

CANCEL: A feature engineering method for churn prediction in a privacy-preserving context

Gabriel T. Coimbra   [Universidade Federal de Viçosa | gabriel.coimbra@ufv.br]

Victor Hugo R. Santos  [Universidade Federal de Viçosa | victor.h.santos@ufv.br]


Pedro A. Maia  [Universidade Federal de Viçosa | pedro.maia@ufv.br]

Leticia O. Silva  [Universidade Federal de Viçosa | leticia.silva1@ufv.br]

Rayanne P. Souza  [Universidade Federal de Viçosa | rayanne.souza@ufv.br]

Fabício A. Silva  [Universidade Federal de Viçosa | fabicio.asilva@ufv.br]

Thais R. M. Braga Silva  [Universidade Federal de Viçosa | thais.braga@ufv.br]

 IEF, Universidade Federal de Viçosa - Campus Florestal, Rodovia LMG 818, km 6, Florestal - MG, 35690-000, Brazil

Received: 23 November 2023 • **Accepted:** 12 July 2024 • **Published:** 04 October 2024

Abstract This paper proposes a solution for predicting churn with privacy preservation by using edge computing. With the increasing popularity of smartphones, users are becoming more demanding regarding mobile app usage. Installing and removing an app are frequent routines and the ease of uninstallation can facilitate churn, which is customer abandonment. Companies seek to minimize churn since the cost of acquiring new customers is much higher than retaining current ones. To predict possible abandonment, organizations are increasingly adopting artificial intelligence (AI) techniques. Nevertheless, customers are becoming more concerned about their data privacy. In this context, we propose a technique called CANCEL, which creates attributes based on users' temporal behavior, with edge computing to predict churn locally, without transmitting users' data. The paper presents the evaluation of CANCEL in comparison to baseline solutions, the development of a mobile app integrated with the proposed method and deployed as an edge computing solution.

Keywords: Churn Prediction, Privacy Preservation, Edge Computing, Mobile App Usage

1 Introduction

With the popularization of smartphones, users are becoming increasingly demanding in terms of mobile application usage. Therefore, the processes of installing and uninstalling an application are frequent routines and easy to execute for anyone using a smartphone. On one hand, the ease of installing an application is a positive point for companies that build software, as long as the proposed solution has added value to become popular. On the other hand, the ease of uninstallation is a significant obstacle, facilitating customer abandonment, known by the term *churn*.

According to Forbes Insights [2011], the corporate environment prioritizes customer retention due to the high cost of acquiring new customers compared to the cost of retaining current ones. Therefore, many organizations seek to understand customer behavior to predict possible churn. Understanding how their customers behave is a challenging task for companies, especially for large corporations, which have thousands or even millions of customers. To solve this problem, Artificial Intelligence (AI) techniques, more specifically Machine Learning, have been used to try to predict user churn.

While companies seek data-oriented solutions to predict customer churn, customers are increasingly concerned about their data privacy [Fox *et al.*, 2022]. Consequently, in recent years, there have been advancements in techniques that reduce the risk of users having their often personal and sensitive data leaked. This concern is reflected in laws being

drafted around the world, such as the General Data Protection Law (LGPD) in Brazil.

In this study, a solution for churn prediction with privacy preservation through the use of edge computing is proposed. Initially, a technique called CANCEL (*Curve-Aware Churn Prediction Models*) is proposed to create attributes based on user behavior over time. This technique is evaluated in comparison with baseline solutions, having achieved superior results in terms of accuracy. Subsequently, the best model that uses CANCEL is integrated into a mobile application so that the prediction of churn is made locally, preventing user data from being transmitted. A web application was developed to present the results of the predictions executed locally on the devices. In summary, this study presents two main contributions: a technique to consolidate user behavior over time and an integrated edge solution to predict churn with privacy preservation.

This study is an extension of the work published in [Coimbra *et al.*, 2023]. In this work, we have expanded the related work section to provide a more comprehensive background and context for our study. Additionally, we have enhanced the data description and characterization to give a clearer understanding of the dataset used for churn prediction. We have also tested two new synthetic functions in our CANCEL method to further refine the attribute creation process. Also, a more detailed description of the results has been provided, along with a thorough analysis of the outcomes. The baseline used for comparison has been modified to include temporal results, allowing a fairer evaluation of the churn prediction

models. Also, we tested CANCEL feature engineering in a public dataset.

The remainder of this text is organized as follows. Section 2 presents related work. Section 3 describes the data used for this study. Section 4 deals with the proposal and results of the attribute engineering method proposed in this study. Section 5 describes the proposal for integrating learning models using edge computing. Finally, Section 6 discusses the final considerations.

2 Related Work

Table 1 presents a comparison of the main studies found in the literature and our proposal. Most studies are not concerned about privacy-preserving and work only on particular scenarios. In the following, we discuss them.

The most common context for automated churn prediction solutions is when the customer dropout rate significantly impacts the business, such as in the telecommunications sector (telephony, Internet, and television) [Jain *et al.*, 2020; Verhelst, 2018]. This scenario is also present in social networking applications [Yang *et al.*, 2018], music streaming [Zhou *et al.*, 2019], games [Milošević *et al.*, 2017], online courses [Tan *et al.*, 2018], credit card provider companies [Rajamohamed and Manokaran, 2018] and other sectors (e-commerce, health, insurance [Bharathi S *et al.*, 2022]) across the industry.

In addition to these sectors, similar to the present work, there are studies that investigate churn prediction in the banking sector [Haddadi *et al.*, 2022; Rahman and Kumar, 2020; Bharathi S *et al.*, 2022; Kavyarshitha *et al.*, 2022; Zaky *et al.*, 2022; Jagad *et al.*, 2023], but using specific data from this environment, such as customer banking transactions, which requires breaching their privacy. The focus on churn prediction in these industries occurs because the financial cost to carry out retention campaigns and prevent churn is often significantly lower than the cost of acquiring new customers [Gupta *et al.*, 2004] so the ability to identify customer churn early and implement marketing strategies using artificial intelligence (AI) systems is crucial for businesses [Wang *et al.*, 2023].

The studies mentioned above aim to improve the accuracy of churn prediction by evaluating state-of-the-art models [Jain *et al.*, 2020; Yang *et al.*, 2018; Zhou *et al.*, 2019; Tan *et al.*, 2018; Rajamohamed and Manokaran, 2018; Kilimci, 2022; Wang *et al.*, 2023], interpreting the results of the models [Yang *et al.*, 2018], or inferring causal relationships [Verhelst, 2018]. However, there is no focus on investigating ways to develop and evaluate these models in environments where data privacy is desirable and churn data represents a typical scenario that frequently encounters privacy concerns, particularly when dealing with sensitive data [Wang *et al.*, 2023]. Also, despite the prediction metrics of our study being lower to those reported in related banking churn predictions in existing literature (Kilimci [2022]; Kavyarshitha *et al.* [2022]; Zaky *et al.* [2022]; Jagad *et al.* [2023]; Haddadi *et al.* [2022]; Rahman and Kumar [2020]), this difference can be readily attributed to our methodology, which does not incorporate transactional or business-specific data while pre-

dicting churn. Thus, our approach is easily applied in other business contexts.

