Cross-Device Identity Resolution using Machine Learning: A Scalable Device Graph Approach

Manasa Priya Koduri, Pei Xuan Lim, Zhen Li, Sandeep Kumar, Muhammad Aamir Saleem, Roger Moy
Consumer Insights and Analytics, Mediacorp Pte Ltd
1 Stars Avenue, Singapore 138507

Abstract

This paper outlines how Mediacorp (Mediacorp 2021), Singapore's public service broadcaster, addresses its cross-device identity challenges using a scalable device graph approach. Research in this area is relevant to the domain of advertising technology as it enables a holistic view of consumers that can be extended to use cases such as improving advertisement targeting, personalized recommendations and demographic predictions. Past research efforts were limited to high-level descriptions of the steps undertaken to create a one-off, static device graph based on data collected over a circumscribed time frame, thus limiting its use in larger-scale commercial applications. In this paper, we propose a scalable solution that enables continuous, incremental revisions of our device graph. We leverage behavioral data captured by Mediacorp across its sites and platforms to build a richer device graph that is updated weekly. First, we introduce additional features and explore various classifiers to improve pairwise probability scores between devices that are likely to belong to the same user. Then, we apply community clustering algorithms to uncover device communities to establish the final device graph. Extensive experiments showed that our additive approach has consistently delivered >90% precision and recall in real-world applications.

1 Introduction

Targeted online advertising is increasingly harder to achieve as users move seamlessly across multiple devices daily. For example, users may have separate work and personal phones or laptops that they use interchangeably throughout the work day. When they return home in the evening, they could continue browsing using their tablets, PCs and smart TVs. While the use of cookies and various advertising identifiers (e.g. Apple's Identifier For Advertisers [IDFA], Google's Advertising ID [GAID], Microsoft's advertising ID) does enable tracking at the device level, *device-level identification* of users is hardly optimal since each identifier is treated as a unique user. Except for logged-in users with a consistent single sign-on identity across platforms, the same users with different identifiers are essentially considered different users. To add further complexity, identity fragmentation could also occur within each device, particularly on mobile devices where multiple identifiers could be assigned depending on the platform utilized. Mobile browsers, mobile apps and embedded browsers within mobile apps can present different identifiers, resulting in splintered identities of the same user. Such fragmentation hampers both user experience and advertising effectiveness. For example, there is wastage when the user is repeatedly served the same advertisement across devices and likely more than the stipulated frequency cap. There are also missed opportunities to enhance user experience when the user traverses across devices. A device graph solution could help alleviate some of these challenges by identifying *inter-device* and *intra-device* linkages to form a unified identity of each user.

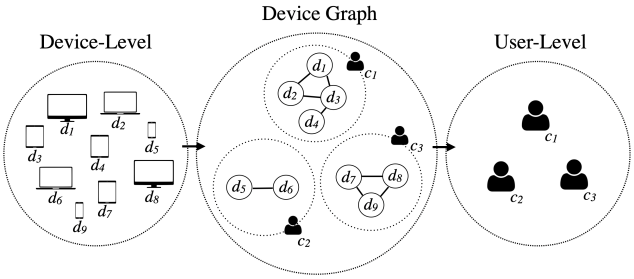

Figure 1: Resolving user identity: Transitioning from device-level to user-level view using device graph.

A device graph comprises nodes that represent each identifier (individual device identifiers or intra-device identifiers) and edges that represent relationships between nodes. There are two approaches to calibrating these relationships – deterministic or probabilistic. The deterministic approach relies on logins or other personally identifiable information available to link multiple identifiers to a single profile confidently. The probabilistic approach, however, *infers* these linkages by using behavioral, activity-based and other data. With an established device graph, we are no longer targeting devices, but actual users (Figure 1). With this new user-centric perspective, an advertisement can now be served more judiciously to the same user across devices by reinforcing the impression with the right frequency and context. Advertising wastage is reduced and more informed media planning decisions can be made.

In this paper, we demonstrate the effectiveness of our ap-

proach using real-world data in production. First-party data gathered across major properties owned by Mediacorp was used in the exercise. We referenced existing methodologies (Malloy et al. 2017; Funkhouser et al. 2018) that utilized IP-related features to generate the device graph. We implemented a new layer in the graph construction process by generating additional behavioral features (e.g. browsing patterns, interests) which were used to train a classifier to predict the pairwise probabilities of devices belonging to the same user. By leveraging methodologies already outlined in existing literature, our experimentation focus is on producing continuously updated device graphs that are more commercially practicable. Training was conducted on Single Sign-On (SSO) data within Mediacorp's data ecosystem. The SSO user pool serves as the ground truth in our evaluation.

