INFORMATICA, 2022, Vol. 33, No. 2, 247–277
© 2022 *Vilnius University*
DOI: https://doi.org/10.15388/22-INFOR484

The Impact of Churn Labelling Rules on Churn Prediction in Telecommunications

Andrej BUGAJEV*, Rima KRIAUZIENĖ, Olegas VASILECAS, Viktoras CHADYŠAS
Vilnius Gediminas Technical University, Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
e-mail: zvex77777@gmail.com, rima.kriauziene@gmail.com, ovasilecas@gmail.com, viktoras.chadysas@vilniustech.lt

Received: February 2022; accepted: May 2022

Abstract. One of the biggest difficulties in telecommunication industry is to retain the customers and prevent the churn. In this article, we overview the most recent researches related to churn detection for telecommunication companies. The selected machine learning methods are applied to the publicly available datasets, partially reproducing the results of other authors and then it is applied to the private Moremins company dataset. Next, we extend the analysis to cover the exiting research gaps: the differences of churn definitions are analysed, it is shown that the accuracy in other researches is better due to some false assumptions, i.e. labelling rules derived from definition lead to very good classification accuracy, however, it does not imply the usefulness for such churn detection in the context of further customer retention. The main outcome of the research is the detailed analysis of the impact of the differences in churn definitions to a final result, it was shown that the impact of labelling rules derived from definitions can be large. The data in this study consist of call detail records (CDRs) and other user aggregated daily data, 11000 user entries over 275 days of data was analysed. 6 different classification methods were applied, all of them giving similar results, one of the best results was achieved using Gradient Boosting Classifier with accuracy rate 0.832, F-measure 0.646, recall 0.769.

Key words: churn prediction, churn definition, telecom, machine learning, binary classification, customer classification, imbalanced learning, RFM.

1. Introduction

Customer churn analysis plays an important role in all business sectors where customers have a choice between different service providers. A well known fact is that often the costs of existing customers retention are much lower than the costs of attracting new customers, especially it is true for telecommunication companies (Ullah *et al.*, 2019). Thus, in a long-term perspective customer churn problem must be addressed more actively than proper advertisements of company services. There are numerous researches on this topic, however, there is a lack of attention paid to some assumptions such as suitability of dif-

[*] Corresponding author.

ferent churn definitions. Also, most of the researches do not analyse temporal data, i.e. instead of using call detail records (CDRs) to produce some features and feed the machine learning methods with it directly (Alboukaey *et al.*, 2020) – this limits the analysis to the scope of classification problem alone, however, the success of classification highly depends on the actual knowledge existing in the data.

In the telecommunication industry acquiring a new subscriber costs 16 times more than retaining an existing one (Ullah *et al.*, 2019). The world wide growth of competition in the telecommunication industry along with the maturation of the service market have turned customer churn management into a main challenge within these industries.

Churn prediction problem usually is formalized as a classification problem. More specifically, for a chosen moment t, using the last data before that moment, the method must decide whether the customer will churn after moment t or not, i.e. whether he must be classified as churner. In other words, the binary classification must be performed in order to split customers into 2 classes: churner and non-churner. In other words, such classification means the prediction telling that the customer will not return to the usage of the services from the chosen moment t, however, it can be generalized by requiring to predict churn before the moment of the last customer activity, introducing a time window relative to the last activity in which the churn must be predicted.

The churned customers usually form a minority – the amount of objects we want to detect is small, creating an additional challenge for a proper measurement of classification results and tuning the methods' parameters. Such data for classification are called imbalanced. As it was pointed out by Chawla (2009) the churn classification data are often imbalanced, i.e. it occurs if objects from one of its categories are 10% or less compared to some other ones. Amin *et al.* (2016) consider churn detection as a classification problem and focus on the class imbalance problem for the churn detection. Authors used on synthetic dataset and the source of other publicly available datasets were not available, thus, the practical application of results is questionable. This can be seen in other articles as well (Keramati and Ardabili, 2011; Keramati *et al.*, 2014). Thus, the data balancing is important for churn classification, however, in many researches this problem was not addressed.

In most of the researches the churn problem in telecommunications comes down to a classification problem, thus, the accuracy of classification methods defines the wellness of final solution. However, the assumptions used for technical formulation of churn classification problem are often very inaccurate and might lead to bigger solution errors than the classification problem solution itself. One of the main such assumption is the simplified churn definition used for data labelling. This research is dedicated to analyse how does the churn definition (directly affecting labelling rules) affect the churn prediction.

The analysis of the results would let us find out how does the prediction accuracy depend on labelling rules due to different churn definitions. Some of the research questions that will follow from the research are:

- **RQ1:** Which labelling rules are sufficient to achieve reasonably small differences in prediction errors comparing to a full churn definition?

- **RQ2:** Which classification methods perform best with different churn definitions?
- **RQ3:** Which classification methods suffer the most from the simplification of churn definition?

1.1. *The Related Work*

There is a wide variety of methods and their combinations were applied to a telecommunication churn prediction. Some of them will be discussed below.

The extensive comparison of different classifiers was done in the research by Adhikary and Gupta (2020) where authors analysed and compared the performance of over 100 classifiers in churn prediction of a telecom company. However, it is important to mention that in the aforementioned research the tuning of different methods was not analysed nor the set hyperparameters were provided – the details were defined by the selected software automatically. Thus, the result of highest accuracy which was given by the Regularized Random Forest classifier is unreliable.

Bose and Chen (2009) have utilized the clustering techniques to improve the decision tree-based churn prediction – clustering was combined with decision trees in such a way that the unsupervised learning technique aided the performance of a supervised learning technique for the classification task of churn prediction.

The other study by Vafeiadis *et al.* (2015) compares different standard machine learning techniques, the tuning of the hyperparameters is performed, the accuracy level 97% was achieved on the publicly available test data. Data are artificially based on claims similar to the real world (Vafeiadis *et al.*, 2015). Note that the results are highly dataset-dependent, we suppose that in this case the high performance is due to dataset specificity.

Some studies show the possibility to combine multiple methods, for example, De Caigny *et al.* (2018) develop a new hybrid classification algorithm that uses a combination of decision trees and logistic regression and that is developed to reduce the weaknesses of DT and LR while maintaining their strengths.

Some research is dedicated to alternative classification methods, such as fuzzy classification methods (Azeem *et al.*, 2017). According to the authors, the achieved level of AUC score of 0.68 is high comparing to other authors, however, achieved values above 0.9 can be found in literature (Ahmad *et al.*, 2019). It is important to note that the results might strongly depend on the dataset specificity, the direct comparison between the results of different researchers is very limited, since different researchers use different datasets. It is important to note that (Azeem *et al.*, 2017) ignore the hyperparameter tuning skipping the opportunity to get better results for every method, leading to incomplete comparison between different methods.

The quality of the results might be strongly conditioned by the data preparation algorithms, according to the research by Coussement *et al.* (2017), the data-preparation technique can strongly affect churn prediction performance. Authors show the improvements of up to 14.5% in the area under the receiving operating characteristics curve (AUC) and 34% in the top decile lift.

In literature, some research that is based on very specific data which is usually not available can also be found. For example, in research by Ahmad *et al.* (2019) the outstanding 70 terabyte dataset from Syrian company SyriaTel was analysed using Hadoop Distributed File System and Spark engine. According to the authors, the extensive expansion of traditional attributes by some data that use a lot of memory has let them improve the results that otherwise were very poor. However, it should be noted that a lot of data had details about customers. The European Union (EU) regulations on data privacy might not let perform similar analysis in EU countries, since a lot of private data in mentioned research was used, such as exact information of migration between companies for each person. Another note is that according to the information specificity, the matrix with the data was very sparse, thus, the storage of these data could be greatly improved by using compressed matrices. Adwan *et al.* (2014) analysed the customer movement from one provider to another. However, nowadays, under EU data privacy regulations the implementation of such research would be hard or nearly impossible. Thus, methods in the aforementioned research are not applicable in practice.

In context of the final goal for business, the reason for churn prediction is the possible customer retention; there are researches in which the churn prediction is accompanied by its further application. Ullah *et al.* (2019) presented the investigation of the existing techniques in machine learning and data mining and proposed a model for customer churn predictions, to identify churning factors and to provide retention strategies. In study by Ahn *et al.* (2020), there are main concepts for churn loss evaluation presented: Customer Acquisition Cost (CAC), Customer Lifetime Value (CLV); also the feature engineering was discussed. There are successful attempts to include the survival analysis methods into the churn prediction, interpreting the churn retention as a subject survival (Routh *et al.*, 2021), the risks raised due to competing events are used in modelling by a random survival forest.

Various machine learning methods are used to determine the churn, as can be seen from the overview in the Table 1. The results obtained in the articles show that the results of the several different methods are good enough. Thus, the efficiency of churn prediction for the actual customer retention might not be bound by the actual (technical) classification performance – the main source of inaccuracy might arise from some assumptions, such as sufficient conditions of churn definition or the assumptions used for the artificial data generation. This is the problem which we are going to address.

In general, there are few most commonly used methods that give a reasonably good results in many researches, moreover, often there were no significant differences between these and some other methods that sometimes perform better in other researches. In our case, several commonly used methods were selected and studied, most of them are tree-based methods.

For example, Adhikary and Gupta (2020) have analysed as many as 100 methods for determining the churn. However, as we have already mentioned, the hyperparameters tuning was ignored. It can be assumed that the hyperparameter optimization was not analysed in the aforementioned research. Thus, it can be considered as a drawback of the considered study.

Another important note can be done based on research by Alboukaey *et al.* (2020) – the RFM (Recency, Frequency, Monetary) features enabled even such method as Random

Table 1
Literature overview.

