A Temporal Approach to Customer Churn Prediction: A Case Study for Financial Services

Marcus Almeida[1], **Mariana Mota**[1], **Wellington Souza**[2], **Marcos Nicolau**[2], **Eduardo Luz**[1], **Gladston Moreira**[1]

[1]Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
35400-000 – Ouro Preto – MG – Brasil

[2]Gerencianet S.A
Ouro Preto – MG – Brasil.

{marcus.daniel, mariana.regina}@aluno.ufop.edu.br

{wellington.souza, marcos.nicolau}@gerencianet.com.br

{eduluz, gladston}@ufop.edu.br

Abstract. *Customer churn prediction models aim to detect customers with a high probability of canceling the contract with the company, based on the use of the offered services. We propose a temporal approach to the labeling stage, based on the frequency reduction of the use of services, through each customer's behavior. We also propose a temporal neural network architecture for the task. The approach was evaluated on a real dataset provided by a Brazilian company in the financial sector. The temporal convolutional neural network achieved an accuracy of* 82.63%*, a sensitivity of* 61.5%*, and a precision of* 41.58%*, outperforming other traditional classifiers (XG-Boost and Random Forest).*

Resumo. *Modelos de previsão de desligamento de clientes visam detectar clientes com alta probabilidade de cancelamento do contrato, com base no uso dos serviços oferecidos. Propomos uma abordagem temporal para a etapa de rotulagem, baseada na redução da frequência de uso dos serviços, por meio do comportamento de cada cliente. Também propomos uma arquitetura de rede neural temporal para a tarefa. A abordagem foi avaliada em um conjunto de dados reais, fornecido por uma empresa brasileira do setor financeiro. A rede neural convolucional temporal alcançou uma acurácia de* $82,63\%$*, uma sensibilidade de* $61,5\%$ *e uma precisão de* $41,58\%$*, superando outros classificadores tradicionais (XG-Boost e Floresta Aleatória).*

1. Introdução

Diminuir a rotatividade de clientes no cenário atual de alta competitividade, tornou-se um desafio para o gerenciamento de clientes nas empresas, uma vez que conquistar novos clientes pode ter um alto custo. Nesse contexto, as empresas precisam adotar estratégias na identificação de clientes propensos a migrar para empresas concorrentes. Previsão de desligamento de clientes (PDC) pode ser de grande importância, visto que uma vez detectado quais clientes pretendem mudar o provedor de serviço, viabiliza-se o desenvolvimento de estratégias comerciais para antecipar e mitigar a evasão [Devriendt et al. 2021, Hadden et al. 2007]. O desligamento de clientes, traduzido aqui

do termo em inglês *Customer Churn*, é definido como a perda ou saída de clientes da base de clientes de uma empresa.

Com o avanço recente dos bancos de dados em nuvem, as companhias têm maior capacidade de gerenciamento de informações históricas de seus clientes. Informações que podem ser usadas para criar modelos estatísticos e de aprendizado de máquina direcionados à gestão de desligamento de clientes [Gregory 2018, Caigny et al. 2018, Caigny et al. 2020].

Em [Amin et al. 2014], os autores definem três tipos de desligamento: (i) voluntário, quando os clientes rompem com a empresa e contratam o serviço do concorrente, (ii) não voluntário, quando é a empresa quem rompe o serviço com o cliente e (iii) desligamentos silenciosos, que se refere aos clientes que diminuem a utilização do serviços e podem vir a ser um desligante em futuro próximo. Os tipos (i) e (ii) têm o taxa de desligamento facilmente calculada, o desafio está em predizer os clientes do tipo (iii). Tais clientes precisam ser identificados antes de apresentar um comportamento explícito de desengajamento para que sejam direcionados para uma campanha de retenção.

