Time series for early churn detection: using similarity based classification for dynamic networks

María Óskarsdóttir[a,*], Tine Van Calster[a], Bart Baesens[a,b], Wilfried Lemahieu[a], Jan Vanthienen[a]

[a] *KU Leuven, Dept. of Decision Sciences and Information Management, Naamsestraat 69, B-3000 Leuven, Belgium*
[b] *University of Southampton, University Road, Southampton SO17 1BJ, United Kingdom*

Abstract

The success of retention campaigns in fast-moving and saturated markets, such as the telecommunication industry, often depends on accurately predicting potential churners. Being able to identify certain behavioral patterns that lead to churn is important, because it allows the organization to make arrangements for retention in a timely manner. Moreover, previous research has shown that the decision to leave one operator for another, is often influenced by the customer's social circle. Therefore, features that represent the churn status of their connections are usually good predictors of churn when it is treated as a binary classification problem, which is the traditional approach.

We propose a novel method to extract time series data from call networks to represent dynamic customer behavior. More precisely, we use call detail records of the customers of a telecommunication provider to build call networks on a weekly basis over the period of six months. From each network, we extract features based on each customer's connections within the network, resulting in individual time series of link-based measures. The time series are then classified using the recently proposed similarity forests method, which we further extend

*Corresponding author

Email addresses: `Maria.Oskarsdottir@kuleuven.be` (María Óskarsdóttir), `Tine.VanCalster@kuleuven.be` (Tine Van Calster), `Bart.Baesens@kuleuven.be` (Bart Baesens), `Wilfried.Lemahieu@kuleuven.be` (Wilfried Lemahieu), `Jan.Vanthienen@kuleuven.be` (Jan Vanthienen)

in various ways to accommodate multivariate time series.

We show that predicting churn with customer behavior represented by time series is a suitable option. According to our results, the similarity forests method together with some of our proposed extensions, perform better than the one-nearest neighbor benchmark for time series classification. Using a time series of a single feature only, the similarity forests method performs as good as traditional churn prediction methods using more features. In fact, compared to traditional methods, similarity forests based approaches perform better when predicting further in the future, and are therefore better at detecting churn early, allowing organizations to make arrangements for retention in a timely manner.

Keywords: Multivariate time series, Churn prediction, Call detail records, Time series classification, Social networks, Dynamic networks

---

1. Introduction

The mobile telecommunication industry (telco) is a saturated and fast-moving market where customers can easily change providers to obtain a subscription which they are satisfied with. At the same time, it is less profitable for providers to attract new customers than to prevent current customers from quitting (De Caigny et al., 2018). Therefore, providers engage in building predictive models in order to identify which customers are most likely to leave –or churn– so they can be offered promotions to persuade them to stay (Verbeke et al., 2011). In business-oriented contexts, where costs and benefits are a concern, it is furthermore valuable to detect the potential churners early enough for the campaigns to achieve optimal success and maximize their return (Verbraken et al., 2013). Research has shown that features that incorporate the influence of prior churners in a customer's social circle –their ego-net– are usually good predictors of churn when it is treated as a binary classification problem (Óskarsdóttir et al., 2017; Verbeke et al., 2014; Phadke et al., 2013). These features are extracted from call networks which are constructed based on call detail records (CDR) by linking together customers who have called or texted each other. The result is a representation of the customers' calling behavior and social circle that can be utilized to predict churn with enhanced accuracy (Óskarsdóttir et al., 2017).

Most studies consider static networks by aggregating the call records for a fixed period of time, e.g. one month (Backiel et al., 2016; Kim et al., 2014; Zhang et al., 2012). This approach has an obvious drawback since it fails to take into account the temporal nature of the networks, which in turn might result in a partial and biased representation (Navarro et al., 2017). As human behavior is susceptible to changes over time, it is reasonable to assume that during the time leading up to a customer's decision to churn, they start behaving differently. Furthermore, the behavior among churners might be similar and distinguishable from the behavior of non-churners (Chen et al., 2012).

In this study, we propose a way of incorporating the time dimension of customer behavior and present a dynamic end-to-end approach to the churn prediction task in telco, which even leads to higher model performance. More precisely, we construct call networks on a weekly basis for a period of six months and extract network features from each customer's ego-net to capture the dynamics of customer churn behavior, using three distinct datasets. This is a novel procedure of processing CDR data as far as we know. To predict customer churn, we then perform classification with the resulting multivariate time series.

Although time series classification is an active research field, an extensive benchmarking study of the numerous methods that have been proposed over the years showed that only complex methods outperformed the one-nearest neighbor (1-NN) with dynamic time warping (DTW) benchmark (Bagnall et al., 2017). However, this 1-NN classifier with DTW as a similarity measure, computes all pairwise similarities between the time series which makes it computationally expensive, especially for a large dataset. Although extensive, the benchmarking study (Bagnall et al., 2017) did not include classification of multivariate time series and in fact, the literature on the topic is scarce. Therefore, we turn to a recently proposed classification method called similarity forests, which has not been applied to time series before (Sathe & Aggarwal, 2017). This extension of random forests, which is applicable to any type of data for which similarity between observations is defined, computes only a fraction of the pairwise similarities and is thus orders of magnitude faster and computationally less expensive than the 1-NN DTW benchmark. In addition, it is robust to noise and missing values and performs well in terms of accuracy. We propose novel extensions to the similarity forests method to make it applicable to multivariate time series by, on the one hand, incorporating a multivariate distance measure for time series and, on the other hand, exploiting the functionality of random forests in two distinct ways.

We apply our extended methods together with the original similarity forests to the multivariate time series to classify the customers and compare their performance with the 1-NN DTW benchmark. Thereby, we perform the analyses with variation in the length of the time series, to better understand the dynamics of the churn process. Our results show that the similarity forests perform substantially better than the benchmark method. Finally, we compare our dynamic approach to the more traditional static one, by extracting features from a static network and predicting churn with two commonly used classifiers, i.e. logistic regression and random forests. We demonstrate that time series classification is better suited for detecting churn early. Concretely, we show that by using our proposed method for representing dynamic behavior of customers together with a powerful binary classifier, i.e. similarity forests, telcos have an optimal chance of detecting churners well in advance to take appropriate action for retention.

In the next section, we discuss related literature on dynamic networks and time series classification. In Section 3 we introduce our proposed method of extracting multivariate time series from CDR data to represent dynamic customer behavior. Thereafter, in Section 4, we describe the similarity forests method and our adaptations of the method for multivariate time series. In Section 5, we explain the setup of the experiments conducted in the paper and present the results in Section 6. Finally, we summarize our findings and contributions and discuss future work in Section 7.

2. Theoretical background

2.1. Dynamic networks

Although data used to build networks typically arrives as streams of time stamped transactions, the analysis is most often conducted on static networks, with the information from the data streams aggregated. However, a more natural way, and one that is more representative of the reality of social networks and human behavior and interaction, is to base the analysis on dynamic or temporal networks. This was the topic of a recent special issue of Machine Learning, that focussed on innovative techniques that handle time-evolving networks, while also acknowledging the challenges that arise with such data (Rouveirol et al., 2017).

In the current literature, there are mainly two approaches for dealing with dynamic networks. One way is to capture the time-dependent network structure in a single network by using temporal edges between the same vertices at different moments. The other approach is to look at time series of networks, i.e. a sequence of static networks, where each network is built by aggregating interactions over a fixed period of time and thus gives snapshots of the network at given times (Santoro et al., 2011; Simmhan et al., 2014).

