2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

Overcoming Data Sparsity in Predicting User Characteristics from Behavior through Graph Embeddings

Munira Syed
University of Notre Dame
msyed2@nd.edu

Daheng Wang
University of Notre Dame
dwang8@nd.edu

Meng Jiang
University of Notre Dame
mjiang2@nd.edu

Oliver Conway
Conde Nast, NY
oliver_conway@condenast.com

Vishal Juneja
Conde Nast, NY
vishal_juneja@condenast.com

Sriram Subramanian
Conde Nast, NY
sriram_subramanian@condenast.com

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

***Abstract*—Understanding user characteristics such as demographic information is useful for the personalization of online content promoted to users. However, it is difficult to obtain such data for each user visiting the website. Since demographic data for some users can be collected, their behavior can be used to predict the attributes of unknown users. Through online news consumption, we can infer the attributes of users from the articles they view. Most existing models take a supervised learning approach to this modeling task. However, by representing the user-URL interactions with a network, we can convert it to a semi-supervised learning problem and learn embeddings for users. Graph embeddings have become popular in recent years, with research mainly focusing on algorithmic developments. However, while we have an intuitive understanding of the problems they may overcome, such as data sparsity, this problem remains unexplored in the domain of demographic prediction using behavior. In this paper, we first investigate the effectiveness of using user embeddings generated from network representation learning for prediction by comparing its performance with other traditional feature sets, including content and item-based features. We find that the embeddings can represent a user generally on two prediction tasks, (1) gender prediction (classification) and (2) age prediction (regression). Second, we explore the advantages of using these embeddings over the other methods in two cases of data sparsity, where (1) the training and testing sets of users are temporally split and (2) the user labels are imbalanced. In both these cases, the embeddings outperform the baseline.**

***Index Terms*—graph embeddings, demographic prediction, network representation learning**

I. Introduction

To improve the user experience, content-hosting organizations strive to personalize the content delivered to users in their news feeds, email newsletters, and advertisements. One such example of personalization is behaviorally targeted advertising [12, 20]. However, for such personalization, user demographic information such as age and gender is often useful [20]. Companies can collect demographic data from some of the users, but, for the majority of content consumers on these sites, demographic information is difficult to obtain [6, 12, 20]. In response, demographic prediction techniques have been proposed in several studies [6, 12, 13, 17, 20], with most of these works focusing on feature engineering. The reason for this effort towards exploring features is that users are not directly represented in their behavior. While we can get a list of clicks on webpages from users, and metadata associated with those webpages, such as content, category, and topic, aggregating them to represent a user is not trivial and can be done in many ways. Thus, for demographic data prediction, multiple methods have been proposed for user representation. While click-level data can be used to predict users' demographics, we focus on item-level predictions as the insights are more useful to the content creators and curators.

Network representation learning has become very popular for classification, recommendation, and link prediction problems recently with deployments even at the industry scale [7]. However, their application towards demographic modeling is underexplored. The methods used in the literature either use different feature sets [6, 20] or bipartite graphs [12] with gender predicted through a Bayesian framework. In this paper, we provide deeper insights into how embeddings learned from a bipartite graph perform in cases of data sparsity in the domain of online news consumption. We will explore and demonstrate the effectiveness of representing users with embeddings learned from network representation with the help of two research questions.

RQ1: *How do embeddings perform in comparison to other feature sets?* First, we will show that user embeddings are at par or better than other methods of representing users, including item-level and content-based representation.

RQ2: *Do the user embeddings suffer from the effects of data sparsity?* Through this research question, we argue that when the data is sparse (temporal split and imbalance), the user embeddings outperform other representations.