The main contributions of this paper include:

- Establishment of a scalable customer identification framework, which enables incremental updates of a master device graph. To our knowledge, this is the first time such a framework has been proposed.
- Building a commercial device graph solution of higher precision and recall with lower latency.
- Application of deep-learning techniques to device-pair classification.

In Section 2, we summarize key learnings from existing literature. Section 3 outlines step-by-step our methodology undertaken to derive a functional and scalable device graph. In Section 4, we describe the datasets, experiments, evaluation and implementation results of our additive approach. Finally, we discuss future areas of potential investigation in Section 5.

2 Related Work

There are only a handful of related research papers in the domain of device graphs. Several organizations (Adbrain 2020; Tapad 2020; Lotame 2020) have been providing identity resolution products since early 2010. For commercial reasons, there was rarely mention of the algorithms used or methodological details behind the products beyond a high-level description.

Methodologies presented by (Malloy et al. 2017; Funkhouser et al. 2018) were built on the concept of IP co-location, by observing co-occurrences of IDs from an IP address at a specific point in time. IP co-location graphs formed the foundation of their device graphs. Various community clustering algorithms are then applied to the IP co-location graphs to detect *communities*, each representing a single user or household. Due to the scale of such graphs, greedy approaches are favored as exhaustive searches are computationally costly.

Classification models were used to predict relationships between devices. Beyond IP and device-related features (Volkova 2017), additional learning features could be gleaned from browsing logs and associated meta-data (Tran 2016; Funkhouser et al. 2018) that may be useful. To improve the classifier, (Tran 2016) proposed several features such as domains visited, actions taken, and time spent while browsing. Such data were readily available in our case. We generated a unique set of learning features, including viewing patterns and interests amongst others, to supplement our IP co-location graph.

One of the biggest challenges of developing device graphs is accurately validating the user-device linkages. In this regard, (Funkhouser et al. 2018) used data from user panels operated by Comscore as a source of ground truth, realized through specialized monitoring software that provided data such as unique hardware identifiers of devices and associated IDs. Such data was unavailable in our context and infeasible for commercial implementation due to the amount of third-party involvement required. Instead, we relied on actual SSO data of users that have logged into Mediacorp's network of sites to validate our results regularly.

From the literature, there also appeared to be limited testing of generated graphs. For example, in (Malloy et al. 2017; Funkhouser et al. 2018), graphs were built using data from Comscore's digital census network consisting of 700 million records collected over six weeks in the United States, but were not evaluated regularly over a period of time. Graphs appeared to be one-off output and were not updated continuously to capture new identifiers.

3 Methodology

Building our device graph involved three key steps. First, we constructed an IP co-location graph, G, based on *pairwise IP co-locations* of IDs. Next, we extracted activity-based, event-based and behavioral features of each pair of IDs in G and updated the pairwise weights based on a *classification model* trained on these features. Finally, we clustered the IDs into appropriate groupings using *community clustering* algorithms. Details are given below.

3.1 IP Co-location Graph

Based on the intuition that devices that shared IP addresses were more likely to be related, we began generating the IP co-location graph by collating device data (*device ID, IP address, time*) over a defined look-back period. Each day, for every IP address in the dataset, an edge was created between each pair of device IDs that appeared together on that IP address on that day.

The weight of the edge is the inverse of the total number of distinct devices, N, that appeared on the IP address over the day. To reduce noise, IP addresses with more than N_{max} distinct devices were disregarded as they were more likely to be public IPs. In our experiments, we defined $N_{max} = 30$. The final weight between two devices is a summation of weights over the look-back period, across all IP addresses.

3.2 Feature Generation

IP co-location can occur between devices within a household, office, or public space. While an IP co-location graph served to identify instances of co-location, it did not allow us to factor other potential relationships between IDs. To better predict the probability that two devices were related, a classification model was built using additional features on top of IP co-location weights.

