2017 International Conference on Data Science and Advanced Analytics

Scalable RFM-Enriched Representation Learning for Churn Prediction

Sandra Mitrović*[1], Gaurav Singh†[2], Bart Baesens*‡[3], Wilfried Lemahieu*[4] and Jochen De Weerdt*[5]
*Department of Decision Sciences and Information Management, KU Leuven, Belgium
†University College London, London, United Kingdom
‡School of Management, University of Southampton, United Kingdom
Email: [1]sandra.mitrovic@kuleuven.be, [2]gaurav.singh.15@ucl.ac.uk, [3]bart.baesens@kuleuven.be,
[4]wilfried.lemahieu@kuleuven.be, [5]jochen.deweerdt@kuleuven.be

***Abstract*—Most of the recent studies on churn prediction in telco utilize social networks built on top of the call (and/or SMS) graphs to derive informative features. However, extracting features from large graphs, especially structural features, is an intricate process both from a methodological and computational perspective. Due to the former, feature extraction in the current literature has mainly been addressed in an ad-hoc and hand-crafted manner. Due to the latter, the full potential of the structural information is unexploited. In this work, we incorporate both interaction and structural information by devising two different ways of enriching original graphs with interaction information, delineated by the well-known RFM model. We circumvent the process of extensive manual feature engineering by enriching the networks and improving the scalability of the renowned node2vec approach to learn node representations. The obtained results demonstrate that our enriched network outperforms baseline RFM-based methods.**

***Index Terms*—Enriched (Social) Networks, Node Representation Learning, RFM, Churn Prediction.**

I. Introduction

Churn prediction in telco is a well-known problem for several decades now, and, consequently, it has an extensive presence in the data mining literature. Recently, with the expansion of social network analytics, the studies on churn prediction have shifted in the direction of using Call Detail Records (CDRs) to generate call networks and extract informative features from these. An additional motivation for expanding the feature space stems from previous churn related work which has proven that an increase in the predictive performance depends on the explanatory features, rather than on the modeling technique used [33].

However, extracting informative features from graphs (so-called graph featurization), can be a quite intricate process due to the complex structure of networks themselves, where together with the topology of the network (we will refer to this as structural information), additional information characterizing relationships between network nodes (to be referred to as interaction information), is usually provided. The literature has seen many different featurizations of both interaction and structural information. Interaction as part of customer behavior is usually delineated with the well-known RFM (**R**ecency, **F**requency, **M**onetary) framework, which has its benefits both in simplicity and predictive performance [1, 2, 4, 35]. Similarly, structural network information is a mainly hand-crafted ad-hoc solution based on degree/closeness/betweenness/eigenvector centrality measures [13, 24, 35]. The problem with such approaches is not only in selecting which features to derive between the variety of all possible ones, but also in the computational burden which increases with the size of the network. For example, a very informative feature, betweenness centrality, becomes computationally intractable for very large networks (the fastest known algorithm for computing betweenness centrality takes $O((e')^2 log^2 n)$, with n number of nodes and e' the upper bound of the number of edges belonging to the shortest path). Therefore, only a small number of related works use both structural and interaction features [4, 8, 16, 22, 23, 35] and not even to their fullest predictive potential.

In this work, we focus on the churn prediction task in telco, with the idea to exploit both structural and interaction information from the CDR graph, while circumventing the process of extensive feature hand-engineering. For this, we propose a novel approach based on node representation learning in RFM-enriched call networks. First, we devise three different operationalizations of RFM variables. Second, we use these to design RFM-enriched variations of the original CDR-graph, to achieve the goal of conjoining interaction information with the original topology. Third, for node representation learning, we propose a scalable adaptation of the existing node2vec approach proposed in [5], to ensure scalability for very large graphs (with 5M edges and more, as in our case). Furthermore, this enables learning node representations in a more automated and task-agnostic manner, which is the purpose of these kind of approaches [5, 27].

To the best of our knowledge, this paper is the first both in using node representations in CDR graphs for churn prediction and applying the RFM framework together with unsupervised learning of node representations. Additionally, we demonstrate that designed extensions of the original graphs using RFM features (in the form of artificial nodes), can lead to significant performance improvements in terms of AUC and lift.

The rest of the paper is organized as follows. Section II provides an overview of related work. Sections III and IV explain our method and experimental setup, respectively. In Section V we present experimental results and shortly discuss them in Section VI. Finally, Section VII provides a conclusion and directions for future research.

978-1-5090-5004-8/17 $31.00 © 2017 IEEE
DOI 10.1109/DSAA.2017.42

II. Related Work

Since the literature related to churn prediction in telecommunications is very extensive, we restrict ourselves only to areas being of interest for this work: interaction features, structural features and representation learning in graphs.

A. Extracting Interaction Features using the RFM Model

The RFM model represents the most typical way of characterizing customer interactions, and customer behavior in general. Given an event of interest, the RFM model defines the following measures on a customer level: 1) how recent the event occurred, i.e. how long is the time interval between the event's last occurrence and the moment of reference in time; 2) how frequently the event occurred, i.e. what is the number of event occurrences in the observed time frame; 3) what is the monetary aspect of the event, i.e. how much money the customer spent related to the event. The RFM model has three important qualities, due to which it has gained its popularity in the literature. First, it relies on a simple concept, which makes it easily understandable and computationally tractable [11]. Furthermore, the only assumption required with RFM variables is that the future behavior and value of a customer can be predicted based on the customer's past behavior [19]. Second, since the definition of the event is flexible and can be adapted to different contexts, the RFM model is applicable to different domains (banking [1], retail [2], telco [28]). For example, in telco, the event is a call between two customers (which can be further aggregated on different levels). Finally, when used as explanatory variables, the RFM variables exhibit very good predictive power [2, 12].

The current literature related to churn prediction in telco devises a wide spectrum of RFM variations, ranging from summary, coarse-grained to more fine-grained features. Examples of summary RFM variables are total call frequency, total call volume (seconds) [4], seconds of use, frequency of use and frequency of SMS in [12]. An overview of fine-grained RFM variables across different dimensions can be seen in Table I. In addition, different works propose different aggregation levels and transformations. The RFM features are engineered both by absolute values and percentages, for example, the call frequency percentage (wrt total) to/from a different operator's network [4], the ratio of interaction frequency with churners to the total interaction frequency [16] and the percentage of minutes of use that were made to on-net callers [26].

