CUsTOMEr: a novel Customer chUrn predicTion methOd based on Mobile application usagE

Nielsen L. R. Machado
Faculty of Informatics - FACIN
Pontifical Catholic University of
Rio Grande do Sul - PUCRS
Brazil, 90619-900
Email: nielsen.machado@acad.pucrs.br

Duncan D. A. Ruiz
Faculty of Informatics - FACIN
Pontifical Catholic University of
Rio Grande do Sul - PUCRS
Brazil, 90619-900
Email: duncan.ruiz@pucrs.br

***Abstract*—Technological innovation and fierce dispute to conquer the mobile device market make manufacturers and telecommunication companies increase their attention to the loyalty of their clients. In this sense, customer churn prediction is particularly important, and recently received more attention from aforementioned companies, mainly those that merchandise mobile devices. Besides helping users in their daily activities, these devices provide to manufacturers a large amount of data, as application usage and battery consumption. It is important for companies to know how customers use their applications. Automated tools are needed to collect this large amount of data and help companies to understand customer's behavior throughout time, seeking to prevent customer abandonment. Based on this, this work proposes an innovative and effective method to manage and predict customer churn based on mobile application usage. In order to investigate the distribution of data throughout time, the proposed method processes application usage data, based on techniques of data stream clustering (where customers are grouped by their behavior) and based on novelty detection. Our tests show an accuracy rate of 87%, a satisfactory performance in this context.**

***Keywords-component*—Customer Churn Prediction; Mobile App Usage; Data Stream Clustering; Novelty Detection; Concept Drift Detection.**

I. Introduction

Customer churn is the term used by a variety of business to denote the movement of a customer from a service provider to another. There are different reasons for this movement, such as service cost and quality, and loyalty to a service provider is directly related to those criteria. In practice, the cost of attracting a new customer is 5 to 6 higher than to keeping an already loyal customer [1]. As a result, companies are always under pressure to reduce customer churn. Customer churn prediction is particularly important and has recently received more attention from a variety of industries such as smartphone manufacturers and telecommunication (telecomm) operators. Advances in these industries such as new services and technologies, as well as advances in the areas of data mining and machine learning, increased the competition in the market. Since customers are the ”source material” of these companies, new methods for detecting customer churn are vital to the survival and development of aforementioned companies [1].

Mobile devices are wireless by definition, which helps in their popularization and omnipresence in everyone's everyday routines. Today smartphones are produced with an ever increasing number of sensors (e.g. accelerometer, compass, gyroscope, etc), and these sensors are capable of producing large amounts of data [2]. One key component of smartphones are their applications (*apps*), programs which are capable of a wide variety of tasks, such as making video calls, playing music, watching videos, and so on [3]. An *app* can be opened, used and closed many times in a single day. From such tasks we can capture each app launch count (i.e the number of times in which the app is used) and the time use of each of them. The number of *apps* available is steadily growing: on Google Play Store and Apple Store iOS, the number of *apps* available in 2015 was four times larger than in 2011 [4]. Data deriving from those *apps* are continuously being generated, making difficult to store and analyze them [2]. The research community [3] points out that it is still difficult to access data from mobile devices, despite their inherent potential: mobile data are usually private or not available due to privacy-preserving regulations [3].

Keeping a customer loyal is one of the most important issues to be addressed by smartphone manufacturers. Methods for tackling this issue consider customer churn prediction as a classification problem, since the available data are mainly based on static data analysis usually known *a priori* [5]. Furthermore, customer churn prediction, as a classification task, considers customers as independent individuals [6].

The large amount of data issued by mobile devices can be approached in a Data Stream fashion. Gama in [7] defines data stream as stochastic processes in that events occur continuously and independently of each other. Such data usually have a large volume, a large variety and are produced at a high velocity (i.e. ongoing and in real time). However, it is necessary to receive this mobile devices' data, from a large number of users, as soon as they are made available. An effective tool, able to perform information analysis and capable of helping researches and manufacturers to extract knowledge about these data streams, is almost mandatory [2]. In this sense, we approach the customer churn prediction as a data stream clustering problem. Data stream clustering can

978-1-5090-4372-9/17/$31.00 ©2017 IEEE

arrange sets of customers into groups (concepts), based on their mobile *app* usage behavior [8]. In a data stream it is important to investigate the changes in its distribution, namely Concept Drift. In this sense, new concepts arise and known concepts may disappear or evolve, which can be discovered applying Novelty Detection techniques [8].

