dimensions are independent. Whereas as the topic modelling is based maximum likelihood function, it works better when each of the dimensions are interlined (like words in a document). Second, the feature groups are ranked based on their utility value – introducing a way to find new patterns of incoming bid requests with new feature values. As a result, the intelligent bidding phase identifies the feature groups with new feature values capable of obtaining a higher CTR as well as a higher number of bid requests (i.e. higher number of total clicks). The supervised bidding phase uses the new click behavior information gathered by the intelligent bidding phase in order to provide more accurate click predictions. In this phase, a rare event logistic regression model is trained to predict the bid requests while giving more weight to the bid requests received from the target audience. With experimental results based on real campaign data from a well-known DSP, we demonstrated that the proposed approach of campaign optimization can attain a considerably higher eCTR and Bidding rate compared to other benchmark approaches currently used in practice. Additionally, the proposed approach can consistently achieve the expected number of clicks from the target audience throughout the campaign without considering whether or not the DSP receives bid requests with new feature values. Therefore, the proposed approach can provide a full-fledged solution to achieve advertisers' targets successfully.

A few limitations exist in the proposed approach. According to Table 18 and Figure 29, when the number of bid requests arrives with many new feature values, the proposed approach may find it difficult to select feature groups with patterns of new feature values due to the high degree of variation. As a result, the highly ranked feature groups would have more common feature values instead of the new feature values. As a result, the proposed approach's bidding rate may increase, and its CTR may reduce. However, according to the results discussed in the previous section, even in such circumstances the proposed approach outperforms the other benchmark approaches. Our dataset does not contain any demographic data. Demographic data is one of the primary information that can bring a higher number of user clicks (Yan et al. 2009). Consequently, in the future, we intend to conduct a more thorough evaluation of our approach, including demographic data and running live comparative A/B test at a DSP. To the best of our knowledge, this is the first study that introduces a two-phase bidding algorithm for click prediction. Likewise, this type of feature grouping model can be implemented alongside with other traditional classification methods in other contexts like contextual recommendation systems where many new features and feature values are introduced to the system in real-time.

# Chapter 5. Conclusion

As explained in the introductory chapter, the field of RTB has become very active in terms of research and development. Hence, as more of the traditional digital media sources (such as TV and the web) move to RTB, and the use of smartphones and handheld devices become more pervasive, the need for a highly optimized computer and data-centric processes that enable highly targeted, personalized and performance-based advertising will become even greater. This dissertation has given a clear and detailed overview of the technologies and business models that are transforming the field of RTB in online display advertising. Also, the key challenges mentioned are important to enable more practically sensible solutions which can transform the current RTB ecosystem into a fully optimized and automated infrastructure.

With such motivation, based on the three main research gaps, this dissertation proposes detailed design science solutions to bridge the practical limitations in the state of the art approaches. In our solutions, we have focused on how ad campaigns are designed more effectively to offer a higher revenue to the advertisers, and how ad networks can be selected to bring a higher return to the publishers. In particular, we proposed novel bidding algorithms to garner more clicks for a lower target spend and to predict audience with a higher click probability. Furthermore, we have elaborated on how the ad networks can be selected at the user level to improve the user click return. The efficacies of proposed methodologies were subjected to rigorous experimental validation, and the results are promising.

The solutions provided in this dissertation includes three studies from both the advertiser side and the publisher side. As the first study of the dissertation, from the advertiser side, a novel bidding strategy called APS is developed to automate the bidding process in the DSP. APS recommends a novel approach that follows a dynamic programming (DP) algorithm, where the bid price and the decision to bid on an impression is decided in real-time through constant adjustment of the decision process based on a more recent stream of impressions and their bidding results. The second study, in particular, is from the publisher side and its algorithm is developed at the app instance level to select the optimal ad networks, which can bring higher revenue to the publisher. The algorithm considers both ad network and mobile app user behaviours. The third study is again from the advertiser side, and it further optimizes the ad campaign process to achieve a higher number of clicks consistently. The solution consists of a classification model and a feature grouping model that addresses both the dilemma of CTR vs. Number of bid requests and the rapid accession of new audiences (feature values). The outcome of these studies will contribute to the growth of the RTB advertising ecosystem into a fully-fledged automated system.