B. Extracting Structural Features

Many previous works on churn prediction in telco have been using features derived from the structure of the underlying networked graph. These features are mainly expressed in the form of different centrality measures since they, in one way or the other, quantify the importance of the node in a network from a topological perspective. For example, degree centrality measures of $1^{st}, 2^{nd}$ and 3^{rd}-order were used in [16]; 2^{nd}-order degree and structural cohesion (also known as clustering coefficient or density) were used in [35]; PageRank in [8]; degree centrality, closeness centrality, eccentricity centrality, clustering coefficient, Shapley value, degree and proximity prestige in [30]. In [13], the authors used eigenvector centrality not explicitly as explanatory variable, but instead to initialize the energy level of a node in the spreading activation process. Even more diversified structural features have been used for analyzing telco call and SMS graphs (PageRank, diameter, number of (strongly) connected components and cliques in [24]), the process of message spreading in viral marketing in telco (betweenness centrality, authorities, hubs, Weighted PageRank, Weighted SenderRank, edge-weighted degree in [14]), classifying the edges on decaying and permanent in the call graph (the number of neighbors that two nodes have in common, employed as explanatory variable in [28]).

The potential of structural features has not been fully exploited in the literature. Despite recognizing their predictive power, in e.g. [36] closeness and betweenness centrality were not taken into account due to their computational requirements.

C. Combining Interaction and Structural Information

Interaction and structural information has already been jointly utilized in the literature, either implicitly or explicitly. For example, in [26], customer behavior is claimed to be independent of the call graph structure but, nonetheless, degree centrality is used to normalize the values of interaction features; in [28], 2^{nd}-order embeddedness is derived as the number of calls that neighbors of one node make to the neighbors of another node, which essentially is a frequency based on 2^{nd}-order neighborhood. Moreover, several studies on telco churn prediction have explicitly employed both interaction and structural features as explanatory variables in their predictive models [4, 8, 16, 22, 23, 35]. Nevertheless, these studies have not exploited the full potential of these two sets of features, as they mostly employ either only degree measures [4, 16, 22, 23] or a slightly extended, but still limited number of structural features [8, 35]. The literature agrees that utilizing more diversified (types of) features leads to better performance [4, 8, 16, 35]. However, to the best of our knowledge, no churn prediction in a telco related study carried out an inquiry to determine which of these two classes of features provides better predictive scores. On the other hand, RFM variables were found to have more importance than structural features in the task of classifying edges in a telco graph [28] and churn prediction in banking [1]. This provides a good motivation for enriching graph topologies with RFM variables.

D. Representation Learning on Graphs

Representation learning is a fast growing field of research aiming not just to automate the feature engineering process, but also to acquire task-independent and high performing feature representations. It has many applications in various domains such as natural language processing, speech recognition, image classification. In fact, some prominent recent works related to representation learning on graphs [5, 27] are based on representation learning achievements in the natural language processing (NLP) field [21], using the analogy between a context of a word in a document and a neighborhood

TABLE I
DIFFERENT VARIATIONS OF FINE-GRAINED RFM VARIABLES PRESENT IN THE CURRENT TELCO CHURN PREDICTION LITERATURE. THE FIRST COLUMN CONTAINS THE DIMENSION (THE COMBINATION OF DIMENSIONS) ACCORDING TO WHICH SLICING IS DONE; THE SECOND COLUMN REPRESENTS A SUB-DIMENSION UTILIZED TO MAKE AN EVEN MORE FINE-GRAINED DISTINCTION. EXAMPLES ARE BORROWED FROM THE REFERENCES IN BOLD TYPEFACE, REFERENCES IN STANDARD TYPEFACE BELONGING TO THE SAME ROW CALCULATE THE SAME VARIABLES BUT USE SLIGHTLY DIFFERENT TERMINOLOGY.

Dimension(s)	Calculated per	Example	Used in	RFM
Time	working days/weekends	Minutes of usage by days of the week (working days and weekend)	**[25]**,[23]	M
	morning/midday/night	Minutes of usage by time of call (morning, midday, night)	**[25]**	M
	peak/off-peak	Duration of calling in busy time	**[7]**,[26]	M
	special events	Duration of calling in festival	**[7]**	M
Direction	incoming/outgoing	Total incoming/outgoing call duration (seconds)	**[4]**,[22]	M
		Number of outgoing/incoming calls	**[4]**,[22, 23]	F
		Number of SMS sent/received	**[22]**	F
		Incoming/outgoing communication volume between i and j	**[6]**,[10]	F/M
	reciprocal calls	Mutual communication volume between i and j	**[6]**	F/M
Destination	home network	In-net call duration	**[10]**	M
	competitor network	External total calls	**[23]**	F
		External total duration	**[23]**	M
	local/national/international	Minutes of local call	**[8]**,[7]	M
		Number of international calls	**[7]**	F
		Minutes of long-distance call	**[8]**	M
Other party	churners/non-churners	Total interaction frequency with churners	**[16]**,[4]	F
		Total call volume to/from churner neighbors (seconds)	**[4]**,[35]	M
		Average of call counts to/from non-churn neighbors	**[35]**	F
Direction+Destination	incoming/outgoing+competitor network	Number of incoming, outgoing calls to/from a different operator's network	**[4]**,[22, 23]	F
		Total incoming/outgoing call duration from a different operator's network (seconds)	**[4]**,[22, 23]	M
Direction+Other party	incoming/outgoing+churners	Number of SMS/calls made/received to/from churners	**[22]**	F
		Total duration of calls made/received to/from churners	**[22]**	M
Time+Destination	peak/off-peak+competitor network	Minutes of use (MOU) during peak period that were made to off-net callers	**[26]**	M

of a node in the graph. Once the node representations are learnt, they can be employed in different predictive tasks, such as multi-label classification, link prediction and so on. For example, node representations are used for multi-label classification tasks in social networks (Facebook [5], Flickr [17, 27], YouTube [17, 27], blogger networks derived from the BlogCatalog website [5, 27]), Wikipedia words co-occurrence networks [5, 17] and citation networks (DBLP) [17]. In link prediction, node representation learning has been applied on the Facebook graph, Protein-Protein Interaction graphs and a collaboration graph from ArXiv in [5].

To the best of our knowledge, representation learning has not been used before in combination with RFM variables. In addition, this paper is the first in using node representations in CDR graphs for churn prediction in telco.

III. Scalable RFM-Enriched Node Representation Learning

In this section, we introduce our approach, in which we incorporate both structural and interaction information, by carefully devising different RFM operationalizations and network designs, as well as a scalable node representation learning method. Hence, our approach is based on three main building blocks: RFM variables (capturing interaction information), enriched call networks (integrating structural and interaction information) and node representation learning methods (learning representations based on these networks). We explain each of these in more detail, as follows.