In this paper, we propose an innovative and effective method to manage and predict customer churn, based on mobile *app* usage, employing data stream clustering and novelty detection techniques. Specifically, these techniques: (a) deal with different types of data from mobile *app* usage, (b) group customers based on their behavior, (c) detect churn behavior by identifying concept drifts, and (d) find the best predictive model that will help to prevent customer turnover. We perform our experiments on a simulated dataset that resembles the patterns we found in our primary collected data. The owner of the primary data, an international smartphone manufacturer which sponsors this research, prohibits third-party access to sensitive (i.e. not anonymous) data.

This paper is organized as follows. Section II reviews related work. Section III presents the proposed method and its phases. In Section IV we show the tests carried out and the results. Finally, Section V draws the conclusions and future work.

II. Related Work

Related work includes mobile device usage and customer churn prediction in telecomm scenario. Bohmer et al. [9] presents the use of *apps* in different contexts (e.g. hours and places). For instance, some apps have intense spikes in related context (e.g. music and social *apps*). The author aims to use its findings to further design an *app* recommendation system. Tang et al. [10] shows a launcher called *iLauncher* in order to lay out *apps* based on each users' individual usage patterns. However, those studies, albeit presenting useful insight on mobile *app* usage from data stream, do not address customer churn prediction.

Verkasalo [11] shows that users make use of some types of *apps* in specific contexts. For example, browsers and media *apps* (photos, videos) are more used when they people are on the move, whereas videogames are played when they are at home. Froehlich et al. [12] shows a framework that collects quantitative and qualitative data on people's personal devices in order to support studies of mobile technology usage and evaluation. However, both studies presents either a small number of users, *apps*, time window or a combination of them. In addition, these studies do not address customer churn prediction.

Li et al. [5] proposes a method to predict customer churn in telecoms, aiming to solve the problem of imbalanced data present in some datasets available. The method uses static datasets from UCI and Orange, that does not aim to analyze user behavior over time windows. Such datasets do not allow handling time intervals, thus making impossible to find drifting concepts. Furthermore, the mobile *app* usage scenario depicted is also not taken into account. On the other hand, Dasgubta et al. [13] presents a technique that predicts potential customer churn by examining the current set of churners and their underlying social network. The author uses a call data records, which is an informative source on how customer communicates. However, there is no information about mobile *app* usage in this type of data, a flaw also present in the work of Li et al.

To the best of our knowledge, the approach proposed in this work is the first of its kind. We combine data stream with the measuring of mobile *app* usage and predict churn based on novelty detection and concept drift. We explore (1) data streams, (2) a large number of users, (3) a large number of *apps*, (4) the detection of concept drift, and (5) the prediction of customer churn, using a long time window.

III. The CUsTOMEr

The exploitation of mobile *app* usage and investigation of concept drift aims to identify customer behaviors and establish different customers profiles. In this context, data stream clustering can help to analyze and answer such situations, allowing manufacturers companies to keep their customers by offering new services and products. The Activity Diagram in Figure 1 shows the pipeline for our method. It can be summarised by two main steps, namely *Data Stream Clustering* and *Customer Churn Prediction*. Both steps are composed of two phases, *Inception / Identification* and *Analysis / Prediction*, respectively.

A. Data Stream Clustering Step

In this step we aim to understand the data stream, and find out the customers distribution.

1) Inception Phase: In the *Inception* phase, all data from the data stream are continuously summarised and preprocessed for dealing with space and memory constraints. Note that some parameters are defined in this phase: (i) the size of the time window that will cover the most recent data; (ii) a starting time; and (iii) an ending time for the data to be captured. The execution procedure of this phase is:

1) Definition of $window\ time$ (i.e. a day, week or month), which is set according to the available computational resources. The $window\ time$ can be variable or fixed. Definition of the $start$ window points to the beginning of customers monitoring, and the definition of an end window indicates when the customers will no longer be monitored.
2) Definition of the number of customers N to be monitored from the $start$ window until the end window. This number can be variable or fixed and will be set according the available computational resources.
3) Initialization of data structures considering that the entire data stream cannot be stored in the main memory.
4) For each $window$ from $start$ to end, data stream events need to be summarised into the $data\ structure$, since not all events from the data stream can be stored in main memory at the same time.