As a next step, the effectiveness of incorporating the APS with an audience prediction algorithm should be evaluated. For example, in APS, target audience is considered at the application level. However, by utilizing the two-phase bidding approach of the third study, APS can be further improved to bid at a lower granularity such as bid request level. As a result, APS will be able to optimize its results not only to reach a higher winning rate for a lower target spend but also to reach a higher Click sensitivity with a higher effective CTR. As the dissertation suggested when the current ecosystem becomes fully automated it easily becomes the target of click frauds (Crussell et al. 2014; Edelman and Brandi 2015; Wilbur and Zhu 2009). Therefore, we need to determine the loopholes in the RTB ecosystem that it can be vulnerable to botnet attacks that create millions of fraudulent clicks/conversions. New solutions ought to be designed to prevent and track such incidents in RTB advertising. Besides click fraud detection, attention needs to be directed towards another important research area on device identification based audience measurement in mobile app advertising (Bojinov et al. 2014; Grace et al. 2012; Stevens et al. 2012). This area of research is still undervalued due to the privacy concerns of determining individual user behaviors (Goldfarb and Tucker 2011; Tucker 2014). Similar to the cookies used in the web advertising to determine the users independently (Lambrecht and Tucker 2013; Unni and Harmon 2007), there is a necessity for a proper mechanism to independently recognize the devices in the mobile app advertising. Thus with such technologies and quality research on privacy management, we can improve the efficacy of the proposed solutions in this dissertation and augment the success of the RTB advertising for each stakeholders, especially advertisers, publishers, and mobile app users.

# References

Adikari, P., and Dutta, K. 2015. “Auto pricing strategy for Real time bidding in advertising: A dynamic programming approach,” in *Proceedings of the International Conference on Information Systems (ICIS),* Fort Worth, USA.

Adikari, S., and Dutta, K. 2015. “Real time bidding in online digital advertisement,” in *Proceedings of the International Conference on Design Science Research in Information Systems and Technology (DESRIST),* Dublin, Ireland.

Adikari, S., Hahn, J., and Dutta, K. 2016. “Temporal feature grouping based campaign optimization in Real-time bidding digital advertising,” in *Workshop on Information Technology and Systems (WITS),* Dublin, Ireland.

Admob. 2015. “Monetize your apps intelligently ,” available at https://www.google.com/admob/.

Adopsinsider.com. 2013. “How ad serving works – Mobile vs. Web environments,” available at http://www.adopsinsider.com/ad-serving/how-ad-serving-works-mobile-vs-web-environments/.

Adspeed.com. 2010. “How to choose a good ad network?,” available at https://www.adspeed.com/Knowledges/1093/Blog/howto_choose_good_ad_network.htmladspeed.com.

Afèche, P., Araghi, M., and Baron, O. 2017. “Customer Acquisition, Retention, and Service Access Quality: Optimal Advertising, Capacity Level, and Capacity Allocation,” *the Journal of Manufacturing & Service Operations Management*, 19(4), pp. 674-691.

Agarwal, N., Athey, S., and Yang, D. 2009. “Skewed bidding in pay-per-action auctions for online advertising,” *The American Economic Review,* 99(2), pp. 441–447.

Ahmed, M. T., and Kwon, C. 2014. “Optimal contract-sizing in online display advertising for publishers with regret considerations,” *Omega,* 42(1), pp. 201–212.

von Alan, R. H., March, S. T., Park, J., and Ram, S. 2004. “Design science in information systems research,” *MIS quarterly,* 28(1), pp. 75–105.

Amiri, A., and Menon, S. 2006. “Scheduling web banner advertisements with multiple display frequencies,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* 36(2), pp. 245–251.

Asdemir, K., Kumar, N., and Jacob, V. S. 2012. “Pricing models for online advertising: CPM vs. CPC,” *Information Systems Research,* 23(3), pp. 804–822.

Bagwell, K. 2007. "The economic analysis of advertising," *Handbook of industrial organization,* (3), pp. 1701–1844.

Bea, F. 2013. "What is a fill rate in mobile advertising?," available at http://appflood.com/blog/about-mobile-ad-fill-rates.

Bertsekas, D. P. 1995. *Dynamic programming and optimal control,* (1), Athena Scientific Belmont, MA.

Bharadwaj, V., Chen, P., Ma, W., Nagarajan, C., Tomlin, J., Vassilvitskii, S., Vee, E., and Yang, J. 2012. "Shale: an efficient algorithm for allocation of guaranteed display advertising," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, pp. 1195–1203.

Bilenko, M., and Richardson, M. 2011. "Predictive client-side profiles for personalized advertising," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, USA, pp. 413–421.

Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. "Latent dirichlet allocation," *the Journal of Machine Learning Research,* (3), pp. 993–1022.

Bleier, A., and Eisenbeiss, M. 2015. "Personalized online advertising effectiveness: The interplay of what, when, and where," *Marketing Science,* 34(5), pp. 669–688.

Bojinov, H., Michalevsky, Y., Nakibly, G., and Boneh, D. 2014. "Mobile device identification via sensor fingerprinting," *arXiv preprint arXiv:1408.1416*.