A. Operationalization of RFM Network Features

In our approach, we first construct a usage graph from the observed monthly CDR data. This is done in a standard way, representing customers as nodes and adding edges between nodes only if the corresponding customers had a call registered in the CDR. Next, to quantify customer interaction behavior, we calculate RFM variables for each customer and observed period (month) using the following definitions:

- Recency: the number of days between the end of the observed month and the customer's most recent call
- Frequency: the number of calls of a customer during the observed month
- Monetary: The monetary feature is calculated as the duration (in seconds) of customer calls during the observed month, given that the actual amount charged per call is not available in our datasets.

We already mentioned in Section II that the literature has seen plenty of RFM variations. In order to retain the simple concept of RFM, we do not aim at fully replicating these approaches, neither at additionally expanding the space of different RFM characterizations. Instead, we opt for three different RFM variants, as follows:

- Summary-RFM (denoted by RFM_s), calculated as the total R/F/M per customer, i.e. overall recency (R), total number of calls (F) and total duration (M) per customer.
- Detailed-RFM (denoted by RFM_d), where each of the R/F/M variables is sliced based on the direction and destination dimension into three subcategories: outgoing towards home network, outgoing towards other networks and incoming (denoted as $R_{out\_h}, R_{out\_o}, R_{in}$ and similarly for M and F), inspired by the approaches in [4, 10, 22, 23].
- Churn-RFM (denoted by RFM_{ch}), where we calculate R/F/M variables only with respect to customers who churned (denoted as R_{ch}, F_{ch}, M_{ch}). Hence, in this case, M_{ch}, for example, represents the total duration of calls to/from churners. A similar characterization of RFM has been done in [4, 22, 35].

B. RFM-Enriched Network Construction

In order to incorporate both interaction information (captured by RFM features) and latent topologically-sensitive information of the network constructed from the CDR, we consider two directions for network construction:

- First, retaining the original call network, while embedding RFM information as the edge weighting factor (referred to as RFM-embedded network);
- Second, augmenting the original call network with additional (artificial) nodes corresponding to RFM variables (referred to as RFM-augmented network).

1) RFM-Embedded Networks: As already mentioned, the aim in this scenario is to retain the original graph topology and incorporate RFM information for computing edge weights. This requires a method for combining RFM features into a single score, which is not very established in the current RFM related literature. In fact, except for ranking customers using the unique RFM score defined as $(R*100)+(F*10)+M$ in [31], we have not encountered any similar consideration. However, similarity between nodes can be determined based on the Euclidean distance between their corresponding vectors, as has been frequently used in previous works [18, 29]. Hence, we devise the following four network designs:

- Embedded Graph with Summary-RFM Euclidean Distance Based Weighting (EG_s^{Eucl})
  In this graph, the weights of the edges are determined as the inverse of the Euclidean distances between vectors $\boldsymbol{v} \in \mathbb{R}^3$ of corresponding nodes. Here, $\boldsymbol{v}^T = (R, F, M)$, where R, F, M are Summary-RFM features for each node (Summary-RFM explained in III-A).
- Embedded Graph with Summary- Plus Churn-RFM Euclidean Distance Based Weighting (EG_{s+ch}^{Eucl})
  Here the weights of the edges are determined as the inverse of the Euclidean distances between enriched node vectors $\boldsymbol{v} \in \mathbb{R}^6$ of the form $\boldsymbol{v}^T = (R, F, M, R_{ch}, F_{ch}, M_{ch})$. Here, R, F, M represent Summary-RFM features, while R_{ch}, F_{ch}, M_{ch} correspond to the R/F/M features calculated with respect to a node's interaction with churners (Churn-RFM in III-A).
- Embedded Graph with Detailed-RFM Euclidean Distance Based Weighting (EG_d^{Eucl})
  In this graph, the weights of the edges are determined as the inverse of the Euclidean distances between vectors $\boldsymbol{v} \in \mathbb{R}^9$ of the corresponding nodes. In this case, a vector v has nine components, since each of R/F/M captures a fine-grained information for: incoming ($X_{in}, X \in \{R, F, M\}$), outgoing towards home network ($X_{out\_home}, X \in \{R, F, M\}$) and outgoing towards other network ($X_{out\_other}, X \in \{R, F, M\}$) (Detailed-RFM in III-A).
- Embedded Graph with Detailed- Plus Churn-RFM Euclidean Distance Based Weighting (EG_{d+ch}^{Eucl})
  The weights of the edges in this graph are determined as the inverse of the Euclidean distances between enriched node vectors $\boldsymbol{v} \in \mathbb{R}^{12}$, since on top of the previous 9 components for Detailed-RFM features, we add R_{ch}, F_{ch}, M_{ch}, which correspond to the R/F/M features calculated with respect to a node's interaction with churners (Churn-RFM in III-A).

It is worth noticing that due to the preservation of the original network topology, the computation of similarities is performed only for those pairs of nodes which are actually connected (and not for all possible pairs of nodes). This makes it computationally tractable, despite the large size of the networks in consideration.

2) RFM-Augmented Networks: The second type of networks we consider, are likewise built on top of the original graph. Nevertheless, their final topology differs from the one of the original graph, since here RFM information is conjoined in the form of artificial nodes, which are added to the original graph. To devise the construction of artificial nodes, we follow the idea frequently used in literature related to customer segmentation and customer lifetime value modeling, where customers are usually segmented partitioning each of their R/F/M variables in five equally-sized groups (corresponding to very high, high, medium, low, very low) [3, 9, 19]. Inspired by this approach, we design four different networks, as follows:

- Augmented Graph with Summary-RFM Artificial Nodes (AG_s)
  In this case, we start with deriving R, F and M features for each node. Then we partition R, F and M values each into five quantiles (similar to [3, 9, 19]), and we assign one new (artificial) node to each quantile, obtaining thus 15 artificial nodes R_i, F_i, M_i, where $i \in \{1, 2, ..5\}$[1]. Next, we expand the original graph by adding these new artificial nodes and connecting each node to exactly one node from the set of artificial R nodes, exactly one node from the set of artificial F nodes and exactly one node from the set of artificial M nodes, based on the appropriate R, F and M quantiles. In this way, the newly obtained graph has at most 15 new nodes, but exactly $3 * |V|$ more edges than the original one (with $|V|$ being number of nodes in the original graph).
- Augmented Graph with one Churn Node and Summary-RFM Artificial Nodes (AG_{s+ch})
  This graph is exactly the same as AG_s, except that one additional artificial node representing churn is added, to which then all the churners are connected. By adding this node, we try to compensate for the information contained in churn-RFM related features, which we do not exploit in this case as we try to keep the number of nodes and edges in the resulting graph not too large.
- Augmented Graph with Detailed-RFM Artificial Nodes (AG_d)
  In the two previous cases, RFM is calculated on a summary level (Summary-RFM), which means that only one value for each of R, F and M is calculated. Here, we first derive Detailed-RFM information, calculating each of R, F, M on a more fine-grained level using direction