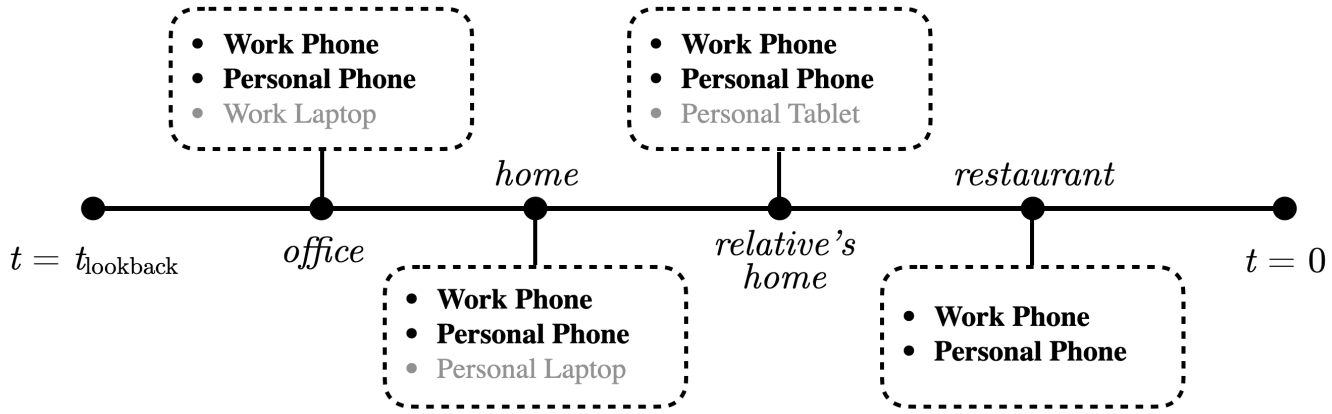

Figure 2: Example of IP co-location patterns of a user's devices. Resultant edge between the user's 'Work Phone'and 'Personal Phone'will be relatively higher due to higher frequency of co-occurrence.

IP-based features IP features were generated based on data of the IP addresses accessed by each device. These features conveyed information on how the IP addresses were shared between the devices. First, statistics like the number of IPs accessed by each device, the number of common IPs accessed by each pair of devices, and the total number of IPs accessed by each pair of devices were derived. Then, additional features like Dice and Jaccard similarity, overlap coefficient, ratio of commonly accessed IPs, and number of concurrent IPs accessed were computed.

Activity-based features Similar to IP-based features, activity-based features were also extracted. Here, we compared the active days of each device.

Event-based features We noticed that the majority of device pairs had only one shared IP address. There could be a few reasons for this observation. Users may not have moved or changed locations during the look-back period, or users had not been accessing the network from personal mobility devices (for example, accessing from PCs/TVs instead of laptops/tablets/phones). There was also the possibility that the co-location occurred by chance. To differentiate these scenarios, we introduced additional features derived from a combination of IP and activity-based events. We generated these features by computing the ratio of common IPs and active days of each pair against the total number of events.

Behavioral features Attributes were extracted for each device and aggregated over the look-back period from the browsing logs (Table 1). For each device pair, we calculated similarity measures such as cosine similarity, L1 norm, and max norm as learning features at each attribute level. To account for different content languages, various natural language processing techniques were also employed to extract user interests for each device from the different sites they had visited.

3.3 Classification Model

For each pair of IDs, an updated pairwise probability score was computed with the above features. We experimented with LightGBM, RandomForest, and Neural Network classifiers and selected the best-performing model to derive the probability scores.

LightGBM (Ke et al. 2017) is a Gradient Boosted Decision Tree (GBDT) built upon weak learners having high bias and low variance since weak learners are typically shallow trees. Gradient boosting iteratively reduces error mainly by rectifying bias. Compared to other popular GBDTs like XGBoost (Chen et al. 2015), LightGBM speeds up the training process by up to 20 folds while achieving similar accuracy. A RandomForest (Liaw, Wiener, and others 2002) is typically built upon fully-grown decision trees having low bias and high variance. It aggregates output from many uncorrelated models, thus helping to reduce variance. However, it cannot reduce bias and as such, the bias of a RandomForest may not be lower than that of an individual decision tree. We also trained a Neural Network classifier for comparison with these machine learning approaches. We adopted an optimal network structure based on 6 hidden layers with 200, 100, 50, 25, 50 and 100 neurons.