Neste trabalho, é apresentado um estudo de caso em um conjunto de dados reais, fornecido por uma empresa brasileira do setor financeiro. Uma abordagem temporal para previsão de desligamento de clientes do tipo silencioso (onde não temos informações exatas e completas para rotular dados), é proposta uma metodologia para a rotulagem da base de dados por meio de informações de recência e frequência de uso dos serviços. Além disso, estamos propondo uma modelagem que leva em consideração a questão temporal (sequencial) e usamos um modelo baseado em rede neural convolucional temporal (*Temporal Convolutional Neural Networks* - TCNN) para avaliação. Os experimentos mostraram que o modelo TCNN avaliado, alcançou uma acurácia de $82,63\%$, uma sensibilidade de $61,5\%$ e uma precisão de $41,58\%$, superando outros classificadores tradicionais.

2. Trabalhos relacionados

Na literatura, várias técnicas tradicionais de classificação, tais como regressão logística [Neslin et al. 2006], Árvores de Decisão [Lima et al. 2009], Máquinas de Vetor de Suporte [Chen et al. 2012], métodos *ensemble* [Coussement and De Bock 2013], têm sido usados para o problema de predição de desligamento de clientes. Uma consistente revisão da literatura sobre o problema de previsão de desligamento de clientes é apresentada em [Verbeke et al. 2011].

[Ullah et al. 2019] propõe um modelo baseado em Floresta Aleatória combinada com a seleção de atributos, usando ganho de informação e filtro de classificação de atributos correlacionados. Dois conjuntos de dados são usados neste trabalho, ambos de provedores de serviços de telecomunicações. Para os dois conjuntos de dados, o modelo proposto foi comparado com outras técnicas, como Regressão logística e Árvores de decisão, apresentando maior quantidade de instâncias classificadas corretamente (89,63%).

[Zhuang 2018] apresenta um modelo baseado no algoritmo XGBoost para a previsão de desligamento de clientes de uma rede social *e-commerce*. O conjunto de dados utilizado nesse trabalho contém atributos de recência, frequência e valor monetário, (modelos RFM), combinados com o valor do efeito de rede de cada cliente. No experimento construído, o modelo XGBoost supera técnicas comumente usadas como Redes Neurais Artificiais (RNAs) e Regressão Logística, com acurácia de 88% e AUC de 89%.

Em [Mena et al. 2019], é proposto um modelo baseado em rede LSTM (*Long-Short Term Memory*) para extrair probabilidade de desligamento de cliente. O modelo usa atributos variáveis no tempo a partir de dados de recência, frequência e valor monetário de um provedor de serviços financeiros. Os dados foram rotulados por meio de análise de clientes que fecharam todos os contratos em uma janela de 12 meses. O modelo LSTM, com os atributos RFM, apresentou uma AUC de 77,9% nos experimentos executados.

Na literatura, as Redes Neurais Convolucionais (CNN) são ainda pouco exploradas para o problema de desligamento de clientes [Zhong and Li 2019, Chouiekh et al. 2020]. Em [Ahn et al. 2020], é proposto um modelo de CNN que considera características temporais dos dados, chamado *Weibull Time To Event Temporal Convolutional Network* (WTTE-TCN). O WTTE-TCN contém camadas TCNNs bidirecionais, camadas de atenção, camadas de processamento de dados textuais. O conjunto de dados utilizado nesse trabalho possui dados de jogos para dispositivos móveis. Nos experimentos realizados, o WTTE-TCN obteve um erro médio absoluto de 11,34 e apresentou vantagens em relação a custo computacional.

Todavia, acreditamos que a abordagem temporal é um caminho promissor para a PDC no setor financeiro, dado que, transações financeiras podem ser sequenciados no tempo. A partir disso, modelos baseados em dados sequenciais podem ser aplicados ao problema, especialmente, baseados em TCNN, que têm superado as redes recorrentes genéricas como a LSTM para várias tarefas de mapeamento de sequências [Bai et al. 2018].

3. Metodologia

Para a resolução do problema apresentado, foi utilizado o seguinte fluxo metodológico apresentado na Figura 1.