[1] Exceptionally, due to the skewed distribution of R/F/M values, one can end up having less than five quantiles per R/F/M.

and destination dimensions to obtain three categories: incoming, outgoing towards the home network and outgoing towards other networks. Next, we repeat the steps from AG_s, whereby each of these three categories is binned into quantiles, thus leading to 45 artificial nodes which are used to enrich the original graph. As with AG_s, each node is connected to the appropriate artificial nodes. The number of additional edges, compared to the original graph, increases with nine times the original number of nodes (since each node becomes connected to three more R artificial nodes - for incoming, home outgoing and other outgoing, and similarly 3 more per F and per M).

- Augmented Graph with one Churn Node and Detailed-RFM Artificial Nodes (AG_{d+ch})
  Similarly to AG_{s+ch}, this graph is exactly the same as AG_d, except for adding one additional artificial node representing churners, to which then all the churners are connected.

All constructed networks are considered to be undirected. This decision is motivated by the fact that our call graph is sparse, and hence, retaining directed graphs would make random walks get stuck at sink nodes. In addition, the preference to undirected call graphs was given as well in some previous works, where only hand-engineered features were utilized (without considering random walks) [13]. The motivation provided there was that, regardless the initiator, both sides share the same information during interaction.

Additionally, we consider enriched graphs unweighted, for two different reasons. First, due to a very different nature between the type of nodes ("real" vs. artificial) in these pseudo-bipartite[2] graphs, it is not easy to determine the corresponding weights as we would not like to bias the process of walk generation neither towards "real" nor towards the artificially added nodes. Second, if we were to make a differentiation of weights for all the edges in the graph, we would need to consider potential different weightings towards the artificial nodes since the RFM-related literature argues weather R/F/M features should be treated with the same [9] or different [32] importance. However, in order to not determine these weightings in an ad-hoc manner, studies do consider them with equal priority [3, 19]. Furthermore, preliminary experiments showed that unweighted networks provide better results than weighted ones, which might stem from wrongly instantiating the weighted parameters, but in any case setting parameters would require more computational effort in order to find an optimal weighting schema.

C. Scalable node2vec-based Representation Learning

For learning node representations, we utilize the node2vec implementation provided by [5], for which we propose modifications, in order to ensure its scalability on the large graphs that we consider. As already mentioned, node2vec is based on the SkipGram model from the NLP domain, where, given the current word, the goal is to predict its surrounding words. The model aims to maximize the probability of finding the words from the same context together and bring the representations of the words from the same context closer than those of the words found in different contexts. More precisely, if we denote by V a set of words in vocabulary (or analogically, a set of nodes in the graph), we are learning a function $f, f : V -> R^d$ such that

$$max \sum_{v \in V} log Pr(C_v | f(v))$$

where C_v is a context of word (node) v. This objective function is further simplified using the independence assumption, while the conditional likelihood is modeled with a softmax function in which a computationally expensive normalization factor is approximated using negative sampling. This procedure is already explained in detail both in [21] and [5].

Another important aspect is the generation of a context, C_v, which in NLP setting is straightforward, but in a graph setting requires more attention. Therefore, different approaches were used for this purpose in the current literature. Both in node2vec ([5]) and DeepWalk ([27]), fixed-length random walks are used to generate contexts (neighborhoods). In [27], the next node in the random walk is determined by uniformly sampling between the current node neighbors. Unlike this, in node2vec, random walks are guided by return and in-out parameters p and q, trying to find the best balance between breadth-first and depth-first sampling strategies. More precisely, if the walk has reached node j, coming from node i, the (unnormalized) transition probability to continue to node k is $w_{jk}, w_{jk}/p, w_{jk}/q$ if d_{ik} equals 1,0,2 respectively, where w_{ij} and d_{ij} represent

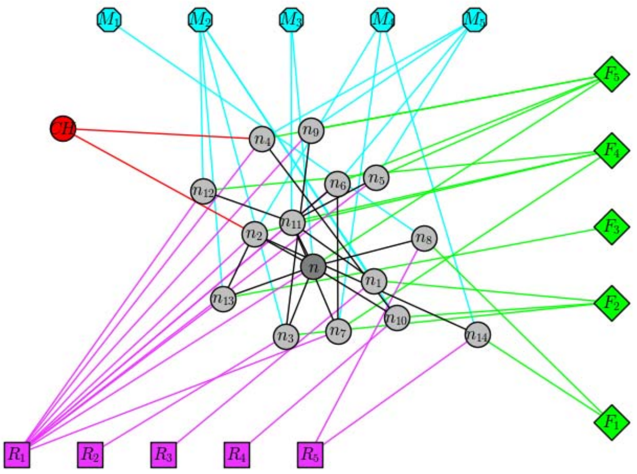

Fig. 1. An extract from the AG_{s+ch} network, depicting a 2^{nd}-level neighborhood of a node n, with artificial nodes magenta (square), green (diamond) and blue (octagon) representing R(ecency), F(requency) and M(onetary), respectively. R_i corresponds to i^{th} quantile with respect to R(ecency), similarly for F_i, M_i. The artificial node corresponding to churners is denoted with CH in red.