The work [Bertens *et al.*, 2017] introduces a technique that allows for real-time training and prediction with low processing cost, but it is not applicable to mobile applications due to the need for a large volume of data for training and it is not implemented in production. Other studies employ techniques to understand the user's relationship with the service and modify the experience in real time to prevent churn, such as in [Li *et al.*, 2021], which estimates the difficulty of an online game through user interaction metrics and makes modifications to alter the game's formality.

In addition to the aspect of prediction, another increasingly important requirement is the protection of sensitive user-generated data [Schlackl *et al.*, 2022]. In most of the studies found, invasive data were used that require extra protection measures when transmitted and stored. Additionally, it is important to note that these measures vary depending on the legislation of the country or state, such as GDPR in Europe and LGPD in Brazil.

In this context, the concept of federated learning [Bonawitz *et al.*, 2017] has emerged to improve data protection and reduce the need for large-scale computational resources (i.e., storage and data transmission) on centralized servers. However, two problems arise: firstly, a higher consumption of computational resources from users' mobile devices is required, which is not always accepted by them or possible due to mobile device restrictions. In addition, the model trained by federated learning has several challenges (i.e., data or model poisoning, reduced accuracy, and high aggregation cost) compared to models trained with all centralized data, as aggregation methods still need to evolve [Mammen, 2021]. It is also worth noting that, unlike some solutions in the literature with federated learning approaches [Shokri and Shmatikov, 2015; Bonawitz *et al.*, 2017] that have greater potential for data protection, since sensitive data are never transmitted over the network, the approach of this work is feasible even in big data contexts and mobile device battery preservation, as no training is done on the mobile device, only prediction is made which is less intensive in computational cost.

It is also worth highlighting that customer data is complex, it contains multi-dimensional characteristics that are dynamically changing and just a small fraction of customers churn among customers, which is a challenge for traditional statistical methods [Xiong *et al.*, 2019]; therefore, methods are needed to facilitate feature engineering, such as the proposal to use synthetic data to resolve data quantity, quality, and availability issues [Wang *et al.*, 2023].

Unlike existing studies, this paper proposes and evaluates an approach for predicting mobile app churn using traditional machine learning algorithms in conjunction with enrichment techniques that allow for the inference of user behavior over time. Also, the proposed approach enhances security and preserves data privacy while making churn predictions by edge computing in addition to being a generalist and flexible solution that can be adopted in different scenarios related to mobile services.

Work	Industry	Metrics	Edge Computing	Privacy	Feature Engineering
Jain et al. [2020]	Telecom	ACC=0.85	-	-	-
Verhelst [2018]	Telecom	AUC=0.73	-	-	Difference and ratio columns
Yang et al. [2018]	Online Platform	-	-	-	Fit a sigmoid curve on time series data
Zhou et al. [2019]	Music Streaming	AUC=0.87	-	-	XGBoost tree model features
Milošević et al. [2017]	Mobile Games	-	-	-	Feature Standardization
Tan et al. [2018]	-	F1=0.76	-	-	Imputation, capping, logarithm, binning, interaction of two variables and products
Rajamohamed and Manokaran [2018]	Banking	ACC=0.96	-	-	-
Haddadi et al. [2022]	Banking	F1=0.83	-	-	-
Rahman and Kumar [2020]	Banking	ACC=0.95	-	-	-
Bertens et al. [2017]	Mobile Games	-	-	-	-
Li et al. [2021]	Online Games	F1=0.94	-	-	Difficulty-aware features
Bharathi S et al. [2022]	Banking	F1=0.92	-	-	-
Kilimci [2022]	Banking, Insurance, Telecommunication	F1=0.90	-	-	-
Kavyarshitha et al. [2022]	Banking	ACC=0.86	-	-	Data Scrubbing and Feature Selection
Zaky et al. [2022]	Banking	F1=0.92	-	-	Label Encoding and Feature Scaling
Jagad et al. [2023]	Banking	ACC=0.96	Train local models	Send only parameter weights and gradients to the server	SMOTE
Wang et al. [2023]	-	AUC=0.76	-	-	Use synthetic data
Xiong et al. [2019]	-	F1=0.64	Multi-terminal Processing	-	-
Our Work	Mobile Services	F1=0.72	Model execution locally	Prediction result sent to the server	CANCEL

Table 1. Related Work

3 The Data

In this study, a database provided by a partner company through a confidentiality agreement is utilized, containing records of 33,817 users of a digital banking application. The available data includes: the application installation date, the banking application usage records, user location records when moving, a list of other applications installed on the user’s device, and the device model. From this data, feature engineering was performed in this study, explained below, to extract metrics and enrich the data.

The decision to classify a user as a churner or non-churner is based on the amount of time they have gone without accessing the application. To better understand, a user is classified as a churner when their last access is equal to or greater than 3 months (i.e., 90 days). Figure 1 presents the CDF (Cumulative Distribution Function) for the interval between accesses for the users. The majority (96%) of users in this context are non-churners, as their access interval, even if long, does not exceed 90 days. This demonstrates that only 4% of our user base are churners. Considering the scenario of banking applications, it is expected that a considerable proportion of users only access the application sporadically, which does not mean that these are considered *churners*. One of

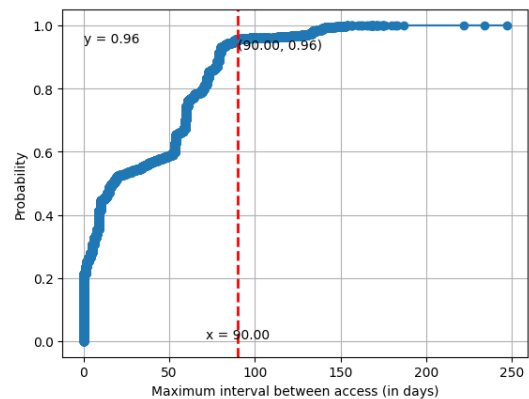


Figure 1. CDF of the maximum interval between accesses.

the reasons for this could be the use of this banking application for investment, especially long-term, which leads to less frequent use of the application. Another reason is that users may use other forms of interaction with the bank that are not through the application, such as cards.

With the usage records and the installation date, the number of distinct days the user accessed the application and the number of days since the application was first installed were calculated. Using the user’s location records, the point of interest detection algorithm from Capanema *et al.* [2021] was used to infer the approximate location of their home. Data from the 2010 IBGE¹ Census were used to enrich with demographic information from the census sectors where the house is located. These demographic data provide information about the age range of the region’s residents, ethnicity, income, and gender². The CrawMobi [Maia *et al.*, 2020] was also used to estimate the price and the release year of the device.

Finally, from the application usage records that were available, sequential information was extracted containing, in chronological order, the frequency of application use and the duration of each use. The decision to extract this sequential information was based on the possibility that this information could well represent the user’s behavior with the application over time, a factor considered relevant for churn prediction, as will be shown in the results.