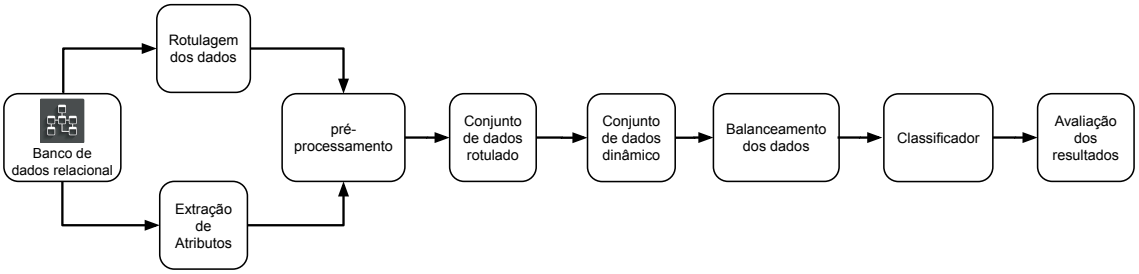


Figura 1. Fluxo metodológico utilizado neste trabalho.

3.1. O conjunto de dados

Os dados foram fornecidos por uma instituição financeira brasileira e disponibilizados em forma de um banco de dados relacional. Movimentações de serviços financeiros utilizados pelos clientes foram registrados compreendendo um período de 52 semanas, de agosto de 2020 a julho de 2021. Ressaltamos que os dados fornecidos não incluem um indicador de desligamento. Respeitando a Lei Geral de Proteção de Dados brasileira, todas as informações foram devidamente anonimizadas e os respectivos valores das transações normalizados.

Os serviços financeiros utilizados pelos clientes e que foram selecionados neste trabalho foram: conta paga, cobranças, Pix enviado e recebido, Transferência Eletrônica Disponível (TED) enviado e recebido.

Visando a criação de um conjunto de dados mais apropriado para aprendizagem supervisionada, foram extraídos atributos com base nas tabelas do banco de dados relacional (por cliente). A extração dos atributos foi realizada para cada serviço separadamente. Aqui, os atributos foram classificados em quatro categorias: (i) atributos monetários, (ii) atributos de atividade, (iii) atributos de recência e (iv) atributos de frequência. A Tabela 1 descreve os atributos utilizados neste trabalho.

Tabela 1. descrição dos atributos extraídos por categoria.

Categoria	Atributos extraídos
Monetários	Valor máximo, médio e mínimo de cada serviço utilizado pelo cliente
Atividade	Total de transações de cada serviço utilizado pelo cliente, total de cobranças com API, total de cobranças com carnê, total de cobranças com boleto, total de cobranças pagas, total de cobranças não pagas, total de cobranças com cartão e total de cobranças canceladas
Recência	Os dias desde a última transação de cada serviço utilizado pelo cliente
Frequência	O máximo, a média e o mínimo de dias entre transações subsequentes de cada serviço realizado pelo cliente

Neste trabalho, a criação do conjunto de dados segue uma abordagem temporal. O conjunto de dados foi dividido em duas janelas temporais. Os registros das transações relativas à janela temporal das 26 semanas mais antigas foram separados para a extração dos atributos para gerar os atributos do conjunto de dados de treinamento. Os registros das transações das outras 30 semanas (as mais recentes) foram reservados para o processo de rotulagem dos dados, visando aprendizado supervisionado. A escolha do tamanho da janela foi definida a partir de conversas com especialistas da empresa baseando-se no padrão de uso dos clientes. A janela deve ser grande o suficiente para capturar o comportamento do cliente. Todavia, um tamanho muito grande de janela pode capturar comportamentos de usos no passado que não representem mais o cliente atual.A Figura 2 apresenta a separação dos dados.

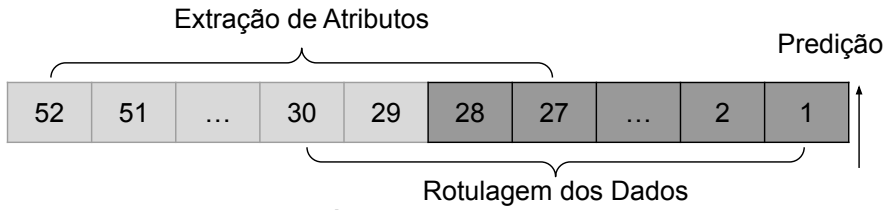


Figura 2. Separação dos históricos de registros: as 26 semanas mais antigas geraram os atributos e as 30 semanas (mais recentes) reservadas para a rotulagem dos dados.