Bouchachia, A., Gabrys, B., and Sahel, Z. 2007. "Overview of some incremental learning algorithms," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, London, United Kingdom pp. 1–6.

Chaffey, D. 2015. "Display advertising clickthrough rates," available at http://www.smartinsights.com/internet-advertising/internet-advertising-analytics/display-advertising-clickthrough-rates/.

Chaitanya, N., and Narahari, Y. 2012. "Optimal equilibrium bidding strategies for budget constrained bidders in sponsored search auctions," *Operational Research,* 12(3), pp. 317–343.

Chakrabarti, D., Agarwal, D., and Josifovski, V. 2008. "Contextual advertising by combining relevance with click feedback," in *Proceedings of the International Conference on World Wide Web*, Beijing, China, pp. 417–426.

Chakraborty, T., Even-Dar, E., Guha, S., Mansour, Y., and Muthukrishnan, S. 2010. "Selective call out and real time bidding," in *Proceedings of the International Workshop on Internet and Network Economics*, Stanford, USA, pp. 145–157.

Chang, C.-C., and Lin, C.-J. 2011. "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3), p. 27.

Chen, B., Yuan, S., and Wang, J. 2014. "A dynamic pricing model for unifying programmatic guarantee and real-time bidding in display advertising," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp. 1–9.

Chen, Y., Berkhin, P., Anderson, B., and Devanur, N. R. 2011. "Real-time bidding algorithms for performance-based display ad allocation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, USA, pp. 1307–1315.

Chen, Y.-J. 2017. "Optimal dynamic auctions for display advertising," *Operations Research*.

Cheng, H., Manavoglu, E., Cui, Y., Zhang, R., and Mao, J. 2012. "Dynamic ad layout revenue optimization for display advertising," in *Proceedings of the International Workshop on Data Mining for Online Advertising and Internet Economy*, Beijing, China, p. 9.

Chickering, D. M., and Heckerman, D. 2003. "Targeted advertising on the web with inventory management," *Interfaces*, 33(5), pp. 71–77.

Cortes, C., and Vapnik, V. 1995. "Support-vector networks," *Machine learning* 20(3), pp. 273–297.

Crussell, J., Stevens, R., and Chen, H. 2014. "Madfraud: Investigating ad fraud in android applications," in *Proceedings of the International Conference on Mobile Systems, Applications, and Services*, Bretton Woods, USA, pp. 123–134.

Datacratic. 2016. "RTB Optimizer.," available at https://www.datacratic.com.

Deane, J., and Agarwal, A. 2012. "Scheduling online advertisements to maximize revenue under variable display frequency," *Omega*, 40(5), pp. 562–570.

Ditzler, G., and Polikar, R. 2013. "Incremental learning of concept drift from streaming imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, 25(10), pp. 2283–2301.

Dyn.com. 2013. "5 Ways To Reduce Advertising Network Latency," available at http://dyn.com/wp-content/uploads/2013/08/5-Ways-to-Reduce-Latency.pdf.

Ebbert, J. 2012. "Define It - What is programmatic buying?," available at http://adexchanger.com/online-advertising/define-programmatic-buying/.

Econsultancy. 2013. "Online advertising survey 2013," available at https://econsultancy.com/reports/online-advertising-survey.

Edelman, B., and Brandi, W. 2015. "Risk, information, and incentives in online affiliate marketing," *Journal of Marketing Research* 52(1), pp. 1–12.

Edelman, B., Ostrovsky, M., and Schwarz, M. 2005. "Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords," *The American economic review,* 97(1), pp. 242-259.

Emarketer.com. 2014. "Driven by Facebook and Google, mobile ad market soars 105% in 2013, available at http://www.emarketer.com/Article/Driven-by-Facebook-Google-Mobile-Ad-Market-Soars-10537-2013/1010690.

Emek, Y., Feldman, M., Gamzu, I., PaesLeme, R., and Tennenholtz, M. 2014. "Signaling schemes for revenue maximization," *ACM Transactions on Economics and Computation,* 2(2), p. 5.

Evans, D. S. 2008. "The economics of the online advertising industry," *Review of Network Economics,* 7(3).

Evans, D. S. 2009. "The online advertising industry: Economics, evolution, and privacy," *the Journal of Economic Perspectives,* 23(3), pp. 37-60.

Firth, D. 1993. "Bias reduction of maximum likelihood estimates," *Biometrika,* 80(1), pp. 27–38.

Fjell, K. 2008. "Online advertising: Pay-per-view versus pay-per-click — A comment," *Journal of Revenue and Pricing Management,* 8(2–3), pp. 200–206.

