

Previsão de *churn* na borda: uma solução com atributos temporais e preservação de privacidade

Gabriel T. Coimbra¹, Victor Hugo R. Santos¹, Pedro A. Maia¹,
Letícia O. Silva¹, Rayanne P. Souza¹, Fabrício A. Silva¹, Thais R. M. Braga Silva¹

¹Laboratório de Inteligência em Sistemas Pervasivos e Distribuídos (NESPED-Lab)
Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa (UFV) - Florestal - MG - Brasil

{gabriel.coimbra, victor.h.santos, pedro.maia,
leticia.silval, rayanne.souza, fabricio.asilva, thais.braga}@ufv.br

Resumo. *Este trabalho propõe uma solução para prever o churn (abandono de usuários) com preservação da privacidade usando a computação de ponta. Com a crescente popularidade dos smartphones, os usuários estão se tornando mais exigentes em relação ao uso de aplicativos móveis. As instalações e desinstalações de aplicativo são eventos frequentes e a facilidade de desinstalação pode facilitar a rotatividade, que é o abandono do cliente. As empresas buscam minimizar a rotatividade, pois o custo de aquisição de novos clientes é muito maior do que o de retenção dos atuais. Para prever o possível abandono, as organizações estão adotando cada vez mais técnicas de inteligência artificial (IA). No entanto, os clientes estão cada vez mais preocupados com a privacidade de seus dados. Nesse contexto, propomos uma técnica chamada CANCEL, que cria atributos com base no comportamento temporal dos usuários, com computação de ponta para prever o churn localmente, sem transmitir os dados dos usuários. O artigo apresenta a avaliação do CANCEL em comparação com as soluções de base, o desenvolvimento de um aplicativo móvel integrado ao método proposto e implantado como uma solução de computação de borda.*

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.*

1. Introdução

Com a popularização dos *smartphones*, os usuários estão se tornando cada vez mais exigentes em relação ao uso de aplicativos móveis. Sendo assim, os processos de instalar e desinstalar um aplicativo são rotinas frequentes e de fácil execução para quem usa um *smartphone*. Por um lado, a facilidade de instalação de um aplicativo é um ponto positivo para as empresas que constroem software, bastando a solução proposta possuir valor agregado para se popularizar. Por outro lado, a facilidade de desinstalação é um grande empecilho, facilitando o abandono do cliente, conhecido pelo termo em inglês *churn*.

Segundo [Insights 2011], o meio corporativo busca a retenção de clientes como uma prioridade devido ao alto custo de aquisição de novos clientes em comparação com o custo de retenção dos atuais. Sendo assim, muitas organizações buscam entender o comportamento dos clientes de modo a prever o possível abandono. Entender como seus clientes se comportam é uma árdua tarefa para as empresas, principalmente para as gigantes do mercado, que têm milhares ou até mesmo milhões de clientes. Para resolver este problema, técnicas de Inteligência Artificial (IA), mais especificamente de Aprendizado de Máquina, têm sido usadas para tentar prever o *churn* de usuários.

Enquanto as empresas buscam soluções orientadas a dados para prever o abandono de clientes, estes estão cada vez mais preocupados com a privacidade dos seus dados [Fox et al. 2022]. Com isso, nos últimos anos têm acontecido avanços em técnicas que diminuam o risco de os usuários terem seus dados, muitas vezes pessoais e sensíveis, vazados. Essa preocupação com leis que estão sendo elaboradas ao redor do mundo, como a Lei Geral de Proteção de Dados (LGPD) no Brasil.

Neste trabalho, é proposta uma solução para previsão de *churn* com preservação de privacidade por meio do uso de computação na borda. Primeiramente, é proposta uma técnica chamada CANCEL (*Curve-Aware Churn Prediction Models*) para criar atributos com base no comportamento dos usuários ao longo do tempo. Essa técnica é avaliada em comparação com soluções base, tendo alcançado resultados superiores em termos de precisão e revocação. Em seguida, o melhor modelo que utiliza o CANCEL é integrado a um aplicativo móvel para que a previsão de *churn* seja feita de forma local, evitando que dados dos usuários sejam transmitidos. Uma aplicação Web foi desenvolvida para apresentar os resultados das previsões executadas localmente nos dispositivos. Em resumo, este trabalho apresenta duas contribuições principais: uma técnica para consolidar o comportamento do usuário ao longo do tempo e uma solução integrada na borda para prever *churn* com preservação de privacidade.