[2] One type of nodes are "real" nodes stemming from the original graph and representing customers, while the other are artificial nodes representing quantiles of R/F/M values. Artificial nodes are indeed mutually disconnected but "real" nodes can be connected among themselves, hence we call these networks "pseudo-bipartite".

the weight of the (i,j) edge and the length of shortest path between nodes i and j respectively. This obviously requires precomputing transition probabilities to allow for efficient sampling afterwards. The sampling procedure itself requires special attention, since a straightforward sampling procedure of randomly choosing an element from a sequence of length n takes O(n) time, which is costly when n is large. Instead, the alias table sampling method [15], which takes only O(1) time, is used. The node2vec approach requires constructing two alias tables, one for nodes and one for edges, since in the beginning, starting from the initial node, it determines the next node in the walk based on the node alias tables, while afterwards, it always decides based on the alias edge table and the edge from which it has reached the current node. In the case of very large graphs (with e.g. 40M of edges, as in our case) this approach, in the provided implementation, turns out to be computationally unfeasible. Hence, on the contrary to the original node2vec idea, we decided not to impose any additional parametrization on random walks except the weights of the adjacent edges. In other words, once the random walk reaches the node v, the probability of moving to its adjacent node u, is proportional to normalized weight of the edge (u, v). Given that we consider our graphs undirected, it is still possible for a random walk to return back to the node from which it has arrived, hence we consider that no special treatment to enforce or impede backtracking is necessary in our case. This modification allows for faster walk generation, since it basically means that only an alias table for the nodes has to be precomputed which significantly decreases computational time, as will be shown in the results section. Moreover, we will show that this relaxation will not deteriorate the predictive performance.

Due to the simpler procedure used for random walk generation in DeepWalk [27], DeepWalk might look as a good alternative to node2vec. However, on the contrary to node2vec, which uses negative sampling for approximating the normalization factor in softmax probabilities while trying to optimize the objective function in SkipGram, DeepWalk uses less efficient hierarchical softmax initially suggested in [20]. It is worth mentioning that our edge weight-driven random walks used for context generation is similar to the incident-edges-weighted sampling procedure performed in LINE [17]. Nevertheless, the other elements of these two approaches are significantly different. Namely, in our approach the context (walk) of full (predefined) length is generated in one pass after which its representation is learnt by the SkipGram model. On the contrary, in [17], each context is generated from two parts, which are constructed independently from first and second-order neighborhood and additionally, optimized separately using two different objective functions. While [5] criticizes previous approaches [17, 27] for not being flexible enough when it comes to generating contexts through sampling, due to the fact that there is no unique best performing sampling and that the way of sampling influences learnt representation, we have to emphasize that the flexibility of generating random walks in node2vec, comes with a high computational setback, as it requires tuning of return and in-out parameters and especially for large networks, requires a lot of time for precomputing transition probabilities.

D. Churn Definition

In the absence of churn labels (except for a very small number of postpaid customers who ported-out), we define churners based on the absence of activity, which can be detected from CDRs. For the customers of month M (representing our customer base) we aim at predicting which of them will churn in the month $M+2$ (we will refer to this as "one-month gap" approach). More precisely, the customer who is active in month M, is a churner in month $M+2$ if (s)he is still active in month $M+1$ (appears in the CDR for this month) and has not been in the list of the ported-out customers for the month $M+1$, while not being active in month $M+2$ or has ported-out during the month $M+2$.

IV. Experimental Setup

In this section, we provide brief insights about our datasets, churn definition, baselines, predictive model, different parameterizations and evaluation method used.

A. Data

We perform our experiments on one prepaid and one postpaid dataset (see Table II), each of which consists of four consecutive months of CDR data. The only usage type available in CDRs are calls (no SMS/MMS/GPRS usage), for which we are provided with (anonymized) information about caller, callee, as well as the date and the (real) duration of call. For postpaid customers, we also have information if and when the customer has ported-out (decided to switch to other provider while retaining the same phone number), but this is the case for only a dozen of customers on a monthly level. Applying the before mentioned churn definition, we end up with around 7,5% of churners for the prepaid and 4.5% of churners for the postpaid dataset.

TABLE II
Statistics of the datasets.

Measure	Prepaid	Postpaid
Number of nodes	4303541	4799149
Number of edges	5936423	9246134
Average degree	2.75886	3.85324
Average clustering coefficient	0.05749	0.06939
Number of connected components	138509	27392
Size of maximal clique	7	7
Degree assortativity coefficient	-0.00110	-0.02290
Power law coeff. (degree distr.)	1.29164	1.67275

B. Baseline Methods

We use four different RFM-based baseline methods, as shown below. We restrain to fairly simple RFM alternatives, where R/F/M features are calculated using the same definitions as provided in Section III-A, just applying different slicing across direction and destination dimensions. These features are

TABLE III
Method Notation.

Type of Information	RFM Features	Enriched Graph	Augmented Graph
Summary	RFM_s	EG_s^{Eucl}	AG_s
Summary+Churn	RFM_{s+ch}	EG_{s+ch}^{Eucl}	AG_{s+ch}
Detailed	RFM_d	EG_d^{Eucl}	AG_d
Detailed+Churn	RFM_{d+ch}	EG_{d+ch}^{Eucl}	AG_{d+ch}

then provided as explanatory variables to a logistic regression model.

- RFM_s: Summary-RFM, which contains only summarized RFM information, the same as defined in III-A.
- RFM_d: Detailed-RFM, where each of the R/F/M dimensions is sliced into three partitions (as defined in III-A).
- RFM_{s+ch}: Summary-RFM information enriched with the same variables calculated with respect to churners; hence, containing all the variables present both in RFM_s and RFM_{ch}.
- RFM_{d+ch}: Detailed-RFM information enriched with the same variables calculated with respect to churners; hence, containing all the features present in RFM_d and RFM_{ch}.

This setup imposes the requirement of considering four consecutive months of data, since we use information from month $M-1$ to identify to-be-churners in month $M+1$, which are still present in CDR for month M and based on which we can calculate RFM features related to churners.

We would like to emphasize that RFM features can be derived from data originating from different sources: either from customer interactions (network features) as described above, or from the individual (local) features of a customer. In case of the latter, the monetary value of the customer could be, for example, calculated as the amount recharged/billed per month (for prepaid/postpaid respectively) [22]. However, since in this work our datasets only consist of CDRs, we consider only RFM features based on customer interactions.

C. Experiments

We perform a set of experiments to help us compare the following methods (for notation see Table III), categorized in four comparison scenarios:

- RFM_s vs. EG_s^{Eucl} vs. AG_s
- RFM_{s+ch} vs. EG_{s+ch}^{Eucl} vs. AG_{s+ch}
- RFM_d vs. EG_d^{Eucl} vs. AG_d
- RFM_{d+ch} vs. EG_{d+ch}^{Eucl} vs. AG_{d+ch}

It is important to mention three sets of parameters that we take into account for our experiments. The first set of parameters is utilized both by our adaptation of node2vec and the original node2vec method: number of walks n and walk length l. We set n=10 and l=30 (in [5]: n=10, but l=80). The second set of parameters refers to the underlying SkipGram model and consists of three parameters: the number of iterations i, the context window size s and the number of dimensions in the resulting representation d. For these, we set $i = 5, s = 10$ and $d = 128$ (same in [5] except for $i = 1$). Finally, the last set consists of return parameter p and in-out parameter q, required only for node2vec. They are both instantiated to 1 in our experiments.