Much of the research on dynamics of social networks focuses on the evolution and discovery of communities, and to discover what drives network formation (Kossinets & Watts, 2006). For example, Aiello & Barbieri (2017) looked at the evolution of ego-nets in social media. Using a sequence of network snapshots, they studied how link recommendations affect the expansion of people's social circles. In the case of community detection, the recently proposed Tiles method extracts communities and tracks their evolutions over time (Rossetti et al., 2017). Furthermore, Lin et al. (2008) proposed FacetNet, a method to analyse, in a single process, communities and how they evolve. Finally, to distinguish peer-to-peer influence from homophily, Aral et al. (2009) studied the diffusion of a mobile service product over a social network. Evidently, these studies are mostly aimed towards descriptive analytics, while prediction and classification in dynamic networks, which is the main application in this paper, is scarce in the literature. A few recent publications include an approach for outlier detection in dynamic networks (Ranshous et al., 2017), link prediction (Ranshous et al., 2017) and the modeling of epidemic spreading (Joneydi et al., 2017) and peer influence (Wölbitsch et al., 2017).

2.2. Time series classification

A lot of research has been conducted on time series classification in the last years and numerous algorithms have been proposed. Many of these were included in an extensive benchmarking study, which showed that the few techniques that outperformed the simple 1-NN classifier with DTW were complex ensembles of methods (Bagnall et al., 2017). The benchmarking study did however not include methods designed for classification of multivariate time series.

Many methods developed for multivariate time series classification depend on either some kind of featurization of the time series data to construct a tabular data set or on dimensionality reduction, combined with a binary classification technique (Weng & Shen, 2008; Batal et al., 2009; Orsenigo & Vercellis, 2010). In an application designed for electro-cardiograms, where early classification is vital, He et al. (2015) proposed a method to detect distinctive shapelets to use as core features in binary classifiers. Similarly, Kadous & Sammut (2005) used constructive induction, which takes advantage of recurring substructures that are often present in multivariate time series. Finally, Esmael et al. (2012) proposed a memory-based classifier that classifies incremental segments of the multivariate time series based on trend and value-based approximation of the segments.

In contrast to these featurization techniques, Wang et al. (2016) used recurrent neural networks and adaptive differential evolution to classify multivariate time series with robust results. Furthermore, Górecki & Łuczak (2015) proposed using a combination of dynamic time warping distance between the time

Table 1: An example of a CDR log. In the actual dataset the phone numbers are encrypted.

Call Start Date	Call Start Time	Call Duration (sec)	From Number	To Number
01MAY2017	14:51:14	715	(202) 555-0116	(701) 555-0191
02MAY2017	14:34:37	29	(803) 555-0129	(202) 555-0116
01MAY2017	20:34:14	9	(803) 555-0117	(406) 555-0137
02MAY2017	20:03:38	89	(701) 555-0148	(803) 555-0129

series and the derivative of the time series, to be used in classification with 1-NN classifiers. The similarity forest method applied in this paper also depends on distances between time series, but used in combination with random forests (Sathe & Aggarwal, 2017).

3. Dynamic representation of calling behavior

3.1. Time series extraction

Social network analytics for churn prediction in telco is usually based on static networks, which means that the network representation is frozen in time. Although these methods have shown to be successful, they fail to take into account the temporal aspect of phone usage. In this section, we propose a method that captures the dynamics of people's calling behavior, while also incorporating influence from churners and non-churners in their social circle.

The approach is based on phone call logs of the customers of a telco, commonly referred to as call detail records (CDR), see Table 1 for examples. These logs are kept for billing purposes, but as numerous studies over the last decade have shown, they can be used in various ways (Naboulsi et al., 2016). In particular, for social network analysis, call traffic for a fixed period of time is aggregated to build social networks. Subsequently, the networks can be used to investigate the structure of interactions among the customers of the telco. To extract dynamic behavioral patterns from the CDR, we follow an approach similar to Santoro et al. (2011). Thus, we achieve an evolution of call networks, by aggregating the logs of one week to build each network. The result is a sequence of static networks. As the CDR spans six months in total, this gives us

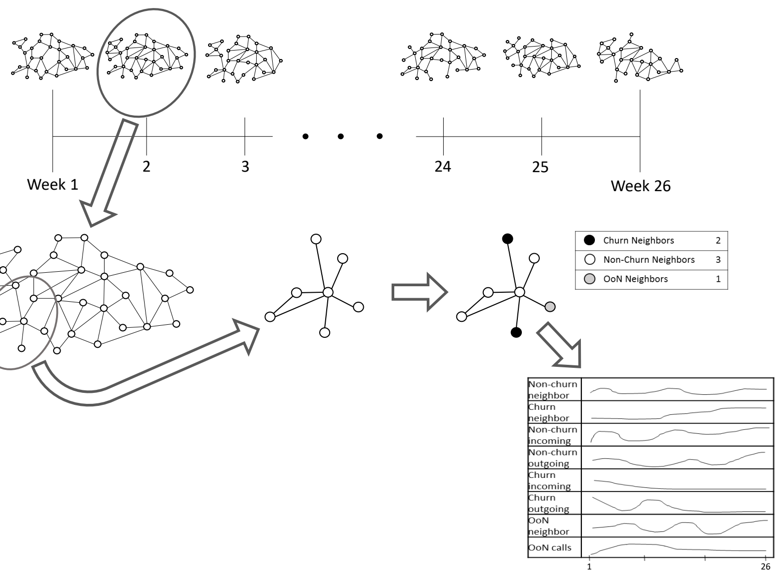

Figure 1: The figure demonstrates the featurization process, which begins with a time series of one-week call networks and ends with eight distinct time series for each customer. In the ego-net the white nodes denote a non-churner, the black nodes denote a churner and the gray nodes denote an out-of-network neighbor.

a time series of call networks with 26 observations. The process is demonstrated in Figure 1.

From the CDR we know when each phone call was made and therefore also when the customers are active and when they stop being active. We use this information to determine churn as follows. Customers are labelled as churners when they have been inactive for 30 consecutive days and the moment of churn is defined as the first of those 30 days, as is common in the literature (Verbeke et al., 2014). As some phone calls are made to people who are not customers of the telco, we refer to them as out-of-network, or OoN, connections. This information is aggregated on a weekly level and thus we know for each week who churned and who is an OoN connection.

Featurization is a process where attributes are derived from a network. For each network in the sequence of call networks, we extract features from the ego-net of each active customer, see Figure 1. Ego-nets are fundamental structures in

Table 2: The features extracted from the ego-nets.