O restante deste texto está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve os dados utilizados para o presente trabalho. A Seção 4 trata da proposta e resultados do método de engenharia de atributos proposto neste trabalho. A Seção 5 descreve a proposta de integração de modelos de aprendizado utilizando computação na borda. Por fim, a Seção 6 discorre sobre as considerações finais.

2. Trabalhos relacionados

O contexto mais comum para soluções automatizadas de previsão de *churn* é quando a taxa de abandono de clientes é impactante para o negócio, como é o caso do setor de telecomunicações (telefonia, Internet e televisão)

[Jain et al. 2020, Verhelst 2018]. Este cenário também é presente em aplicativos de redes sociais [Yang et al. 2018], *streaming* de música [Zhou et al. 2019], jogos [Milošević et al. 2017], cursos online [Tan et al. 2018] e de empresas fornecedoras de cartão de crédito [Rajamohamed and Manokaran 2018]. Além desses setores, de forma similar ao presente trabalho, há estudos que investigam a previsão de *churn* no setor bancário [Haddadi et al. 2022, Rahman and Kumar 2020], porém usando dados específicos deste ambiente, como transações bancárias dos clientes, o que demanda a quebra de privacidade dos mesmos. O foco da previsão de *churn* nessas indústrias acontece, pois muitas vezes o custo financeiro para realizar campanhas de retenção e evitar o *churn* é significativamente menor que o custo de aquisição de novos clientes [Gupta et al. 2004].

Os estudos citados acima visam melhorar a precisão na previsão de *churn* por meio da avaliação de modelos de estado da arte [Jain et al. 2020, Yang et al. 2018, Zhou et al. 2019, Tan et al. 2018, Rajamohamed and Manokaran 2018], interpretação dos resultados dos modelos [Yang et al. 2018] ou inferência de relações causais [Verhelst 2018]. No entanto, não há enfoque em investigar maneiras de desenvolver e avaliar esses modelos em ambientes onde a privacidade dos dados é desejável.

Além do aspecto de previsão, outro requisito cada vez mais importante é a proteção aos dados sensíveis gerados pelo usuário [Schlackl et al. 2022]. Na maioria dos trabalhos encontrados, foram utilizados dados invasivos que necessitam de medidas extras de proteção ao serem transmitidos e armazenados. Além disso, é importante ressaltar que essas medidas variam dependendo da legislação do país ou estado como, por exemplo, a GDPR na Europa e a LGPD no Brasil. Neste contexto, o conceito de aprendizado federado [Bonawitz et al. 2017] tem surgido para melhorar a proteção aos dados e diminuir a necessidade de recursos computacionais (i.e., armazenamento e a transmissão dos dados) de grande escala em servidores centralizados. Porém, dois problemas surgem: primeiramente, é necessário um maior consumo de recursos computacionais dos dispositivos móveis dos usuários, o que nem sempre é aceito por eles ou possível devido a restrições dos dispositivos móveis. Além disso, o modelo treinado por aprendizado federado tem diversos desafios (i.e., envenenamento de dados ou modelo, acurácia reduzida e alto custo de agregação) se comparados com modelos treinados com todos os dados centralizados, já que os métodos de agregação ainda precisam evoluir [Mammen 2021].

Diferentemente dos trabalhos existentes, neste trabalho é proposta e avaliada uma abordagem de previsão de *churn* de aplicativos móveis utilizando algoritmos tradicionais de aprendizado de máquina em conjunto com técnicas de enriquecimento que permitem a inferência do comportamento do usuário ao longo do tempo. Além disso, a abordagem proposta permite aumentar a segurança e preservar a privacidade dos dados ao mesmo tempo em que é feita a previsão de *churn* por meio da computação na borda.

3. Os Dados

Neste trabalho, é utilizada uma base de dados fornecida por uma empresa parceira por meio de um acordo de confidencialidade, contendo registros de 33.817 usuários de um aplicativo de banco digital. Os dados disponíveis são: a data de instalação do aplicativo, os registros de uso do aplicativo do banco, registros de localizações do usuário ao se deslocar, a lista de outros aplicativos instalados no dispositivo do usuário e o modelo do dispositivo. A partir desses dados, foi feita neste trabalho uma engenharia de atributos,

explicada a seguir, para extrair métricas e enriquecer os dados.