II. Related Work

In previous works, a variety of features are used for predicting users' demographic information. Browsing history can

IEEE/ACM ASONAM 2020, December 7-10, 2020
978-1-7281-1056-1/20/$31.00 © 2020 IEEE

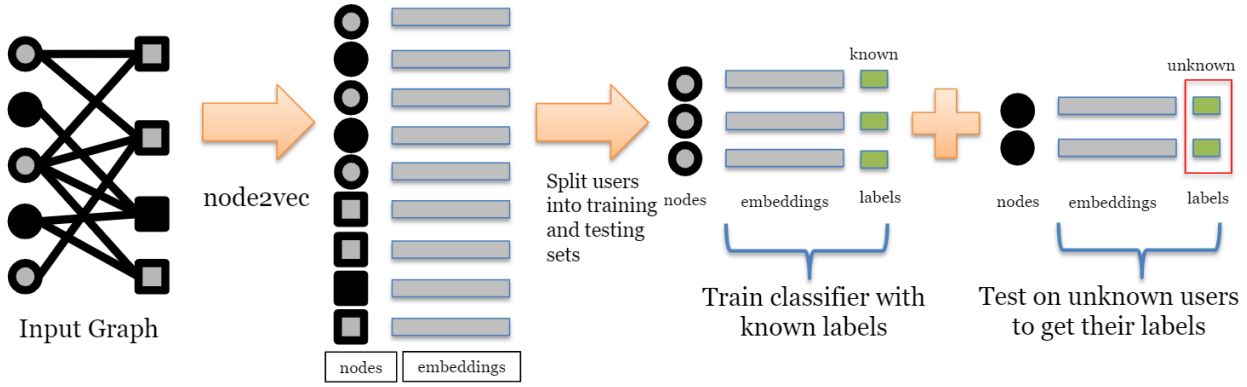

Fig. 1. Pipeline. The circular nodes represent users and the squares are URLs. The solid black nodes are unlabeled, whereas the gray nodes are labeled.

be used to learn about unknown users. In general, three types of browsing information are used to predict various user attributes: click features such as the number of clicks and timestamp [6, 20], item-level features such as the products, URLs, and items viewed by users [6, 12], and content-based features such as the text of articles and search queries [3, 13]. The two kinds of content-based features are the content generated by users [22] and the content consumed [20]. However, only 8% of the users create content (blogs) on the Internet [12]. Thus, content consumption-based features cover a larger user-base whose demographics can be predicted. While a previous study proposes a Bayesian framework to predict a user's age and gender [12], other works mainly focus on user representation with well-established models for prediction [6, 13, 17, 20].

With the recent advancements in representation learning on graphs [23], we can use the network of users and URLs for user representation. Deepwalk [19], Node2vec [10] and Metapath2vec [5] use random walks for generating embeddings. LINE [21] generates embeddings by preserving the first and second order proximities between pairs of nodes.

One of our data sparsity problems is imbalance, in which fewer samples of minority users exist in the dataset. A way of dealing with the imbalance problem is resampling the data. Various resampling techniques exist to improve predictive performance on data with imbalanced class distributions [18] in a supervised classification problem. However, the idea of using semi-supervised learning for imbalanced classification has been explored in some other works. For example, Li et. al. [14] uses a transductive semi-supervised learning method for their imbalance problem. Other work on this topic exists in gene function prediction [8, 9]. However, the imbalance problem is unexplored in the behavior modeling domain.

TABLE I
10-FOLD CROSS-VALIDATION WITH DIFFERENT FEATURE SETS.

Features	Type	F1 Score	Accuracy	AUROC
Node2vec	**UE**	**0.660 (0.0223)**	**0.661 (0.0226)**	**0.718 (0.0258)**
Deepwalk	**UE**	**0.669 (0.0223)**	**0.670 (0.0222)**	**0.727 (0.0260)**
Metapath2vec	**UE**	**0.662 (0.0498)**	**0.665 (0.0522)**	**0.720 (0.0621)**
Title Words	CB	0.649 (0.0047)	0.652 (0.0049)	0.698 (0.0034)
LDA 150	CB	0.604 (0.0036)	0.605 (0.0036)	0.637 (0.0050)
NMF 1500	CB	0.618 (0.0032)	0.620 (0.0034)	0.660 (0.0028)
Node2vec + Title Words	HG	0.665 (0.0219)	0.667 (0.0218)	0.724 (0.025)
Node2vec + LDA 150	HG	0.662 (0.0216)	0.663 (0.0218)	0.720 (0.0250)
Top URLs + NMF 1500	HG	0.666 (0.0044)	0.666 (0.0045)	0.722 (0.0030)
Top URLs	**IL**	**0.669 (0.0048)**	**0.669 (0.0038)**	**0.723 (0.0030)**