In this way, the churn decision is enriched by using the 100 most installed applications, allowing the model to leverage this information to determine whether a customer has churned or not. After that, we have 32,324 non-churner users compared to 1,493 churners, and each of these users has a list of installed applications. Figure 2 presents the top 20 most installed apps for churners and non-churners. It should be noted that the chance of a user churning due to having other banks (Digital Bank 0 or Digital Bank 1)³ installed on their phone is high. This is why these attributes significantly enhance the model’s ability to decide whether a user is a potential churner or not.

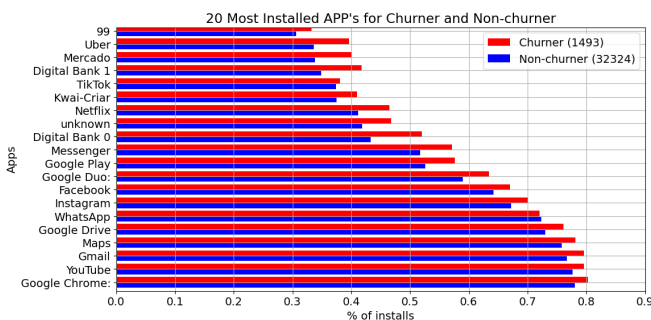


Figure 2. 20 app’s more installed.

The imbalance between *churners* (96%) and *non-churners* (4%) adds an extra difficulty to the problem, which will be addressed in the solution. It is worth noting that, although the data used pertains to users of a digital bank, the proposed solution is flexible to be adopted in different scenarios, pro-

vided that the period of inactivity for churn indication is adjusted.

4 Churn prediction solution

The problem of predicting churn in other industries such as telecommunications companies, has been extensively studied in the literature using traditional machine learning algorithms. However, one of the main limitations found is the lack of a *baseline* that can be applied in different scenarios, since each problem has its own characteristics and data availability. Thus, the main task in a churn prediction problem is feature engineering, responsible for identifying and preparing the relevant attributes for the predictive model. With this, in Section 4.1 a method is proposed that allows traditional machine learning models to use temporal data. In Section 4.2 this method is evaluated in comparison to other techniques in the context of churn prediction.

4.1 CANCEL: Curve-Aware Churn Prediction Models

This work proposes CANCEL (*Curve-Aware churn Prediction modELs*), a method to be applied during feature engineering to enable traditional machine learning models, which work with static data, to make inferences about temporal behaviors. For this, a technique similar to that of [Yang *et al.*, 2018; Lin *et al.*, 2003] was used. However, in this work, the values calculated by CANCEL are used in a supervised learning context (i.e., churn prediction) and not for interpreting user behaviors in applications while clustering like in Yang *et al.* [2018]. Also, we propose using curve fitting as the basis of our method instead of symbolic approximations like in Lin *et al.* [2003].

4.1.1 Preliminaries

CANCEL is a generalist proposal to easily allow the inference of sequential information from time series in traditional machine learning algorithms already known in the literature. In summary, this method consists of finding parameters of synthetic functions that minimize the distance between the real data and these functions. With this, these parameters and errors of the curve fitting process can summarize the behavior of the functions and can be used directly in machine learning algorithms to provide information about the time series. Beyond generalization, when reducing sequential features to a constant number of scalar features, CANCEL also implicitly provide regularization and increase the training speed of traditional machine learning algorithms.

$$f_{linear}(x) = \alpha_l \times x + \beta_l \tag{1}$$

$$f_{sine}(x) = \sin(\alpha_s \times x) + \beta_s \tag{2}$$

$$f_{quadratic}(x) = \alpha_q \times x^2 + \beta_q \times x \tag{3}$$

$$f_{sigmoid}(x) = \frac{1}{1 + e^{-(\alpha_i \times x + \beta_i)}} \tag{4}$$

¹Brazilian Institute of Geography and Statistics

²<https://censo2010.ibge.gov.br/sinopseporsetores/>

³The name of the apps was removed due to privacy concerns.

In the development of the CANCEL method, the selection of synthetic functions was an important decision. The chosen functions, namely linear (Eq. 1), sine (Eq. 2), quadratic (Eq. 3), and sigmoid (Eq. 4), were selected. The combination of these four functions is expected to be effective in capturing the temporal behaviors of interest. Secondly, these functions, when combined, can represent a wide range of behaviors. They are capable of modeling periodic, constant, increasing, decreasing behaviors, or a combination of these. Furthermore, they can capture the nuances of these behaviors, such as constant acceleration (quadratic), constant rate (linear), and deceleration (sigmoid) in the temporal behaviors. This versatility is important in the context of user behavior, which can vary significantly and unpredictably. Thirdly, the parameters obtained from these functions, along with the error from the function fitting from the original data, provide valuable information for the model. They allow the model to determine which parameters are most suitable in a given situation. Lastly, while it is possible to use more functions, this can lead to overfitting. By limiting the number of functions, we reduce this risk. Also, given the diversity of behaviors that users can exhibit, it is reasonable to think that more complex functions would not be necessary to represent these behaviors accurately.

We note that some functions were made specifically for this work. The quadratic function (3) does not have a constant parameter as is commonly used. The constant term can be useful for specifying the intercept at the Y-axis. However, as all values were scaled beforehand, this constant term is not needed and would add an additional term for the optimization and as a feature in the model.

4.1.2 Formal Definitions

A more formal description of how these functions are used follows: let $\hat{f}_v(x)$ be the observed behavior for the behavior category v (e.g., quantity or duration of accesses in the time window) of the user in the time-space x , where $x \in \mathbb{N}$ and $1 \leq x \leq N$, and N represents the number of time windows (e.g., days, weeks or months) of sequential data collected for this user. The goal now is to minimize ϵ from equation 5 using non-linear least squares algorithms on the parameters of the synthetic functions (e.g., $\alpha_l, \beta_l, \alpha_s$ and β_s for the linear and sine functions). With this, we will have the best parameters for the synthetic functions that explain the real behavior.

In addition, in Equation 5, $v \in \{\text{frequency}, \text{duration}\}$ and $c \in \{\text{sine}, \text{linear}, \text{sigmoid}, \text{quadratic}\}$, considering the scope of this churn prediction work.

$$\epsilon_v^c = \frac{\sum_{i=1}^N (\hat{f}_v(i) - f_c(i))^2}{N} \quad (5)$$

In addition to the function parameters (in the case of this work $\alpha_l, \beta_l, \alpha_s, \beta_s, \alpha_q, \beta_q, \alpha_i$ and β_i), the mean square errors ϵ_v^c for each function are also used as attributes during the training of the supervised models. With this, it is possible to summarize the user's trend for a specific attribute in a few parameters, which reduces the chances of overfitting. The values of the function parameters are used by the model to summarize the sequential behavior of the users, while ϵ allows the model to consider the magnitude of the error. With

this, it is possible for the model to assign a greater weight to the parameters when the distance between the user's behavior and the synthetic function is smaller.