Authors	Methods	Atributes
Bose and Chen (2009)	Two-stage hybrid models: the first stage – unsupervised clustering technique (KM, KMD, FCM, and SOM), the second stage – C5.0 tree with boosting	**Revenue contribution**: mean monthly revenue (charge amount); percentage change in monthly revenue versus previous three months average; total revenue, billing adjusted total revenue over the life of the customer, etc. **Service usage**: percentage change in monthly minutes of use versus previous three months average, mean number of attempted voice calls placed, mean number of received voice calls, etc.
Keramati and Ardabili (2011)	Binomial logistic regression	Number of failed calls, subscription length, customer complaints, amount of charge, length of all calls, second of use, number of calls, frequency of SMS, frequency of use, number of distinct calls, type of service, group age , status, churn.
Keramati *et al.* (2014)	Decision Tree (DT), Artificial Neural Networks (ANN), K-Nearest Neighbours (KNN), Support Vector Machine's (SVM)	Call failure (CF), number of complains (Co), subscription length (SL), charge amount (CA), seconds of use (SU), frequency of use (FU), frequency of SMS (FS), distinct calls number (DCN), age group (AG), type of service (TS), status (St), churn (Ch).
Vafeiadis *et al.* (2015)	Back-Propagation algorithm (BPN) (case of the ANN classifier) , Support Vector Machine's (SVM) and Decision Tree C5.0 (DT) with and without boosting, Naïve Bayes (NB), Logistic regression (LR)	Number of months active user, total charge of evening calls, area code, total minutes of night calls, international plan, total number of night calls, voice mail plan, total charge of night calls, number of voice-mail messages, total minutes of international calls and etc.
Amin *et al.* (2016)	Oversampling techniques (SMOTE, ADASYN, MTDF, ICOTE, MWMOTE and TRkNN) and they compare the performance of four rules-generation algorithms (Exhaustive Algorithm, Genetic Algorithm, Covering Algorithm and RSES LEM2 Algorithm)	4 data sets
Azeem *et al.* (2017)	Fuzzy classifiers: FuzzyNN, VQNN, OWANN and FuzzyRoughNN	Days since last recharge, voice bucket revenue, active days since last call, sms bucket revenue, total revenue voice, revenue on net, active days since recharge, total revenue, sms charged outgoing count, GPRS bucket revenue, revenue sms, crbt revenue, charged off net, minute of use off-net, balance average daily, balance last recharge, off net outgoing minute of use, charged off net minute of use, free minute of use, free on net minute of use, free sms, revenue fix, active days recharge, recharge count, recharge value, last recharge value, act days minute of call, promo opt in, loan count, active days since loan, inactive days calls, inactive days sms, inactive days data.
Coussement *et al.* (2017)	LOGIT-DPT, Bagged CART, Bayesian network, Decision Tree, Neural netwok, Naïve Bayes, Random Forest, Support Vector Machine's, Stochastic gradient boosting	156 categorical and 800 continuous variables

(*continued on next page*)

Table 1
(*continued*)

Authors	Methods	Atributes
De Caigny *et al.* (2018)	Decision tree (DT), Logistic model tree (LMT), Logistic regression (LR), Random forests (RF)	Mean number of call waiting calls, change in minutes of use, dummy if change in minutes of use is imputed, low credit rating, mean number of customer care calls, mean number of director assisted calls, number of days of the current equipment, mean number of inbound voice calls, models issued, mean monthly minutes of use, mean number of in and out off-peak voice call, mean number of outbound voice calls, handsets issued, mean total recurring charge, number of calls previously made to the retention team, mean monthly revenue, missing data on handset price, handset is web capable
Ullah *et al.* (2019)	Random forest vs other machine learning techniques	2 data sets
Ahmad *et al.* (2019)	Decision Tree (DT), Random Forest (RF), Gradient Boosted Machine Tree (GBM) and Extreme Gradient Boosting (XGBOOST)	10 000 variables
Adhikary and Gupta (2020)	100 classifiers	57 attributes

Forests to perform surprisingly well, however, this method was not among the best ones in other researches. The results without RFM features were much worse in aforementioned research. Thus, in this article we compute RFM features as an important part of dataset for all experiments as well, a detailed description is provided in Section 3.1.

1.2. *Some of Requirements for a Proper Churn Prediction*

A big amount of flaws can be noted in other researches, which can be seen as research gaps – here we will try to categorize them by providing the appropriate research requirements. From the overview of the results of other researchers, we highlight these requirements which should be fulfilled:

1. the temporal data is used when the features are derived from actual CDR and/or payment data keeping the information about behaviour dynamics. Data aggregation over the whole period removes the information about the temporal changes in customers' behaviour leading to the loss of the discriminative ability of the classification methods. Note that the temporal data was not utilized in this research, the most of features lack the information about behaviour dynamics.
2. the dataset must be not synthetic and the labelling rules and data filtering should be defined, otherwise the usefulness for practical purposes is questionable. The big amount of publicly available datasets are synthetic, the sources of datasets are not properly described, thus, the practical application of results is questionable.
3. hyperparameter tuning is performed, since methods with default parameter values might perform far from perfect in many cases. The research ignores the hyperparam-

Table 2
Hyperparameter ranges of classification methods. Here + or − refer to requirement fulfillment or non-fulfillment, accordingly, +/− – means the partial fulfillment of the requirement, the unknown state of the fulfillment of provided requirements is referred as nan.

Article	Temporal data was used	Data is non-synthetic	Hyperparameters were tuned	Data balancing was performed
Adhikary and Gupta (2020)	−	+	−	+
Ahmad *et al.* (2019)	+	+	−	+
Coussement *et al.* (2017)	nan	+	+	−
De Caigny *et al.* (2018)	+/−	+	+	−
Ullah *et al.* (2019)	+/−	+	−	−
Amin *et al.* (2016)	+/−	+/−	−	−

eter tuning skipping the opportunity to get better results for every method, leading to incomplete comparison between different methods.

4. data balancing is performed as an important part of minority churn class detection. Churn labelling leads to imbalanced data, thus, proper balancing techniques are required to utilize the full power of machine learning methods.

From Table 2 the lack of the proper attention to the aspects mentioned above can be seen. However, it must be noted that the good results can still be achieved without fulfilling all of these requirements. For example, some of the methods are less prone to the data imbalance problem and might perform relatively well even without balancing, especially if a proper measurement was used (for example, if it is sensitive to the Recall metric such as F-measure). However, according to other authors, these requirements are recommended to be fulfilled and in current research one of the goals is to fulfill these requirements.

1.3. *Churn and Partial Churn Definitions*

Often a churner in the mobile telecommunications is defined as a customer that stops doing revenue generating events during the next 90 days (Alboukaey *et al.*, 2020), while he was active during the observation period. Churn directly affects the profit of the company, thus a customer retention is an important part of company strategy. However, to be able to apply any customer retention, it is necessary to detect the churn event on its early stage, i.e. during the first days of this 90 day period or even before this period has started. In order to estimate the consequences of the churn, it is important to estimate the loss of the profit due to some specific churn event, i.e. churners might have different weights depending on the amount of profit being generated by these customers. However, it is important to note that in most cases the one month period is sufficient to identify the churn case, i.e. the inactivity during 30 days is followed by another 60 day inactivity in most cases. Thus, the definitions of a partial churner as a customer who doesn't use the services for 30 days might be used in practice. On the one hand, the shorter period lets us label more recent cases, thus it lets a model achieve a faster reaction to the changes in the behavioural patterns. On the other hand, the quality of the possible prediction using shorter time interval might decrease due to possible errors in labels of the churners, i.e. a small portion of partial churners

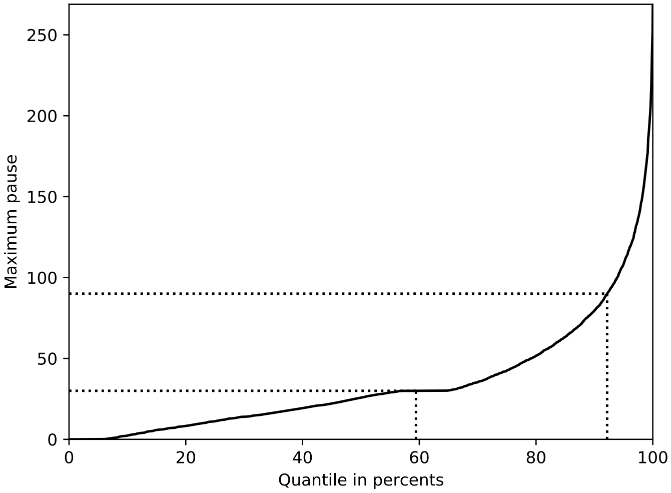

Fig. 1. The values of quantiles of maximum pauses during 275 days time interval according to Moremins data.

does not fulfill the true churner criteria which means a good machine learning approach will produce a portion of false positives and potentially these false positives could even increase false negatives as well due to the shifts in the model focus while learning from some false data.

The above said implies that it is necessary to find a compromise between relevancy and certainty. This topic has not been investigated before, some authors only mention that in their specific case the churner definition and the partial churner definitions are equivalent in over 80% cases, however, they achieve over 80% accuracy in their predictions using machine learning models, thus it is not clear what is the true accuracy and how does the differences between churner and partial churner affect the final results. On technical level, the aforementioned differences of definitions lead to different labelling rules, thus different training and test data for the classification problem. This is one of the main questions this article is dedicated to.

The main reason behind the churner detection and prediction is the customer retention in order to save the portion of the profit, since in most cases the costs of customer retention are usually much lower than the costs of new user attraction (Barrett, 2003; Lu, 2002). However, the success of retention highly depends on how fast the churn detection was performed. Some authors focus on early churn detection calling it early prediction (Zhang *et al.*, 2016). Therefore, only analysis of retention effectiveness can lead to clues about analysis of suitability and usefulness of different churner definitions, but this is out of scope of the current research. We will assume that the standard definition is the most commonly used definition of 3 months customer activity absence leading to a benchmark rule for labelling the churner. Any other alternative definitions along with their appropriate labelling rules will be considered as approximations of the standard definition, the precision of such approximations is the main topic of the current research.