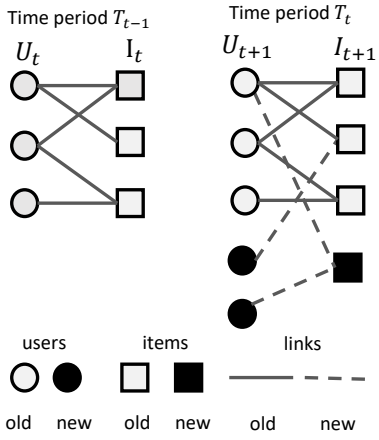

Fig. 2. Temporal Split

III. Data Description

A. Dataset

We use the clickstream data of page views on an online magazine for our experiments. In all the experiments, we retained clicks by users with at least 10 pageviews for consistent performance. For the experiments in RQ1, we use data from May 2018. We use the gender of 84,380 users in which 49.35% are female and the age of 64,102 users with a mean age of 58.73 and a standard deviation of 15.97. Thus, while the distribution of gender is quite balanced, there are more users in the range of 60-80 years. For RQ2, in the week-wise split experiments and imbalance experiments, we use the data from May. In the month-long split experiment, we use the click data

from May 2018 for training and June 2018 for testing.

B. Features

1) Item-level features based models (IL): We can represent users as a feature vector of URL items. The user representation matrix U is given with the indicator matrix I, in which the URLs that the user views are indicated by counts. Let I have the dimensions $n * m$, where n is the number of users and m is the total number of URLs that are present in the click dataset. **Top URLs** is the set of most popular URLs based on the number of total clicks on them.

2) User embeddings (UE): We generate user embeddings by training a bipartite graph of users-items using node2vec. For our experiments, we use the same hyperparameter setting throughout with 100 walks per source of length 20 each, and a window size of 10. The p and q parameters of node2vec are set to the default of 1 because they did not affect performance significantly. We also use Principal Component Analysis (PCA) [1] in RQ2.We compare PCA with node2vec as both of them use clicks from both the training and testing for generating embeddings sets unlike the traditional features. The features included are **Node2vec** embeddings generated from the user-URL bipartite graph, **Metapath2vec [5]** embeddings generated from the user-URL bipartite graph using the metapath user-item-user, **Deepwalk [19]** embeddings generated using from the user-URL bipartite graph, and **PCA** with 128 components, same as the size of the embedding vector generated through node2vec.

3) Content-based features (CB): We represent the URLs themselves in terms of their content as matrix C of dimensions $m * r$, where r is the dimension of the URL-based features (e.g., r words or r topics). In this case, the user matrix U is represented by $U = I.C$ and has the dimensions $n * r$. Thus, the user representation is calculated by averaging the item representation of all the items that are associated with the user. This becomes the user feature vector U and is the input to the classification model. We also train an LDA (Latent Dirichlet Allocation) [4] and NMF (Non-negative Matrix Factorization) [2] model to derive the topic probabilities of documents and represent users by averaging the probabilities of the articles that each user reads. The features used in this category are **Title Words** which is a bag-of-Words representation of articles from titles, **LDA 150** (LDA topic model with 150 topics), and **NMF 1500** (NMF topic model with 1500 topics), where the number of topics were chosen experimentally.

C. Heterogeneous Features (HG)

We can also incorporate other features into the feature vector by concatenating (augmenting) them to the user matrix U. For example, if matrix L of size $n * l$ denotes the user-location matrix with l total locations, the user representation would be given as $[U|L]$ and have dimensions $n * (m + l)$. In Table I, the heterogeneous features have a $+$ in the Features column and are generated through concatenation.

IV. Model Description

A. Base Model

After experimenting with different models, we selected Logistic Regression for gender prediction and Beta Regression for age prediction since user ages lie between 0 and 100. We measure performance on gender prediction mainly with accuracy, and also report on f1 score and area under receiver operator characteristics (AUROC). For age prediction, a regression problem, we use mean squared error (mse), r^2, mean absolute error (mae) and root mean squared error (rmse). For cross-validated experiments, the mean of the metrics is reported with the standard deviation in parentheses.