Performing step 4 is a key step to preserve the meaning of all events without actually storing them [14]. Moreover,

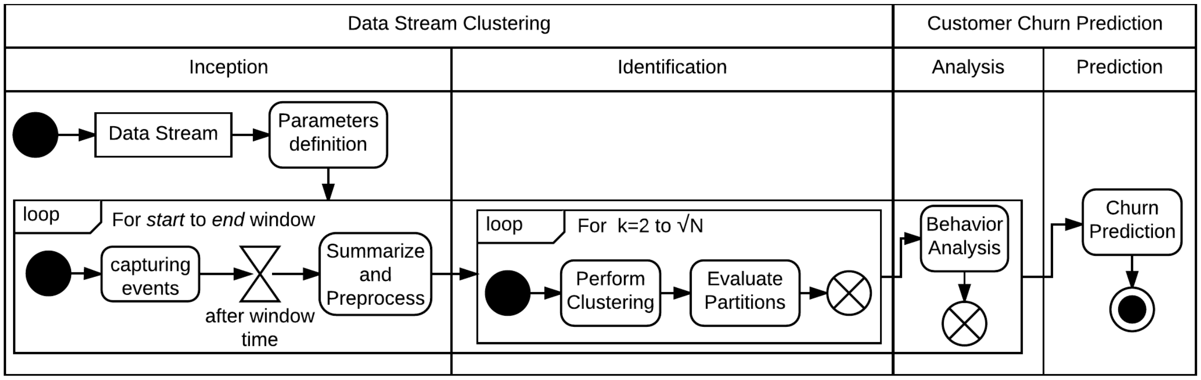

Fig. 1. The Activity Diagram of the proposed method.

preprocessing techniques may be performed aiming at the treatment of data if necessary.

2) Identification Phase: After performing the *Inception* phase, clustering algorithms receive as input the previously generated summary statistics in the *Identification* phase. Initially, the *Identification* is performed with the defined time window. Clustering algorithms will produce several partitions, with varying number of groups. This phase is carried out to ensure the choice of the best partition for each time window. The execution procedure of this phase is:

1) For each window from the $start$ to the $end\ window$, for k = 2 to $\sqrt{N}$ (where N is the number of customers chosen [15]), use the a given clustering algorithm $\sqrt{N}-1$ times.
2) Collect partitions and cluster centroids for each one of the algorithm's execution.
3) Evaluate partitions using chosen validity criteria.
4) Select the partition which performs the best in the validity criteria.

At the end of this phase, all customers have been mapped to one of the k clusters. Each one of the groups' prototypes represents a typical customer behavior. By analysing such behaviors, one can employ techniques able to identify possible churn candidates. In order to tackle this, we consider that churning is an atypical behavior, and describe how we aim to identify it in the next step.

B. Customer Churn Prediction Step

In this step we aim to predict the likelihood of a given customer abandoning his or her smartphone in the future, at the *Prediction Phase*, based on the definition of churn behaviors in the *Analysis* Phase.

1) Analysis Phase: The *Analysis* phase is responsible for the novelty detection. It allows the detection of both concept drift and potential churn behaviors based on the life curve of each customer. We define a life curve as the sequence of profiles (i.e. partition) of a customer over the analyzed time windows. The execution procedure of this phase is:

1) Create a data structure D to receive and store likely churn behaviors that may be found in steps 4 and 5.
2) For each *window*, from $start$ to end, select all labels from the best partition with k groups, chosen by its performance in the validity criteria. These labels represent customer clusters and their profiles in the window.
3) After processing all predefined windows, create a life cycle l for each customer.
4) For each l, if the curve l has a C (a *change* in the cluster profile to which the customer belongs), between two or more consecutive windows, select the l to be stored in D. If the curve has no C then go to the next life cycle.
5) After selecting all life cycles that have C, define the ones that are *outliers curves* and *churn curves*.