D. Model and Evaluation Methods

Our predictive models are generated using logistic regression with $l2$ regularization, where the regularization hyperparameter was determined using 10-fold cross validation. The motivation for using logistic regression is two fold. First, not only churn prediction studies have shown that applying logistic regression provides comparatively good predictive performance to some other, more complex, predictive models [33], but it was also used in studies [5, 27] which demonstrate that learnt representations perform well even with this fairly simple predictive technique.

We evaluate our models using AUC and lift scores (at 0.5%), performing out-of-sample evaluation, as done in [35]. We also provide an outlook into computational aspects regarding random walk generation and node representation with respect to the number of walks.

V. Experimental Results

In this section, we present the results of our experiments and analyze them from both a predictive and computational performance perspective.

A. Predictive Performance

The results in terms of AUC and lift (at 0.5%) for Summary- and Detailed-RFM based approaches are displayed in Tables IV and V, respectively. Consequently, following are the results for the four comparison scenarios (mentioned in Section IV-C):

1) RFM_s vs. EG_s^{Eucl} vs. AG_s
In terms of AUC, AG_s performs best, while EG_s^{Eucl} scores worst, for both datasets (Table IV). The situation is similar for lift for the postpaid dataset. For the prepaid dataset, EG_s^{Eucl} outperforms the other two approaches in terms of lift, while RFM_s performs worst.
2) RFM_{s+ch} vs. EG_{s+ch}^{Eucl} vs. AG_{s+ch}
Similarly as in the previous case, AG_{s+ch} outperforms the two other approaches in terms of AUC, while EG_{s+ch}^{Eucl} is the worst (Table IV). In case of lift, AG_{s+ch} again has the best performance. EG_{s+ch}^{Eucl} performs better than RFM_s for the prepaid dataset, while the opposite holds for postpaid.
3) RFM_d vs. EG_d^{Eucl} vs. AG_d
In this case, RFM_d outperforms the other two approaches both in terms of AUC and lift (at 0.5%) (Table V). On the other hand, AG_d outperforms EG_d^{Eucl}.
4) RFM_{d+ch} vs. EG_{d+ch}^{Eucl} vs. AG_{d+ch}
In this case EG_{d+ch}^{Eucl} scores the worst in terms of both AUC and lift, on both datasets. For the prepaid dataset, AG_{d+ch} outperforms RFM_{d+ch} in terms of AUC, while for postpaid the opposite holds.

The overall best performance in terms of AUC is achieved by AG_{s+ch} for the postpaid and by AG_{d+ch} for the prepaid dataset. Furthermore, augmented network-based results are

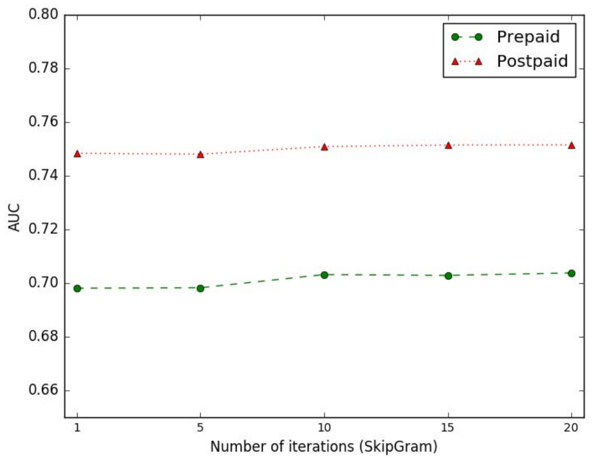

Fig. 2. AUC scores for SkipGram parameter number of iterations i instantiated with 1,5,10,15,20 using AG_{s+ch} network as an example. AUC scores are stable for varying i.

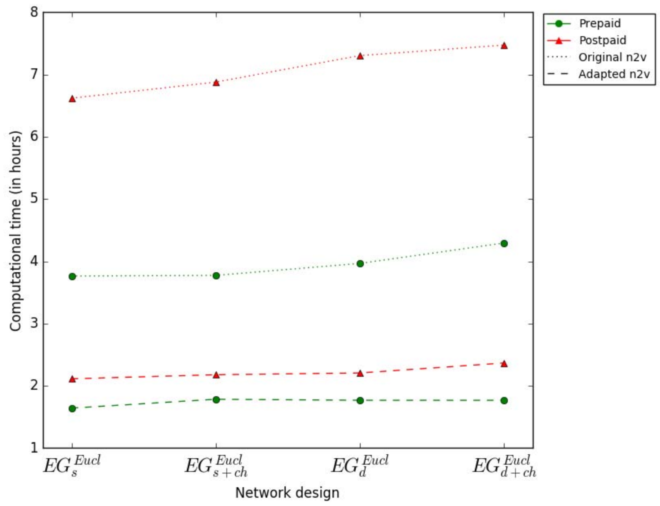

Fig. 3. Computational time needed for random walk generation with the original node2vec approach and our adaptation across different network designs.

always better than the embedded network-based ones. Additionally, in case of both datasets, augmenting with churn information seems most important. Surprisingly, augmenting with Detailed-RFM does not remarkably improve predictive performance (as compared to augmenting with Summary-RFM), although Detailed-RFM outperforms other baselines. Original *node2vec* for p=q=1 and our adaptation *s-node2vec* yield very similar performances. It is worth mentioning that the regularization hyper-parameter in the logistic regression model is tuned as to maximize average AUC score across 10 folds, hence it is not optimized for lift.

Performance is stable regardless of the number of iterations in the SkipGram model as illustrated in Figure 2 for AG_{s+ch}.

B. Computational Performance

Due to a large number of parameters that would have to be taken into account for a thorough computational analysis across different methods, we restrain ourselves to providing the outlook into computational performance. Figure 3 shows the computational time needed to generate random walks using the original node2vec approach (in the available implementation, setting p=q=1) and our adaptation, throughout different embedded networks. The computational efficiency of our method is obvious. Moreover, generation of random walks with the original node2vec method was not feasible for augmented networks within a reasonable time frame (24h). Therefore, these are omitted from this analysis.