Forums.makingmoneywithandroid.com. 2013. "Multiple ad networks in same app?," available at http://forums.makingmoneywithandroid.com/advertising-networks/1298-multiple-ad-networks-same-app.html.

Fridgeirsdottir, K., and Asadolahi, S. 2007. "Revenue management for online advertising: Impatient advertisers," Working paper, *London Business School*.

Friedman, M. 1937. "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association,* 32(200), pp. 675–701.

Gartner. 2013. "Forecast: Mobile App Stores, Worldwide, 2013 Update," available at https://www.gartner.com/doc/2584918/forecast-mobile-app-stores-worldwide.

Geyik, S. C., Dasdan, A., and Lee, K.-C. 2015. "User Clustering in Online Advertising via Topic Models," *arXiv preprint arXiv:1501.06595*.

Ghosh, A., McAfee, P., Papineni, K., and Vassilvitskii, S. 2009. "Bidding for representative allocations for display advertising," in *Proceedings of the International Workshop on Internet and Network Economics*, Rome, Italy, pp. 208–219.

Ghose, A., and Todri, V. 2016. "Towards a Digital Attribution Model: Measuring the Impact of Display Advertising on Online Consumer Behavior," *MIS quarterly,* 40(4)

Gilks, W. R. 2005. "Markov chain monte carlo," Wiley Online Library.

Goes, P. B. 2014. "Editor's comments: Design science research in top information systems journals," *MIS quarterly,* 38(1), pp. 3–8.

Goldfarb, A., and Tucker, C. E. 2011. "Privacy regulation and online advertising," *Management Science,* 57(1), pp. 57–71.

Grace, M. C., Zhou, W., Jiang, X., and Sadeghi, A.-R. 2012. "Unsafe exposure analysis of mobile in-app advertisements," in *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks*, Tucson, USA, pp. 101–112.

Graepel, T., Candela, J. Q., Borchert, T., and Herbrich, R. 2010. "Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine," in *Proceedings of the International Conference on Machine Learning (ICML)*, Washington, USA, pp. 13–20.

Greenacre, M., and Blasius, J. 2006. "Multiple correspondence analysis and related methods," CRC Press.

Gregor, S., and Hevner, A. R. 2013. "Positioning and presenting design science research for maximum impact," *MIS quarterly,* 37(2), pp. 337–356.

Gulyani, A. 2016. "Extensive List of Mobile Ad Network Companies 2016," available at http://gulyani.com/complete-list-of-mobile-ad-networks-companies/.

He, X., Dai, W., Cao, G., Tang, R., Yuan, M., and Yang, Q. 2015. "Mining target users for online marketing based on App Store data," in *Proceedings of the IEEE International Conference on Big Data*, Santa Clara, USA, pp. 1043–1052.

Heavlin, W. D., and Radovanovich, A. 2012. "Risk-aware revenue maximization in display advertising," in *Proceedings of the International Conference on World Wide Web*, Lyon, France, pp. 91–100.

Hegeman, J., Yan, R., and Badros, G. J. 2011. "Budget-based advertisment bidding," Google Patents.

Henderson-Sellers, B., and Gonzalez-Perez, C. 2010. "Granularity in Conceptual Modelling: Application to Metamodels," in *Proceedings of the International Conference on Conceptual Modeling*, Vancouver, Canada, pp. 219–232.

Hojjat, A., Turner, J., Cetintas, S., and Yang, J. 2014. "Delivering Guaranteed Display Ads under Reach and Frequency Requirements," in *Proceedings of the Conference on Association for the Advancement of Artificial Intelligence (AAAI),* Québec City, Canada, pp. 2278-2284.

IAB. 2013. “IAB 2013 Global mobile advertising revenue report,” available at http://www.iab.net/globalmobilerevenue.

IAB. 2014. “Openrtb API specification version 2.2. ,” available at http://www.iab.net/media/file/OpenRTBAPISpecificationVersion2_2.pdf.

Inmobi. 2015. “The discovery platform for a mobile-first world,” available at http://www.inmobi.com/.

Japkowicz, N., and Stephen, S. 2002. “The class imbalance problem: A systematic study,” *Intelligent Data analysis*, 6(5), pp. 429–449.

Jeffreys, H. 1946. “An invariant form for the prior probability in estimation problems,” in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, (186), pp. 453–461.

Ji, S., Li, W., Srivatsa, M., and Beyah, R. 2014. “Structural data de-anonymization: Quantification, practice, and implications,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, USA, pp. 1040–1053.

Jordan, P. R., and Wellman, M. P. 2010. “Designing an ad auctions game for the trading agent competition,” in *Agent-Mediated Electronic Commerce. Designing Trading Strategies and Mechanisms for Electronic Markets*, Berlin, Heidelberg, pp. 147–162.