Os atributos foram gerados de forma sequencial, semana-a-semana. Tem-se, então, versões de um mesmo atributo extraído em períodos diferentes. Uma versão tabular do conjunto de dados sequenciais foi gerada (conjunto de dados dinâmico), consistindo em um único arquivo *Comma-Separated-Values* (CSV) orientado por cliente, com total de 60607 clientes com 62 tipos de atributos, cada um com 26 versões, totalizando 1612 atributos de histórico de transações.

3.2. Rotulagem dos Dados

Uma boa abordagem para previsão de desligamento de clientes é aprender os padrões de comportamento do cliente (transações realizadas) a partir dos dados históricos. Cada cliente possui padrões próprios de uso dos serviços. Neste sentido, definimos o engajamento de cada cliente baseado na redução da frequência de uso dos serviços.

A partir dos registros das últimas 30 semanas (ver Figura 2), definimos um limiar para cada cliente, em cada serviço, como sendo o percentil 60 dos dias entre suas transações. Neste trabalho definimos o percentil 60% empiricamente como um valor acima da mediana (uma redução significativa da frequência de uso), pensado na classificação do cliente desligado antes de um comportamento de desengajamento explícito. Formalmente definimos o engajamento E do cliente C_i no serviço s_j, como:

$$E(C_i, s_j) = \begin{cases} 1, & \text{se (dias desde a última transação)} < \text{(percentil 60 dos dias entre as transações)} \\ 0, & \text{caso contrário} \end{cases} \tag{1}$$

Se $E(C_i, s_j) = 1$, o cliente C_i usa o serviço s_j como de costume, ou seja, definindo o cliente C_i engajado no serviço s_j. Por outro lado, se $E(C_i, s_j) = 0$, o cliente C_i não usa o serviço s_j como de costume, definindo o cliente C_i desengajado no serviço s_j.

Formalmente, a rotulagem proposta neste trabalho é definida como:

$$R(C_i) = \begin{cases} 1, & \text{se } \sum_{j=1}^{n} E(C_i, s_j) = 0 \\ 0, & \text{se } \sum_{j=1}^{n} E(C_i, s_j) > 0 \end{cases} \tag{2}$$

em que n é o número total de serviços selecionados para a base de dados. Assim, se $R(C_i) = 0$ o cliente C_i é rotulado como não desligado, ou seja, está engajado em pelo menos um serviço, e se $R(C_i) = 1$ o cliente está desengajado em todos os serviços e rotulado como desligado.

A intenção dessa abordagem é prever se um cliente vai reduzir o uso dos serviços mesmo antes de apresentar um comportamento de desligamento explícito.

3.3. Modelagem de Sequência e Previsão de Sequência

Naturalmente, como a construção dos atributos deste trabalho segue uma abordagem sequencial, esses dados podem ser submetidos a técnica de modelagem de sequência. Por exemplo, dado uma entrada sequencial $x_0, ..., x_T$, onde a informação em x_i com $i < T$, representa um valor no passado, quando se considera o período de tempo T. Se um modelo quer predizer uma saída futura ($\hat{y}_t$) com base no histórico de dados do passado ($x_0, ..., x_t$), ele deve, idealmente, levar em consideração a ordem em que os dados são amostrados, ou seja, o fator temporal. De maneira formal, um modelo sequencial é uma função $f : X^{T+1} \rightarrow Y^{T+1}$ que gera o mapeamento

$$\hat{y}_T = f(x_0, ..., x_T)$$

se essa satisfaz a restrição causal de que y_t depende apenas de $x_0, ..., x_t$ e não de nenhuma entrada subsequente $x_{t+1}, ..., x_T$.

Desse modo, espera-se construir modelos sequenciais baseados em TCNN com convoluções causais para previsão de desligamento futuro a partir de dados sequenciais presentes no conjunto de dados dinâmico construído.