Name	Description
Non-churn neighbor	Number of neighbors that did not churn in the week
Churn neighbor	Number of neighbors that did churn in the week
Non-churn incoming	Duration of phone calls received from neighbors that did not churn in the week
Non-churn outgoing	Duration of phone calls made to neighbors that did not churn in the week
Churn incoming	Duration of phone calls received from neighbors that did churn in the week
Churn outgoing	Duration of phone calls made to neighbors that did churn in the week
OoN neighbor	Number of neighbors that are customers of a different telco
OoN calls*	Duration of phone calls made to neighbors that are customers of a different telco

*Since some CDR only contain phone calls made by their customers, we do not make a distinction between incoming and outgoing traffic of OoN neighbors.

social networks because they portray the social interaction that occur between individual people (Aiello & Barbieri, 2017). In our setup, the ego-net of a customer is the set of all people that they were in contact with, i.e. via phone calls, during the one-week period and we refer to the other people in the ego-net as neighbors. We use the information about churners and OoN neighbors in the respective week to label the people in the ego-net and aggregate information from the CDR to assign weights to the connections between them. In this way, we define and extract eight features for the customers in each one-week network, as listed in Table 2. Together, these features capture different aspects of calling behavior with respect to communication with non-churners, churners and OoN neighbors. Churners are only considered as churners in the week in which they churn and thus can only affect the ego-net of others at one point in time. Their churn week marks the end of their time series and afterwards they are viewed as OoN neighbors. Our proposed approach results in a multivariate time series for each customer of the telco. The starting point for each individual is the first week in which they are active and the end point is either the last week in the CDR data or the week where the customer churns. Because we specified the featurization with eight features, the multivariate time series have eight dimensions.

3.2. Time series notation and distance measures

We will now briefly explain the notation for multivariate time series that we use in the paper. In addition, we specify the distance measures which are deployed in the classification methods.

Let O be a dataset with N multivariate time series of dimension M, i.e. each observation $O_i \in O$ in the dataset consist of M time series

$$O = \begin{pmatrix} O_1 \\ \vdots \\ O_N \end{pmatrix} = \begin{pmatrix} O_{11} & \cdots & O_{1M} \\ \vdots & \ddots & \vdots \\ O_{N1} & \cdots & O_{NM} \end{pmatrix} \tag{1}$$

where $O_{ij} = P = (p(t))_{t=1}^{T}, i = 1, \ldots, N, j = 1 \ldots, M$ is a time series with T time steps. The time series of each observation, i.e. customer, has the same length, but the time series of different observations can vary in length. Furthermore, each observation $O_i \in O$ is labelled, meaning that it belongs to one of the two classes: churn or non-churn, denoted by c_1 and c_0 respectively. Thus, there is a label vector

$$y = \begin{pmatrix} l_1 \\ \vdots \\ l_N \end{pmatrix}$$

for each $l_i \in \{c_0, c_1\}, i = 1 \ldots, N,$.

There are several ways to compute the distance between two time series depending on whether or not the time series are multidimensional. Here, we discuss two commonly used measures for comparison of time series and time series classification together with their multidimensional counterpart.

The Euclidean distance (ED) between two univariate time series P and Q of equal length is defined as

$$ED(P, Q) = \sqrt{\sum_{t=1}^{T} (p(t) - q(t))^2}.$$

Analogously, the Frobenius distance (FR) between two multivariate time series $P = (P_1, \ldots, P_M)$ and $Q = (Q_1, \ldots, Q_M)$ of equal length and dimension M is defined as

$$FR(P, Q) = \sqrt{\sum_{t=1}^{T} \sum_{m=1}^{M} (p_m(t) - q_m(t))^2}.$$

Although the Euclidean distance is a straightforward way to compare two time series, it has been shown to give suboptimal results when used for the classification of time series (Bagnall et al., 2017).

Dynamic time warping is an alternative measure that overcomes the one-to-one alignment constraint and allows for one-to-many alignment to find an optimal path between the two time series. To compute the dynamic time warping distance between the time series P and Q of lengths T and S, a matrix of size $T \times S$ is constructed where the element in cell (i, j) is the squared distance between elements p_i and p_j or $(p_i - p_j)^2$. The method finds a path

$$w = w_1, \ldots w_t$$

from cell $(0, 0)$ to cell (T, S). While aggregating the values in the cells, it goes through the matrix to compute the path's warping cost. Finally, the dynamic time warping distance is defined as the path with minimum cost, or

$$DTW(P, Q) = \min \sum_{i=1}^{t} w_t.$$

The optimal path is found using dynamic programming and the recurrence

$$D(i, j) = (p_i - q_j)^2 + \min(D(i-1, j-1), D(i, j-1), D(i-1, j))$$

to compute the cost of the warping path. Euclidean distance is a special case of this approach where the minimum in the equation above is replaced with $D(i-1, j-1)$ only. Multivariate dynamic time warping is a multidimensional generalization of DTW where two approaches are possible, depending on whether the dimensions are considered to be independent or not (Shokoohi-Yekta et al., 2017). In the former case, which we assume here, it is defined as

$$MDTW(P,Q) = \sum_{m=1}^{M} DTW(P_m, Q_m)$$

or the cumulative distance of each dimension.

4. Multivariate similarity forests

4.1. The similarity forests method

Similarity forests are a novel supervised machine learning technique for the binary classification of objects in an arbitrary dataset, where the only requirement is that similarities between the objects can be obtained (Sathe & Aggarwal, 2017). The technique is an extension of random forests that in a similar way constructs a multitude of decision trees at training time and combines the results to obtain a prediction for each object (Breiman, 2001). Contrary to random forests, similarity forests perform a split of a node based on similarity between the objects in the node as described below.

Let $O_1, O_2, \ldots O_n$ be n observations or data objects of a dataset and assume that they can be embedded theoretically as data points $\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_n$ in some multidimensional space. The exact representation of the data objects in the multidimensional space is not important as long as there is a way to obtain a similarity between each pair of objects. We denote with S_{ij} the similarity between objects O_i and O_j. Finally, we also assume that each observation is labelled, i.e. it belongs to one of the two classes c_0 and c_1.

When a node $\mathcal{N}$ is to be split, a pair of observations $(O_i, O_j) \in \mathcal{N}$, with opposite labels, is randomly selected to determine the direction of the split. According to Sathe & Aggarwal (2017), selecting opposite labels leads to a more discriminative split. The quality of the split is evaluated by projecting the rest of the observations in the node along the direction from O_i to O_j. If $O_k \in \mathcal{N}$, then the projection of O_k along the direction from O_i to O_j is

$$P(\bar{X}_k) = (\bar{X}_k - \bar{X}_i) \frac{\bar{X}_j - \bar{X}_i}{\|\bar{X}_j - \bar{X}_i\|}$$

$$
\begin{aligned}
&= \frac{\bar{X}_k \cdot \bar{X}_j - \bar{X}_k \cdot \bar{X}_i - \bar{X}_i \cdot \bar{X}_j + \bar{X}_i \cdot \bar{X}_i}{\sqrt{\|\bar{X}_j\|^2 + \|\bar{X}_i\|^2 - 2\bar{X}_i \cdot \bar{X}_j}} \\
&= \frac{S_{kj} - S_{ki} - S_{ij} + S_{ii}}{\sqrt{S_{ii} + S_{jj} - 2S_{ij}}} \\
&\propto S_{kj} - S_{ki} + C
\end{aligned}
$$

where C is a constant. Therefore, the projection only depends on the difference between the similarities of the given observation with each of the two randomly selected observations. So, to compute a split relative to the pair (O_i, O_j), the observations in the node $O_k \in \mathcal{N}$ need to be sorted in order of $(S_{kj} - S_{ki})$. The split of the node is determined by dividing this ordered vector of differences into two parts, so the weighted Gini quality,

$$GQ(\mathcal{N}_1, \mathcal{N}_2) = \frac{n_1 G(\mathcal{N}_1) + n_2 G(\mathcal{N}_2)}{n_1 + n_2}, \tag{2}$$

of the two children nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ with n_1 and n_2 values, respectively, is minimized. In the equation, the Gini index of node $\mathcal{N}$ is given by

$$G(\mathcal{N}) = 1 - p_0^2 - p_1^2$$

where p_0, p_1 are the fraction of objects with labels c_0 and c_1 respectively. Each tree in the forest is constructed recursively in this way until the leaf nodes are pure in terms of class labels.