VI. Discussion

It can be hypothesized that the performance comparability of our representation learning approach and the node2vec approach is purely empirically verified, requiring a complete grid search fine-tuning of return and in-out parameters p and q to be performed. However, we strongly believe that there are grounded reasons for the obtained results. Namely, we would like to remind the reader that the main motivation for using parameters p and q in node2vec was to guide walks to better account for homophily and structural equivalence. Homophily refers to the concept that similar people (in this case, nodes) tend to connect to those who they perceive as similar. The idea behind the SkipGram model, used to learn node representations in node2vec, is essentially the same as it hypothesizes that similar words (again, nodes in this setting) appear in similar contexts. Hence, we strongly believe that, for a sufficient number of walks, the SkipGram architecture itself can produce a similar (if not even a better) effect as the one achieved by force-steering random walks in certain directions. Another potential reason for not losing predictive power despite the significant simplification in our approach might be the fact that for churn prediction in particular structural equivalence is of much less importance. Indeed, while there are previous works [34] which make a connection between churn prediction in telco and homophily, to the best of our knowledge, there are none which mention something similar for structural equivalence.

Additionally, we would like to emphasize that our method achieved similar performance in terms of AUC as the original node2vec approach, despite the fact that we were using shorter walk lengths (30 vs. 80, as used in [5]). The value of parameters differing from the original node2vec setting (l=30 and i=5) were discovered through experimental evaluation.

Furthermore, we would like to mention that using a combination of defined RFM-based (interaction) and several ad-hoc structural features as a baseline, was not possible due to the computational intractability of e.g. closeness centrality and betweenness centrality.

Finally, observe that we aimed at predicting churners in

TABLE IV

COMPARISON IN TERMS OF AUC AND LIFT (AT 0.5%) BETWEEN BASELINES (RFM_s AND RFM_{s+ch}), ORIGINAL NODE2VEC $n2v$ AND OUR SCALABLE ADAPTATION (DENOTED AS $s{-}n2v$) ACROSS NETWORK DESIGNS EG_s^{Eucl}, AG_s, EG_{s+ch}^{Eucl}, AG_{s+ch} BASED ON **SUMMARY-RFM** FEATURES. SYMBOL "-" DENOTES THAT RANDOM WALKS COULD NOT BE CALCULATED WITHIN A REASONABLE TIME FRAME. THE BEST SCORES PER SCENARIO ARE MARKED IN BOLD.

Dataset	RFM_s		EG_s^{Eucl}				AG_s				RFM_{s+ch}		EG_{s+ch}^{Eucl}				AG_{s+ch}			
			n2v		s-n2v		n2v		s-n2v				n2v		s-n2v		n2v		s-n2v	
	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift
Prepaid	0.668	1.993	0.636	**2.182**	0.636	2.165	-	-	**0.679**	2.070	0.669	1.995	0.633	2.133	0.635	2.106	-	-	**0.698**	**2.439**
Postpaid	0.726	3.784	0.669	3.553	0.668	3.579	-	-	**0.748**	**4.153**	0.727	3.779	0.664	3.292	0.663	3.289	-	-	**0.748**	**4.430**

TABLE V

COMPARISON IN TERMS OF AUC AND LIFT (AT 0.5%) BETWEEN BASELINES (RFM_d AND RFM_{d+ch}), ORIGINAL NODE2VEC $n2v$ AND OUR SCALABLE ADAPTATION (DENOTED AS $s{-}n2v$) ACROSS NETWORK DESIGNS EG_d^{Eucl}, AG_d, EG_{d+ch}^{Eucl}, AG_{d+ch} BASED ON **DETAILED-RFM** FEATURES. SYMBOL "-" DENOTES THAT RANDOM WALKS COULD NOT BE CALCULATED WITHIN A REASONABLE TIME FRAME. THE BEST SCORES PER SCENARIO ARE MARKED IN BOLD.

Dataset	RFM_d		EG_d^{Eucl}				AG_d				RFM_{d+ch}		EG_{d+ch}^{Eucl}				AG_{d+ch}			
			n2v		s-n2v		n2v		s-n2v				n2v		s-n2v		n2v		s-n2v	
	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift	AUC	lift
Prepaid	**0.687**	**2.087**	0.625	1.937	0.628	1.954	-	-	0.666	1.953	0.686	2.087	0.626	1.933	0.624	1.926	-	-	**0.699**	**2.500**
Postpaid	**0.744**	**4.322**	0.586	1.954	0.583	1.905	-	-	0.732	3.792	**0.744**	**4.322**	0.585	1.981	0.583	1.916	-	-	0.733	3.922

month $M{+}2$ (a "one-month gap" approach) in order to make prediction more in-time and hence, more valuable from a business perspective. However, due to the fact that we use data in month $M{+}1$ to identify churners, we are actually in a position to make a prediction only for the next month, which, nevertheless, coincides with standard approaches [7, 34].

VII. CONCLUSION

The contributions of this paper are four-fold. First, we design RFM-enriched extensions of original graphs which enable conjoining both interaction and structural information. Second, we adapt the original node2vec approach to relax random walk generation and grid search tuning for two additional parameters, making it scalable for very large graphs. Third, conducted experiments showcase the performance benefits which stem from constructing RFM-augmented networks and learning node representations from these. Finally, this study is the first both in using node representations in CDR graphs for churn prediction and in applying the RFM framework together with unsupervised learning of node representations, in general.

This work is a preliminary study which inspires a series of interesting research questions. For our future work, we would like to explore how different ways of creating artificial nodes influence the predictive performance, as well as the effects that various similarity measures might have for the RFM-embedded networks. We are interested in a more profound analysis of the weak performance obtained using fine-grained RFM enriched networks. Additionally, it would be interesting to see whether the change in parameters, that is, the number of walks and walk length, would lead to an increase in AUC/lift scores. Finally, determining what impact a particular selection of random walk has on capturing homophily and structural equivalence phenomena in networks is still an open question.

ACKNOWLEDGMENTS

The authors would like to thank Aditya Grover and Jure Leskovec for making their code reusable.

REFERENCES

[1] Benoit, D. F., & Van den Poel, D. (2012). *Improving customer retention in financial services using kinship network information*. Expert Systems with Applications, 39(13), 11435-11442.

[2] Buckinx, W., & Van den Poel, D. (2005). *Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting*. European Journal of Operational Research, 164(1), 252-268.

[3] Cheng, C. H., & Chen, Y. S. (2009). *Classifying the segmentation of customer value via RFM model and RS theory. Expert systems with applications, 36(3), 4176-4184.*

[4] Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A. A., & Joshi, A. (2008, March). *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 (pp. 668-677). ACM.