3.4 Community Clustering

After establishing the probabilistic relationships between pairs of devices, we progressed with different community clustering approaches (affinity propagation, label propagation, and connected components) to build the final device graph.

Affinity Propagation (Frey and Dueck 2007) takes in the pairwise probability scores between data points and finds clusters by maximizing the total similarity between data points and their surrounding points. Label Propagation (Xiaojin and Zoubin 2002) is an iterative algorithm that propagates labels throughout the dataset with the assumption that closer data points tend to have similar class labels. The last approach uses Connected Components, which is an exhaustive search of all data points connected. To granularize the resulting communities, final community sizes was fine-tuned by varying the probability thresholds.

Table 1: Behavioral features

Attribute Name	Description
browser language	Browser language set on the device E.g. Chinese, English, Malay etc.
site	Distribution of Mediacorp's sites accessed on the device
media language	Distribution of language of the content visited on the device [Chinese, English, Malay, Tamil]
interests	Interests are extracted based on the content visited E.g. Entertainment, News, Education etc.
hour	Distribution of page/video views during each hour of the day
weekday	Distribution of page/video views during each day of the week
device model	Device manufacturer. E.g. Apple, Samsung, LG etc.
platform	Device platform. E.g. Mobile, PC, Tablet, TV
browser	Browser of the device from which user accessed the content. E.g. Google Chrome, Safari etc.
operating system	Device operating system. E.g. Android, iOS, macOS, Windows etc.
time spent	Average time spent on different sites
video completion pattern	Distribution of completion milestone of video [25%, 50%, 75%, 90%, 100%]
viewing pattern	Distribution of visit frequency
page access pattern	Distribution of number of pages accessed in a visit
referral traffic	Distribution of referrer source to the content. E.g. Search, Social, Internal, Direct etc.

4 Evaluation

In this section, we describe the results of our experimentation and implementation process. Our goal is to demonstrate the performance of our methodology using real-world data.

4.1 Dataset and Protocols

Mediacorp operates a suite of TV channels, radio stations, and multiple digital platforms. Data collected from digital platforms comprise of IDs, IP addresses, timestamps and other browsing details. If a user is logged in, all device IDs will be linked to the user's SSO ID deterministically. If a user is not logged in, each device ID will be assumed to be from a different user. As such, we aim to link the device IDs to reflect inter-device as well as intra-device relationships. SSO IDs are deterministic and hence, will serve as our ground truth data (validation set). Models were trained using data collected from May to August 2020. Evaluation was conducted weekly over the following nine weeks.

4.2 Evaluation Metrics

For evaluation, we computed the precision and recall of the generated graph with a methodology similar to (Funkhouser et al. 2018), using the graph generated from the SSO IDs as ground truth. Let N_g be the number of SSO users, H be the community of devices corresponding to a user, u. Each SSO ID community was hence identified as $H_{u,g}$ (ground truth community). We then compared this with $H_{u,t}$, the community generated from our device graph (generated community).

For each user U, different devices D were captured in the ground truth data and generated graphs. For example, ground truth community $H_{u,g}$ consisted of d_1, d_2, d_4 and generated community $H_{u,t}$ consisted of d_1, d_2, d_3, d_5. Based on the above, precision and recall scores were computed in Eq. 1 and Eq. 2 as follows:

$$Precision = \frac{1}{|N_g|} \sum_u Precision_u \tag{1}$$

where

$$Precision_u = \frac{|H_{u,g} \cap H_{u,t}|}{|H_{u,t}|}$$

$$Recall = \frac{1}{|N_g|} \sum_u Recall_u \tag{2}$$

where

$$Recall_u = \frac{|H_{u,g} \cap H_{u,t}|}{|H_{u,g}|}$$

In addition, we have defined an additional metric *coverage*, which represents the percentage of devices included in the device graph. Let C be the number of communities formed in the device graph, S_{mean} be the average size of the communities and N be the total devices count. *Coverage* is defined as:

$$Coverage = \frac{C \times S_{mean} \times 100\%}{N} \tag{3}$$

High precision and recall scores, coupled with high coverage score, would be ideal as this would indicate that the device graph had managed to link a large proportion of the devices correctly. Low coverage, however, would suggest a large proportion of devices had not been linked successfully. In this scenario, high precision and recall scores would be less meaningful.