In Fig. 1, the values of quantiles for maximum pauses per customer are presented. This lets us evaluate the difference of partial churn and churn, more specifically, every customer with maximum pause between 1 and 3 months would mean at least one false positive label if we substitute the churn by partial churn in some specific time window. If we define a churner to be a customer leaving the services forever, then it can be seen that 30 day length interval would let us correctly label up to 59.4% of churners, 90 day interval – would be correct for up to 92.2% of cases. It must be noted that these data are derived from the whole interval, however, at some selected moment there will be only a part of these falsely potentially labelled churners. In other words, the provided estimates show the pessimistic lower bound of correctly labelled churners' percentage. Thus, the three month period can be seen as quite robust and reliable comparing to labelling in 30 day period. It can be concluded that for the considered data churn and partial churn definitions show how different they can be, despite some authors (Alboukaey *et al.*, 2020) claiming otherwise based on the distribution of churners' activity during the last thirty days – such analysis can be misleading because the churners form a minority and there might be other customers' minority group that has a similar activity patterns (e.g. with periodic peaks of activity). The provided rough lower bounds do not prove the suitability or unsuitability of the partial churn definition either. In order to compare the performance of different churner definitions, both definitions with according labelling rules can be used for training the classification algorithm, however, for the final measurement the full churner definition can be used to compare the results of different classification algorithms. Also, it is an open question whether we should consider a churner which stopped using the services for 3 months and have returned afterwards, since it can be considered as churn from competitor services.

Summarizing the said above, there is no perfect churner definition itself and appropriate criteria to define a churner might differ depending on many factors. However, the analysis of suitability and usefulness of different churner definitions is out of scope of the current research, thus, as a standard we will use the most commonly used definition of 3 month customer activity absence and use it as a reference labelling rule. For this research we generalize the churner definition by introducing the labelling rule parameter T_c which is equal to the minimal length of the absence time interval which is needed for to proclaim the customer to be a churner.

1.4. *Paper Roadmap*

The remainder of this paper is organized as follows: the datasets used in this research are described in Section 2, Section 3 provides the method-related information for this study, results are provided in Section 4 followed by discussion in Section 5.

2. Dataset Overview

The most of the analysed articles have either missing references to the dataset sources or the links to the sources of different datasets are broken. Thus, the amount of publicly available datasets that were used by other authors is quite limited, here we will overview the main of them.

One of the datasets is data from IBM (2020). It is a fictional telco company that provided home phone and Internet services to 7043 customers. The churn column indicates whether the customer left within the last month or not. Other columns include gender, dependents, monthly charges, and information about the types of services each customer has. One of the advantages of these data is column “churn reason”, which we wouldn’t have in real data without additional sophisticated analysis. IBM data have 33 columns and 7043 rows (which is equal to the number of customers).

Amin *et al.* (2016), Ullah *et al.* (2019), Xu *et al.* (2021a) mention a dataset consisting of 3333 instances and 21 attributes, one of which is the churn tag. The links in these articles are not working. However, these data can be found in the Kaggle platform. Dataset is available at (2021).

1. state – customer state;
2. total day minutes – total minutes of talk during the day, can be generalized into a value aggregated in other time interval, e.g. total month;
3. total day calls – number of calls in the day, can be generalized into a value aggregated in other time interval, e.g. month;
4. total day charge – call charges during the day;
5. total eve minutes – total minutes of talk last night;
6. total eve calls – number of calls last night;
7. total eve charge – charges for calls last night;
8. total night minutes – night total call minutes;
9. total night calls – total number of calls in the evening;
10. total night charge – total charge for calls at night.

Moremins dataset has been taken from a company Moremins. This company is oriented to the niche of customers that migrate between countries, the services are based on cheap calls between different countries, in some of the countries this company has the status of a mobile operator, in others, it has status of a mobile virtual network operator (MVNO). Moremins dataset is not available to public because of the restriction applied on it from Moremins company, since the license was granted for research purposes only. The data is available to researchers in Moremins company and will be available for others after getting the permission from the company (2021). Dataset contains 275 day usage information, the period covered is from 2020-11-10 to 2021-08-12. The data was derived by aggregating Call Detail Records (CDRs) and payment history information. More than 1200000 phone calls and more than 58000 payment records were examined and characterized. The daily aggregation of these data by users yielded 11100 vectors with values of 426 features of daily behaviour.

The attributes of the prepared dataset for the experiments are given in the Table 3. The daily data are calculated for these parameters:

1. call count,
2. the amount of minutes,
3. payment count,
4. the costs of payments.

Table 3
Structure of the prepared dataset.

Attribute	Values or their range	Data type	Description
X1	0-26751	Numerical	The sum of minutes from all calls through whole period
X2	1-7032	Numerical	The amount of calls through whole period
X3	0-1475	Numerical	The sum of costs of customers payments through whole period
X4	0-255	Numerical	The amount of payments through whole period
X5	0-90	Numerical	The average of minutes from all calls during the day
X6	1-104	Numerical	Activity provided by company
X7	0-73	Numerical	Usefulness provided by company
X8	0-47	Numerical	Involvement provided by company
X9	0-266	Numerical	The maximum pause in days of customer activity
X10	0, 1, 2, 3, 4	Categorical	Customers classes provided by company
X11	0-275	Numerical	Duration of activities in days
X12, X13, ..., X18	–	Numerical	The amounts of calls of different types (7 different)
X19, X20, ..., X66	–	Numerical	RFM features
X67, X68, ..., X426	–	Numerical	Daily parameters for the last 90 days

Table 4
Dataset review.

Dataset	Instances	Number of attributes	Attributes
Dataset 1 Amin *et al.* (2016); Ullah *et al.* (2019) (BigML)	3333	21	State; account length; area code; phone number; international plan; voice mail plan; number vmail messages; total day minutes; total day calls; total day charge; total eve minutes; total eve calls; total eve charge; total night minutes; total night calls; total night charge; total intl minutes; total intl calls; total intl charge; customer service calls; churn.
Dataset 2 IBM (2020)	7043	33	LoyaltyID; Customer ID; Senior Citizen; Partner Dependents; Tenure; Phone Service; Multiple Lines; Internet Service; Online Security; Online Backup; Device Protection; Tech Support; Streaming TV; Streaming Movies; Contract; Paperless Billing; Payment Method; Monthly Charges; Total Charges; Churn
Moremins dataset	11100	426	

The aforementioned parameters are calculated for last 90 days of data resulting in 360 features in total.

Dataset overview is summarized in Table 4.

The churn rate in different datasets. In order to evaluate the imbalanced data problem for different datasets, here we provide the percentage of churned customers among all customers:

1. BigML – 14.49%,
2. IBM – 26.54%, which makes it a slightly imbalanced data,
3. Moremins – 20.21%.

As it can be seen, the imbalance in the considered data is not very high, this will be taken into account later in Section 4.2.

3. Methods

3.1. *Feature Extraction*

A proper feature extraction might be the key for the performance of machine learning methods. Here we discuss the features used in the current research.

Raw data. Here we present the most detailed form of data feeding the machine learning methods in this research. For that role we have chosen to aggregate the most important parameters for every day per each customer, according to other researches (De Caigny *et al.*, 2018; Alboukaey *et al.*, 2020), the most significant and usually sufficient information is the one which is directly related to the usage of services (minutes, amount of SMS, etc.) or charges for the services (customer payments), thus, a daily data is used and is described as:

1. The sum of minutes from all calls during the day;
2. The amount of calls during the day;
3. The sum of costs of customer payments during the day;
4. The amount of payments during the day.

Thus, we consider this data to create the basis for all future analysis. Next, the appropriate time window must be defined, i.e. how many days must be considered in order to create a sufficient amount of data.

RFM features. As it was mentioned before, the absence threshold for partial churn is 30 days partial, for such labelling in some other researches Alboukaey *et al.* (2020) RFM-based classification has given one of the best classification results achieving over 90% accuracy.

Compared to usual operators, MVNO clients tend to perform non-regular usage of the services, since clients use services from traditional operators as well. Which means the explicit aggregation of parameters in longer time intervals could give the regularization effect making adding the robustness to the solution.

Recency, Frequency and Monetary (RFM) features are well-known for their suitability for churn modelling (Gupta *et al.*, 2006; Khajvand *et al.*, 2011). These features are the values obtained by aggregation of the considered parameter in the selected time interval in three different ways. Let us assume that we have daily data, i.e. the sums of some parameter for different days, we denote t_1, t_2 to be integers representing the first and last day of investigated interval, then R, F, M features in interval $t = [t_1, t_2]$ are obtained using these formulas:

$$R(t_1, t_2) = \begin{cases} \sum_{ts+1}^{t2} 1, & \text{if } ts < t_2, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$
$$ts = \max\bigl\{i : f_i > 0,\, f_i \in \{f_{t_1}, f_{t_1+1}, \dots, f_{t_2}\}\bigr\},$$

i.e. R (Recency) is equal to the amount of the last days with non-zero values of selected parameter during the interval $[t_1, t_2]$.

$$F(t_1, t_2) = \sum_{t=t_1}^{t_2} \begin{cases} 1, & \text{if } f_t > 0, \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

i.e. F (Frequency) is the amount of days with non-zero values of selected parameter.

$$M(t_1, t_2) = \sum_{t=t_1}^{t_2} f_t, \tag{3}$$

which is a basic sum, in other words, it transforms daily data to some data aggregated in the same way, but for the bigger time interval.