The minimization for the more complex functions (i.e., sigmoid) is performed by the Levenberg-Marquardt algorithm [Moré, 2006] (LM) implemented in the MINPACK library⁴ with the default value. This algorithm is the default choice of the `curve_fit` function⁵ when the number of observed values is greater than the number of variables and the optimization problem has no constraints on the image of the optimization function; these conditions are met in the dataset used here. The LM is used because it allows the fitting of various functions, not necessarily being the best algorithm to minimize the linear or sine function. When using this minimization method, it is important to be careful with the choice of the domain x of f , as ideally it should be possible to represent any value of f given the appropriate parameters for \hat{f} . For example, if the function $\hat{f}(x, a) = \sin(x * a)$ was chosen, the convergence of the minimization is more difficult if the codomain of f is outside the interval $[-1, 1]$. Therefore, the function $\hat{f}_v(x)$ is normalized between -1 and +1 using the min-max normalization, which facilitates convergence at the time of curve fitting, especially for the case of the sine function. We note, however, that algorithms such as LM are not necessary for curve fitting of some of the described functions, such as linear and quadratic, because they have a non-convergence needed closed formula solutions.

The Levenberg-Marquardt method generally converges with quadratic or linear rate Ueda and Yamashita [2010]. With CANCEL's method, this optimization procedure must be done for each temporal feature compared to simply using the feature over a time window in a naive approach. We consider this a good tradeoff between accuracy and computational resources.

In summary, when using CANCEL, the optimized parameters and the error for each function (i.e., sine, linear, sigmoid, and quadratic) and by behavior category (i.e., access frequency and access duration) are included. Therefore, in the present study, when using CANCEL the dataset from Section 3, referred to as *baseline*, is incremented with 24 (3 parameters \times 4 functions \times 2 categories) new attributes generated from sequential data.

4.2 Evaluation

4.2.1 Setup

For the evaluation of CANCEL, five algorithms were used: Random Forest (RF), XGBoost, SVM, Logistic Regression (LR), and *Naïve Bayes* (NB). Each of these algorithms was adopted in two implementations: *baseline*, where only the original static data were used, and CANCEL, where the sequential data transformed by the CANCEL method described in Section 4.1 were used. This will enable the evaluation of the impact of CANCEL from different perspectives.

From the training data, to replace null values in the static data, a simple imputation is performed using the strategy

⁴<https://netlib.org/minpack/>

⁵https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.curve_fit.html

of replacing with the most frequent value, as this technique works with both numerical and categorical data. No imputation is performed on the sequential data; if the user has less than 3 weeks of data, the sequential values are considered null in the models due to restrictions imposed by the LM optimization algorithm, used by CANCEL. Categorical values, such as city name, and state name, among others, are encoded into distinct numbers. Also, to reduce the computational cost during the prediction at the edge described in Section 5, only the 100 most common applications were considered. And, to avoid balancing problems, a random *downsampling* of the *non-churners* user class was performed.

In order to establish a comparative benchmark for the proposed CANCEL method, a baseline approach was developed. This baseline approach utilizes the same static features, such as the demographic information from the estimated home location of the user. However, instead of employing the CANCEL method to extract information from the sequential data, the baseline approach aggregates the time series data in a variety of ways. The time series data is first arranged in chronological order, after which four statistical metrics are extracted: the mean, maximum, minimum, and standard deviation. These four metrics are calculated over four distinct periods: from the present day to 30 days prior, from 30 to 60 days prior, from 120 to 150 days prior, and from 150 to 180 days prior. This results in the generation of 16 new features for each user. This baseline approach provides a comparison point for evaluating the effectiveness and efficiency of the CANCEL method in extracting meaningful information from time series data.

The metrics shown in the following figures were obtained with 10 repetition *5-fold* cross-validation. Also, in all rounds of the cross-validations, the same hyperparameters used in the models with the information provided by CANCEL are used in the *baseline* models. The presented *F1-score* is calculated using the Precision and Recall of both classes of users (e.g., churners and non-churners). The only difference between both models generated by the same algorithm are the attributes used, emphasizing that the solutions that use CANCEL add the new attributes based on the sequential data.

It is important to note that, for a fair comparison, the *baseline* models include information calculated on the sequential data such as count of values, minimum, maximum, mean, and standard deviation. These values are commonly used in works where there are sequential data but in a static fashion [Milošević et al., 2017; Bertens et al., 2017]. The bars height in the following plots represent the metric value and the standard deviation of the sample is also shown.

4.2.2 Results

The CANCEL method has shown significant improvements in several metrics of traditional machine learning models. As shown in Figure 3 for the F1-Macro Score, the mean precision increased for all models when CANCEL was applied. The most significant increase was observed in the Naive Bayes model, from 55.3% to 71.9%. The standard deviation also decreased for all models, indicating more consistent performance.

Figure 4 shows the Recall for churners. The CANCEL

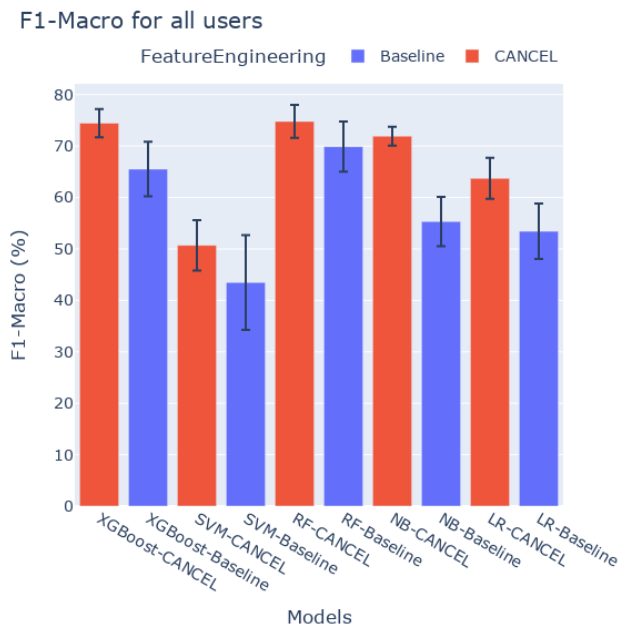


Figure 3. Difference between F1 Macro for feature engineering methods CANCEL and baseline computed for all users.

technique had varying impacts. For Logistic Regression and Naive Bayes, the mean recall improved slightly. However, for the Random Forest model, the mean recall decreased, but the model’s consistency improved. The SVM model showed a slight decrease in mean recall but a significant reduction in standard deviation, indicating more consistent performance.

In Figure 5, is shown that for Recall for non-churners, the CANCEL technique generally improved the performance of the models, particularly for Logistic Regression and Naive Bayes. The Random Forest model showed a slight decrease, but an increase in model stability. The SVM and XGBoost models also showed improvements.

For Figure 6, the precision for churning users, the most significant improvement was observed in the Naive Bayes model, where the mean precision increased from 56% to 77%. The Logistic Regression and XGBoost models also showed substantial improvements, while the SVM model showed the least improvement.

For Figure 7, the precision for non-churner users, all models but the Random Forest one showed an increase in mean precision when the CANCEL technique was applied. The standard deviation of precision generally decreased, except for the SVM model, where it increased slightly.