B. User Embeddings

We generate a bipartite graph $G(V, E)$ shown in Figure 1 where V are the user and URL vertices, and the edge E between a user and a URL exists if the user viewed that URL. We generate node embeddings by the following steps as outlined by node2vec [10]: (1) Random walk on the graph with a fixed walking length (20) starting at each vertex and no. of walks per starting vertex (100). (2) Input these random walks into the word2vec [16] model with a given context size (10) to generate embeddings for each node. (3) Train a classifier with the embeddings of the training vertices as the input and predict gender based on the embeddings of the testing vertices. The generated embeddings can be used in the baseline predictors for training and testing, as shown in Figure 1.

V. Analysis

A. Comparison of Different Feature Sets

To answer RQ1, we focus on gender prediction for space considerations. Table I shows the performance of different feature sets for gender prediction. Through experimentation, we discovered that only using the top 8% most popular URLs to represent users in the Top URLs representation achieves the best performance and slightly drops if we include more URLs, likely due to logistic regression overfitting on a highly dimensional feature vector. Table I lists some of the best-performing feature sets. Overall in both age and gender prediction, the performance of the embedding methods deepwalk, node2vec, and metapath2vec are at par with Top URLs, the best performing features among the traditional features. The best performing threshold for age prediction is at 6% of the most viewed URLs. We see that node2vec features get a slight boost when combined with content features, suggesting that models incorporating heterogeneous features have the potential to outperform models that only use one type of feature.

TABLE II
Age Features

Metrics	Node2vec	Top URLs	Metapath2vec	Deepwalk
mse	201.6 (9.4)	205.6 (5.8)	198.6 (2.4)	197.5 (3.3)
r2	0.208 (0.04)	0.197 (0.02)	0.221 (0.01)	0.226 (0.01)
mae	11.2 (0.3)	11.3 (0.2)	11.1 (0.1)	11.0 (0.1)
rmse	14.1 (0.3)	14.3 (0.2)	14.1 (0.1)	14.1 (0.1)

B. Data Sparsity Conditions

In this section, we answer RQ2. In the domain of user modeling, the data available may be sparse, either because users do not visit many URL items or because fewer users of a particular demographic group engage with the website. These problems impact the performance of models on certain predictive tasks. We describe two examples of tasks impacted by data sparsity and showcase how traditional supervised learning frameworks succumb to these issues, while the user embeddings in our framework are resilient to them. Since Top URLs are the best performing of the traditional features and node2vec has the lowest performance among the embedding methods, we use those for our experiments.

1) Temporally split training and testing set users: Given some users whose demographic information we have, we can infer the demographic information about unknown users that access the same articles. However, since new articles are released weekly, the time duration in which a URL is actively viewed is limited. When this time period is smaller than the time window in which we train and test user features for prediction, item-level features are no longer feasible for use due to very few users in the future viewing them. Thus, the sparsity issue of user-URL clicks, in which the number of users viewing URLs follows a power-law distribution, is exacerbated when a temporal split is considered. This problem is motivated by the observation that days apart from each other have a smaller intersection of URL clicks than closer days [11, 15].

Let the users at time t be represented by U_t and the URLs at time t be represent by I_t as shown in Figure 2. First, we compare two of the best feature sets on the cross-validated experiments, namely Top URLs and node2vec embeddings, which use structural rather than content-based features.