As it was proposed by Alboukaey *et al.* (2020), we calculate RFM features in intervals $t \in [1, 90], t \in [1, 30], t \in [31, 60], t \in [61, 90]$. Which means there are 4 features calculated per each parameter, we use these parameters:

1. call count,
2. the amount of minutes,
3. payment count,
4. the costs of payments.

In order to label the data for some time moment t, some amount of future days must be known. This means that if we use partial churn definition to label the data we can do that only for the moment t that is at least 31 days old (compared to the actual real moment). For the RFM feature usage we need the data aggregation in 5 time windows: 4 for RFM features, and one 30 day window $t \in [91, 180]$ for churn labelling. The RFM features application in the full test procedure context looks like that:

1. RFM features R_1, F_1, M_1 are calculated for $t \in [1, 90]$;
2. labels C_1 are derived from data for $t \in [91, 180]$;
3. RFM features are added to data, the set is split into train and test parts;
4. using R_1, F_1, M_1 the standard 5-folds cross validation technique is applied with a chosen classification method to a training set part, hyperparameter tuning is performed;

5. the final performance estimation is performed by applying the model on the test part of data labels C_1.

As can be seen, for the full analysis of the selected time moment the data from 180 days is needed, the selected time moment to perform the classification is the beginning of 91th day.

Other features. Here we present the list of some simple, yet efficient features suitable for churn prevention, mostly taken from publicly available datasets (Xu *et al.*, 2021a) or based on them:

1. monthly revenue;
2. monthly minutes;
3. state – customer state;
4. payment method;
5. monthly change;
6. percentage change in monthly minutes of use vs. previous three months average;
7. mean monthly revenue over the data collection period;
8. mean number of monthly minutes of use;
9. mean monthly revenue.

3.2. *Classification Algorithm*

Here we provide the sequence of procedures defining the classification algorithm. It must be noted that some standard steps were omitted from discussion since the motivation of their usage is well known and these are done in most of other researches. More specifically we include into the algorithm: data normalization, random oversmapling balancing method, dimensionality reduction method PCA (Principal component analysis). Data normalization is important for variance-based PCA, PCA can increase the performance of some methods such as SVM. Thus, the steps to perform the churn prediction are:

1. RFM and other feature extraction,
2. data labelling according to different churn labelling rules,
3. unsupervised method application: feature vectors' normalization by standard scaler (division by standard deviation), application of PCA (Principal component analysis),
4. the construction of classification method based on these steps: random oversampler (churner entries duplicating), the selected method,
5. execution of Algorithm 1 passing to it data and a method,
6. saving the metrics.

In Algorithm 1 some of functions reflect the functionality of Pandas dataframes and scikit-learn library, more specifically:

- method drop() refers to removing the rows (entries);
- train_test_split randomly splits the data into test and training/validation samples with the given proportion (0.1 to the test part);

Algorithm 1: The algorithm for metrics extraction

```
Input: data, method, grid, labels
Output: metrics, true_metrics
for $l$ in labels do
    D = data.copy()
    d = D entries which are satisfying the churn labelling rule $l$
    D.drop(d); ans = D[$l$]
    d['answer'] = 1
    remove all columns with names [labels] from D
    $X$, $Y$, $X_t$, $Y_t$ = train_test_split(D, ans, test_size = 0.1)
    model = GridSearchCV(method, param_grid = grid).fit(X,Y)
    $Y_a$ = model.predict($X_t$)
    metrics.append(evaluate($Y_a$, ans))
    d = resample(d, n_samples = 0.1*len(d))
    $Y_a$.append(d['answer'])
    ans.append(d[l])
    true_metrics.append(evaluate($Y_a$,ans))
end
```

- GridSearchCV iterates through all possible combinations of the given ranges of parameters;
- resample refers to the random sampling from the given set of entries at the given proportion, i.e. it leaves 0.1 of the initial rows to sustain the constant proportion of removed and total entries between the initial and test data.

Also, the true metrics concept is indirectly present in Algorithm 1, it will be described in details in Section 3.4.

3.3. *Classification Methods*

Taking into account the results obtained by other authors, we notice that the methods based on the decision tree give a good enough result.

- GradientBoostingClassifier (GBC) is a stochastic gradient boosting algorithm, which was proposed by Friedman (2001, 2002). The weak learner of this method is a decision tree. On each iteration, trees are fit on the negative gradient of the loss function evaluated at the previous iteration.
- XGBClassifier (XGBoost) was proposed by Chen and Guestrin (2016). This model grows trees level-wise. It was developed as a method for computation speed and performance. It is state-of-the-art in many articles.
- LGBMClassifier (LGBM) – a gradient boosting model (Ke *et al.*, 2017). This model grows trees leaf-wise. It chooses the leaf which is predicted to give the largest improvement for the loss function.

- RandomForestClassifier (RF) was proposed by Breiman (2001). This method creates a forest of random trees. L. Breiman minimized the generalization error for forests, because the number of trees in the forest increases.
- KNeighborsClassifier (KNN) was first developed by Fix and Hodges (1951). The idea of this method is to assign classes to new data (test data) on the basis of data already classified (learning data).
- SVM – supervised classification method for classification in two groups. SVM was proposed by Cortes and Vapnik (1995). The training data are divided in the high dimension feature space so that the distance between the classes is the greatest. New data (test data) is displayed in the same space. The assignment of test data is based on which side of the gap it is displayed.

3.4. *Performance Metrics*

We find a lack of description of different metrics' importance in other researches, thus, here we will fill this gap by discussing different metrics in the context of the churn problem.

Accuracy. Accuracy is the most common metric to measure the performance of classification method. What it lacks is the sensitivity for imbalanced data, i.e. if, for example, there are 10% of churners in the data, and there will be no single churner classified, the accuracy will still show value 0.90 which is usually a good result in the case with balanced data. For churn prediction the focus must be made on churners, the minority of objects, so the accuracy doesn't show the performance very well.

Recall. Recall or true positive rate (TPR) is calculated as $TPR = TP/(TP + FN)$. If true positives are very important in the context of the research, attention must be paid to recall since it measures that true positive rate among all positives. This metric doesn't take into account false positive though, so it is unsuitable as a standalone metric, i.e. classifying all objects as churners will give you a perfect recall. In the context of churn prediction this is an important metric, i.e. we want to be sure that if a customer is going to leave, then a retention technique should be applied, even if it will lead to a side effect of applying the retention to some of non-churners falsely classified as churners.

Specificity. Specificity or true negative rate (TNR) is calculated as $TNR = TN/(TN+FP)$. it doesn't hold very important information if we want to find churners and apply retention techniques to them, however, if we want to limit the amount of retention applied to non-churners, this characteristic measures that indirectly, since you can derive false positive rate from it $FPR = 1 - TNR$.

Precision. Precision or positive predictive value (PPV) is calculated as $PPV = TP/(TP + FP)$, it shows how well the positive result is determined. Usually this characteristic is important in cases when it is important to avoid false positives. Thus, in the case of churn prediction this metric is not important, however, the indirect usage of it might be still useful, for example, as a criteria to limit the recall – even if we want recall to be as big as possible, we should control the precision and keep it from being unreasonably small.

Balanced accuracy. Balanced accuracy is a compromise between recall and specificity, more specifically, it is an average of the aforementioned two metrics $BA = (TPR + TNR)/2$. It means, the rates for classification of minority and majority have the same weights, making balanced accuracy somewhat more suitable for imbalanced data comparing to accuracy metric. This is based on obvious observation that model can be tuned to have either big recall or specificity, so this characteristic seeks to find a compromise. However, this characteristic in some cases do not represent the situation well, a well-known fact is that arithmetic mean has drawbacks when it is applied to the rate-type characteristics, moreover, this metric is sensitive to majority which usually is not important, thus nowadays this metric is rarely used as primary.

F-measure. F-measure has the same motivation behind it as the balanced accuracy has, but the counterpart of recall is selected to be the precision which is far more suitable to describe the success of minority prediction. Moreover, attention should be paid to the fact that we are talking about rates, so the significance of the changes of absolute values depend on the values themselves, e.g. decrements of 0.1 have a more dramatic effect for value 0.11 than for value 0.9. In such situations a more appropriate way to find a compromise is to use harmonic mean instead of arithmetic average. Thus, F-measure is the harmonic mean of precision and recall $F_1 = 2 \cdot PPV \cdot TPR/(PPV + TPR)$. This is a very popular metric to measure the churn prediction classification in recent researches, therefore, this metric will be the main measurement instrument in the current research.

AUC. The aforementioned metrics are calculated using a finally tuned and fixed classification method, however, often methods give you the estimations of likelihood of some object being belonged to one or other class, the final classification is performed by selecting discrimination threshold which must be stepped over in order to assign the object to some class. This means, that, for example, you can sacrifice the accuracy for a bigger recall if you shift the discrimination threshold for the final decision. In other words, after the learning process is finished, you can create a series of classification algorithms by choosing different thresholds and draw the ROC curve describing the dependence of TPR on FPR, both values will increase from 0 to 1 forming a curve. The area under ROC curve (AUC) is a good way to estimate the overall machine learning technique independently of discrimination threshold. The closer AUC value to 1 – the better, AUC equal 0.5 means the method's performance is similar to random classifier.

True metrics. In the context of the current research there is a need to distinguish between the classification metrics and the actual metrics in terms of the true (full) churn definition. The reason is the partial churn definition and generalized definitions which let the churner be labelled based on the inactivity time interval which is less than 3 months. However, it is done with the intention to predict the churner based on the full churner definition assuming that the differences between aforementioned definitions are small.

For correct formulation of classification problem we must remove the entries for which the answer is already known. It is assumed that the definition itself is changed to an alternative one with other labelling rule, so if the object fulfills this definition the prediction is not needed, the answer is known. Such removal of the entries is an exact representation

Table 5
The labels defining different labelling rules for different churn definitions.