4.3 Experiments

Device Pair Classification We experimented with the LightGBM, RandomForest, and Neural Network algorithms and compared the macro-averaged F1-scores on testing data comprising of device pairs. Results are shown in Table 2.

Though results were largely similar, we decided to use the Neural Network classifier due to marginally better performance.

Table 2: Results of classification models

Macro-Averaged	F1	Precision	Recall
LightGBM	0.93	0.95	0.92
RandomForest	0.92	0.91	0.93
Neural Network	0.94	0.94	0.94

Community Clustering Due to sheer size of the data, affinity propagation did not scale well and took over 16 hours to generate 4,000 clusters. In real-world deployment, this may not be ideal as the device graphs will need to be updated periodically in our proposed approach. With the connected components method, we varied the probabilistic weight threshold to evaluate the results. Table 3 shows the precision and recall values of the validation set at various thresholds. As the threshold increased, precision increased but recall decreased. $threshold = 0.5$ was selected for its higher recall and coverage, with negligible precision trade-off.

Table 3: Results of connected components approach

Threshold	**0.5**	0.7	0.9
Precision	**0.983**	0.992	0.997
Recall	**0.931**	0.878	0.784

Preliminary evaluation showed that the connected components approach yielded better results and scaled better at the same time. As such, the connected components approach was eventually chosen for production. Table 4 is a summary of evaluation results using different clustering approaches.

Table 4: Results of community clustering algorithms

Algorithm	Precision	Recall
IP co-location	0.596	0.995
Affinity Propagation	0.832	0.914
Label Propagation	0.957	0.563
Connected Components	**0.983**	**0.931**

4.4 Implementation

Existing studies tested their methodologies by using static device graphs. For commercial purposes, we believe that our device graphs will need to be continuously updated to remain robust and effective. To achieve this, we performed incremental weekly updates to the device graph by repeating the same process over a 14-day look-back period. For each generated community, we assign a new community ID (PeopleID) that has a one-to-many relationship with the existing device-level IDs. If a generated community already exists in the current device graph, they are merged. Otherwise, the new community is added to the graph as shown in Figure 3. This way, we will be able to account for new IDs by incrementally assigning them to a PeopleID. This ensures that the device graph remains updated and relevant.

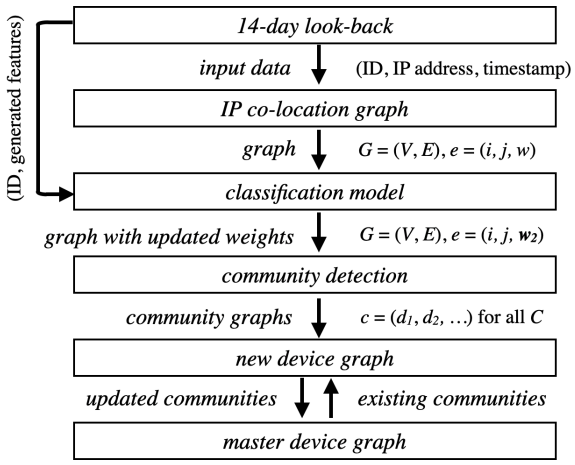

Figure 3: Weekly device graph update process

Table 5: Overall ID distribution two months after implementation

Total Unique Device IDs	15,808,740
Total Unique SSO Device IDs	979,873
Coverage	6.19%
Device IDs included in IP co-location Graph	9,684,073
Coverage	61.25%
Device IDs included in Device Graph	6,922,044
Coverage	43.78%

Results There were a total of 15,808,740 unique Device IDs whose activity lifetime was at least one day. Out of these, 979,873 of them were deterministically linked to SSO IDs, representing coverage of 6.19% before application of the device graph. After generating the IP co-location graph, 9,684,073 (61.25%) device IDs were subsequently linked. The IP co-location graph only generated links based on device IDs that appeared on the same IP address at the same time, subject to conditions such as disregarding IPs with large number of connections as these could be public IPs (e.g. offices, shopping malls). A total of 6,922,044 (43.78%) device IDs were linked in the final device graph after the classification and community clustering steps. The device graph enabled a seven-fold increase in linked devices from 6.19% (SSO IDs only) to 43.78% after deployment.