Normal customers (i.e. presents *loyal curves*) are the ones that belong to a group with similar *app* usage behavior, throughout several time windows. *Outlier customers* are the ones that can change from a group to another throughout the time windows. It is difficult to group all *outlier customers* in a single group after the first time window because their profiles are changing on each window. Finally, *churn curves* can be characterized by the migration of a customer from a group to another within a time windows. This behavior may be interpreted as a difficulty in the adaptation of the customer with the product, or a dissatisfaction.

2) Prediction Phase: In this last phase, potential churn behaviors previously selected from life cycles are used to predict the likelihood of each customer churning or not. The execution procedure of this phase is:

1) Select the N customers and D data structures produced in the previously phase.
2) For each n in N and for each *churn curve* in D, verify if the customer n presents one of the *churn curves* in D by comparing the customer's life cycle l with the $churn\ curves$ held in D.
3) If the customer n presents one of the *churn curves* in D, then the customer n is labeled as a potential churn.

IV. Tests

The state-of-the-art in customer churn prediction is mainly based on static data analysis, which does not correspond to real conditions. In a real-world scenario, many customers carry

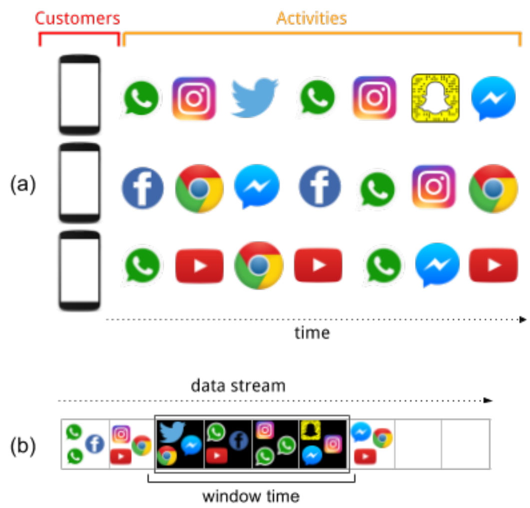

Fig. 2. Real-world mobile *app* usage data stream representation. (a) Different customers perform various activities. (b) Activities are captured for different time windows.

out activities through a mobile device (see Figure 2). Each data stream event corresponds to one out of several activities produced by those costumers (Figure 2 (a)).

Such activities are performed by different customers, with a variable volume of usage, depending on the user we are analyzing (Figure 2 (b)). Thus, it is evident that each event (*app* activity) can not be considered as a complete representation of an independent object (customer); However, this is how traditional data stream algorithms interpret them.

In this section, in order to mimic the problem of customer churn in the manufacturer scenario, we analyze a data stream from an UCI dataset about Taxi Service Trajectory[1] [16]. This data stream represents the launching count of *apps* from a real-world scenario provided by our sponsor. We test the effectiveness of our method in this dataset due to privacy and sensitiveness issues imposed by our sponsor company in its dataset (i.e. our primary data source).

A. Data description

The data stream used is from a taxi company from the city of Porto, Portugal. Such data stream contains 442 taxis and each data chunk arrives with the following attributes:

- TRIP_ID: a unique identifier for each trip;
- CALL_TYPE: it may be either *A*, if the trip was demanded from the telephonic central; *B*, if the trip was demanded directly to a taxi driver at a given stand; or *C* for none of these cases;
- ORIGIN_STAND: a unique identifier for the starting location of the trip if CALL_TYPE is *B*;
- TAXI_ID: a unique identifier for each taxi;
- TIMESTAMP: starting time for a trip;
- ORIGIN_CALL, DAYTYPE, MISSING_DATA and POLYLINE were not used in our test.

[1] https://goo.gl/fSTuZ7

The data were continuously collected over a period of twelve months, from July 2013 to June 2014. We selected the taxis that have starting trips from one of the 63 possible taxi default locations (i.e. taxi stands). Because of existing regulations in the city, the drivers must choose a specific stand to wait for passengers and are forbidden to randomly searching for passengers. The mapping from mobile customers to taxis is done as follows:

Observation 1: We map *Customers* to taxis, since taxis stay at some stands one or more times in given day, and customers also use mobile *apps* one or more times in a given day.