Com os registros de uso e a data de instalação, foram calculados por quantos dias distintos o usuário acessou o aplicativo e quantos dias se passaram desde a instalação do aplicativo pela primeira vez. Utilizando os registros de localizações do usuário, o algoritmo para detecção de pontos de interesse de [Capanema and Silva 2021] foi utilizado para inferir a localização aproximada da sua casa. Foram utilizados dados do Censo IBGE de 2010 para enriquecer com informações demográficas dos setores censitários onde a casa está localizada. Essas informações demográficas fornecem dados sobre a faixa etária dos residentes da região, etnia, renda e sexo¹. Também foi utilizado o Crow-Mobi [Maia et al. 2020] para estimar o preço e o ano de lançamento do dispositivo.

Por fim, a partir dos registros de uso do aplicativo, foram extraídas informações sequenciais contendo, em ordem cronológica, a frequência de uso do aplicativo e a duração de cada uso. A decisão de extrair essas informações sequenciais foi fundamentada na possibilidade dessas informações representarem bem o comportamento do usuário com o aplicativo ao longo do tempo, fator considerado relevante para a previsão de *churn*, como será mostrado nos resultados.

Neste trabalho, foram rotulados como *churners* os usuários que não acessaram o aplicativo nos últimos dois meses. É importante notar que, uma vez que esses dois últimos meses são usados para a definição do objetivo da previsão de *churn*, eles foram descartados dos dados utilizados para treinamento. Com essa definição, dos 33.817 usuários, 1.493 são classificados como *churners*, o que resulta em uma taxa de *churn* de aproximadamente 4%. Esse desbalanceamento entre *churners* e não-*churners* traz uma dificuldade extra ao problema, que será tratada na solução.

Vale destacar que, apesar de os dados utilizados serem referentes a usuários de um banco digital, a solução proposta é flexível para ser adotada em diferentes cenários, bastando que o período de inatividade para indicação de *churn* seja ajustado.

4. Solução de Previsão de *churn*

O problema de previsão de *churn* em outras indústrias como, por exemplo, empresas de telecomunicações, tem sido amplamente estudado na literatura utilizando algoritmos tradicionais de aprendizado de máquina. No entanto, uma das principais limitações encontradas é a falta de um *baseline* que possa ser aplicado em diferentes cenários, já que cada problema possui suas próprias características e disponibilidade de dados. Assim, a principal tarefa em um problema de previsão de *churn* é a engenharia de atributos, responsável por identificar e preparar os atributos relevantes para o modelo preditivo. Com isso, na Seção 4.1 é proposto um método que permite que modelos tradicionais de aprendizado de máquina usem dados temporais. Na Seção 4.2 esse método é avaliado em comparação a outras técnicas no contexto de previsão de *churn*.

4.1. CANCEL: *Curve-Aware Churn Prediction Models*

Este trabalho propõe o CANCEL (*Curve-Aware churN prediction modELs*), um método a ser aplicado durante a engenharia de atributos para permitir que modelos tradicionais de aprendizado de máquina, que trabalham com dados estáticos, façam inferências

¹<https://censo2010.ibge.gov.br/sinopseporsetores/>

sobre comportamentos temporais. Para isso, foi utilizada uma técnica similar à de [Yang et al. 2018]. Porém, neste trabalho os valores calculados por CANCEL são utilizados em um contexto de aprendizado supervisionado (i.e., previsão de *churn*) e não para interpretação dos comportamentos dos usuários nos aplicativos.

O CANCEL é uma proposta generalista para facilmente permitir a inferência de informações sequenciais provenientes de séries temporais em algoritmos de aprendizado de máquina tradicionais já conhecidos na literatura. Em resumo, esse método consiste em encontrar parâmetros de funções sintéticas que minimizem a distância entre os dados reais e essas funções. Com isso, estes parâmetros resumem o comportamento das funções e podem ser utilizados diretamente em algoritmos de aprendizado de máquina para fornecer informações sobre as séries temporais.