The time it takes to split a node in similarity forests, is linear in terms of the number of points in the node. Hence, the time to create a level in a decision tree is also linear. Therefore, if the tree is of height $O(\log(n))$, $O(n \log(n))$ is the time required to build it. Compared to the 1-NN DTW benchmark for time series classification which has complexity $O(n^2)$, similarity forests thus provide a substantial decrease in complexity.

Although similarity forests depend on similarities between observations, they can also be applied using distances, which is more natural in the case of time series. To convert distances to similarities, Sathe & Aggarwal (2017) discuss both an exact approach, using the cosine law, and an approximate approach. As the first approach is computationally very expensive, especially for large data sets, we apply the second one, which entails using squared distances instead of similarities, since the self-similarity of each object in our datasets is the same. For a more detailed discussion, we refer to (Sathe & Aggarwal, 2017).

4.2. Multivariate similarity forests

The similarity forests method described above, can be applied directly to univariate time series by using the distance measures described in Section 3.2. We denote this method by SF_S. In addition, in this subsection, we propose three ways of extending the similarity forests method to multivariate time series. In what follows, we assume that $O_i = (O_{i1}, \dots O_{iM})$ and $O_j = (O_{j1}, \dots O_{jM})$ are two multivariate time series of dimension M.

Multivariate distance measures. The first approach uses multidimensional distance measures to compute the distance between the objects O_i and O_j in a single number. This is a straightforward extension of similarity forests, since the key idea is that only the similarity between objects is known. Thus, when splitting a node, the similarity between observations is computed using either the FB or the MDTW distance as defined in Section 3.2. We denote this method with SF_{MV}.

Pick the best feature at each split. The second approach considers the predictive capabilities of each feature as a tree is constructed. When splitting a node, Equation 2 is used to compute the Gini quality for each feature separately. The feature that results in the lowest value, is chosen for the split, similar to a regular decision tree. More precisely, for a node $\mathcal{N}$ and each $m \in M$, a split is determined relative to the randomly selected pair of objects (O_{km}, O_{lm}), giving the Gini qualities $GQ_m, m = 1, \dots, M$. Subsequently, the feature m such that $\min_{m \in M} GQ_m$ is selected to determine the split. As the features are considered separately and the splits are based on single-dimensional time series, the EU and DTW distances are used, see Section 3.2.

This approach combines the strengths of decision trees and similarity forests within each tree. Furthermore, it inherently incorporates a feature selection

Table 3: Description of the datasets

Name	# Customers	Year	Churn rate (4)	Churn rate (8)	Churn rate (12)	Contract type
Dataset 1	8.6×10^4	2015	38.75%	21.53%	10.90%	Postpaid
Dataset 2	1.5×10^6	2010	35.04%	22.85%	9.57%	Prepaid
Dataset 3	1.7×10^5	2013	15.63%	8.42%	4.38%	Prepaid

process by building trees using the features that are most discriminative. This method is denoted by SF_{PB}.

Ensemble of similarity forests. The final approach exploits the ensemble property of random forests. For each feature in the multidimensional time series, a number of trees are built using similarity-based splitting, as described in Section 4.1, and the EU and DTW distances, as defined in Section 3.2. Afterwards the predictions are averaged. In this case, separate similarity forests with k trees are applied to the observations $(O_{1m}, \ldots, O_{Nm}), m = 1, \ldots, M$. The resulting forests are then applied to the test set to produce class probabilities $prob_1, \ldots, prob_M$. Finally, the class probabilities of this ensemble approach are obtained by averaging the probabilities of the individual similarity forests

$$prob = \frac{1}{M} \sum_{m=1}^{M} prob_m.$$

This approach is faster than the previous two since no additional computations are needed when building the trees. However, knowledge about which variables are the best predictors, is lost. This method is denoted by SF_E.

5. Experimental setup

We analyse three distinct CDR datasets from European telcos. Table 3 provides a summary of the datasets. They all contain six consecutive months of phone call traffic between the telco's customers and vary in size and churn rates as Table 3 shows. The method described in Section 3 was used to extract eight distinct time series from the call networks. This resulted in a multidimensional time series for each customer in the dataset. We note that the time series of

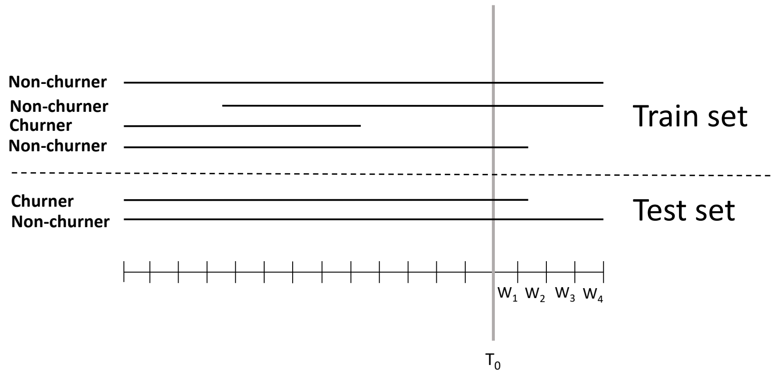

Figure 2: Timeline for the experimental setup.

each customer do not all have the same length, as some of them are active at the beginning of the six months while others appear later on, as seen in Fig. 2. Furthermore, as soon as a customer churns, his time series ends. Although there are 26 weeks of CDR data, not all of them can be used in our experiments. Because of the churn definition, the last five weeks are needed to construct the labels and are not used for the analyses. In addition, the out-of-time experimental setup described below requires four weeks of data points after training the models. In order to determine how long the churn process is and therefore, for how long customer behavior must be observed for accurate predictions, we conduct experiments with time series of different lengths, i.e. spanning 4, 8 and 12 weeks. These values represent observation periods of one, two and three months. The table displays the churn rates in the train sets with respect to the length of the time series. We note that these churn rates are higher than in comparable studies (Verbeke et al., 2014; Backiel et al., 2016). As explained below, we look at individual behavior and not fixed time points. This results in a pooling effect, since everyone who churned during the observation time is counted as a churner in the train set.

In all cases, we predict churn in the next four weeks –or one month– following a specific point in time T_0, for which time series before T_0 are long enough. Fig. 2 shows the experimental setup. The train and test sets are constructed in the following way. The non-churners in the train set are defined as all customers who had not churned before time T_0 and the churners in the test set are all customers who did churn before the same time. After the labelling, we draw a random sample, that preserves the churn rate of the original dataset, of 20000 customers on which to train the models. In a similar way, the non-churners in the test set are defined as all customers who did not churn during the four weeks after time T_0 and churners as customers who did churn during that same time. For evaluation purposes, we also keep track of the week in which churn took place. To create the test set, 5000 customers are randomly sampled. This leads to a 80/20% split of train and test set. In addition, because there are churners in the test set that might be labelled as non-churners in the train set, this is an out-of-time experimental setup which further facilitates this dynamic approach for predicting churn. As a result, there are 3 sets of train and test sets for each dataset, one for each length of time series. Table 3 shows the churn rate in each train set. Furthermore, each dataset contains the eight time series features with the same customers for each feature.