Observation 2: We map *apps* to taxi stands, since the relation of taxis-to-stands is many-to-many (i.e. a stand serves many taxis in a day, and a taxi may be served by several stands in a day), as well as the relation of customers-to-apps is many-to-many (i.e. a customer uses several *apps* in a day, and an *app* can be used by several users in a day).

Observation 3: We map the *app launch count* to the number of times that a taxi picks up some passenger in a given stand. Is important to note that, in a real-world scenario, we have a large number of *apps*, but few of them are effectively used [17]. In this case, many *apps* are used by a small number of users, not producing a minimum number of useful activities.

B. Inception Phase

We select rides that were directly obtained at the stands (CALL_TYPE B) as input/output. We also chose a month (i.e. 30 days) as the size of the $window\ time$. In addition, due to Portugal's academic calendar, we define September 2013 as the $start$ window and January 2014 as the end window. Additionally, we decide to not consider all 442 available taxis, since that some of them have less than 3 months of data available. In this sense, we analyze 412 taxis (93%) that have more than 3 months of data between September 2013 and January 2014. Finally, we define a data matrix to summarise available data, where objects are *customers* (i.e. 412 taxis), features are *apps* (i.e. 63 stands) and each taxi have its ride count (i.e. *app launching count*) for each one of the taxi stands it was served. In Table I we present the total number of stands each taxi used to wait for passengers for each time window (i.e. month).

After summarizing the data we perform different preprocessing procedures – such as standardization, normalization and logarithmic transformation [18] – since some predictors are sensitive to feature scales (i.e. maximum and minimum values), even though these scales may be arbitrary. In our

TABLE I
NUMBER OF STANDS THAT EACH TAXI USED TO WAIT FOR PASSENGERS, FOR EACH MONTH.

taxi_id	Sep-13	Oct-13	Nov-13	Dec-13	Jan-14
20000534	15	19	19	19	23
20000596	33	31	27	33	27
⋱	⋱	⋱	⋱	⋱	⋱
20000693	9	10	9	9	13

primary data (i.e. real world scenario), standardization by logarithm transformation obtained the best result. However, in the simulated scenario, normalization by logarithm transformation obtained the best result.

C. Identification Phase

Over the last few decades, some data stream algorithms have been proposed or extended, such as *K-Means*, *STREAM* and *CluStream* [18]. Some algorithms consider data stream events as a complete representation of independent objects. However, these algorithms can not be applied in some real-world scenarios, such as the one addressed in this study. For this case, data stream events are records (activities) performed by a single object (device).

We use *K-Means*, *WARD*, and *DBSCAN*, since these algorithms can adapt to different usage scenarios. In our tests, *K-Means* presents the best performance according to the *validity criteria*. We also adopt evaluation metrics for cluster analysis: *SWC*, *DBI*, *DUNN*, and *CH* [19]. Based on these evaluation metrics we chose the best number of clusters, for each window. *SWC* presented the best behavior by suggesting similar number of groups for all of the time windows analyzed. In summary, we decide to employ *K-Means* and evaluate with *SWC* throughout the whole process.

In Table II we present the grouped instances for each cluster, and the best number of groups for each time window, by using *SWC*. As we can see, the number of groups may change throughout time, representing the rise of new concepts and/or the end of known concepts. These changes can reveal drifting concepts and delineate the behavior of a churning customer. We identify and present the *churning curves* in the next phase.

D. Analysis and Prediction Phases

In order to distinguish potential churn behaviors, in the *Analysis* phase we design all possible life cycles, based on the life cycle of each customer. We end up producing 19 different life cycles. The suggested curves C are 1 *outlier curve*, 8 *churn curves* and 10 *normal curves*. We select and store the *churn curves* in D to use them in the last phase of our method; we discard the other curves since they will not help us detect churning customers. In summary, we can perform the identification of C by analyzing different shapes of curves according to the behavior of all customers.