O conjunto de dados dinâmico foi adaptado para se adequar a entrada do modelo TCNN proposto. Tem-se, para cada atributo, uma sequência de 26 elementos. O conjunto de dados completo é um *tensor* de formato $(tamanho\_de\_lote, dimensao\_de\_entrada, numero\_canais)$, em que o $tamanho\_de\_lote$ é igual ao número de instâncias de dados de clientes, a $dimensao\_de\_entrada$ é igual ao número de passos (em semanas) e o $numero\_canais$ é igual ao número de atributos, ilustrado na Figura 3.

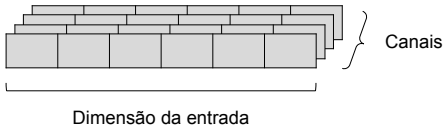


Figura 3. Ilustração de uma instância de entrada para uma TCNN.

3.4. Redes Convolucionais Temporais

As Redes Neurais Convolucionais Temporais (do inglês *Temporal Convolutional Neural Networks* - TCNN) não é uma arquitetura nova de redes convolucionais, mas sim, um termo que descreve um conjunto de arquiteturas [Bai et al. 2018]. Esse tipo de rede é capaz de olhar para um passado distante e realizar uma previsão usando camadas convolucionais dilatadas e causais. Segundo [Bai et al. 2018], duas características definem uma rede como TCNN, são elas: (i) as convoluções na arquitetura devem ser causais (ii) a arquitetura pode receber uma sequência de qualquer tamanho e mapeá-la para uma sequência de saída.

Atendendo ao primeiro critério, para que uma convolução seja causal, não se pode ter informação do futuro na operação. Dessa forma, deve-se garantir que para uma saída no tempo t, apenas elementos no tempo t ou anterior podem fazer parte da operação de convolução, conforme Figura 4a.

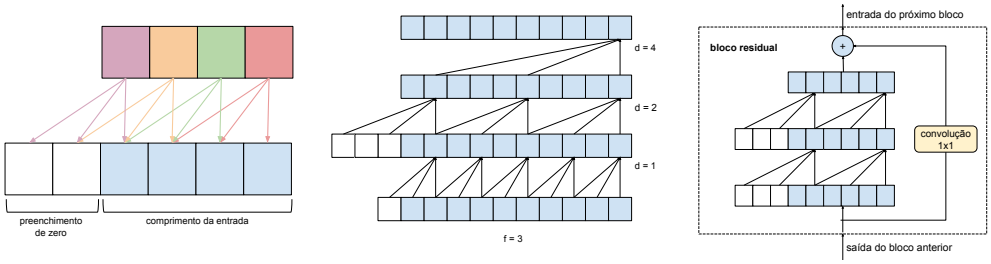
Ao se usar uma rede de convolução tradicional para um tarefa de previsão, fica-se limitado ao campo receptivo das operações de convolução. A fim de se aproveitar melhor o tamanho do comprimento do dado de entrada, as TCNNs geralmente utilizam convoluções dilatadas [Bai et al. 2018]. Uma operação de convolução dilatada é o equivalente a introduzir um distanciamento fixo entre os elementos da sequência que são enxergados pelo filtro (Ver A Figura 4b).

Finalmente, as TCNNs empregam conexões residuais para atender ao segundo critério de que a sequência de entrada e saída das convoluções tenham o mesmo tamanho. Essas conexões se formam a partir da saída de uma operação de convolução 1x1 adicionada a saída das camadas convolucionais dilatadas. Essa junção de camadas é chamada de bloco residual (ilustrado na Figura 4c).

4. Experimentos

4.1. Pré-processamento dos dados

O conjunto de dados passou por um processo de pré-processamento. Os valores ausentes foram substituídos por zero, para atributos de atividade e monetários quando o cliente



(a) Convoluções causais. **(b)** Convoluções Dilatadas. **(c)** Bloco residual.

Figura 4. Representação das convoluções causais, convoluções dilatadas e bloco residual. O Autor.

não realizou nenhuma transação serviço naquela semana. Para os atributos de recência e frequência, os valores ausentes foram representados pelo pior caso da janela (sete dias) quando nenhuma transação do serviço foi encontrada naquela semana. Os clientes que não usaram nenhum serviço durante o período de observação da rotulagem foram removidos considerados clientes ”fantasmas” dada a abordagem baseada em recência e frequência. Por fim, os dados foram normalizados entre 0 e 1. O conjunto de dados depois deste pré-processamento ficou distribuído da seguinte forma: 34712 clientes não desligados (classe 0) e 2801 clientes desligados (classe 1).