We apply the three variations of multivariate similarity forests as described in Section 4.2 to each dataset. In addition, we deploy the original similarity forests method using each of the eight single-dimensional time series and finally we compare the results to the 1-NN DTW benchmark as well as 1-NN EU, 5-NN DTW and 5-NN EU Bagnall et al. (2017). We also tried the multivariate version of the nearest neighbor classifier but since it rarely gave results that are better than a random model, it was excluded from the experiments.

We use three distinct performance measure to evaluate our models. Firstly, the area under the receiver operating characteristic curve (AUC) is used, since it summarizes the trade-off between model sensitivity and specificity, regardless of the cut-off value between churners and non-churners, and it is frequently used when evaluating churn prediction models. Secondly, the commonly used top decile lift (lift) represents how much better a prediction model is at identifying churners, compared to a random sample of customers by comparing the ratio of churners in the 10% of the highest predicted probabilities to the ratio

Table 4: Performance measured in AUC. The three highest values for each dataset are denoted in boldface.

Dataset		Dataset 1			Dataset 2			Dataset 3		
TS length		4	8	12	4	8	12	4	8	12
Method	dist									
SF_S	EU	0.871	**0.883**	**0.888**	0.634	0.637	**0.650**	**0.659**	0.606	0.627
	DTW	0.873	0.882	**0.903**	0.637	0.641	**0.649**	**0.656**	0.613	0.648
SF_E	EU	0.792	0.786	0.829	0.618	0.637	0.625	**0.651**	0.620	0.593
	DTW	0.777	0.780	0.809	0.616	**0.656**	0.627	0.633	0.603	0.598
SF_{PB}	EU	0.793	0.700	0.707	0.575	0.598	0.549	0.644	0.609	0.599
	DTW	0.773	0.674	0.700	0.594	0.595	0.539	0.639	0.595	0.572
SF_{MV}	EU	0.540	0.575	0.626	0.570	0.634	0.603	0.540	0.581	0.544
	DTW	0.542	0.551	0.565	0.555	0.627	0.599	0.525	0.546	0.545
1-NN	EU	0.529	0.526	0.522	0.513	0.513	0.509	0.510	0.501	0.505
	DTW	0.584	0.623	0.635	0.520	0.518	0.502	0.500	0.549	0.531
5-NN	EU	0.535	0.500	0.533	0.515	0.517	0.510	0.508	0.512	0.503
	DTW	0.731	0.691	0.663	0.538	0.537	0.514	0.516	0.521	0.518

of churners in the actual customer base. Finally, the expected maximum profit measure (EMP) is a recent performance measure, specially designed to measure performance of churn prediction models while taking into account the cost and expected return of a retention campaign (Verbraken et al., 2013). When computing the EMP, we use the default parameter values, $\alpha = 6, \beta = 14, \mathrm{CLV} = 200, d = 10, \mathrm{and} f = 1$ as suggested by Verbraken et al. (2013).

6. Results and discussion

6.1. Comparison of methods

The results in Table 4 show the performance measured in AUC of all similarity forests techniques together with the 1-NN benchmark and 5-NN applied to the three datasets and three lengths of time series using the two types of distance measures. Tables 5 and 6 show the results measured in top decile lift and EMP, respectively. For the univariate techniques, i.e. SF_S and all the NN, we only report the results of the model with the feature that gave the highest performance. A few observations can be made when looking at these tables and comparing different components. Firstly, dataset 1 performs overall much

Table 5: Performance measured in top decile lift. The three highest values for each dataset are denoted in boldface.

Dataset		Dataset 1			Dataset 2			Dataset 3		
TS length		4	8	12	4	8	12	4	8	12
Method	dist									
SF_S	EU	6.05	**6.09**	**7.01**	**2.15**	1.87	2.11	1.80	**2.56**	2.08
	DTW	5.50	**6.09**	**6.69**	**2.21**	2.02	1.91	1.92	**2.32**	2.01
SF_E	EU	4.50	4.38	5.51	1.92	2.11	1.91	2.04	**2.26**	1.70
	DTW	4.57	4.53	4.88	2.11	2.11	1.71	1.86	**2.26**	1.89
SF_{PB}	EU	4.73	3.28	3.15	0.35	1.30	1.01	1.98	2.13	1.38
	DTW	4.26	3.44	3.78	0.38	1.20	0.92	1.68	1.77	0.88
SF_{MV}	EU	2.02	1.48	1.57	**2.15**	1.78	2.05	1.86	1.95	1.13
	DTW	2.17	1.72	1.26	2.11	2.02	1.91	1.62	1.65	0.94
1-NN	EU	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	DTW	1.86	2.66	2.83	0.00	0.75	0.00	0.00	1.59	0.94
5-NN	EU	0.00	0.00	1.18	0.00	0.00	0.00	0.84	0.00	0.00
	DTW	0.00	4.30	3.46	0.00	0.00	1.24	1.14	0.73	0.75

Table 6: Performance measured in EMP. The three highest values for each dataset are denoted in boldface.

Dataset		Dataset 1			Dataset 2			Dataset 3		
TS length		4	8	12	4	8	12	4	8	12
Method	dist									
SF_S	EU	0.2887	0.2617	**0.4097**	0.0990	0.0754	**0.1156**	0.0000	**0.0219**	0.0000
	DTW	0.2607	0.2759	**0.4046**	0.0788	**0.1028**	0.1011	0.0000	**0.0177**	0.0024
SF_E	EU	0.2585	0.1236	0.2926	0.0670	**0.1032**	0.1011	0.0000	0.0058	0.0000
	DTW	0.2417	0.1976	**0.3178**	0.0761	0.0799	0.0546	0.0000	0.0033	0.0000
SF_{PB}	EU	0.1038	0.0301	0.0344	0.0001	0.0082	0.0058	0.0000	0.0001	0.0008
	DTW	0.0811	0.0327	0.0817	0.0074	0.0016	0.0091	**0.0272**	**0.0177**	0.0000
SF_MV	EU	0.0194	0.0000	0.0004	0.0816	0.0734	0.0728	0.0007	0.0079	0.0000
	DTW	0.0133	0.0191	0.0000	0.0954	0.0881	0.0858	0.0038	0.0009	0.0000
1-NN	EU	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	DTW	0.0843	0.1672	0.2519	0.0000	0.0059	0.0000	0.0000	0.0000	0.0000
5-NN	EU	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	DTW	0.0105	0.1505	0.2828	0.0001	0.0003	0.0001	0.0000	0.0000	0.0000

better than the other two. This can be explained by the fact that the contracts in this dataset are postpaid whereas the other datasets have prepaid contracts. As previous research has shown, performance of models using postpaid call networks tends to be higher, because the networks are more densely connected and therefore contain more information (Óskarsdóttir et al., 2016). The second observation is the difference in performance between the two types of distance measures, or rather, the lack of difference. The distance measure only seems to make a difference for the nearest neighbor classifiers, where using DTW gives higher performance. The same does not hold for the similarity forests methods. To test whether there is a significant difference between using the two types of distance measures, we apply a Kruskal-Wallis rank sum test which is a non-parametric method for testing whether samples originate from the same distribution. Furthermore, we use the non-parametric Friedman test to statistically compare the differences among the results (Garcia & Herrera, 2008). The Kruskal-Wallis test showed no statistical difference in performance measured in both AUC (p-value = 0.2531) and lift (p-value = 0.2289). However, there is statistical difference when performance is measured in EMP at the 95% confidence level, but not at the 99% confidence level (p-value = 0.03483), most likely because of how often the result of EMP is zero. According to the Friedman test, there is no statistical difference in performance for all of the three metrics. As a result, we assume that there is no difference between using on one hand the Euclidean distances and the dynamic time warping distances on the other hand. Furthermore, we compare performance between time series of different lengths. Again, we apply the Kruskal-Wallis rank sum test, which shows that also in this case, there is no statistical difference in performance (AUC: p-value = 0.9534, lift:p-value = 0.7246, EMP: p-value = 0.6264). The same holds for the Friedman test. This means that we only need the four last time steps of a time series in order to predict a change in behavior. This approach results in a minimum loss of data without sacrificing any predictive performance. To conclude these comparisons, in what follows, we will focus on the dynamic time warping distances, even though they are slower, since they give more distinction among the

NN classifiers and on time series of length four, since they require less data and are faster and easier to compute.