Karande, C., Mehta, A., and Srikant, R. 2013. “Optimizing budget constrained spend in search advertising,” in *Proceedings of the ACM International Conference on Web Search and Data Mining*, Rome, Italy, pp. 697–706.

Kenett, D., and Preis, T. 2012. “Dependency network and node influence: application to the study of financial markets,” *International Journal of Bifurcation and Chaos*, *22*(7), p. 1250181.

Khalaf, S. 2013. “Flurry five-year report: It’s an app world. The web just lives in it,” available at http://flurrymobile.tumblr.com/post/115188952445/flurry-five-year-report-its-an-app-world-the.

King, G., and Zeng, L. 2001. “Logistic regression in rare events data,” *Political Analysis*, 9(2), pp. 137–163.

King, M., and Mercer, A. 1985. “Problems in determining bidding strategies,” *Journal of the Operational Research Society*, pp. 915–923.

Kohno, T., Broido, A., and Claffy, K. C. 2005. “Remote physical device fingerprinting,” *IEEE Transactions on Dependable and Secure Computing*, 2(2), pp. 93–108.

Kwon, C. 2011. “Single-period balancing of pay-per-click and pay-per-view online display advertisements,” *Journal of Revenue & Pricing Management*, 10(3), pp. 261–270.

Lambert, M. C., and Woods, K. T. 2007. “Advertiser portal interface,” Google

Patents.

Lambrecht, A., and Tucker, C. 2013. "When does retargeting work? Information specificity in online advertising," *Journal of Marketing Research,* 50(5), pp. 561–576.

Lee, K.-C., Jalali, A., and Dasdan, A. 2013. "Real time bid optimization with smooth budget delivery in online advertising," in *Proceedings of the* ACM *International Workshop on Data Mining for Online Advertising*, New York, USA, p. 1.

Lee, K., Orten, B., Dasdan, A., and Li, W. 2012. "Estimating conversion rate in display advertising from past erformance data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Beijing, China, pp. 768–776.

Lew, A., and Mauch, H. 2006. "Dynamic programming: A computational tool," (38), Springer.

Li, C., Lu, Y., Mei, Q., Wang, D., and Pandey, S. 2015. "Click-through prediction for advertising in twitter timeline," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* Sydney, Australia, pp. 1959–1968.

Li, X., and Guan, D. 2014. "Programmatic buying bidding strategies with win rate and winning price estimation in real time mobile advertising," in *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining,* Tainan, Taiwan, pp. 447–460.

Lin, C.-T., and Hsu, P.-F. 2003. "Adopting an analytic hierarchy process to select Internet advertising networks," *Marketing Intelligence & Planning,* 21(3), pp. 183–191.

Lin, Y.-L., and Chen, Y.-W. 2009. "Effects of ad types, positions, animation lengths, and exposure times on the click-through rate of animated online advertisings," *Computers & Industrial Engineering,* 57(2), pp. 580–591.

Liu, D., and Chen, J. 2006. "Designing online auctions with past performance information," *Decision Support Systems,* 42(3), pp. 1307–1320.

Liu, D., Chen, J., and Whinston, A. B. 2010. "Ex ante information and the design of keyword auctions," *Information Systems Research,* 21(1), pp. 133–153.

LLP, P. water house C. 2015. "IAB internet advertising revenue report," available at https://www.iab.com/insights/iab-internet-advertising-revenue-report-conducted-by-pricewaterhousecoopers-pwc-2/.

Mack, G. A., and Skillings, J. H. 1980. "A Friedman-type rank test for main effects in a two-factor ANOVA," *Journal of the American Statistical Association,* 75(372), pp. 947–951.

McAfee, R. P., Papineni, K., and Vassilvitskii, S. 2013. "Maximally representative allocations for guaranteed delivery advertising campaigns," *Review of Economic Design,* 17(2), pp. 83–94.

Menache, I., Ozdaglar, A., Srikant, R., and Acemoglu, D. 2009. "Dynamic online-advertising auctions as stochastic scheduling," available at https://dspace.mit.edu/handle/1721.1/87056.

Miracle, R. 2013. "Tutorial: Using multiple ad networks," available at https://coronalabs.com/blog/2013/11/19/tutorial-using-multiple-ad-networks/.

Mohri, M., and Medina, A. M. 2016. "Learning algorithms for second-price auctions with reserve," *Journal of Machine Learning Research,* 17(74), pp. 1–25.

Mookerjee, R., Kumar, S., and Mookerjee, V. S. 2016. "Optimizing Performance-Based Internet Advertisement Campaigns," *Operations Research,* 65(1), pp. 38–54.