Label	T_c
Churner	90
Churner1	60
Churner2	50
Churner3	40
Churner4	30
Churner5	15

of what is done in other researches, Alboukaey *et al.* (2020) excluded inactive customers for the last 30 days. I.e. in the training data there should be no customers who are inactive for the last $k \leqslant m$ days, where m is the number of days of inactivity that must be passed for the customer to be labelled as churner according to one or another generalized churner definition. However, these dropped entries should be still present in data labelled using full churn definition's labelling rules.

Therefore, to unify the metrics for different churn definitions we introduce the true metrics concept – before the final measurement, the entries, which where excluded due to decreased time interval in labelling rules, must be returned to the dataset, such objects must be labelled as churners.

4. Experimental Results

4.1. *Data Preparation*

As it was mentioned in Sections 1.3 and 3.4, the churn definition, which is used for labelling, affects the dataset. A single parameter can be used to describe different labelling rules for different churn definitions used in the current research – the amount of days of customer absence T_c which was presented already in Section 1.3. In Table 5 the list of label names according to different churn definitions is provided along with the value of the aforementioned parameter T_c.

- Churner – the churner according to the standard 90 day absence definition,
- Churner4 – the label according to a partial churner definition which is often used as an alternative due to many reasons and assumptions,
- Churner1, Churner2, Churner3, Churner4 – the labels used to describe the compromise between the standard (full) and partial churner definitions,
- Churner5 – the label used to investigate the extreme case with a very short churner detection window of 15 days.

Note that in order to label the data, additional 90 days from the future were used, expanding the actual time interval used to create the dataset from 275 to 365 days, last 90 of which were used only for labelling.

Table 6
Hyperparameter ranges of classification methods.

Method	Hyperparameters	Ranges
GBC	*min_samples_leaf*:	[3, 5, 7]
	n_estimators:	[256, 512]
	max_depth:	[2, 3, 5, 7]
	n_iter_no_change:	[5]
	tol:	[0.0001]
XGBoost	*booster*:	[*gbtree*]
	nthread:	[1]
	use_label_encoder:	[*False*]
	max_bin:	[128, 256, 512]
	max_depth:	[2, 3, 5, 7]
	subsample:	[0.5, 1]
	eval_metric:	[‘*logloss*’]
	tree_method:	[*hist*]
LGBM	*boosting_type*:	[‘*gbdt*’]
	n_jobs:	[1]
	n_estimators:	[128, 256, 512]
	max_depth:	[5, 10, 15, 20]
	learning_rate:	[0.1]
	subsample:	[1]
	learning_rate:	[0.05, 0.1, 0.15]
	subsample:	[0.5, 1]
RF	*n_estimators*:	[128, 256, 512]
	min_samples_leaf:	[3, 5, 7]
	max_depth:	[15, 30]
KNeighborsClassifier	*n_neighbors*:	[10, 15, 20, 25]
	algorithm:	[‘*auto*’, ‘*ball_tree*’]
	leaf_size:	[3, 5, 10, 15]
svm.SVC()	*tol*:	[$1e-04$]
	kernel:	[‘*poly*’, ‘*rbf*’, ‘*sigmoid*’]

4.2. *Hyperparameter Values for Different Methods*

The explored hyperparameter ranges in current research are presented in Table 6.

Hyperparameter values for Moremins dataset are presented in the Table 15 and for other datasets in Table 8. The appropriate parameters were selected by trying all possible combinations using the Grid search method, as a comparison criterion F-measure was used.

As it was already presented in Section 2, the imbalance in the considered data is not very high. Thus, metrics for imbalanced data, such as F-measure, should not be blindly considered as the only right way to seek for an optimal solution. It is possible to optimize the F-measure metric at the cost of Accuracy via thresholding techniques or even the usage of special loss functions, however, the optimality of results obtained in that way can be still questionable as there is no single perfect way to measure the performance of the methods. Thus, the approach used in this research is to let methods converge using loss functions which are known to be most suitable for according methods, i.e. default ones, however, to use F-measure to perform the final selection of combination of hyperparameters.

Table 7
Results of different datasets using different methods.

Method	Datasets	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
GBC	BigML	0.925	0.762	0.741	0.784	0.961	0.851
	IBM	0.746	0.637	0.818	0.522	0.719	0.769
XGBoost	BigML	0.922	0.745	0.704	0.792	0.964	0.834
	IBM	0.757	0.642	0.797	0.537	0.743	0.77
LGBM	BigML	0.931	0.768	0.704	0.844	0.975	0.839
	IBM	0.769	0.637	0.745	0.556	0.778	0.761
RF	BigML	0.91	0.737	0.778	0.7	0.936	0.857
	IBM	0.787	0.639	0.693	0.594	0.823	0.758
SVMds	BigML	0.91	0.732	0.759	0.707	0.939	0.849
	IBM	0.74	0.627	0.802	0.515	0.717	0.76
KNeighborsClassifier	BigML	0.835	0.604	0.778	0.494	0.846	0.812
	IBM	0.718	0.611	0.812	0.489	0.682	0.747

4.3. *Method Comparison for Different Datasets*

Comparison of different methods for different datasets is presented in Tables 9–13.

As it can be seen from Table 7, the achieved accuracy score for dataset BigML is close to the values which were presented for this particular dataset by other authors, which in most cases are between 0.91 and 0.98 (Xu *et al.*, 2021b; Śniegula *et al.*, 2019), as for IBM dataset, its accuracy values can be found in literature to be between 0.69 and 0.80 (Singh *et al.*, 2021; Pamina *et al.*, 2019). However, in some sources there is a lack of information about the F-measure scores which were used in the current research as the criterion to choose parameters, thus, in this research, a bit of accuracy might be sacrificed to achieve a better F-measure. Thus, from now we assume that the achieved results along with the quality of the selected methods are similar to the ones existing in other researches. In Table 8 the used parameters are presented.

The results of different metrics with different churn labelling rules using different methods are provided in Tables 9–14. In these tables we provide the metrics values for different methods with different labelling rules, for each rule there are two different metrics sets presented: the standard one which is used in all other researches and the True metrics (which were introduced in Section 3.4). The general note on the provided tables is that different methods show similar performance in many cases. Some special cases of exceptional values might be related to the specificity of the data in the provided dataset, thus, next, the attempt to provide general insights of tendencies in the results will be performed.

As the T_c of according labels decreases (row numbers in tables increase), the additional entries with positive churner answers according to the considered approach do the following:

- As it can be expected, true Recall decreases, true Specificity increases. There are some exceptions. The Specificity with labelling rule Churner2 and Gradient Boosting Classifier method worked surprisingly well due to some reasons that are hidden from the researcher, such as better suitability of the provided set for convergence. As it can be seen from Table 15, it takes the highest N-estimators and different parameters than with

Table 8
Hyperparameters of model for different datasets.

	Datasets	
Model	BigML	IBM
	Parameters	
SVMds	‘kernel’:‘rbf’ ‘tol’:0.0001	‘kernel’:‘rbf’ ‘tol’:0.0001
KNeighborsClassifier	‘algorithm’:‘auto’ ‘leaf_size’:15 ‘n_neighbors’:20	‘algorithm’:‘auto’ ‘leaf_size’:15 n_neighbors:20
GBC	‘max_depth’:7 ‘min_samples_leaf’:5 ‘n_estimators’:512 ‘n_iter_no_change’:5 ‘tol’:0.0001	‘max_depth’:2 ‘min_samples_leaf’:3 ‘n_estimators’:256 ‘n_iter_no_change’:5 ‘tol’:0.0001
XGBoost	‘booster’:‘gbtree’ ‘eval_metric’:‘logloss’ ‘max_bin’:128 ‘max_depth’:7 ‘nthread’:1 ‘subsample’:1 ‘use_label_encoder’:False	‘booster’:‘gbtree’ ‘eval_metric’:‘logloss’ ‘max_bin’:128 ‘max_depth’:2 ‘nthread’:1 ‘subsample’:1 ‘use_label_encoder’:False
LGBM	‘boosting_type’:‘gbdt’ ‘learning_rate’:0.1 ‘max_depth’:15 ‘n_estimators’:128 ‘n_jobs’:1 ‘subsample’:1	‘boosting_type’:‘gbdt’ ‘learning_rate’:0.1 ‘max_depth’:5 ‘n_estimators’:128 ‘n_jobs’:1 ‘subsample’:1
RF	‘max_depth’:15 ‘min_samples_leaf’:7 ‘n_estimators’:256	‘max_depth’:30 ‘min_samples_leaf’:7 ‘n_estimators’:128

Table 9
Results of dataset with different churn labelling rules using GBC.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.832	0.646	0.769	0.557	0.848	0.809
	True	0.832	0.646	0.769	0.557	0.848	0.809
Churner1	Standard	0.819	0.597	0.745	0.498	0.836	0.79
	True	0.803	0.622	0.829	0.497	0.796	0.813
Churner2	Standard	0.808	0.571	0.71	0.477	0.83	0.77
	True	0.786	0.612	0.847	0.48	0.77	0.809
Churner3	Standard	0.83	0.578	0.665	0.512	0.865	0.765
	True	0.813	0.67	0.879	0.541	0.794	0.837
Churner4	Standard	0.821	0.543	0.734	0.431	0.835	0.785
	True	0.743	0.558	0.909	0.403	0.707	0.808
Churner5	Standard	0.763	0.574	0.698	0.487	0.782	0.74
	True	0.614	0.499	0.955	0.338	0.527	0.741