In the *Prediction* phase, we need to identify the customers that have the same kind of behavior present in one of the 8 *churn curves*. In Table III we present the number of customers, number of C changes, % of users from data and the type of each life cycle. To represent such curves we consider $LC = \{LC_1, LC_2, ..., LC_{19}\}$ to be each one of the produced life cycles. To identify customer churn, we execute a process that compares the life cycle from each customer with each one of the stored life cycles considered *churn curves* in D. When a customer presents the same behavior of some of the *churn curves*, we predict such customer as possible a churn.

E. Evaluation Measures

We use a confusion matrix to describe the performance of our prediction method. Due the lack of a *true* class (i.e. whether the customer will churn or not) in the simulated data, we determine that 79 registers are churning customers, after performing an analysis in three time windows of the simulated data stream: February, March and April 2014. We use the following criteria to determine if a customer will perform churn or not: (a) if a customer presents some C in two or three consecutive months, we label his or her class as *true*; (b) if a customer presents some C at most in one of these three months, we label his or her class as *false*. These criteria are derived from our analysis in the real-world dataset, where we know whether customers are churn candidates or not. In summary, only customers that present behavioral changes during two or more time windows will have their classes labeled as churn.

We predict the class of 412 customers, as shown in Table IV. We predict 41 churners and 371 non-churners, although we know, by our analysis, that 79 customers are indeed churners. To test whether the proposed strategy is effective in detecting churners, we compute some evaluation measures [18]: accuracy (87%), precision (81%), sensitivity 42%, specificity (98%) and False Positive Rate (2%). With a false positive rate at 2%, we can guarantee that, when our method is used by manufacturers companies, it provides a tolerable disturbance level (i.e. only 2% of loyal customers will be targeted by

TABLE II
TOTAL OF GROUPED INSTANCES FOR EACH WINDOW.

Window	Clusters						
	C0	C1	C2	C3	C4	C5	C6
Sep-13	68	131	100	56	57	-	-
Oct-13	54	60	85	78	109	25	-
Nov-13	64	96	107	92	53	-	-
Dec-13	66	63	60	160	62	-	-
Jan-14	47	38	64	51	116	50	44

TABLE III
DESCRIPTION OF EACH LC.

LC	Cs	Customers	LC Type	% from data
LC_1	3	14	churn	3.40%
LC_2	2	8	churn	1.94%
LC_3	3	5	churn	1.21%
LC_4	2	4	churn	0.97%
LC_5	3	3	churn	0.73%
LC_6	3	3	churn	0.73%
LC_7	3	3	churn	0.73%
LC_8	3	1	churn	0.24%
LC_9	4	5	outlier	1.21%
LC_{10}	0	166	normal	40.29%
LC_{11}	1	55	normal	13.35%
LC_{12}	1	34	normal	8.25%
LC_{13}	2	34	normal	8.25%
LC_{14}	2	22	normal	5.34%
LC_{15}	2	21	normal	5.10%
LC_{16}	2	20	normal	4.85%
LC_{17}	2	12	normal	2.91%
LC_{18}	1	1	normal	0.24%
LC_{19}	1	1	normal	0.24%

TABLE IV
CONFUSION MATRIX OF THE SIMULATED DATA STREAM.

Actual \ Predicted	*Non Churners*	*Churners*	**Total**
Non Churners	TN = 325	FP = 8	**333**
Churners	FN = 46	TP = 33	**79**
Total	**371**	**41**	**412**

marketing campaigns), and with a precision as high as 81%, we are successfully identifying churning customers.

V. Concluding Remarks

In this paper, we introduce an effective method for predicting churning customers in mobile *app* usage scenario. Previous works typically used for this task were developed targeting other application domains, and using static data. Mobile *app* usage data is characterized as a data stream mining problem, and interpreting such data as being static may impair the applicability of such methods.

Predicting whether customers may migrate from a service provider to another is essential to manufacturers and telecomm companies, since the cost of attracting new customers is 5 to 6 times greater than maintaining the current ones [1].

In order to analyse customers' behaviors, we group them based in their activity patterns. This strategy is the opposite the ones used by telecomm companies, which use analysis techniques solely based in individuals [6]. A group of individuals that presents some concept drift (i.e. change in their behavior) is likely to represent a churning set of customers.

We test the effectiveness of our method in a simulated dataset that emulates the patterns found in our real-world data. We had to do this because our primary source of data, provided by our sponsor company, is confidential data. By simulating our primary data in an already well-known public dataset, we ensure the reproducibility of our tests.