Table 7: Avereage ranks of the six methods, as measured by the three measures. The best value for each performance measure is shown in boldface.

Method	AUC	lift	EMP
SF_S	**1.00**	**1.00**	**2.33**
SF_E	2.33	2.17	2.67
SF_{PB}	2.67	3.33	3.00
SF_{MV}	4.67	3.50	2.67
1-NN	5.67	5.50	4.83
5-NN	4.67	5.50	5.50

We now compare the different methods. Table 7 shows the average rank of each method measured by the three performance measures which show consistent results. Clearly, the original similarity forests method SF_S using the best feature consistently performs best. The next best method is the ensemble method SF_E. The worst performing methods are the nearest neighbor classifiers. For both the SF_S and 1-NN, the single feature that outperformed was either OoN neighbors or OoN calls, see Table 2. This is an interesting result for two different reasons. Firstly, the best predictor of churn is how many OoN neighbors one has or how much time one spends calling people that are out of network. Secondly, to achieve optimal performance only a single feature is required.

6.2. Dynamic versus static churn prediction

Table 8: Performance of traditional models and the best similarity forests methods.

Measure		AUC			Lift			EMP	
Dataset	1	2	3	1	2	3	1	2	3
SF_S	0.87	0.63	0.66	6.05	2.15	1.80	0.29	0.10	0.00
SF_E	0.79	0.62	0.65	4.50	1.92	2.04	0.26	0.07	0.00
Logistic regression	0.86	0.65	0.67	6.06	2.21	2.74	0.30	0.49	0.01
Random forests	0.85	0.65	0.67	5.92	2.14	2.67	0.28	0.44	0.01

In addition to comparing the proposed methods to the time series classification benchmark, we compare their performance to the more traditional churn prediction models. For churn prediction in telco, binary classification models can be built using network features extracted from call networks to predict churn with good performance (Óskarsdóttir et al., 2017). In this case, we aggregated call details of four weeks to build a training network and extracted all eight features in Table 2. These were then used to build a model to predict churn in the following four week time frame, and in addition, in each of those four weeks. We adopted two commonly used classification methods, namely logistic regression and random forests, as they offer a trade-off between complexity and predictive capability from opposite angles Verbraken et al. (2014). We experimented with other classification methods, such as extreme gradient boosting and support vector machines, but none of them demonstrated superior performance. To maintain a balance between the new and the traditional techniques, we decided to only include logistic regression and random forests. Both have shown to perform well when predicting churn in telco as well as being popular in both literature and industry (Verbraken et al., 2013). The models were evaluated out-of-time, by applying them to features extracted from the call network of the following month and predict churn in the corresponding periods.

The results for the four-week prediction time frame can be seen in Table 8 together with the two best similarity forests methods from Table 7. Clearly, both classifiers, logistic regression and random forests, perform as well as the single similarity forests method SF_S, and sometimes even better. Note however that these models make predictions with eight variables whereas the best similarity forests method depends on only one.

As mentioned before, by construction, the predictive performance of the similarity forests can be evaluated for each week in the four week prediction time frame. We compare these by-week performances to the corresponding predictions of the logistic regression and random forests models. Figures 3, 4 and 5 show the performance measured in AUC, lift and EMP for the three datasets.

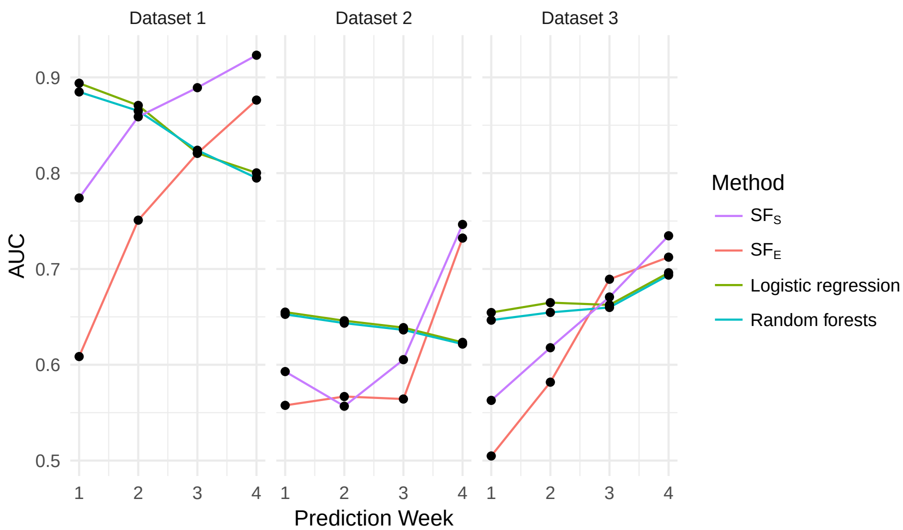

Figure 3: Performance by week (AUC)

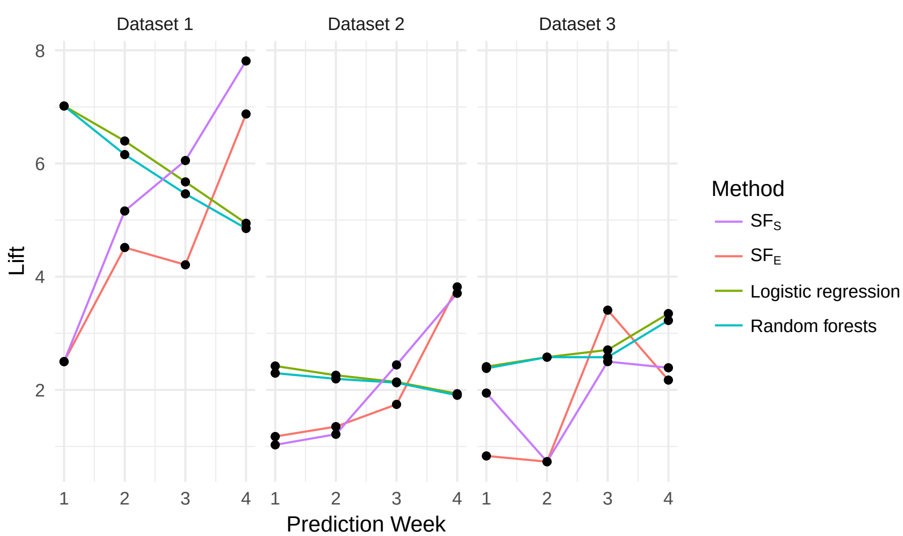

Figure 4: Performance by week (lift)