Due to the better metrics using CANCEL, the Random Forest model (i.e., *RandomForestCANCEL*, represented in the blue bars in the Figure) was chosen to be deployed at the edge in Section 5.

4.2.3 Feature Importance Analysis

The analysis of the feature importance as determined by the Random Forest model with the Gini Importance provides insightful information about the relevance of different attributes in the context of churn prediction. In Figure 8 is possible to observe that, for CANCEL, the temporal features are the most important group of features, and for the baseline, the demographic features are more important than the tem-

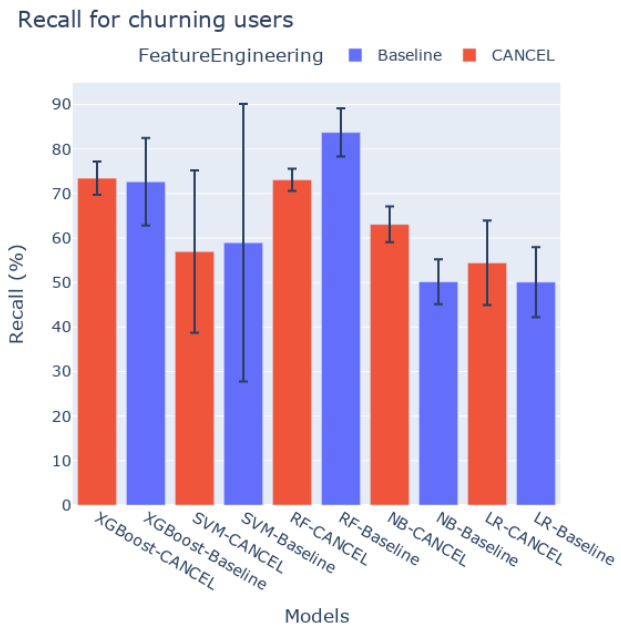


Figure 4. Difference between recall for feature engineering methods CANCEL and baseline computed for churning users.

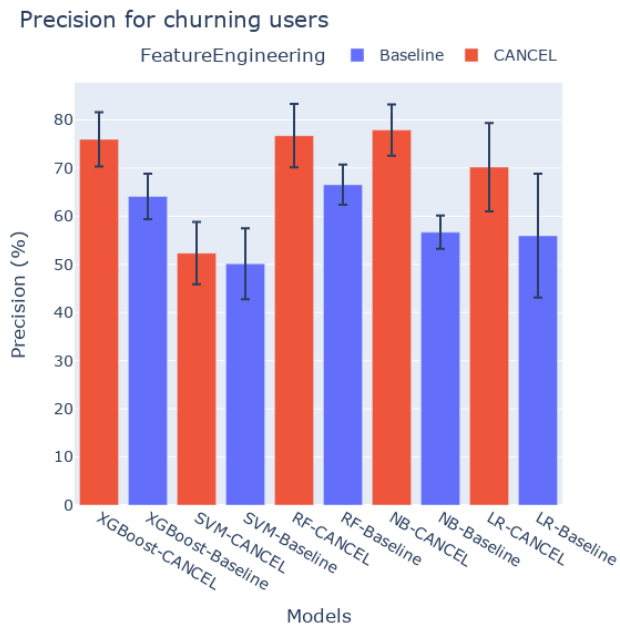


Figure 6. Difference between precision for feature engineering methods CANCEL and baseline computed for churning users.

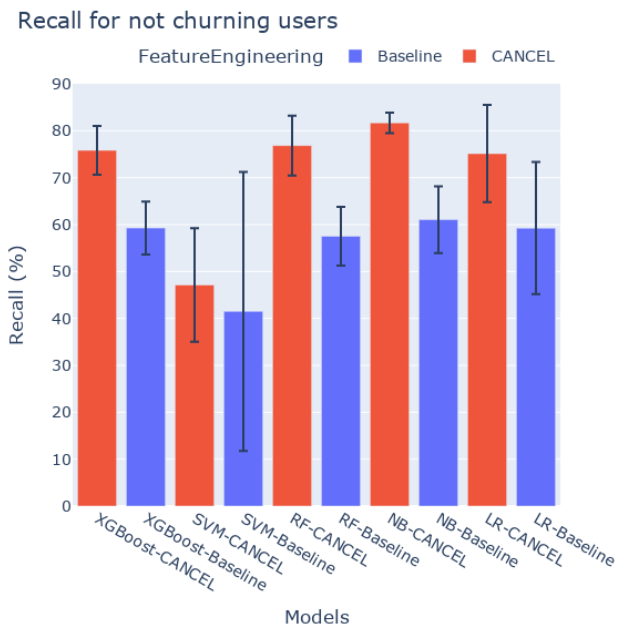


Figure 5. Difference between recall for feature engineering methods CANCEL and baseline computed for not churning users.

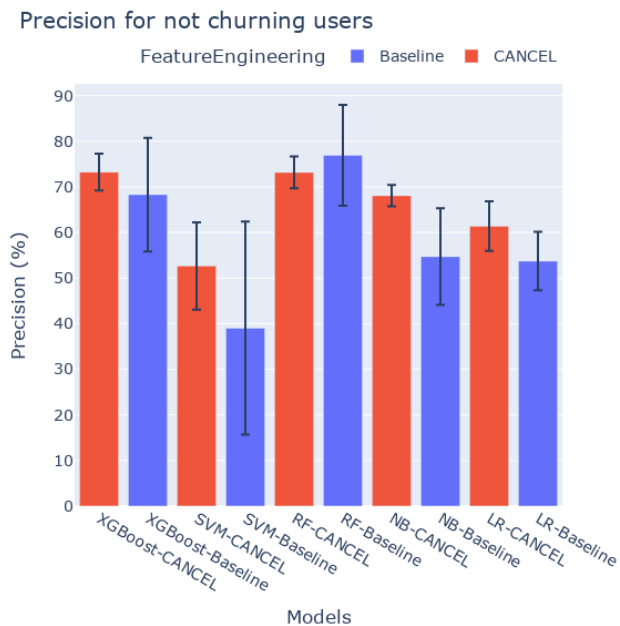


Figure 7. Difference between precision for feature engineering methods CANCEL and baseline computed for not churning users.

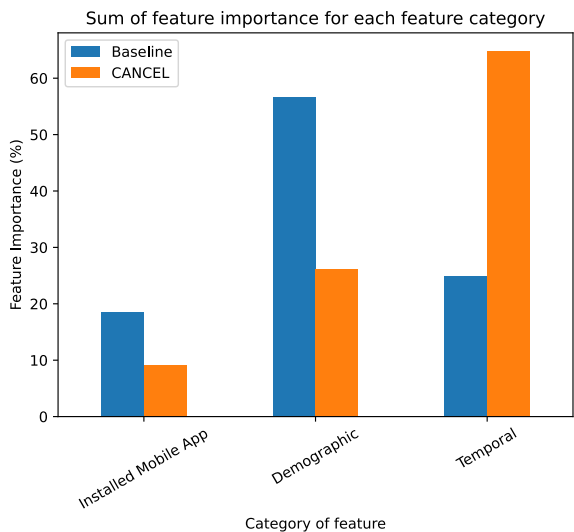


Figure 8. Most important groups of features according to Random Forest gini measure.