After detecting drifting concepts and behavioral changes in customers, we predict that less than 10% of them are likely to churn, with an accuracy of 87% and precision of 81% – a remarkable metrics considering this kind of application. Our method has a false positive rate at only 2%, meaning that manufacturers using our method can avoid disturbing large amount of customers with its marketing campaigns, saving money.

As future work, we intend to develop a churn prediction system that will perform on-the-fly predictions. We also intend to build a new algorithm for data stream clustering tailored specifically for the context of mobile *app* usage. Finally, we plan to increase the range of features used in our method, such as Internet traffic, storage level, battery level, and sensors.

Acknowledgment

We gratefully acknowledge Motorola Mobility for its support to this research.

References

[1] A. M. Almana, M. S. Aksoy, and R. Alzahrani, "A survey on data mining techniques in customer churn analysis for telecom industry," *International Journal of Engineering Research and Applications*, vol. 45, pp. 165–171, 2014.

[2] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1–5, 2013.

[3] V. D. Blondel, A. Decuyper, and G. Krings, "A survey of results on mobile phone datasets analysis," *EPJ Data Science*, vol. 4, no. 1, pp. 1–55, 2015.

[4] The Statistics Portal, "Number of apps available in leading app stores," http://goo.gl/3FLn4f, online; accessed 05 January 2017.

[5] P. Li, S. Li, T. Bi, and Y. Liu, "Telecom customer churn prediction method based on cluster stratified sampling logistic regression," in *Proceedings of the International Conference on Software Intelligence Technologies and Applications & Frontiers of Internet of Things 2014*. IET, 2014, pp. 282–287.

[6] B. Bahmani, G. Mohammadi, M. Mohammadi, and R. Tavakkoli-Moghaddam, "Customer churn prediction using a hybrid method and censored data," *Management Science Letters*, vol. 3, no. 5, pp. 1345–1352, 2013.

[7] J. a. Gama, *Knowledge discovery from data streams*. Boca Raton: CRC Press, 2010.

[8] E. R. Faria, J. Gama, and A. C. Carvalho, "Novelty detection algorithm for data streams multi-class problems," in *Proceedings of the 28th annual ACM symposium on applied computing*. ACM, 2013, pp. 795–800.

[9] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer, "Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage," in *Proceedings of the 13th international conference on Human computer interaction with mobile devices and services*. ACM, 2011, pp. 47–56.

[10] L.-Y. Tang, P.-C. Hsiu, J.-L. Huang, and M.-S. Chen, "ilauncher: an intelligent launcher for mobile apps based on individual usage patterns," in *Proceedings of the 28th Annual ACM Symposium on Applied Computing*. ACM, 2013, pp. 505–512.

[11] H. Verkasalo, "Contextual patterns in mobile service usage," *Personal and Ubiquitous Computing*, vol. 13, no. 5, pp. 331–342, 2009.

[12] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay, "Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones," in *Proceedings of the 5th international conference on Mobile systems, applications and services*. ACM, 2007, pp. 57–70.

[13] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjea, A. A. Nanavati, and A. Joshi, "Social ties and their relevance to churn in mobile telecom networks," in *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. ACM, 2008, pp. 668–677.

[14] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. de Carvalho, and J. Gama, "Data stream clustering: A survey," *ACM Computing Surveys (CSUR)*, vol. 46, no. 1, p. 13, 2013.

[15] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1. Oakland, CA, USA., 1967, pp. 281–297.

[16] L. Moreira-Matias, J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas, "Predicting taxi–passenger demand using streaming data," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 3, pp. 1393–1402, 2013.

[17] H. Li, X. Lu, X. Liu, T. Xie, K. Bian, F. X. Lin, Q. Mei, and F. Feng, "Characterizing smartphone usage patterns from millions of android users," in *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*. ACM, 2015, pp. 459–472.

[18] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[19] L. Vendramin, R. J. Campello, and E. R. Hruschka, "Relative clustering validity criteria: A comparative overview," *Statistical Analysis and Data Mining*, vol. 3, no. 4, pp. 209–235, 2010.