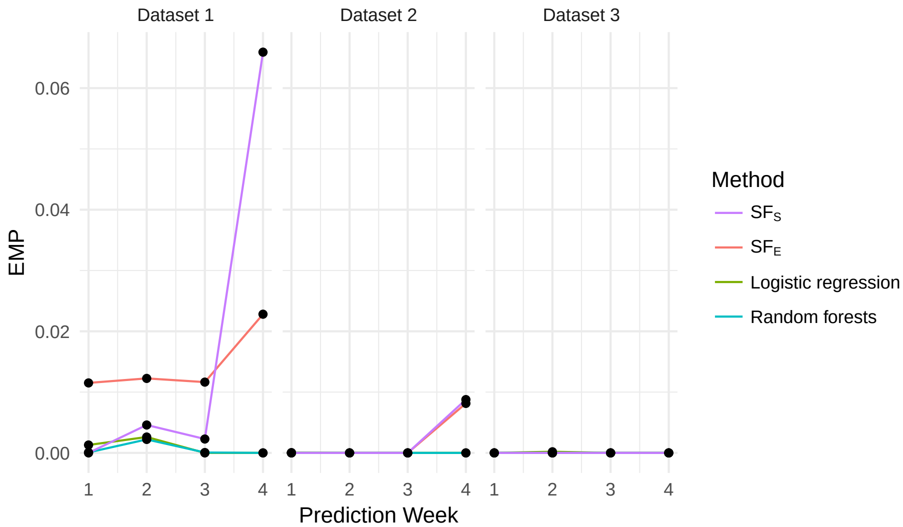

Figure 5: Performance by week (EMP)

From these figures we make the following observations. The performance of the traditional methods, logistic regression and random forests does not change much from week to week, although it does go slightly down as predictions are made further in the future. Contrastively, the performance of the similarity forests methods changes substantially from the first week to the last week, with the highest performance in the last week. In the case of the dataset 1, AUC values for the single method increase from 0.79 to 0.92 and the lift values from 3.1 to over 8. There is also great increase in the performance values of the dataset 2, with AUC changing from 0.6 to 0.74 and lift from 0.9 to 3.8. The change is least drastic for dataset 3, with AUC changing from 0.68 to 0.74 and lift going from 1.7 in week one to 2.4 in week four. We note that the traditional methods show better performance in terms of lift for dataset 3. The results show that the similarity forests methods are better at predicting churn further in the future, which is beneficial for telco that need to design retention campaigns and take action in a timely manner. Since the similarity forests methods depend on the customer behavior, they are better at detecting intention to churn early on. The static methods, which depend on a single value for each feature are better at just-in-time predictions.

7. Conclusion

7.1. Main contributions

In short, the goal of this paper is to predict customer churn in telco using time series that represent the dynamics of customer behavior. Our main contributions consist of two methodological insights, which can be useful for other domains and applications, and two application-specific observations, which are relevant for the field of churn prediction in particular.

As our first methodological contribution, we present a novel way of extracting multivariate time series data from call detail records, thus integrating the dynamic aspect of networks into churn prediction. This feature extraction approach can also be used in other classification tasks and domains in order to capture relevant changes in salient variables. As a next step, the resulting time series can then be combined with any time series classification technique. While this paper does not provide a comprehensive overview or benchmark of existing time series classification methodologies, the general approach of using this research field to represent dynamic behavior is interesting to explore in previously static classification applications. Secondly, we propose three distinct ways in which the similarity forests method can be adapted to handle multivariate similarities. All of these similarity forests methods perform substantially better than the time series classification benchmark, i.e. 1-NN DTW. This is even the case when using the Euclidean distance measure, which makes the technique a lot faster than the methods that are based on dynamic time warping. This result proves that the three novel similarity forests techniques are a valid option for multivariate time series classification. However, we should note that not all of the presented approaches proved to be successful for our application, although they might still prove to be effective in other settings. In particular, the multivariate distance version SF_{MV}, which depends on computing multivariate time series distances, performed only slightly better than a random model.

Nonetheless, this is also an interesting result, as it shows that, at least in this case, it is not enough to be able to only compute similarities between objects as is claimed for the original similarity forests method (Sathe & Aggarwal, 2017).

In terms of the specific application of this paper, our findings indicate that while the similarity forests do not outperform the traditional churn prediction models, i.e. logistic regression and random forests, some of them are able to perform equally on all three datasets. Interestingly, the best performing classification method is the original similarity forests method with a single feature: the duration of calls to out-of-network neighbors. This feature is the weekly time series that only takes into account the customer's contacts in other telcos. The similarity forest with only one feature is therefore able to achieve similar results to the traditional static techniques that contain all eight extracted network features. This conclusion leads to a model that is less computationally heavy and easier to interpret. Furthermore, it is much easier to define which customers are out of network than which customers are churners, as this can be perceived as an in-between state. However, the traditional methods are better at detecting churn that happens shortly after the historic time frame in which the features are extracted, while the similarity forests approach has higher performance for churn that occurs later in the prediction time frame. This leads to our final contribution, which indicates that, the similarity forest approach is better at detecting churn early, which is crucial for the business reality of striving to design and deploy successful retention campaigns. Overall, this paper has shown that the recent behavior amongst churners is similar and distinguishable from the behavior of non-churners.

7.2. Limitations and future work

Our work is not without limitations and offers many possibilities for future research. So far, we have only applied the similarity forests to time series from one domain and for one particular task, i.e. customer behavior and churn prediction respectively. It would be interesting to apply the method to time series from other fields, with completely different applications.

Furthermore, the networks that we derive from the CDR data provide an approximation of the customers' actual social network. The structure of a social network does not evolve drastically over time, especially not from week to week. However, the features that we extract from the networks are only representative of a customer's calling behavior to different types of people, which can change quickly over time, as our results show. Additionally, by using weekly networks, we manage to accumulate enough information from the churn neighbors, which are very few with a monthly churn rate of only 1% to 5%. Daily networks, however, would increase the risk of not capturing enough influence of neighboring churners.

Motivations to churn are many and varied, and we are not able to address all of them in this paper. The main reason for this shortcoming is that we solely base our analysis on individual calling behavior, without taking into account customer characteristics or explicit social influence. In order to accommodate other types of churn as well, we would have to incorporate other customer features, such as socio-demographics and payment history. It would be interesting to use our approach to record changes in these types of variables as well. Furthermore, it is indicated in the literature (Verbeke et al., 2014) and commonly believed in the industry, that customers that churn because of 'social contagion' are harder to retain. In a future study, our approach could be applied to identify the source of this social contagion, e.g. by extracting different types of features that represent the structure of the social network and the individual's position within it.

We have shown that our approach performs well when predicting further into the future, but only for certain types of churn. The combination of our early churn detection technique using dynamic behavior and traditional classification techniques based on customer data, would result in a holistic approach to the churn prediction problem that captures both short- and long-term churn for different reasons. Thus, this approach would allow organizations to retain the most customers and maximize the return on their marketing investment.

Acknowledgments

This work was supported by the Coca-Cola Company and a large Belgian bank.

References

Aiello, L. M., & Barbieri, N. (2017). Evolution of ego-networks in social media with link recommendations. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (pp. 111–120). ACM.

Aral, S., Muchnik, L., & Sundararajan, A. (2009). Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences*, *106*, 21544–21549. URL: `http://www.pnas.org/content/106/51/21544.abstract`. doi:10.1073/pnas.0908800106. `arXiv:http://www.pnas.org/content/106/51/21544.full.pdf`.