Os comportamentos de usuários nos aplicativos podem ser variados: alguns podem acessar o aplicativo em intervalos pequenos e regulares, outros têm uma tendência de diminuir a quantidade de acessos ou a duração do tempo de sessão em cada acesso. Para converter o comportamento de acessos ao aplicativo para parâmetros estáticos, foram utilizadas duas funções sintéticas, a linear e a seno:

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

$$f_{seno}(x) = seno(\alpha_s \times x) + \beta_s \quad (2)$$

Seja $\hat{f}_v(x)$ o comportamento observado para a categoria de comportamento v (e.g., quantidade ou duração de acessos na janela de tempo) do usuário no espaço de tempo x , em que $x \in \mathbb{N}$ e $1 \leq x \leq N$, e N representa a quantidade de janelas de tempo (e.g., dias, semanas ou meses) de dados sequenciais coletados para este usuário. O objetivo agora é a minimização de ϵ da equação 3 utilizando algoritmos de mínimos quadrados não-lineares sobre os parâmetros das funções sintéticas (e.g., α_l , β_l , α_s e β_s para as funções linear e seno). Com isso, teremos os melhores parâmetros para as funções sintéticas que explicam o comportamento real. Além disso, na equação 3, $v \in \{\text{frequência}, \text{duração}\}$ e $c \in \{\text{seno}, \text{linear}\}$, considerando o escopo deste trabalho de previsão de *churn*.

$$\epsilon_v^c = \frac{\sum_{i=1}^N (\hat{f}_v(i) - f_c(i))^2}{N} \quad (3)$$

Além dos parâmetros das funções (no caso deste trabalho α_l , β_l , α_s e β_s), também são utilizados os erros quadráticos médios ϵ_v^c como um atributo durante o treinamento dos modelos supervisionados. Com isso, é possível resumir a tendência do usuário para um atributo específico em poucos parâmetros, o que diminui as chances de sobreajustamento. Os valores dos parâmetros das funções são utilizados pelo modelo para resumir o comportamento sequencial dos usuários, já ϵ permite ao modelo considerar a magnitude do erro. Com isso, é possível ao modelo atribuir um peso maior aos parâmetros quando a distância entre o comportamento do usuário e a função sintética é menor.

Além das funções seno e linear, outras poderiam ter sido utilizadas, como a função sigmoide, como é feito em [Yang et al. 2018]. No presente trabalho foi escolhida a função seno por ajustar melhor em dados onde o comportamento do usuário é periódico. Já a função linear representa um comportamento crescente, constante ou decrescente. A

função linear tem um propósito similar à função sigmoide em [Yang et al. 2018], porém é mais facilmente otimizada e interpretável quando utilizada em conjunto com a função seno. Ressalta-se que a combinação dos parâmetros otimizados obtidos das funções linear e seno permitem aos modelos codificar diversas situações como: comportamento muito periódico e pouco crescente, comportamento pouco periódico e muito decrescente, comportamento pouco periódico e estável. Com isso, os atributos gerados pelo CANCEL tendem a ser mais adequados para serem utilizados em modelos de aprendizado de máquina devido à simplicidade e consistência dos mesmos se comparado a técnicas mais complexas (e.g., técnicas de previsão de séries temporais, como o ARIMA).

Resumindo, ao utilizar o CANCEL, são incluídos os parâmetros otimizados e o erro para cada função (i.e., seno e linear) e por categoria de comportamento (i.e., frequência de acesso e duração de acesso). Portanto, no presente trabalho, ao utilizar o CANCEL o conjunto de dados da Seção 3, chamado de *baseline*, é incrementado com 12 (3 parâmetros \times 2 funções \times 2 categorias) novos atributos gerados a partir de dados sequenciais.

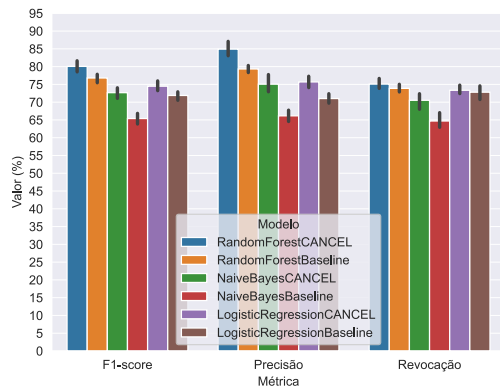
4.2. Resultados

Para avaliação do CANCEL, foram utilizados três algoritmos: Floresta Aleatória, Regressão Logística e *Näive Bayes*. Cada um desses algoritmos foi adotado em duas implementações: *baseline*, em que apenas os dados originais estáticos foram usados, e CANCEL, em que os dados sequenciais transformados pelo método CANCEL descrito na Seção 4.1 foram utilizados. Com isso, será possível avaliar o impacto do CANCEL em diferentes perspectivas.