Table 10
Results for Moremins dataset with different churner labelling rules using XGBoost.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.821	0.625	0.751	0.535	0.838	0.795
	True	0.821	0.625	0.751	0.535	0.838	0.795
Churner1	Standard	0.819	0.59	0.723	0.498	0.84	0.782
	True	0.808	0.63	0.819	0.511	0.805	0.812
Churner2	Standard	0.818	0.576	0.688	0.496	0.847	0.767
	True	0.798	0.624	0.838	0.497	0.788	0.813
Churner3	Standard	0.829	0.596	0.72	0.509	0.852	0.786
	True	0.798	0.649	0.904	0.506	0.771	0.837
Churner4	Standard	0.791	0.48	0.664	0.376	0.813	0.738
	True	0.749	0.588	0.896	0.437	0.712	0.804
Churner5	Standard	0.763	0.576	0.704	0.487	0.781	0.742
	True	0.617	0.509	0.957	0.346	0.528	0.742

Table 11
Results for Moremins dataset with different churner definitions using LGBM.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.842	0.635	0.688	0.589	0.881	0.784
	True	0.842	0.635	0.688	0.589	0.881	0.784
Churner1	Standard	0.848	0.589	0.609	0.571	0.9	0.754
	True	0.84	0.645	0.74	0.572	0.864	0.802
Churner2	Standard	0.844	0.529	0.489	0.577	0.922	0.705
	True	0.826	0.607	0.703	0.534	0.855	0.779
Churner3	Standard	0.852	0.546	0.506	0.593	0.926	0.716
	True	0.838	0.675	0.789	0.59	0.851	0.82
Churner4	Standard	0.846	0.477	0.484	0.47	0.9	0.696
	True	0.803	0.618	0.835	0.49	0.795	0.815
Churner5	Standard	0.795	0.537	0.519	0.556	0.878	0.698
	True	0.676	0.538	0.921	0.38	0.612	0.767

Table 12
Results for Moremins dataset with different churner definitions using RF.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.85	0.614	0.602	0.627	0.911	0.756
	True	0.85	0.614	0.602	0.627	0.911	0.756
Churner1	Standard	0.842	0.506	0.451	0.576	0.927	0.689
	True	0.841	0.611	0.621	0.602	0.896	0.758
Churner2	Standard	0.849	0.513	0.443	0.609	0.938	0.69
	True	0.844	0.639	0.692	0.593	0.882	0.787
Churner3	Standard	0.857	0.546	0.488	0.62	0.936	0.712
	True	0.837	0.663	0.771	0.582	0.854	0.812
Churner4	Standard	0.854	0.446	0.406	0.495	0.93	0.668
	True	0.801	0.587	0.781	0.47	0.805	0.793
Churner5	Standard	0.783	0.513	0.5	0.526	0.867	0.683
	True	0.658	0.51	0.896	0.357	0.598	0.747

Table 13
Results for Moremins dataset with different churner definitions using KNN.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.688	0.517	0.837	0.374	0.651	0.744
	True	0.688	0.517	0.837	0.374	0.651	0.744
Churner1	Standard	0.697	0.509	0.875	0.359	0.658	0.766
	True	0.691	0.549	0.925	0.391	0.631	0.778
Churner2	Standard	0.667	0.471	0.824	0.33	0.633	0.728
	True	0.655	0.519	0.912	0.363	0.589	0.75
Churner3	Standard	0.689	0.477	0.805	0.338	0.664	0.735
	True	0.664	0.534	0.915	0.377	0.597	0.756
Churner4	Standard	0.647	0.401	0.812	0.266	0.619	0.716
	True	0.608	0.482	0.922	0.326	0.531	0.727
Churner5	Standard	0.609	0.486	0.809	0.347	0.55	0.679
	True	0.488	0.435	0.961	0.281	0.366	0.663

Table 14
Results for Moremins dataset with different churner definitions using SVM.

Label	Type of metrics	Accuracy	F-measure	Recall	Precision	Specificity	Balanced accuracy
Churner	Standard	0.811	0.62	0.774	0.517	0.82	0.797
	True	0.811	0.62	0.774	0.517	0.82	0.797
Churner1	Standard	0.814	0.596	0.761	0.49	0.826	0.793
	True	0.802	0.63	0.846	0.501	0.791	0.818
Churner2	Standard	0.804	0.571	0.727	0.471	0.821	0.774
	True	0.78	0.611	0.85	0.478	0.762	0.806
Churner3	Standard	0.819	0.585	0.726	0.49	0.839	0.782
	True	0.785	0.633	0.892	0.49	0.757	0.824
Churner4	Standard	0.798	0.497	0.688	0.389	0.817	0.752
	True	0.735	0.556	0.889	0.404	0.7	0.794
Churner5	Standard	0.756	0.571	0.71	0.477	0.77	0.74
	True	0.591	0.466	0.952	0.308	0.508	0.73

Table 15
Hyperparameters of methods for Moremins dataset.

	Labelling rules for churners					
Method	Churner	Churner1	Churner2	Churner3	Churner4	Churner5
	Parameters					
GBC	max_depth:3 min_samples_leaf:7 n_estimators:256 n_iter_no_change:5 tol:0.0001	max_depth:2 min_samples_leaf:7 n_estimators:256 n_iter_no_change:5 tol:0.0001	max_depth:2 min_samples_leaf:7 n_estimators:256 n_iter_no_change:5 tol:0.0001	max_depth:2 min_samples_leaf:5 n_estimators:512 n_iter_no_change:5 tol:0.0001	max_depth:2 min_samples_leaf:3 n_estimators:256 n_iter_no_change:5 tol:0.0001	max_depth:2 min_samples_leaf:3 n_estimators:256 n_iter_no_change:5 tol:0.0001
XGBoost	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:256 max_depth:3 nthread:1 subsample:1 use_label_encoder:False	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:512 max_depth:2 nthread:1 subsample:1 use_label_encoder:False	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:128 max_depth:2 nthread:1 subsample:1 use_label_encoder:False	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:256 max_depth:2 nthread:1 subsample:1 use_label_encoder:False	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:128 max_depth:2 nthread:1 subsample:0.5 use_label_encoder:False	booster:‘gbtree’ eval_metric:‘logloss’ max_bin:256 max_depth:2 nthread:1 subsample:1 use_label_encoder:False
LGBM	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:10 n_estimators:128 n_jobs:1 subsample:1	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:15 n_estimators:128 n_jobs:1 subsample:1	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:10 n_estimators:128 n_jobs:1 subsample:1	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:10 n_estimators:128 n_jobs:1 subsample:1	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:5 n_estimators:128 n_jobs:1 subsample:1	boosting_type:‘gbdt’ learning_rate:0.1 max_depth:20 n_estimators:128 n_jobs:1 subsample:1
RF	max_depth:15 min_samples_leaf:4 n_estimators:128	max_depth:15 min_samples_leaf:4 n_estimators:128	max_depth:15 min_samples_leaf:4 n_estimators:128	max_depth:15 min_samples_leaf:5 n_estimators:256	max_depth:15 min_samples_leaf:4 n_estimators:512	max_depth:5 min_samples_leaf:2 n_estimators:128
KNN	algorithm:‘auto’ leaf_size:10 n_neighbors:20	algorithm:‘auto’ leaf_size:3 n_neighbors:20	algorithm:‘auto’ leaf_size:15 n_neighbors:20	algorithm:‘ball_tree’ leaf_size:5 n_neighbors:20	algorithm:‘ball_tree’ leaf_size:5 n_neighbors:25	algorithm:‘auto’ leaf_size:10 n_neighbors:20
SVM	kernel:‘rbf’ tol:0.0001	kernel:‘rbf’ tol:0.0001	kernel:‘rbf’ tol:0.0001	kernel:‘rbf’ tol:0.0001	kernel:‘rbf’ tol:0.0001	kernel:‘rbf’ tol:0.0001

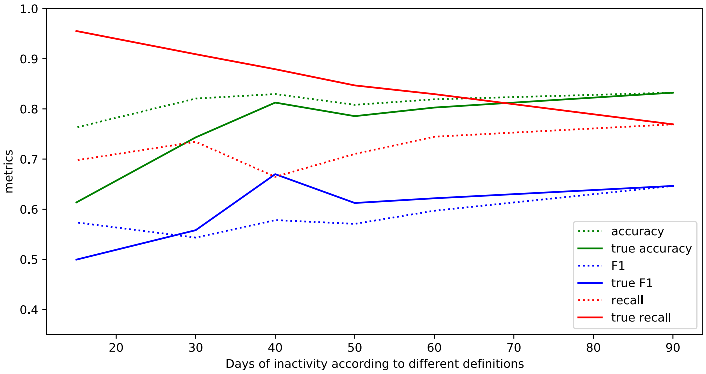

Fig. 2. The dependency of results on the amount of days of customer inactivity T_c used in labelling rules using GBC.

other churn labelling rules leading to a better result. The same stands for Recall for these cases: XGBoost with Churner4, LGBM with Churner2.

- True precision values depend on the precision of classification and the rate of churners among the additional entries. More specifically, the low values of precision with labelling rule according to full churner definition mean more chances to be improved by the big rate of true positives among the additional entries. The overall picture from the Tables is clear – the precision decreases together with T_c with some aforementioned exceptions, this means that precision was higher than the rate of churners among the additional entries.
- True accuracy parameter decreases, more specifically, partial churner (with corresponding label Churner4) comparing to full churner results in these accuracy drops: 0.089 for GBC, 0.072 for XGBoost, 0.039 for LGBM, 0.049 for RF, 0.08 for KNN, 0.076 for SVM.