Backiel, A., Baesens, B., & Claeskens, G. (2016). Predicting time-to-churn of prepaid mobile telephone customers using social network analysis. *Journal of the Operational Research Society*, *67*, 0.

Bagnall, A., Lines, J., Bostrom, A., Large, J., & Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, *31*, 606–660.

Batal, I., Sacchi, L., Bellazzi, R., & Hauskrecht, M. (2009). Multivariate time series classification with temporal abstractions. In *FLAIRS Conference*.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Chen, Z.-Y., Fan, Z.-P., & Sun, M. (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of operational research*, *223*, 461–472.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, .

Esmael, B., Arnaout, A., Fruhwirth, R. K., & Thonhauser, G. (2012). Multivariate time series classification by combining trend-based and value-based approximations. In B. Murgante, O. Gervasi, S. Misra, N. Nedjah, A. M. A. C. Rocha, D. Taniar, & B. O. Apduhan (Eds.), *Computational Science and Its Applications – ICCSA 2012: 12th International Conference, Salvador de Bahia, Brazil, June 18-21, 2012, Proceedings, Part IV* (pp. 392–403). Berlin, Heidelberg: Springer Berlin Heidelberg. URL: https://doi.org/10.1007/978-3-642-31128-4_29. doi:10.1007/978-3-642-31128-4_29.

Garcia, S., & Herrera, F. (2008). An extension on"statistical comparisons of classifiers over multiple data sets"for all pairwise comparisons. *Journal of Machine Learning Research*, *9*, 2677–2694.

Górecki, T., & Łuczak, M. (2015). Multivariate time series classification with parametric derivative dynamic time warping. *Expert Systems with Applications*, *42*, 2305 – 2312. URL: http://www.sciencedirect.com/science/article/pii/S0957417414006927. doi:https://doi.org/10.1016/j.eswa.2014.11.007.

He, G., Duan, Y., Peng, R., Jing, X., Qian, T., & Wang, L. (2015). Early classification on multivariate time series. *Neurocomputing*, *149*, 777 – 787. URL: http://www.sciencedirect.com/science/article/pii/S092523121401008X. doi:https://doi.org/10.1016/j.neucom.2014.07.056.

Joneydi, S., Khansari, M., & Kaveh, A. (2017). An opportunistic network approach towards disease spreading. In *International Workshop on Complex Networks and their Applications* (pp. 314–325). Springer.

Kadous, M. W., & Sammut, C. (2005). Classification of multivariate time series and structured data using constructive induction. *Machine Learn-*

ing, *58*, 179–216. URL: https://doi.org/10.1007/s10994-005-5826-5. doi:10.1007/s10994-005-5826-5.

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

Kossinets, G., & Watts, D. J. (2006). Empirical analysis of an evolving social network. *science*, *311*, 88–90.

Lin, Y.-R., Chi, Y., Zhu, S., Sundaram, H., & Tseng, B. L. (2008). Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proceedings of the 17th international conference on World Wide Web* (pp. 685–694). ACM.

Naboulsi, D., Fiore, M., Ribot, S., & Stanica, R. (2016). Large-scale mobile traffic analysis: a survey. *IEEE Communications Surveys & Tutorials*, *18*, 124–161.

Navarro, H., Miritello, G., Canales, A., & Moro, E. (2017). Temporal patterns behind the strength of persistent ties. *EPJ Data Science*, *6*, 31.

Orsenigo, C., & Vercellis, C. (2010). Combining discrete svm and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, *43*, 3787 – 3794. URL: http://www.sciencedirect.com/science/article/pii/S0031320310002694. doi:https://doi.org/10.1016/j.patcog.2010.06.005.

Óskarsdóttir, M., Bravo, C., Verbeke, W., Sarraute, C., Baesens, B., & Vanthienen, J. (2016). A comparative study of social network classifiers for predicting churn in the telecommunication industry. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on* (pp. 1151–1158). IEEE.

Óskarsdóttir, M., Bravo, C., Verbeke, W., Sarraute, C., Baesens, B., & Vanthienen, J. (2017). Social network analytics for churn prediction in telco:

Model building, evaluation and network architecture. *Expert Systems with Applications*, *85*, 204–220.

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*, 63–75.

Ranshous, S., Chaudhary, M., & Samatova, N. F. (2017). Efficient outlier detection in hyperedge streams using minhash and locality-sensitive hashing. In *International Workshop on Complex Networks and their Applications* (pp. 105–116). Springer.

Rossetti, G., Pappalardo, L., Pedreschi, D., & Giannotti, F. (2017). Tiles: an online algorithm for community discovery in dynamic social networks. *Machine Learning*, *106*, 1213–1241. URL: `https://doi.org/10.1007/s10994-016-5582-8`. doi:`10.1007/s10994-016-5582-8`.

Rouveirol, C., Pensa, R. G., & Kanawati, R. (2017). Introduction to the special issue on dynamic networks and knowledge discovery. *Machine Learning*, *106*, 1131–1132. URL: `https://doi.org/10.1007/s10994-017-5655-3`. doi:`10.1007/s10994-017-5655-3`.

Santoro, N., Quattrociocchi, W., Flocchini, P., Casteigts, A., & Amblard, F. (2011). Time-varying graphs and social network analysis: Temporal indicators and metrics. In *3rd AISB Social Networks and Multiagent Systems Symposium (SNAMAS)* (pp. 32–38).

Sathe, S., & Aggarwal, C. C. (2017). Similarity forests. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 395–403). ACM.

Shokoohi-Yekta, M., Hu, B., Jin, H., Wang, J., & Keogh, E. (2017). Generalizing dtw to the multi-dimensional case requires an adaptive approach. *Data Mining and Knowledge Discovery*, *31*, 1–31.

Simmhan, Y., Wickramaarachchi, C., Kumbhare, A., Frincu, M., Nagarkar, S., Ravi, S., Raghavendra, C., & Prasanna, V. (2014). Scalable analytics over distributed time-series graphs using goffish. *arXiv preprint arXiv:1406.5975*, .

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

Verbeke, W., Martens, D., Mues, C., & Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, *38*, 2354–2364.

Verbraken, T., Bravo, C., Weber, R., & Baesens, B. (2014). Development and application of consumer credit scoring models using profit-based classification measures. *European Journal of Operational Research*, *238*, 505–513.

Verbraken, T., Verbeke, W., & Baesens, B. (2013). A novel profit maximizing metric for measuring classification performance of customer churn prediction models. *IEEE Transactions on Knowledge and Data Engineering*, *25*, 961–973.

Wang, L., Wang, Z., & Liu, S. (2016). An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm. *Expert Systems with Applications*, *43*, 237 – 249. URL: http://www.sciencedirect.com/science/article/pii/S0957417415006120. doi:https://doi.org/10.1016/j.eswa.2015.08.055.

Weng, X., & Shen, J. (2008). Classification of multivariate time series using locality preserving projections. *Knowledge-Based Systems*, *21*, 581 – 587. URL: http://www.sciencedirect.com/science/article/pii/S0950705108000634. doi:https://doi.org/10.1016/j.knosys.2008.03.027.

Wölbitsch, M., Walk, S., & Helic, D. (2017). Modeling peer influence in time-varying networks. In *International Workshop on Complex Networks and their Applications* (pp. 353–364). Springer.

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