4.2. Estratégias para contornar o desbalanceamento do conjunto de dados

O conjunto de dados apresentou um alto desbalanceamento das classes, o que é natural. Para contornar o desbalanceamento, duas combinações de técnicas foram aplicadas para a partição de treinamento dos dados: (i) subamostragem aleatória de 50% das instâncias da classe majoritária, combinada com sobre-amostragem da classe minoritária com uso da Técnica de Sobre-amostragem Minoritária Sintética (*Synthetic Minority Oversampling Technique* - SMOTE) [Chawla et al. 2002] da biblioteca *imbalanced-learn* e (ii) subamostragem de 50% da classe majoritária combinada com compensação por meio da técnica *compute_sample_weight* do *scikit-learn* que usa os valores dos rótulos para ajustar pesos inversamente proporcionais às frequências das classes nos dados. A primeira abordagem resultou em 13884 instâncias por classe. Para a segunda abordagem, a classe 0 contou com 13884 instâncias enquanto a classe 1 com 2241 instâncias.

4.3. Implementação dos Modelos

Os modelos XG-Boost e Floresta Aleatória, considerados estado-da-arte para problemas de predição com dados tabulares, foram selecionados para a comparação com o modelo TCNN proposto aqui neste trabalho. A linguagem de programação *Python* foi utilizada para o desenvolvimento dos algoritmos. O modelo TCNN foi construído a partir da biblioteca *Keras-TCN* [Remy 2020] baseada na API *Keras* da biblioteca *Tensorflow*. Os outros dois modelos foram implementados a partir da biblioteca de aprendizado de máquina *scikit-learn* [Pedregosa et al. 2011] pelos algoritmos *ensemble RandomForestClassifier* e *GradientBoostingClassifier* (com *objective binary:logistic* e *eval_metric logloss*).

4.3.1. Busca de Melhores Hiper-parâmetros

Os hiper-parâmetros dos modelos XG-Boost e Floresta Aleatória foram ajustados usando o algoritmo *GridSearchCV* da biblioteca *scikit-learn*. O objetivo da busca é de obter o maior *F-score* dispondo o conjunto de dados rotulados, com subamostragem de 50% da classe majoritária divididos em 80% treino e 20% teste em uma validação cruzada interna de 3 *k-folds*.

As variações e os hiper-parâmetros encontrados (destacados em negrito) são: XG-Bosst (*min_child_weight* (**1**, 5, 10), *gamma* (0.5, 1, 1.5, 2, **5**), *subsample* (0.6, 0.8, **1.0**), *colsample_bytree* (0.6, **0.8**, 1.0), *max_depth* (3, 4, **5**) e o maior *F-score* obtido de 40.98); Floresta Aleatória (*n_estimators* (**200**, 300, 400, 500); *criterion* (***gini***, *entropy*); *max_features* (***auto***,4,5,6,7,8); *max_depth* (4,5,6,7,**8**) e o maior *F-score* obtido de 36.08).

Para o modelo TCNN a escolha dos hiper-parâmetros foi feita de forma empírica. Pelo fato do TCNN ser um modelo de outra natureza (*deep learning*) o ajuste dos hiper-parâmetros depende, acima de tudo, da arquitetura. E encontrar boas arquiteturas é um problema não trivial. Nesse caso, métodos de pesquisa de arquitetura neural (*Neural Architecture Search* - NAS) podem ser empregados. Contudo, métodos de NAS são muito custosos computacionalmente e serão alvo de investigação futura. Neste trabalho a construção do modelo partiu da avaliação de novos hiper-parâmetros partindo da configuração padrão implementada pelo autor da biblioteca [Bai et al. 2018].