[5] Grover, A., & Leskovec, J. (2016). *node2vec: Scalable feature learning for networks.* In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 855-864). ACM.

[6] Haenlein, M. (2013). *Social interactions in customer churn decisions: The impact of relationship directionality.* International Journal of Research in Marketing, 30(3), 236-248.

[7] Huang, B., Kechadi, M. T., & Buckley, B. (2012). *Customer churn prediction in telecommunications.* Expert Systems with Applications, 39(1), 1414-1425.

[8] Huang, Y., Zhu, F., Yuan, M., Deng, K., Li, Y., Ni, B., Dai, W., Yang, Q. & Zeng, J. (2015). *Telco churn prediction with big data.* In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 607-618). ACM.

[9] Hughes, A. M. (1994). *Strategic Database Marketing: The Masterplan for Starting and Managing a Profitable, Customer-based Marketing Program.*, Probus Publishing Co., Chicago, IL.

[10] Hung, S. Y., Yen, D. C., & Wang, H. Y. (2006). *Applying data mining to telecom churn management.* Expert Systems with Applications, 31(3), 515-524.

[11] Kahan, R. (1998). *Using database marketing techniques to enhance your one-to-one marketing initiatives.* Journal of Consumer Marketing, 15(5), 491-493.

[12] Keramati, A., Jafari-Marandi, R., Aliannejadi, M., Ahmadian, I., Mozaffari, M., & Abbasi, U. (2014). *Improved churn prediction in telecommunication industry using data mining techniques.* Applied Soft Computing, 24, 994-1012.

[13] Kim, K., Jun, C. H., & Lee, J. (2014). *Improved churn prediction in telecommunication industry by analyzing a large network.* Expert Systems with Applications, 41(15), 6575-6584.

[14] Kiss, C., & Bichler, M. (2008). *Identification of influencers-measuring influence in customer networks.* Decision Support Systems, 46(1), 233-253.

[15] Kronmal, R. A., & Peterson Jr, A. V. (1979). *On the alias method for generating random variables from a discrete distribution.* The American Statistician, 33(4), 214-218.

[16] Kusuma, P. D., Radosavljevik, D., Takes, F. W., & van der Putten, P. (2013). *Combining customer attribute and social network mining for prepaid mobile churn prediction.* In Proc. the 23rd Annual Belgian Dutch Conference on Machine Learning (BENELEARN) (pp. 50-58).

[17] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015). *Line: Large-scale information network embedding.* In Proceedings of the 24th International Conference on World Wide Web (pp. 1067-1077). ACM. Chicago.

[18] Ma, J., Lu, J., & Zhang, G. (2014). *A three-level-similarity measuring method of participant opinions in multiple-criteria group decision supports.* Decision Support Systems, 59, 74-83.

[19] McCarty, J. A., & Hastak, M. (2007). *Segmentation approaches in data-mining: A comparison of RFM, CHAID, and logistic regression.* Journal of business research, 60(6), 656-662.

[20] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space.* arXiv preprint arXiv:1301.3781.

[21] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality.* In Advances in neural information processing systems (pp. 3111-3119).

[22] Modani, N., Dey, K., Gupta, R., & Godbole, S. (2013). *CDR analysis based telco churn prediction and customer behavior insights: A case study.* In International Conference on Web Information Systems Engineering (pp. 256-269). Springer Berlin Heidelberg.

[23] Motahari, S., Jung, T., Zang, H., Janakiraman, K., Li, X. Y., & Hoo, K. S. (2014). *Predicting the influencers on wireless subscriber churn.* In Wireless Communications and Networking Conference (WCNC), 2014 IEEE (pp. 3402-3407). IEEE.

[24] Nanavati, A. A., Singh, R., Chakraborty, D., Dasgupta, K., Mukherjea, S., Das, G., Gurumurthy, S. & Joshi, A. (2008). *Analyzing the structure and evolution of massive telecom graphs.* IEEE Transactions on Knowledge and Data Engineering, 20(5), 703-718.

[25] Owczarczuk, M. (2010). *Churn models for prepaid customers in the cellular telecommunication industry using large data marts.* Expert Systems with Applications, 37(6), 4710-4712.

[26] Phadke, C., Uzunalioglu, H., Mendiratta, V. B., Kushnir, D., & Doran, D. (2013). *Prediction of subscriber churn using social network analysis.* Bell Labs Technical Journal, 17(4), 63-75.

[27] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014). *Deepwalk: Online learning of social representations.* In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710). ACM.

[28] Raeder, T., Lizardo, O., Hachen, D., & Chawla, N. V. (2011). *Predictors of short-term decay of cell phone contacts in a large scale communication network.* Social Networks, 33(4), 245-257.

[29] Sağlam, B., Salman, F. S., Sayın, S., & Türkay, M. (2006). *A mixed-integer programming approach to the clustering problem with an application in customer segmentation.* European Journal of Operational Research, 173(3), 866-879.

[30] Saravanan, M., & Raajaa, G. V. (2012, December). *A Graph-Based Churn Prediction Model for Mobile Telecom Networks.* In International Conference on Advanced Data Mining and Applications (pp. 367-382). Springer Berlin Heidelberg.

[31] Saravanan, M., Manoj, P., Smitha, G. B., & Lakshmi, V. (2015). *Aatish-A New Profile-Based Recommendation Services for Mobile Telecom Network Subscribers.* In Network Intelligence Conference (ENIC), 2015 Second European (pp. 160-164). IEEE.

[32] Stone, B. (1994). *Successful Direct Marketing Methods.* NTC Business Books. McGraw-Hill Education, New York.

[33] Verbeke, W., Dejaeger, K., Martens, D., & Baesens, B. (2010). *Customer churn prediction: does technique matter?.* In Proceedings of the Joint Statistical Meeting, JSM2010, Vancouver, Canada.

[34] Verbeke, W., Martens, D., & Baesens, B. (2014). *Social network analysis for customer churn prediction.* Applied Soft Computing, 14, 431-446.

[35] Zhang, X., Zhu, J., Xu, S., & Wan, Y. (2012). *Predicting customer churn through interpersonal influence.* Knowledge-Based Systems, 28, 97-104.

[36] Zhu, T., Wang, B., Wu, B., & Zhu, C. (2011). *Role defining using behavior-based clustering in telecommunication network.* Expert Systems with Applications, 38(4), 3902-3908.