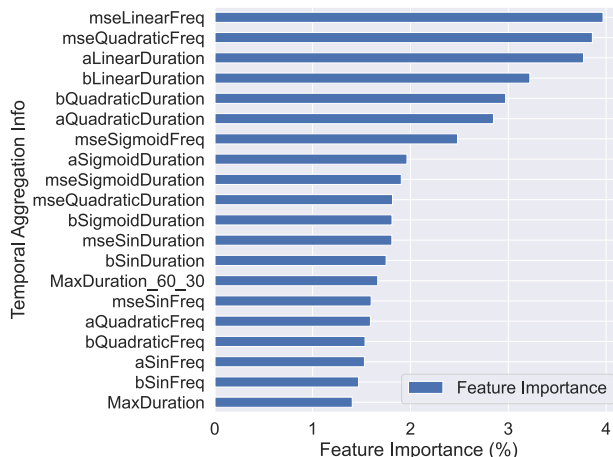


Figure 10. Feature importance of each temporal related feature using Random Forest gini measure.

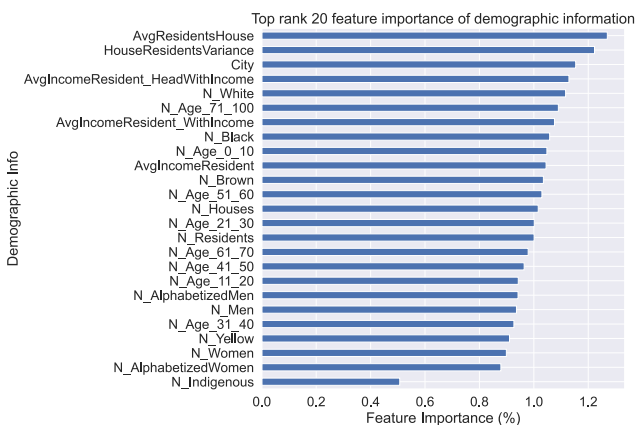


Figure 9. Feature importance of each installed app using Random Forest gini measure.

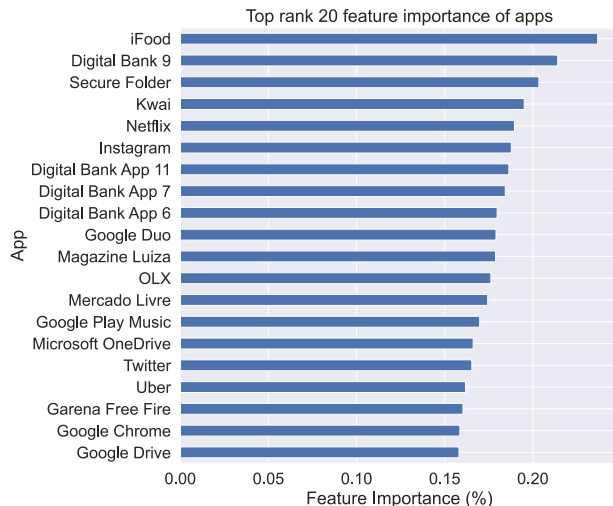


Figure 11. Feature importance of each mobile app using Random Forest gini measure.

poral ones. This shows that temporal attributes computed by the CANCEL feature engineering method are more important for churn prediction. Even so, demographic information is also relevant in this scenario, as shown in Figure 9.

The top five important features are dominated by temporal aggregation attributes. These attributes, which are derived from the CANCEL method, have shown significant importance with values ranging from 2% to 4% (see Figure 10). This indicates that the temporal behavior of users, as captured by these attributes, plays an important role in predicting churn. The most important feature is the error from the fitting process of the linear function from Equation 1, which can be interpreted only if the user’s behavior increases or decreases in the period. However, other functions such as quadratic and sigmoid also provide similar information for the random forest model and were also important. In general, the scalars describing the sequential behavior of the user’s duration were the most important attributes for churn prediction, while the scalars describing the error of the fitting for the access frequency were also important since they provided less valuable data in this context compared to the user’s access duration.

In addition to the temporal attributes, demographic features such as city, variance of residents per house, average

income per resident and average residents with income also contribute to the model, albeit with less importance compared to the temporal attributes. The importance values are also relatively high, suggesting that while these demographic factors do contribute to churn prediction, they are less influential than the temporal behavior of users.

The feature importance of installed apps, as shown in Figure 11 including Digital Bank App 9, iFood, Messenger, Digital Bank App 2, and Google Chrome, are relatively low, ranging from 0.15% to 0.20%. This suggests that while the presence of these apps does have some influence on churn prediction, their impact is less compared to the temporal and demographic attributes.

4.2.4 CANCEL’s generalization

To evaluate the generalizability of the CANCEL methodology across different contexts or datasets, the publicly available Banking Transactions dataset from Kaggle was se-

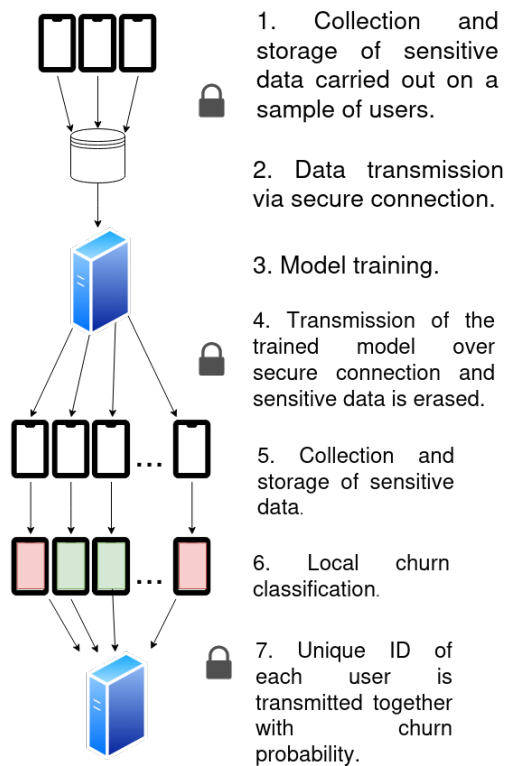


Figure 12. Process for churn classification

lected⁶. The dataset comprises over two million rows of transactional data dating back to 2005.

Given the focus on privacy-preserving churn prediction, only the transaction date and bank account ID were utilized from the dataset. Transactions were treated as equivalent to customer interactions with the bank, and indicates active usage of the bank’s services. Data were aggregated monthly and chronologically ordered to develop features for the CANCEL model. The baseline model was constructed similarly to the approach described in Section 4.2, utilizing only the temporal count of accesses, as each access is a binary indicator and other statistical measures (e.g., mean, maximum, standard deviation) would not provide additional insights beyond the monthly access counts.