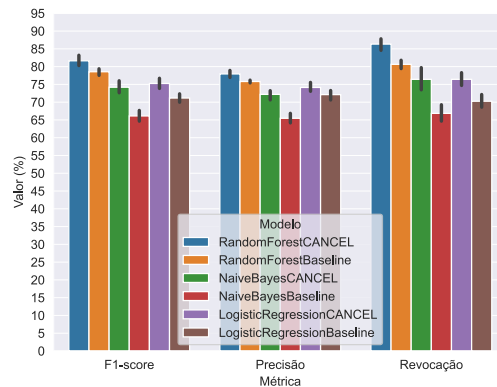
Dos dados de treinamento, para substituir valores nulos nos dados estáticos é feita uma imputação simples usando a estratégia de substituição pelo valor mais frequente, pois essa técnica funciona com dados numéricos e categóricos. Não é feita imputação nos dados sequenciais; caso o usuário tenha menos de 3 semanas de dados os valores sequenciais são considerados como nulos nos modelos devido a restrições impostas pelo algoritmo de otimização LM, usado pelo CANCEL. Os valores categóricos, como nome da cidade, nome do estado, dentre outros, são codificados em números distintos. Além disso, para reduzir o custo computacional durante a previsão na borda descrita na Seção 5, foram considerados apenas os 100 aplicativos mais comuns. Para evitar problemas de balanceamento, foi feito um *downsampling* aleatório da classe de usuários não *churners*.

A Figura 1 mostra a média e desvio-padrão das métricas obtidas com a validação cruzada *5-fold*. Em cada rodada, os mesmos hiper-parâmetros utilizados nos modelos com as informações fornecidas pelo CANCEL são utilizados nos modelos da *baseline*. O *F1-score* apresentado é calculado utilizando a Precisão e Revocação de somente uma classe de usuários. Com isso, a única diferença entre ambos os modelos gerados por um mesmo algoritmo são os atributos usados, ressaltando que as soluções que utilizam CANCEL acrescentam os atributos novos com base nos dados sequenciais. É importante destacar que, para uma comparação justa, os modelos *baseline* incluem informações calculadas sobre os dados sequenciais como: contagem de valores, mínimo, máximo, média e desvio-padrão. Esses valores são comumente utilizados em trabalhos onde existem dados sequenciais, mas de forma estática [Milošević et al. 2017, Bertens et al. 2017].

Como pode ser visto na Figura 1, a utilização das informações fornecidas pelo



(a) Métricas - usuários *churners*



(b) Métricas - usuários não *churners*

Figura 1. Resultados da comparação entre modelos CANCEL e *baselines*.

método CANCEL permitem melhorias no desempenho dos modelos (veja barras azuis, verdes e roxas), tanto para a classe de *churners* (Figura 1a) quanto para os não *churners* (Figura 1b). Devido às melhores métricas utilizando CANCEL, o modelo de Floresta Aleatória (i.e., *RandomForestCANCEL*, representado nas barras azuis na Figura) foi escolhido para ser implantado na borda na Seção 5.