Mungamuru, B., and Garcia-Molina, H. 2008. "Predictive pricing and revenue sharing," in *Proceedings of the International Workshop on Internet and Network Economics,* Shanghai, China, pp. 53–60.

Natarajan, N., Shin, D., and Dhillon, I. S. 2013. "Which app will you use next?: collaborative filtering with interactional context," in *Proceedings of the ACM conference on Recommender Systems*, Hong Kong, China, pp. 201–208.

Nath, S., Lin, F. X., Ravindranath, L., and Padhye, J. 2013. "SmartAds: bringing contextual ads to mobile apps," in *Proceeding of the Annual International Conference on Mobile Systems, Applications, and Services*, Taipei, Taiwan, pp. 111–124.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. 1996. "Applied linear statistical models," (4), Irwin Chicago.

Pechyony, D., Jones, R., and Li, X. 2013. "A joint optimization of incrementality and revenue to satisfy both advertiser and publisher," in *Proceedings of the International Conference on World Wide Web,* Rio de Janeiro, Brazil, pp. 123–124.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., and others. 2011. "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, 12(Oct), pp. 2825–2830.

Pepelyshev, A., Staroselskiy, Y., and Zhigljavsky, A. 2015. "Adaptive targeting for online advertisement," in *Proceedings of the International Workshop on Machine Learning, Optimization and Big Data*, Sicily, Italy, pp. 240–251.

Perlich, C., Dalessandro, B., Hook, R., Stitelman, O., Raeder, T., and Provost, F. 2012. "Bid optimizing and inventory scoring in targeted online advertising," in *Proceedings of the ACM SIGKDD International*

*Conference on Knowledge Discovery and Data Mining*, Beijing, China, pp. 804–812.

Ploner, M., Dunkler, D., Southworth, H., Heinze, G., and Heinze, M. G. 2006. "The logistf Package.," available at https://cran.rstudio.com/web/packages/logistf/logistf.pdf

Powell, W. B. 2014. "Clearing the jungle of stochastic optimization," in *Bridging Data and Decisions*, INFORMS, pp. 109–137.

Programmatic. 2016. "How to optimize RTB campaigns?," available at http://programmatic.cz/docs/Programmatic_explained_EN.pdf

PubMatic. 2016. "RTB and Yield Optimization," available at https://pubmatic.com/solutions/automated-buying-selling/

Radovanović, A., and Zeevi, A. 2010. "Revenue maximization through smart inventory management in reservation-based online advertising," *ACM SIGMETRICS Performance Evaluation Review*, 38(2), pp. 33–35.

Rainey, C. 2016. "Dealing with separation in Logistic regression models," *Political Analysis*, 24(3), pp. 339–355.

Regelson, M., and Fain, D. 2006. "Predicting click-through rate using keyword clusters," in *Proceedings of the Workshop on Sponsored Search Auctions*, p.9623.

Rencher, A. C. 2003. "Methods of multivariate analysis," (492), John Wiley & Sons.

Richardson, M., Dominowska, E., and Ragno, R. 2007. "Predicting clicks: estimating the click-through rate for new ads," in *Proceedings of the International Conference on World Wide Web*, Banff, Canada, pp. 521–530.

Rodgers, S., and Thorson, E. 2000. "The interactive advertising model: How users perceive and process online ads," *Journal of Interactive Advertising*, 1(1), pp. 41–60.

Roels, G., and Fridgeirsdottir, K. 2009. "Dynamic revenue management for online display advertising," *Journal of Revenue & Pricing Management*, 8(5), pp. 452–466.

Rogers, A., David, E., Payne, T. R., and Jennings, N. R. 2007. "An advanced bidding agent for advertisement selection on public displays," in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, Honolulu, Hawai, p. 51.

Ruiz, I. M., Nagappan, M., Adams, B., Berger, T., Dienst, S., and Hassan, A. 2014. "On the relationship between the number of ad libraries in an Android app and its rating," *IEEE Software*, (1), p. 1.

Sahel, Z., Bouchachia, A., Gabrys, B., and Rogers, P. 2007. "Adaptive mechanisms for classification problems with drifting data," in

*Knowledge-Based Intelligent Information and Engineering Systems*, Berlin, Germany, pp. 419–426.

Salganicoff, M. 1993. "Density-adaptive learning and forgetting," in *Proceedings of the International Conference on Machine Learning (ICML)*, Amherst, USA, pp. 276–283.

Scocco, D. 2014. "How many ad networks are there?," available at http://www.dailyblogtips.com/how-many-ad-networks-are-there/.

Shan, L., Lin, L., Sun, C., and Wang, X. 2016. "Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization," *Electronic Commerce Research and Applications*, (16), pp. 30–42.