TABLE III
10-FOLD CROSS-VALIDATION WEEK-WISE FOR GENDER (ACCURACY)

Week	Top URLs	Node2vec	PCA
1	0.596 (0.008)	0.589 (0.019)	0.589 (0.007)
2	0.604 (0.006)	0.597 (0.014)	0.601 (0.005)
3	0.648 (0.004)	0.645 (0.019)	0.647 (0.003)
4	0.620 (0.005)	0.618 (0.014)	0.606 (0.006)

TABLE IV
10-FOLD CROSS-VALIDATION WEEK-WISE FOR AGE (MSE)

Week	Top URLs	Node2vec	PCA
1	231.1 (7.42)	228.3 (9.19)	227.4 (4.45)
2	227.2 (7.56)	222.4 (6.68)	225.9 (4.59)
3	223.6 (4.26)	219.5 (6.16)	221.3 (4.71)
4	230.4 (5.37)	224.9 (4.297)	224.4 (5.38)

Tables III and IV shows these results for gender and age prediction respectively. For node2vec and PCA, embeddings are generated using the behavior of the individual week only. We see that all the methods have similar performance. Then, we train on users in an earlier week and test on users in a later time period. Tables V and VI shows these results, where each row shows the prediction result across a single training-testing split based on time. We find that Top URLs drops in performance more dramatically than node2vec when the training and testing time periods are different for both gender and age prediction problems. The last column N2v_strict shows the performance of node2vec if we exclude nodes corresponding to URLs that only appear in the testing week(s). The performance drops considerably in both gender and age experiments, which emphasizes to us the importance of including all clicks in the model.

TABLE V
WEEK-WISE SPLIT FOR AGE (MSE)

Train on	Test on	Top URLs	Node2vec	PCA	N2v_strict
Week 1	Week 2	279.4	225.9	273.4	240.4
Week 1	Week 2+3	278.7	220.4	252.2	245.6
Week 2	Week 3	235.5	221.4	227.4	239.0
Week 2	Week 3+4	242.3	217.7	225.9	239.0
Week 3	Week 4	279.3	234.2	245.6	246.1
May	June	256.9	221.1	231.7	242.1

TABLE VI
WEEK-WISE SINGLE-SPLIT FOR GENDER (ACCURACY)

Train on	Test on	Top URLs	Node2Vec	PCA	N2v_strict
Week 1	Week 2	0.553	0.629	0.554	0.557
Week 1	Week 2+3	0.553	0.613	0.603	0.552
Week 2	Week 3	0.602	0.623	0.628	0.578
Week 2	Week 3+4	0.580	0.630	0.628	0.579
Week 3	Week 4	0.584	0.609	0.599	0.585
May	June	0.540	0.639	0.605	0.575

2) Imbalanced Classification: Imbalanced classification problems suffer from a sparsity in the number of minority class samples. We evaluate the performance of embeddings on imbalanced data by artificially creating an imbalance in the dataset. We subsample male users such that the ratio of males to all users is 0.10 and systematically repeat the experiments 9 times with different subsets of male users. From Figure 3 we see that for most subsets, node2vec outperforms the baseline classifier. The error bars indicate the standard deviation. Thus,

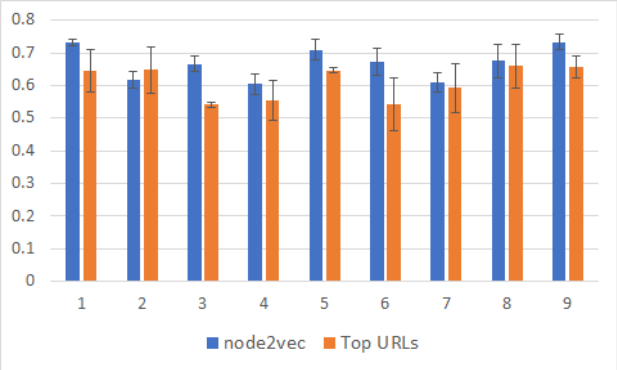

Fig. 3. Imbalanced Classification comparing baseline (top URLs) and node2vec on a 10-fold cross-validation

node2vec does not suffer from the imbalance problem as much as the Top URLs features. Since the node2vec embeddings are trained on all the users, more data is used to generate embeddings compared to the Top URLs feature vector.

VI. Conclusion

With a bipartite graph constructed from the user-URL data, we converted the problem from a supervised learning problem with URL features to a semi-supervised node classification problem. The URL information may be more predictive than the actual content possibly because many URLs are shared on social media networking or other means which have a higher demographic bias. In that case, an article could have a higher occurrence in a demographic group, irrespective of the content. We also discovered that only the most popular URLs are necessary for gender and age prediction tasks, which is good for scalability.