In Figs. 2–7 there are presented results from Tables 9–14 for Recall, Accuracy and F-measure metrics. The interesting observation is that the high recall with lower T_c results in a better F-measure, especially it can be seen with label Churner3. As it was mentioned before, we do not optimize the results in terms of F-measure in all possible ways, thus, it is possible that similar result can be achieved without decreasing T_c, but with a proper method optimization via thresholding techniques and special loss functions towards a better F-measure. Thus, the decrease of T_c can lead to indirect methods tuning towards a better F-measure at the cost of Accuracy. However, decreasing T_c below 40 results in worse performance for all considered methods: for GBC 0.088, for XGBoost 0.037, for LGBM 0.025, for RF 0.027, for KNN 0.035, for SVM 0.064. It can be concluded that the 30 day inactivity period T_c is not sufficient to keep the same performance of classification methods as it is achieved using full churn definition labelling rule. However, for considered case the $T_c = 40$ performed surprisingly well. Please note that such conclusion is valid for Moremins data case only.

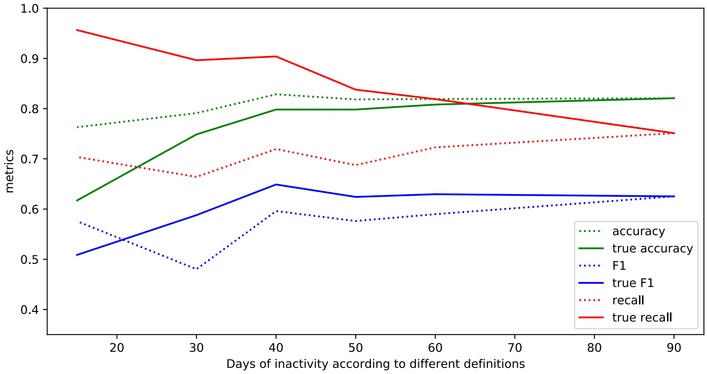

Fig. 3. The dependency of results on the amount of days of customer inactivity T_c used in churn labelling rules using XGBoost.

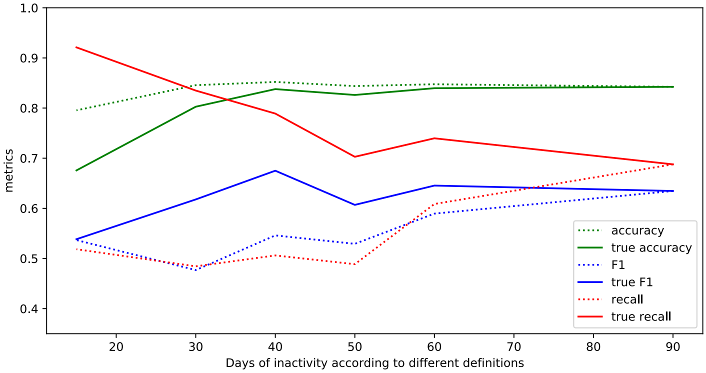

Fig. 4. The dependency of results on the amount of days of customer inactivity T_c used in churn labelling rules using LGBM.

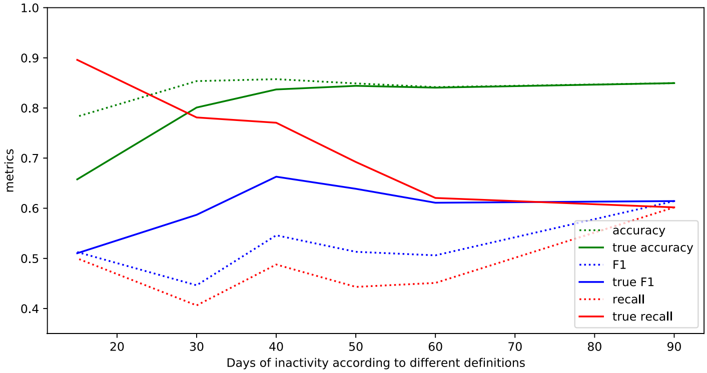

Fig. 5. The dependency of results on the amount of days of customer inactivity T_c used in churn labelling rules using RF.

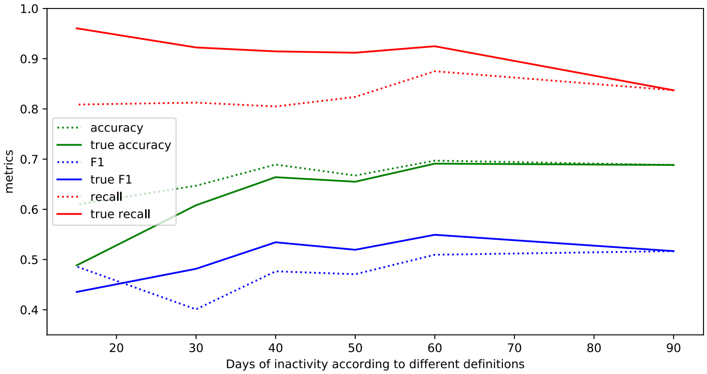

Fig. 6. The dependency of results on the amount of days of customer inactivity T_c used in churn labelling rules using KNN.

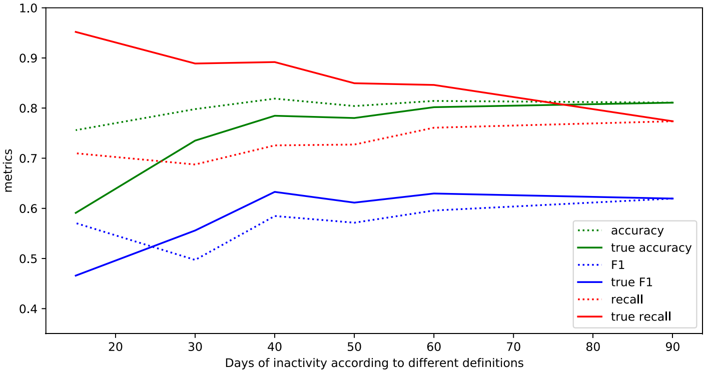

Fig. 7. The dependency of results on the amount of days of customer inactivity T_c used in churn labelling rules using SVM.

5. Discussion

In this research we have mainly focused on tree-based classification methods, as these performed well in many other researches in the context of churn prediction for telecommunication data. These methods were applied to publicly available datasets, partially reproducing the results of other authors. There was no intention to improve the results of other authors directly via methods and their usage development, the main focus was made to investigation of churn definitions and their according labelling rules suitability or unsuitability in the context of churn prediction for telecommunication companies. For this purpose, the data from Moremins company was used, this company specializes on MVNO services.

Due to imbalanced data nature, one of the main selected metrics of performance in this research is F-measure which represents the goal of churn detection very well, it is derived from both precision and recall which are both oriented towards positive answer estimations, i.e. it fits well the churn minority detection. However, the imbalance in the considered data is not very high, since the churn rate is 20.21%, thus, it was decided to not use it for method thresholding in order to optimize this metric, but F-measure was used to select the best combinations of parameters during the Grid Search procedure which was used to tune the model.

According to full churn definition, the best F-measure 0.646 was achieved with GBC method with accuracy 0.832, the best accuracy was achieved using Random Forest classifier with F-measure 0.614. Note that in similar research by Alboukaey *et al.* (2020) one of the best results was achieved using RFM-based Random forest classifier with F-measure 0.525. Thus, we can conclude that Random Forest Classifier supported by RFM features extraction gives reasonably good results despite being considered less advanced method than some other methods.

The results have shown that the reduction of the time interval used for churn labelling rule from 90 to 30 days results in a drop of machine learning performance rates: up to 0.089 for accuracy and 0.088 for F-measure in case of GBC method. However, an interesting observation can be made if that period of 30 days is extended to 40 days – in this case the losses of accuracy are much lower and F-measure greatly increases due to raise of Recall, which is expected due to additional positive answers in the data. However, this effect might be related to Moremins data specificity, so in order to generalize the conclusions, it is necessary to verify these results with other datasets.

It is important to note that all publicly available datasets do not have temporal data, such as daily activity of customers. Moreover, there are many datasets that were created synthetically, production steps of the rest datasets are unknown or are hard to find, the definitions and labelling rules used in them are not clear either. In fact, this lack of knowledge for the nature and derivation of the data related to churn in telecommunication industry raise many questions of data applicability in practice. All the aforementioned data issues create a challenge to make strong and general conclusions with fact verification. However, the methodological steps provided in current research contribute to further development of the general principles of churn prediction for telecommunication companies.

Summarizing the said above, this research makes step forward from methodological point of view for prediction of churn in telecommunications. We showed that omitting the inaccuracies in churn definition might lead to misleading results, small inactivity interval solves a problem which differs a lot comparing to original problem. The accuracy in other researches is better due to some false assumptions, i.e. labelling rules derived from definition leads to a very good classification accuracy, however, it does not imply the usefulness for such churn detection in the context of further customer retention.

The findings in this study raise other questions that might be considered as research gaps:

- Do companies actually need a binary classification of churners, if the result is sensitive to the assumptions that look natural? Some sort of alternative classification generalization could be considered. Especially it can be true since nowadays changing operators is easy, also new eSIM technology possibilities appeared, the loyalty to some services of companies might be much more fuzzy than it was a couple of decades ago.

- The changes of behavioural patterns might greatly affect the proper classification procedure, thus data getting old can significantly affect the results, however, in most studies even the time period is not presented. I.e. even the fact of year season might greatly affect the behavioural patterns of clients, for example, in winter due to Christmas and other socially important events the behavioural pattern might differ a lot comparing to periods during summer time.