Shankar, V., and Hollinger, M. 2007. "Online and mobile advertising: current scenario, emerging trends, and future directions," *Marketing Science Institute*, 31(3), pp. 7–206.

Shen, H., Li, Y., and Chen, Y. 2014. "Robust ad delivery plan for guaranteed display advertising," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA)*, Hulun Buir, China, pp. 1125–1130.

Shen, J., Geyik, S. C., and Dasdan, A. 2015. "Effective audience extension in online advertising," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, NSW, Australia, pp. 2099–2108.

Shen, Q., and Miguel Villas-Boas, J. 2017. "Behavior-Based Advertising," *Management Science*,

Shi, K., and Ali, K. 2012. "Getjar mobile application recommendations with very sparse datasets," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, Beijing, China, pp. 204–212.

stackoverflow.com. 2013. "Integrating multiple ad networks on iOS and Android app," available at http://stackoverflow.com/questions/18474215/integrating-multiple-ad-networks-on-ios-and-android-app/1847624.

Statista. 2014. "Display advertising average display time monthly in the United Kingdom (UK) from October 2010 to April 2014 (in seconds)," available at http://www.statista.com/statistics/313289/display-advertising-average-display-rate-in-the-united-kingdom/.

Statista. 2015a. "Global advertising revenue from 2007 to 2016 (in billion U.S. dollars)," available at http://www.statista.com/statistics/237797/total-global-advertising-revenue/.

Statista. 2015b. "Number of apps available in leading app stores as of May 2015," available at http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

Statista. 2016. “Real-time bidding digital advertising revenues in the United States from 2013 to 2018, by channel (in billion U.S. dollars),” available at http://www.statista.com/statistics/411489/rtb-digital-ad-revenues-by-channel-usa/.

Stevens, R., Gibler, C., Crussell, J., Erickson, J., and Chen, H. 2012. “Investigating user privacy in android ad libraries,” in *Proceedings of the Workshop on Mobile Security Technologies (MoST)*, San Francisco, USA, p. 10.

Sun, Z., Dawande, M., Janakiraman, G., & Mookerjee, V. S. 2016. The Making of a Good Impression: Information Hiding in Ad Exchanges. *MIS Quarterly*, 40(3), pp. 717-739.

Sun, Z., Dawande, M., Janakiraman, G., and Mookerjee, V. 2017. “Not Just a Fad: Optimal Sequencing in Mobile In-App Advertising,” *Information Systems Research*.

Tagami, Y., Ono, S., Yamamoto, K., Tsukamoto, K., and Tajima, A. 2013. “Ctr prediction for contextual advertising: Learning-to-rank approach,” in *Proceedings of the International Workshop on Data Mining for Online Advertising*, New York, USA, p. 4.

Toptenreviews.com. 2016. “2016 Best internet service provider comparisons and teviews,” available at http://isp-review.toptenreviews.com/.

Tucker, C. E. 2014. “Social networks, personalized advertising, and privacy controls,” *Journal of Marketing Research*, 51(5), pp. 546–562.

Turner, J. 2012. “The planning of guaranteed targeted display advertising,” *Operations research*, 60(1), pp. 18–33.

Unni, R., and Harmon, R. 2007. “Perceived effectiveness of push vs. pull mobile location based advertising,” *Journal of Interactive advertising*, 7(2), pp. 28–40.

Vallina-Rodriguez, N., Shah, J., Finamore, A., Grunenberger, Y., Papagiannaki, K., Haddadi, H., and Crowcroft, J. 2012. “Breaking for commercials: characterizing mobile advertising,” in *Proceedings of the ACM Conference on Internet Measurement Conference*, Boston, USA, pp. 343–356.

Viscuso, G. 2013. “Multiple ad networks with one SDK on Android,” available at https://dzone.com/articles/multiple-ad-networks-one-sdk.

Wang, X., Li, W., Cui, Y., Zhang, R., and Mao, J. 2010. “Click-through rate estimation for rare events in online advertising,” *Online Multimedia Advertising: Techniques and Technologies*, pp. 1–12.

Wei, X., & Croft, W. B. (2006). “LDA-based document models for ad-hoc retrieval,” in *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*, Seattle, USA, pp.

178-185.

Weide, K. 2011. “Real-Time bidding in the United States and western Erope, 2010–2015,” available at http://info.pubmatic.com/rs/pubmatic/images/IDC_Real-Time Bidding_US_Western Europe_Oct2011.pdf.

Widmer, G., and Kubat, M. 1996. “Learning in the presence of concept drift and hidden contexts,” *Machine learning,* 23(1), pp. 69–101.

Wilbur, K. C., and Zhu, Y. 2009. “Click fraud,” *Marketing Science,* 28(2), pp. 293–308.