In this specific dataset, a 60-day period without any banking interactions suffices to classify a user as a churner. A seven-month segment from February 17, 2010, to September 15, 2010, was used for feature generation, and the subsequent period up to November 17, 2010, was used to determine churn status. Users with no transactions recorded during this latter period were labeled as churners. This time-span was selected to guarantee a sufficiently large and relatively balanced dataset, resulting in 9,758 churners and 11,097 non-churners. The evaluation employed Random Forest with fixed hyperparameters with 10-times repeated 5-fold cross-validation method, yielding a median F1-score of 77.2% for the CANCEL model, against 67.8% for the baseline model.

5 Edge computing architecture

This section describes the integration of the best-found model (RandomForestCANCEL) with a mobile application for churn classification on the user’s device. Figure 12 shows the necessary steps for churn classification. In steps 1 and 2, the collection and transmission of sensitive data from a sample of users necessary for model training are performed. In step 3, the model training is done in a central server. In step 4, sensitive data can be deleted after the model, already trained, is sent back to the smartphones to perform the churn prediction. With the trained model, in step 5 the necessary data for classification are collected to be able to integrate it into a mobile application in step 6 to take advantage of the benefits of edge computing. In a traditional approach, all users would send their data to a server, which would be responsible for making the churn prediction. With this, the users’ data, often sensitive, would be exposed to attacks and privacy invasion if the servers were compromised. With edge computing, the execution of the model is performed locally on the user’s device, and only the result of the prediction is sent to the server.

The proposed method enhances privacy, security and cloud computing resources by explicitly collecting only a sample of user data necessary for training the model, rather than the entirety of user data. This sample is transmitted to a central server for processing and, once the model is trained, the data is deleted to reduce the risk of data leakage and privacy loss. The resulting model, which is compact in size, is then distributed back to users. For ongoing improvements, daily updates involve minimal data transmission. This approach not only significantly reduces bandwidth usage and server storage demands compared to traditional methods but also incorporates online learning techniques to continuously refine the model without retaining large volumes of data. To evaluate this proposal, a demonstration mobile Android application was developed that incorporates the model responsible for churn prediction. The use cases of this application are illustrated in Figure 13. First, the application performs data collection while simulating an interaction with the services offered, and as the data are collected, the churn prediction is made. After this, the churn classification is sent to a server along with the date when this classification was made and a unique user identification.

The application, when run for the first time, requests the user’s permission to collect GPS sensor data. If the authorization is denied, this information is filled with null values, which does not prevent classification by the model, although it may affect the quality of the result. The other data necessary for local classification, listed in Section 3, are collected together with the location. The location coordinates are processed with a local database⁷ to perform reverse geocoding and the census tract associated with the user.

In Figure 12, in step 5 the local data stored on the Android’s app specific storage, which is accessible only to the owning application, it is possible to make the churn prediction. The training and evaluation of the models described

⁶<https://www.kaggle.com/datasets/jackwalker12/banking-transactions>

⁷<https://github.com/AReallyGoodName/OfflineReverseGeocode>



Figure 13. Flow between Application Screens

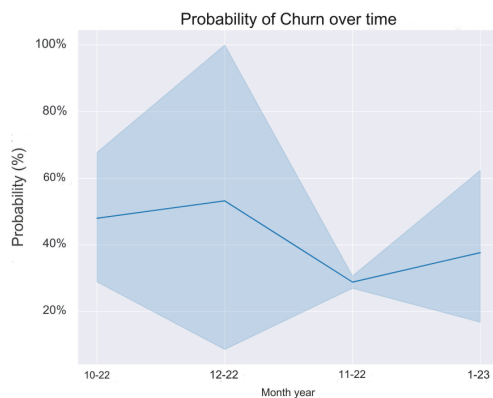


Figure 14. Graph displayed on the Dashboard with the mean and standard deviation of the churn probability of all users over the year.

in Section 4.1 used the *Sklearn*⁸ library in Python, so it was necessary to find a way to export the models trained on the central server and import them on the mobile device. For this, the *Sklearn-ONNX*⁹ and *ONNX*¹⁰ tools were used to, respectively, export the model already trained by *Sklearn*, and import it and use it for predictions in the Android application.

In Figure 12, in steps 6 and 7 the churn prediction is made locally once a month per user and sent to a central server through an encrypted HTTP connection using Transport Layer Security (TLS) along with a unique identifier provided by the user's mobile application. To display these data, a Web application was created that contains an API and a Flask dashboard. The API receives the churn prediction information from the smartphones and presents the graph for display on the dashboard as shown in Figure 14. Although a demonstration application is used, deployment in a real company's application is trivial, through the integration of the code developed in this work as a library in the company's project. In addition, with the prediction model implemented at the edge, there can be a data transmission saving of up to 20 KiBytes/user for each prediction, considering the application implemented in large companies with millions of users, this saving is significant.

6 Conclusions and future work

In this work, a feature engineering method named CANCEL and a privacy-preserving churn prediction architecture at the

edge are presented, enhancing security and reducing the need for computational resources. Five traditional machine learning algorithms were evaluated using naïve temporal metrics and CANCEL, and it was shown that the use of the proposed technique, resulted in better performance in several metrics. Considering the simplicity and generalized characteristics of CANCEL, it could be used in more supervised problems other than churn prediction.

The best model, which uses the Random Forest algorithm with the CANCEL method, was integrated into a test application so that the prediction is made at the edge, preventing sensitive data from being sent to a server. In this case, only the prediction result is sent, which reduces the need for transmission capacity, central processing, and storage, in addition to protecting user privacy.

In the future, it is intended to implement the retraining of the model on the mobile application with the data collected on the user's device. It is also expected that the training, currently done centrally, will be implemented with the federated strategy, further protecting the privacy of users. Also, the proposed CANCEL method could be investigated on other contexts than churn prediction.

Declarations

Acknowledgements

The authors express their gratitude to the company Cinnecta, the Secretariat for Professional and Technological Education of the Ministry of Education (SETEC), CAPES, and FAPEMIG, for the provision of data and funding.

Authors' Contributions

Gabriel T. Coimbra: Conceptualization, Methodology, Implementation, Validation, Writing.

Victor Hugo R. Santos: Methodology, Implementation, Validation, Writing.

Pedro A. Maia: Methodology, Implementation, Validation, Writing.

Leticia O. Silva: Methodology, Implementation, Validation, Writing.

Rayanne P. Souza: Methodology, Implementation, Validation.

Fabício A. Silva: Supervision, Conceptualization, Methodology, Writing, Review.

Thais R. M. Braga Silva: Conceptualization, Methodology, Writing, Review.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

The datasets are private and are not available.