4.3.2. Arquitetura do Modelo TCNN

O modelo TCNN proposto foi construído a partir de blocos de camadas convolucionais dilatadas, seguida de uma camada totalmente conectada de saída tamanho 1 e a ativação *sigmoid*. As configurações construídas foram diferentes para as abordagens de balanceamento. Para o treinamento com o SMOTE, as camadas convolucionais e os hiper-parâmetros escolhidos foram: *nb_filters* (3); *dilations* (1, 2, 4, 8, 16, 32, 64); *dropout_rate* (0.4); *activation* (*gelu*); *kernel_initializer* (*glorot_uniform*); *use_vlayer_norm* (*False*). O otimizador utilizado foi o *Adam* com taxa de aprendizagem inicial de 0.01 e a *loss binary_crossentropy*. Para o treinamento com peso nas classes algumas modificações foram aplicadas, são essas: *dilations* (1, 2, 4, 8, 16); *dropout_rate* (0.6); *use_vlayer_norm* (*True*); taxa de aprendizagem de 0.001. Os demais hiper-parâmetros seguiram a configuração padrão da implementação original. A Figura 5 ilustra uma das arquiteturas desenvolvidas.

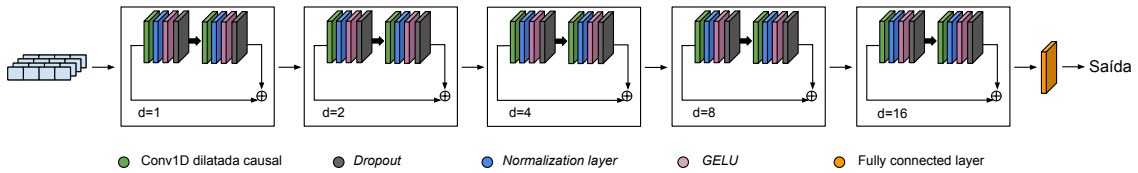


Figura 5. Ilustração da arquitetura da TCNN proposta, usando balanceamento com pesos na classe.

4.4. Treinamento dos Modelos

Os modelos foram treinados em um esquema de validação cruzada de 5 *k-folds*, com as duas abordagens de balanceamento. Para os modelos RF e XG-Boost o conjunto de

dados para cada *k-fold* foi dividido em 80% para treino e 20% para o teste. Já para o modelo TCNN, o conjunto de dados pra cada *k-fold* foi dividido em 70% para treino, 10% para validação e 20% para o teste. As métricas extraídas dos modelos são as médias das métricas de todos os *k-folds*. O modelo de TCNN para o balanceamento com pesos na classes foi treinado por 150 épocas e tamanho de lote (*batch_size*) 64. Para a versão com o SMOTE, foram utilizadas 500 épocas e taxa de aprendizado variando-se de 0.1 a 0.000001, ou seja, taxa de aprendizado multiplicada por 0.1 a cada 100 épocas.

4.4.1. Métricas de desempenho

Métricas de avaliação utilizadas são: Acurácia - $Acc = \frac{(VP+VN)}{VP+FN+VN+FP}$; Sensibilidade - $Se = \frac{VP}{(VP+FN)}$; Especificidade - $Es = \frac{VN}{(FP+VN)}$; Precisão - $Pr = \frac{VP}{(VP+FP)}$; Taxa de Falso Positivos - $TFP = \frac{FP}{VN+FP}$ e ***F-score*** = $2 \times \frac{(Pr \times Se)}{(Pr+Se)}$, em que FN é o número de falsos negativos; FP o número de falsos positivos; VN o número de verdadeiros negativos; e VP o número de verdadeiros positivos.

4.5. Resultados e Discussão

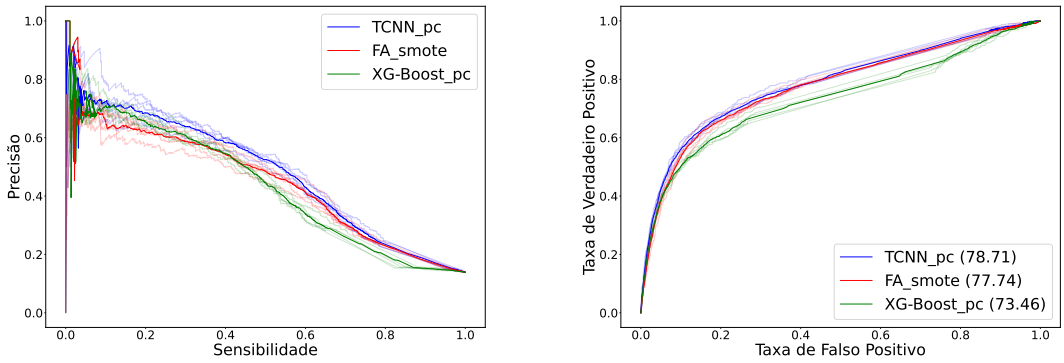
Um sumário do resultado experimental pode ser visto na Tabela 2.