From RQ1, we saw that user embeddings can represent users well. Since these embeddings were generated in an unsupervised fashion, they could theoretically be used for any downstream application. We consider the two prediction tasks – gender (classification) and age (regression), and observe that the user embeddings perform well in both these tasks. Thus, we see that the user embeddings were general enough to fit well to different tasks. However, some shortcomings in using this method could potentially be addressed in the future. Most of the existing network representation techniques are transductive and thus cannot be trained on streaming data. Another disadvantage is that there is an explicit training stage for the user representation, which is not required with some other techniques, such as Top URLs. So in situations where network representation would not offer an advantage, it may be more prudent to use less expensive methods of user representation. Nonetheless, more sophisticated modeling that leverages the heterogeneous data could potentially improve performance compared to simpler network representation techniques.

References

[1] H. Abdi and L. J. Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4), 2010.

[2] S. Arora, R. Ge, and A. Moitra. Learning Topic Models–Going Beyond SVD. In *2012 IEEE 53rd Annual Symposium on Found. of Comput. Sci.* IEEE, 2012.

[3] B. Bi, M. Shokouhi, M. Kosinski, and T. Graepel. Inferring the demographics of search users: Social data meets search queries. In *Proc. 22nd Int. Conf. WWW*. ACM, 2013.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *NIPS*, 2002.

[5] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proc. 23rd ACM SIGKDD*, pages 135–144, 2017.

[6] D. Duong, H. Tan, and S. Pham. Customer gender prediction based on e-commerce data. In *2016 8th Int. Conf. KSE*. IEEE, 2016.

[7] C. Eksombatchai, P. Jindal, J. Z. Liu, Y. Liu, R. Sharma, C. Sugnet, M. Ulrich, and J. Leskovec. Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In *TheWebConf*. Int. World Wide Web Conf. Steering Committee, 2018.

[8] M. Frasca, A. Bertoni, M. Re, and G. Valentini. A neural network algorithm for semi-supervised node label learning from unbalanced data. *Neural Networks*, 43, 2013.

[9] M. Frasca, A. Bertoni, and G. Valentini. An unbalance-aware network integration method for gene function prediction. In *MLSB 2013-Machine Learning for Systems Biol.*, 2013.

[10] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In *Proc. 22nd ACM SIGKDD*. ACM, 2016.

[11] J. A. Gulla, C. Marco, A. D. Fidjestøl, J. E. Ingvaldsen, and Ö. Özgöbek. The intricacies of time in news recommendation. In *UMAP (Extended Proceedings)*, 2016.

[12] J. Hu, H.-J. Zeng, H. Li, C. Niu, and Z. Chen. Demographic prediction based on user's browsing behavior. In *Proc. 16th Int. Conf. WWW*. ACM, 2007.

[13] S. Kabbur, E.-H. Han, and G. Karypis. Content-based methods for predicting web-site demographic attributes. In *2010 IEEE ICDM*. IEEE, 2010.

[14] F. Li, C. Yu, N. Yang, F. Xia, G. Li, and F. Kaveh-Yazdy. Iterative nearest neighborhood oversampling in semisupervised learning from imbalanced data. *The Scientific World J.*, 2013, 2013.

[15] C. Lin, R. Xie, X. Guan, L. Li, and T. Li. Personalized news recommendation via implicit social experts. *Inf. Sci.*, 254, 2014.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[17] J. Misztal-Radecka. Building semantic user profile for polish web news portal. *Comput. Sci.*, 19, 2018.

[18] A. More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

[19] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proc. 20th ACM SIGKDD*. ACM, 2014.

[20] T. M. Phuong. Gender prediction using browsing history. In *Knowledge and Systems Eng.* Springer, 2014.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proc. 24th Intl. Conf. WWW*, 2015.

[22] C. Zhang and P. Zhang. Predicting gender from blog posts. *Univ. Massachussetts Amherst, USA*, 2010.

[23] D. Zhang, J. Yin, X. Zhu, and C. Zhang. Network representation learning: A survey. *IEEE TBDATA*, 2018.