6. Conclusions

The performed research has let us answer relevant questions and make conclusions, some of which are following:

1. If the full churner definition must be avoided for different reasons, such as changes in user behavioural patterns, then the definitions based on 40 day inactivity interval can be a reasonable compromise to achieve reasonably good prediction accuracy, the main sources of errors in such case will likely be the classification problem solution.
2. According to the full churn definition, the best F-measure 0.646 was achieved with GBC method with accuracy 0.832, the best accuracy was achieved using Random Forest classifier with F-measure 0.614.
3. In terms of F-measure True metric, the best result was achieved with LGBM method using Churner3 label according to definition based on 40 day interval absence. It is important to note that labelling according to full churner definition gave worse F-measure result, although accuracy is better.
4. The most significant differences in True and standard metrics due to differences of churn definitions can be seen in cases of usage of LGBM and RF methods. For illustration, as a reference we will use Churner4 label derived from 30 day churn definition as it is done in other researches. For LGBM, the Recall metric standard one equal to 0.484, True – 0.835, F-measure standard and True are equal to 0.477 and 0.618, accordingly. There are similar big differences in case of RF method: the Recall standard and True metrics are equal to 0.406 and 0.781, accordingly; the F-measure standard and True metrics are equal to 0.446 and 0.587, accordingly.

References

Adhikary, D.D., Gupta, D. (2020). Applying over 100 classifiers for churn prediction in telecom companies. *Multimedia Tools and Applications*, 1–22.

Adwan, O., Faris, H., Jaradat, K., Harfoushi, O., Ghatasheh, N. (2014). Predicting customer churn in telecom industry using multilayer preceptron neural networks: modeling and analysis. *Life Science Journal*, 11(3), 75–81.

Ahmad, A.K., Jafar, A., Aljoumaa, K. (2019). Customer churn prediction in telecom using machine learning in big data platform. *Journal of Big Data*, 6(1), 1–24.

Ahn, J., Hwang, J., Kim, D., Choi, H., Kang, S. (2020). A survey on churn analysis in various business domains. *IEEE Access*, 8, 220816–220839.

Alboukaey, N., Joukhadar, A., Ghneim, N. (2020). Dynamic behavior based churn prediction in mobile telecom. *Expert Systems with Applications*, 162, 113779.

Amin, A., Anwar, S., Adnan, A., Nawaz, M., Howard, N., Qadir, J., Hawalah, A., Hussain, A. (2016). Comparing oversampling techniques to handle the class imbalance problem: a customer churn prediction case study. *IEEE Access*, 4, 7940–7957. https://doi.org/10.1109/ACCESS.2016.2619719.

Azeem, M., Usman, M., Fong, A.C.M. (2017). A churn prediction model for prepaid customers in telecom using fuzzy classifiers. *Telecommunication Systems*, 66(4), 603–614.

Barrett, J. (2003). *US Mobile Market Intelligence*. Parks Associates, Dallas, TX.

Bose, I., Chen, X. (2009). Hybrid models using unsupervised clustering for prediction of customer churn. *Journal of Organizational Computing and Electronic Commerce*, 19(2), 133–151.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45, 5–32. https://doi.org/10.1023/A:1010933404324.

Chawla, N.V. (2009). Data mining for imbalanced datasets: an overview. In: Maimon, O., Rokach, L. (Eds.), *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA, pp. 875–886.

Chen, T., Guestrin, C. (2016). XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16*. Association for Computing Machinery, New York, NY, USA, pp. 785–794. 9781450342322. https://doi.org/10.1145/2939672.2939785.

Cortes, C., Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297. https://doi.org/10.1007/BF00994018.

Coussement, K., Lessmann, S., Verstraeten, G. (2017). A comparative analysis of data preparation algorithms for customer churn prediction: a case study in the telecommunication industry. *Decision Support Systems*, 95, 27–36. https://doi.org/10.1016/j.dss.2016.11.007.

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*, 269(2), 760–772. https://doi.org/10.1016/j.ejor.2018.02.009.

Fix, E., Hodges, J.L. (1951). *Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine, Report.

Friedman, J.H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232. https://doi.org/10.1214/aos/1013203451.

Friedman, J.H. (2002). Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2.

Gupta, S., Hanssens, D., Hardie, B., Kahn, W., Kumar, V., Lin, N., Ravishanker, N., Sriram, S. (2006). Modeling Customer Lifetime Value. *Journal of Service Research*, 9(2), 139–155. https://doi.org/10.1177/1094670506293810.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.-Y. (2017). LightGBM: a highly efficient gradient boosting decision tree. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Keramati, A., Ardabili, S.M.S. (2011). Churn analysis for an Iranian mobile operator. *Telecommunications Policy*, 35(4), 344–356. https://doi.org/10.1016/j.telpol.2011.02.009.

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. https://doi.org/10.1016/j.asoc.2014.08.041.

Khajvand, M., Zolfaghar, K., Ashoori, S., Alizadeh, S. (2011). Estimating customer lifetime value based on RFM analysis of customer purchase behavior: case study. *Procedia Computer Science*, 3, 57–63. World Conference on Information Technology. https://doi.org/10.1016/j.procs.2010.12.011.

Lu, J. (2002). Predicting Customer Churn in the Telecommunications Industry —- An Application of Survival Analysis Modeling Using SAS. In: *Proceedings of the Twenty-Seventh Annual SAS Users Group International Conference*. Retrieved from https://support.sas.com/resources/papers/proceedings/proceedings/sugi27/p114-27.pdf.

Moremins (2021). https://www.moremins.com/en.

Pamina, J., Raja, B., SathyaBama, S., S, Soundarya, Sruthi, M.S., S, Kiruthika, V J, Aiswaryadevi G, Priyanka (2019). An effective classifier for predicting churn in telecommunication. *Journal of Advanced Research in Dynamical and Control Systems*, *11*.

Routh, P., Roy, A., Meyer, J. (2021). Estimating customer churn under competing risks. *Journal of the Operational Research Society*, 72(5), 1138–1155.

Singh, D., Jatana, V., Kanchana, M. (2021). *Survey Paper on Churn Prediction on Telecom*. Available at SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3849664.

Śniegula, A., Poniszewska-Marańda, A., Popović, M. (2019). Study of machine learning methods for customer churn prediction in telecommunication company. In: *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services*, pp. 640–644.

Telco custumer churn (2020). https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fpublic.dhe.ibm.com%2Fsoftware%2Fdata%2Fsw-library%2Fcognos%2Fmobile%2FC11%2Fdata%2FTelco_customer_churn.xlsx&wdOrigin=BROWSELINK.

Telco data (2021). https://bigml.com/user/francisco/gallery/dataset/5163ad540c0b5e5b22000383.

Ullah, I., Raza, B., Malik, A.K., Imran, M., Islam, S.U., Kim, S.W. (2019). A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE Access*, 7, 60134–60149.

Vafeiadis, T., Diamantaras, K.I., Sarigiannidis, G., Chatzisavvas, K.C. (2015). A comparison of machine learning techniques for customer churn prediction. *Simulation Modelling Practice and Theory*, 55, 1–9. https://doi.org/10.1016/j.simpat.2015.03.003.

Xu, T., Ma, Y., Kim, K. (2021a). Telecom churn prediction system based on ensemble learning using feature grouping. *Applied Sciences*, 11(11), 1–12. https://doi.org/10.3390/app11114742.

Xu, T., Ma, Y., Kim, K. (2021b). Telecom churn prediction system based on ensemble learning using feature grouping. *Applied Sciences*, 11(11), 4742.

Zhang, J., Fu, J., Zhang, C., Ke, X., Hu, Z. (2016). Not too late to identify potential churners: early churn prediction in telecommunication industry. In: *Proceedings of the 3rd IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, BDCAT '16. Association for Computing Machinery, New York, NY, USA, pp. 194–199. 9781450346177. https://doi.org/10.1145/3006299.3006324.

A. Bugajev in 2015 has defended the dissertation on a topic "The investigation of efficiency of physical phenomena modelling using differential equations on distributed systems". In his dissertation the computational efficiency problems were solved – the efficient parallel algorithms were created and examined, the stability of the algorithms was investigated. The research interest covers theory of algorithms, parallel algorithms, machine learning. He has published 12 papers in journals with Impact Factors indexed in the "Web of Science" database, 7 of them during the last 5 years.

R. Kriauzienė in 2020 has defended dissertation on the topic "Parallel algorithms for non-classical problems with big computational costs". The dissertation is devoted to parallel algorithms that help to solve the problems of memory resource and computation time. Effective parallel algorithms were developed and analysed. At the Young Scientists' Conference of the Lithuanian Academy of Sciences "Interdisciplinary Research in Physical and Technological Sciences: 7th Conference of Young Scientists", her work was rated high, she was included in the list of laureates and awarded the INFOBALT second degree award. She has published 5 articles with Impact Factors indexed in the "Web of Science" database, 4 of them during the last five years.

O. Vasilecas is a senior researcher at the Institute of Applied Informatics of Vilnius Gediminas Technical University (Vilnius Tech). He is the author of more than 329 research papers and 5 books in the field of information systems development. His research interests: knowledge, including business rule and ontology, based information systems development, and Data Science. He delivered lectures in 7 European universities including London, Barcelona, Athens and Ljubljana. O. Vasilecas carried out an apprenticeship in Germany, Holland, China, and last time in Latvia and Slovenia universities. He supervised 13 successfully defended doctoral theses and now is supervising 2 doctoral students. He was the leader of many international and local research projects. Last time he led the "Business Rules Solutions for Information Systems Development (VeTIS)" project carried out under the High Technology Development Program.

V. Chadyšas involves its research areas with a comprehensive analysis of data and the application of different statistical methods in various areas of life. In 2010, he defended the doctoral thesis on the topic "Statistical estimators of the finite population parameters in the case of sample rotation". During his scientific career, Viktoras Chadyšas has prepared and published over 20 scientific articles in mathematics journals. The results of the research were presented at more than 20 scientific conferences held in different Lithuanian and foreign cities. In 2005, Viktoras Chadyšas received the Lithuanian Academy of Science Prize in mathematics, physics and chemistry section for the work "Viktoro Chadyšo 2005 publications". From 2006 Viktoras Chadyšas is a member of the Society of Lithuanian Mathematicians.