Yahalom, S., and Stopel, D. 2011. “Bidding for impressions,” Google Patents.

Yan, B., and Chen, G. 2011. “AppJoy: personalized mobile application discovery,” in *Proceedings of the International Conference on Mobile Systems, Applications, and Services,* Washington, USA, pp. 113–126.

Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., and Chen, Z. 2009. “How much can behavioral targeting help online advertising?,” in *Proceedings of the International Conference on World Wide Web,* Madrid, Spain, pp. 261–270.

Yang, J., Vee, E., Vassilvitskii, S., Tomlin, J., Shanmugasundaram, J., Anastasakos, T., and Kennedy, O. 2010. “Inventory allocation for online graphical display advertising,” *arXiv preprint arXiv:1008.3551*.

Yuan, S., Abidin, A. Z., Sloan, M., and Wang, J. 2012. “Internet advertising: An interplay among advertisers, online publishers, ad exchanges and web users,” *arXiv preprint arXiv:1206.1754*.

Yuan, S., and Wang, J. 2012. “Sequential selection of correlated ads by pomdps,” in *Proceedings of the ACM International Conference on Information and Knowledge Management,* Kuala Lumpur, Malaysia, pp. 515–524.

Yuan, S., Wang, J., and Zhao, X. 2013. “Real-time bidding for online advertising: measurement and analysis,” in *Proceedings of the ACM International Workshop on Data Mining for Online Advertising,* New York, USA, p. 3.

Yuan, Y., Wang, F., Li, J., and Qin, R. 2014. “A survey on real time bidding advertising,” in *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI),* Qingdao, China, pp. 418–423.

Zhang, C., Shao, X., and Li, D. 2013. “Knowledge-based Support Vector Classification Based on C-SVC,” *Procedia Computer Science,* (17), pp. 1083–1090.

Zhang, W., Chen, L., and Wang, J. 2016. “Implicit Look-alike Modelling in Display Ads: Transfer Collaborative Filtering to CTR Estimation,” *arXiv preprint arXiv:1601.02377*.

Zhang, W., Yuan, S., and Wang, J. 2014. "Optimal real-time bidding for display advertising," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, USA, pp. 1077–1086.

Zhu, Z. A., Chen, W., Minka, T., Zhu, C., and Chen, Z. 2010. "A novel click model and its applications to online advertising," in *Proceedings of the ACM International Conference on Web Search and Data Mining*, New York, USA, pp. 321–330.

# Appendix A

**Sample Bid request**


{"bidRequest":"
{\"app\":
{\"cat\":[\"IAB1\"],\"id\":\"65821984\",\"name\":\"Singapore_iOS_FamilySafe\",\"publisher\":
{\"id\":\"923867039\",\"name\":\"inmobi\"}
},\"at\":2,\"badv\":[\"king.com\"],\"bcat\":[\"IAB7-28\",\"IAB19-30\",\"IAB22-1\",\"IAB17-18\",\"IAB26\",\"IAB25\",\"IAB24\",\"IAB9-9\",\"IAB7-42\",\"IAB23\"],\"device\":
{\"connectiontype\":0,\"devicetype\":1,\"geo\":
{\"country\":\"SGP\",\"lat\":1.3061,\"lon\":103.833,\"type\":1
},\"ip\":\"14.100.132.124\",\"js\":0,\"make\":\"Apple\",\"model\":\"iPhone\",\"os\":\"iOS\",\"osv\":\"7.0\",\"ua\":\"Mozilla/5.0 (iPhone; CPU iPhone OS 7_0_2 like Mac OS X) AppleWebKit/537.51.1 (KHTML, like Gecko) Mobile/11A501\"
},\"ext\":
{\"carriername\":\"SingTel Mobile\",\"coppa\":0,\"operaminibrowser\":0,\"udi\":
{\"idfa\":\"4B3B9A08-EC8B-4E49-823C-B5A270CA240C\",\"idfatracking\":1
}
},\"id\":\"DqrVB5bDrJ\",\"imp\":[
{\"banner\":{\"battr\":[1,3,5,8,9],\"btype\":[3],\"h\":50,\"mimes\":[\"image/gif\",\"image/jpeg\",\"image/png\",\"text/html\",\"text/plain\"],\"pos\":0,\"w\":320
},\"displaymanager\":\"SOMA\",\"id\":\"1\",\"instl\":0
}],\"user\":
{\"gender\":\"M\",\"keywords\":\"Social Networking,Lifestyle\"
}
}","mwBidReqid":"smaato-DqrVB5bDrJ","exchangeName":"Smaato","hadBidResponse":false,"isTimeout":false,"location":"SG","datetime":1402235413234
}