References

Bertens, P., Guitart, A., and Periañez, Á. (2017). Games and big data: A scalable multi-dimensional churn pre-

⁸<https://scikit-learn.org>

⁹<https://onnx.ai/sklearn-onnx/>

¹⁰<https://onnx.ai/>

- diction model. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 33–36. DOI: 10.1109/CIG.2017.8080412.
- Bharathi S, V., Pramod, D., and Raman, R. (2022). An ensemble model for predicting retail banking churn in the youth segment of customers. *Data*, 7(5). DOI: 10.3390/data7050061.
- Bonawitz, K., Ivanov, V., Kreuter, B., Marcedone, A., McMahan, H. B., Patel, S., Ramage, D., Segal, A., and Seth, K. (2017). Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS '17*, page 1175–1191, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3133956.3133982.
- Capanema, C. G. S., Silva, F. A., Silva, T. R. B., and Loureiro, A. A. (2021). Dcluster: Geospatial analytics with poi identification. *Journal of Information and Data Management*, 12(2). DOI: 10.5753/jidm.2021.1952.
- Coimbra, G. T., Santos, V. H. R., Maia, P. A., Silva, L. O., Souza, R. P., Silva, F. A., and Silva, T. R. B. (2023). Previsão de churn na borda: uma solução com atributos temporais e preservação de privacidade. In *Anais do XV Simpósio Brasileiro de Computação Ubíqua e Pervasiva*, pages 121–130. SBC. Available at: https://www.researchgate.net/publication/372947246_Previsao_de_churn_na_borda_uma_solucao_com_atributos_temporais_e_preservacao_de_privacidade.
- Forbes Insights (2011). Bringing 20/20 foresight to marketing. Available at: <https://www.iciworld.com/articles/forbes-bringing-foresight-to-marketing%5B1%5D.pdf>.
- Fox, G., van der Werff, L., Rosati, P., Takako Endo, P., and Lynn, T. (2022). Examining the determinants of acceptance and use of mobile contact tracing applications in brazil: An extended privacy calculus perspective. *Journal of the Association for Information Science and Technology*, 73(7):944–967. DOI: 10.1002/asi.24602.
- Gupta, S., Lehmann, D. R., and Stuart, J. A. (2004). Valuing customers. *Journal of marketing research*, 41(1):7–18. DOI: 10.1509/jmkr.41.1.7.25084.
- Haddadi, S. J., Mohammadi, M. O., Bahrami, M., Khoeini, E., Beygi, M., and Khoshkar, M. H. (2022). Customer churn prediction in the iranian banking sector. In *2022 International Conference on Applied Artificial Intelligence (ICAPAI)*, pages 1–6. DOI: 10.1109/ICAPAI55158.2022.9801574.
- Jagad, C., Jain, C., Thakore, D., Naik, O., and Sawant, V. (2023). *Federated Machine Learning-Based Bank Customer Churn Prediction*, chapter 6. CRC Press. DOI: 10.1201/9781003390220-6.
- Jain, H., Khunteta, A., and Srivastava, S. (2020). Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science*, 167:101–112. DOI: 10.1016/j.procs.2020.03.187.
- Kavyarshitha, Y., Sandhya, V., and Deepika, M. (2022). Churn prediction in banking using ml with ann. In *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1191–1198. DOI: 10.1109/ICICCS53718.2022.9788456.
- Kilimci, Z. H. (2022). The effectiveness of homogeneous classifier ensembles on customer churn prediction in banking, insurance, and telecommunication sectors. *International Journal of Computational and Experimental Science and Engineering*, 8(3):77–84. DOI: 10.22399/ijcesen.1163929.
- Li, J., Lu, H., Wang, C., Ma, W., Zhang, M., Zhao, X., Qi, W., Liu, Y., and Ma, S. (2021). A difficulty-aware framework for churn prediction and intervention in games. *KDD '21*, page 943–952, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3447548.3467277.
- Lin, J., Keogh, E., Lonardi, S., and Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11. DOI: 10.1145/882082.882086.
- Maia, W., Silva, F., and Silva, T. (2020). Um estudo sobre a relação entre smartphones e dados demográficos. In *Anais do IV Workshop de Computação Urbana*, pages 302–315, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/courb.2020.12371.
- Mammen, P. M. (2021). Federated learning: Opportunities and challenges. *arXiv preprint arXiv:2101.05428*. DOI: 10.48550/arXiv.2101.05428.
- Milošević, M., Živić, N., and Andjelković, I. (2017). Early churn prediction with personalized targeting in mobile social games. *Expert Systems with Applications*, 83:326–332. DOI: 10.1016/j.eswa.2017.04.056.
- Moré, J. J. (2006). The levenberg-marquardt algorithm: implementation and theory. In *Numerical Analysis: Proceedings of the Biennial Conference Held at Dundee, June 28–July 1, 1977*, pages 105–116. Springer. DOI: 10.1007/BFb0067700.
- Rahman, M. and Kumar, V. (2020). Machine learning based customer churn prediction in banking. In *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1196–1201. DOI: 10.1109/ICECA49313.2020.9297529.
- Rajamohamed, R. and Manokaran, J. (2018). Improved credit card churn prediction based on rough clustering and supervised learning techniques. *Cluster Computing*, 21(1):65–77. DOI: 10.1007/s10586-017-0933-1.
- Schlackl, F., Link, N., and Hoehle, H. (2022). Antecedents and consequences of data breaches: A systematic review. *Information & Management*, 59(4):103638. DOI: 10.1016/j.im.2022.103638.
- Shokri, R. and Shmatikov, V. (2015). Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, CCS '15*, page 1310–1321, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/2810103.2813687.
- Tan, F., Wei, Z., He, J., Wu, X., Peng, B., Liu, H., and Yan, Z. (2018). A blended deep learning approach for predicting user intended actions. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 487–496. DOI:

- 10.1109/ICDM.2018.00064.
- Ueda, K. and Yamashita, N. (2010). On a global complexity bound of the levenberg-marquardt method. *Journal of optimization theory and applications*, 147:443–453. DOI: 10.1007/s10957-010-9731-0.
- Verhelst, T. (2018). Churn prediction and causal analysis on telecom customer data. Available at: https://theoverhelst.com/assets/documents/theo_verhelst_master_thesis.pdf.
- Wang, A. X., Chukova, S. S., and Nguyen, B. P. (2023). Data-centric ai to improve churn prediction with synthetic data. In *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, pages 409–413. DOI: 10.1109/ICCCR56747.2023.10194217.
- Xiong, A., You, Y., and Long, L. (2019). L-rbf: A customer churn prediction model based on lasso + rbf. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 621–626. DOI: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00121.
- Yang, C., Shi, X., Jie, L., and Han, J. (2018). I know you’ll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’18*, page 914–922, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3219819.3219821.
- Zaky, A., Ouf, S., and Roushdy, M. (2022). Predicting banking customer churn based on artificial neural network. In *2022 5th International Conference on Computing and Informatics (ICCI)*, pages 132–139. DOI: 10.1109/ICCI54321.2022.9756072.
- Zhou, J., Yan, J.-f., Yang, L., Wang, M., and Xia, P. (2019). Customer churn prediction model based on lstm and cnn in music streaming. *DEStech Transactions on Engineering and Technology Research*, 5. Available at: https://www.researchgate.net/publication/333252132_Customer_Churn_Prediction_Model_Based_on_LSTM_and_CNN_in_Music_Streaming.