5. Arquitetura de computação na borda

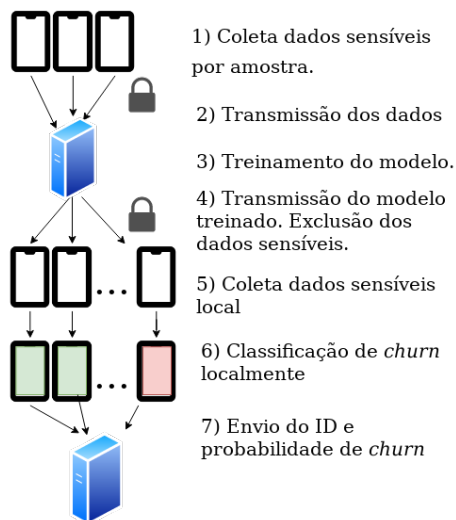


Figura 2. Processo para classificação de *churn*

Nesta seção será descrito como é feita a integração do melhor modelo encontrado (*RandomForestCANCEL*) com um aplicativo móvel para a classificação de *churn* no dispositivo do usuário. A Figura 2 mostra as etapas necessárias para a classificação de *churn*. Nas etapas 1 e 2, são feitas a coleta e o envio dos dados sensíveis de uma amostra de usuários necessários para o treinamento do modelo, na etapa 3, em um servidor central. Na etapa 4, dados sensíveis podem ser apagados após o modelo, já treinado, ser enviado de volta aos *smartphones* para realizar a previsão de *churn*. Com o modelo treinado, na etapa 5 são coletados os dados necessários para a classificação para ser possível integrá-lo em um aplicativo móvel na etapa 6 para aproveitar os benefícios da computação na

borda. Em uma abordagem tradicional, todos os usuários enviariam os seus dados para um servidor, que seria responsável por realizar a previsão de *churn*. Com isso, os dados dos usuários, muitas vezes sensíveis, estariam expostos a ataques e invasão de privacidade caso os servidores sejam comprometidos. Com a computação na borda, a execução do modelo é realizada localmente no dispositivo do usuário, e apenas o resultado da previsão é enviado ao servidor.

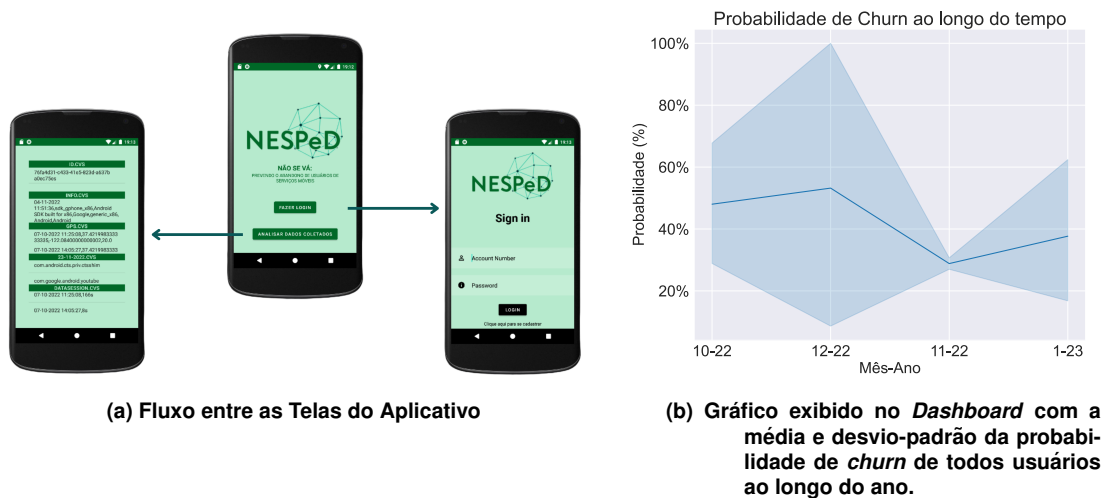


Figura 3. *Dashboard* e aplicativo desenvolvidos para teste em produção.

Para avaliar essa proposta, foi desenvolvido um aplicativo de demonstração que incorpora o modelo responsável pela previsão de *churn*. Os casos de uso deste aplicativo são ilustrados na Figura 3a. Em primeiro lugar, o aplicativo realiza a coleta de dados enquanto simula uma interação com os serviços oferecidos, e à medida que os dados vão sendo coletados, é feita a previsão de *churn*. Após isso, é enviado para um servidor a classificação de *churn* em conjunto com a data em que essa classificação foi feita e um ID único do usuário.

O aplicativo, ao ser executado pela primeira vez, requisita do usuário permissão para coletar dados do sensor de GPS. Caso a autorização seja negada, essa informação é preenchida com valores nulos, o que não impede a classificação pelo modelo, apesar de poder afetar a qualidade do resultado. Os outros dados necessários para classificação local, listados na Seção 3, são coletados em conjunto com a localização. É importante ressaltar que, nesta implementação, a casa do usuário foi estimada utilizando as coordenadas mais frequentes no histórico de latitude e longitude. Essas coordenadas utilizadas são processadas com um banco de dados local² para realizar a geocodificação reversa e o setor censitário associado ao usuário.