We further analysed the effectiveness of the device graph in actual advertising campaigns by measuring *deduplication*. We define *deduplication* as:

$$Deduplication = (1 - \frac{Y}{X + \delta X}) \times 100\% \qquad (4)$$

where X is the number of devices to which an advertisement was served, Y is the number of users that the devices are linked to and δX are the additional devices that potentially belong to Y according to the device graph. The deduplication rate represents the potential decrease in related devices that the advertisement needs to be served to reach the same users. A higher deduplication rate lowers the frequency of

unwanted ad exposure. Across ten recent campaigns, there was an average of 49% deduplication rate achieved in our experiments.

Table 6: Statistics of device graph two months after implementation

No. of Clusters	1,609,111
Avg. Cluster Size	4.301
Std. Cluster Size	4.075

Beyond ground truth data, we also benchmarked the community statistics of our device graph (Table 6). While we are unable to evaluate if the average number of clusters were representative of the population, a previous survey on global device ownership (Buckle 2016) estimated the average number of devices per user at 3.64. This suggests that our average cluster size of 4.301 is a reasonable figure.

5 Summary and Future Work

A reliable device graph is fundamental to cross-device identity resolution in a world of fragmented devices. In this paper, we demonstrated a practical approach to maintaining an updated device graph in a scalable manner. Our weekly additive approach to device graph construction also ensures our graph remains comprehensive, without incurring significant overheads.

We introduced an additional classification layer in our methodology that leverages not only IP-related features but also behavioral features to better improve linkages between devices. Our approach has produced consistent results on an ongoing basis with respect to the validation set users. While it is virtually impossible to ascertain that the generated device graph is an accurate representation of the population (such data does not exist in reality), we can infer that the output graph is reasonable based on available empirical benchmarks.

In the same vein, future work may include gathering more third-party data to better evaluate constructed device graphs. Such data could be collected via surveys, telcos, and other external collaborations focused on users within the country of study. Further experimentation on new datasets could also help determine the portability of our proposed methodology. With device graphs, a holistic set of behavioral features of users across linked devices are also now accessible. This could benefit other adjacent areas of investigation such as personalization, content recommendations, and demographic predictions.

Despite all the attendant benefits of a device graph, industry developments may require our approach to evolve. Consumer concerns have led many players in the industry to roll out new privacy measures. For example, Google (Google 2020) has recently announced plans to phase out third-party cookie support, joining other popular browsers like Safari and Firefox. While our approach relies mainly on first-party data and such privacy-driven measures may not immediately impact our baseline, nonetheless we will need to continue monitoring developments in the industry at large and adapt accordingly.

References

Adbrain. 2020. Solving customer identity. Available at https://www.adbrain.com (Accessed: 2020/08/04).

Buckle, C. 2016. Digital consumers own 3.64 connected devices. Available at https://blog.globalwebindex.com/chart-of-the-day/digital-consumers-own-3-64-connected-devices/ (Accessed: 2020/10/04).

Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2* 1(4).

Frey, B., and Dueck, D. 2007. Clustering by passing messages between data points. *Science* 315:972 – 976.

Funkhouser, K.; Malloy, M.; Alp, E.; Poon, P.; and Barford, P. 2018. Device graphing by example. 273–282.

Google. 2020. Building a more private web: A path towards making third party cookies obsolete. Available at https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html (Accessed: 2020/10/05).

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Neural Information Processing Systems*.

Liaw, A.; Wiener, M.; et al. 2002. Classification and regression by random forest. *R news* 2(3):18–22.

Lotame. 2020. Lotame cartographer. Available at https://www.lotame.com/products/lotame-connect/people-based-id-solution/ (Accessed: 2020/11/13).

Malloy, M.; Barford, P.; Alp, E.; Koller, J.; and Jewell, A. 2017. Internet device graphs. 1913–1921.

Mediacorp. 2021. Mediacorp. Available at https://www.mediacorp.sg/en/ (Accessed: 2021/03/29).

Tapad. 2020. The tapad graph. Available at https://www.tapad.com/the-tapad-graph (Accessed: 2020/08/02).

Tran, N. K. 2016. Classification and learning-to-rank approaches for cross-device matching at cikm cup 2016. *arXiv preprint:1612.07117*.

Volkova, E. 2017. Cross-device tracking with machine learning. Master's thesis.

Xiaojin, Z., and Zoubin, G. 2002. Learning from labeled and unlabeled data with label propagation. *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*.