Com os dados locais preparados, é possível fazer a previsão de *churn*. O treinamento e avaliação dos modelos descritos na Seção 4.1 utilizou a biblioteca *Sklearn*³ em Python, por isso foi necessário encontrar um meio de exportar os modelos treinados no servidor central e importá-los no dispositivo móvel. Para isso, foram utilizadas as ferramentas *Sklearn-ONNX*⁴ e *ONNX*⁵ para, respectivamente, exportar o modelo já treinado pelo *Sklearn*, e importá-lo e utilizá-lo para as previsões no aplicativo em Java.

A previsão de *churn* é feita localmente uma vez ao mês por usuário e enviada a um servidor central juntamente com um identificador único disponibilizado pelo aplicativo móvel do usuário. Para mostrar esses dados, foi criada uma aplicação Web que contém

²<https://github.com/AReallyGoodName/OfflineReverseGeocode>

³<https://scikit-learn.org>

⁴<https://onnx.ai/sklearn-onnx/>

⁵<https://onnx.ai/>

uma API e um *dashboard* em *Flask*. A API recebe as informações de previsão de *churn* dos *smartphones* e apresenta o gráfico para a exibição no *dashboard* como mostrado na Figura 3b. Apesar de ser utilizado um aplicativo de demonstração, a implantação em um aplicativo de uma empresa real é trivial, por meio da integração do código desenvolvido neste trabalho como uma biblioteca no projeto da empresa. Além disso, com o modelo de previsão implantado na borda, pode acontecer uma economia de transmissão de dados de até 20 KiBytes/usuário para cada previsão, considerando a aplicação implementada em grandes empresas com milhões de usuários, essa economia é significativa.

6. Conclusões e Trabalhos Futuros

Neste trabalho é apresentado um método de engenharia de atributos e uma arquitetura de previsão de *churn* na borda com preservação de privacidade, aumento da segurança e diminuição da necessidade de recursos computacionais. Foram avaliados três algoritmos, e mostrado que o uso da técnica proposta, chamada CANCEL, resultou em melhores valores para precisão, revocação e f1-score.

O melhor modelo, que utiliza o algoritmo de Floresta Aleatória com o método CANCEL, foi integrado a um aplicativo de teste para que a previsão seja feita na borda, evitando que dados sensíveis sejam enviados a um servidor. Neste caso, apenas o resultado da previsão é enviado, o que reduz a necessidade de capacidade de transmissão, processamento central e armazenamento, além da proteção da privacidade do usuário.

Futuramente, pretende-se implementar o retreinamento do modelo no *smartphone* com os dados coletados no aparelho do usuário. Também é esperado que o treinamento, atualmente feito de forma centralizada, seja implementado com a estratégia federada, protegendo ainda mais a privacidade dos usuários.

Agradecimentos

Os autores agradecem à empresa Cinnecta, à Secretaria de Educação Profissional e Tecnológica do Ministério da Educação (SETEC) e à CAPES, pelo fornecimento de dados e financiamento.

Referências

- Bertens, P., Guitart, A., and Periañez, Á. (2017). Games and big data: A scalable multi-dimensional churn prediction model. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 33–36.
- 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.
- Capanema, C. and Silva, F. (2021). Detecção de pontos de interesse e predição de próximo local de visita de usuários móveis com base em dados esparsos. In *Anais Estendidos do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 129–136, Porto Alegre, RS, Brasil. SBC.

- 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.
- Gupta, S., Lehmann, D. R., and Stuart, J. A. (2004). Valuing customers. *Journal of marketing research*, 41(1):7–18.
- 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.
- Insights, F. (2011). Bringing 20/20 foresight to marketing. *May* (http://www.forbes.com/forbesinsights/bringing_foresight_to_marketing/).
- Jain, H., Khunteta, A., and Srivastava, S. (2020). Churn prediction in telecommunication using logistic regression and logit boost. *Procedia Computer Science*, 167:101–112.
- 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.
- Mammen, P. M. (2021). Federated learning: Opportunities and challenges. *arXiv preprint 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.
- 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.
- 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.
- Schlackl, F., Link, N., and Hoehle, H. (2022). Antecedents and consequences of data breaches: A systematic review. *Information & Management*, 59(4):103638.
- 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.
- Verhelst, T. (2018). Churn prediction and causal analysis on telecom customer data.
- 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.
- 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.