Tabela 2. Comparação entre médias das métricas obtidas (desvio padrão), pelos modelos XG-Boost (XGB), *Floresta Aleatória (FA)* e TCNN. *(Balanceamento - pc = com pesos nas classes, smote = com SMOTE).

Modelo	Acc	Se	Es	TFP	Pr	*F-Score*
FA_pc	83.1(0.39)	59(1.89)	86.99(0.38)	13.01(0.38)	42.24(0.97)	49.23(1.22)
FA_smote	83.76(0.5)	57.29(2.03)	88.03(0.38)	11.97(0.38)	43.56(1.33)	49.49*(1.55)
XGB_pc	84(0.57)	51.29(1.52)	89.28(0.82)	10.72(0.82)	43.61(1.39)	47.11(0.78)
XGB_smote	**86.94**(0.29)	32.14(1.29)	**95.78**(0.3)	**4.22**(0.3)	**55.14**(1.86)	40.6(1.31)
TCNN_pc	82.63(0.67)	**61.5**(2.23)	86.04(1.06)	13.96(1.06)	41.58(1.17)	**49.58**(0.7)
TCNN_smote	82.41(0.9)	59.96(2.07)	86.03(1.25)	13.97(1.25)	40.98(1.67)	48.65(1.13)

Para os resultados com balanceamento SMOTE tem-se destaque para a métrica de especificidade do método XGB (95,78%). Contudo, o modelo TCNN com pesos nas classes obteve melhores métricas de sensibilidade (61,5%), o que indica melhor capacidade para classificar clientes desligados. Com relação ao *F-Score*, o modelo de TCNN com pesos nas classes obteve melhor métrica (49.58%). As curvas de $Precisão \times Sensibilidade$ foram geradas para os três modelos, e indicam o compromisso entre a precisão e a sensibilidade (ver Figura 6a). Adicionalmente, foram geradas as curvas ROC (*Receiver Operating Characteristic*) para os três modelos, e apresenta a performance entre entre as taxas de verdadeiro positivo e falso positivo (ver Figura 6b). Em ambos os gráficos, o modelo TCNN_pc sobrepõem os outros dois modelos na maioria dos limites.

A arquitetura TCNN proposta superou os outros métodos bem populares para o problema de previsão de desligamentos de clientes. Contudo, ressaltamos que a TCNN tem espaço para melhorias. Um caminho promissor de melhorias é o emprego de mecanismo



(a) Curva de Precisão x Sensibilidade. **(b)** Área sob a curva ROC.

Figura 6. Nos gráficos são apresentados as curvas para os 5 *k-folds* e em destaque a curva média para os três modelos. No gráfico (b) o valor entre parenteses representa a área sobre a curva ROC média do modelo. O Autor.

de atenção [Vaswani et al. 2017] para selecionar, de forma dinâmica, os melhores atributos (canais) para cada inferência. Os métodos baseados em Floresta Aleatória já efetuam uma seleção de atributos de forma intrínseca. Todavia, acreditamos que o caminho mais promissor para melhorias seja por meio de métodos de Busca por Arquiteturas de Redes (do inglês *Network Arquitecture Search* ou NAS) [Tan et al. 2019], por exemplo investigar metodologia multi-objetivo para buscas de arquiteturas (MNAS), otimizando simultaneamente o *F-score* e o tamanho da rede.

Finalmente, alguns pontos sobre as vantagens da utilização de modelos baseados em TCNN, para o contexto da indústria, precisam ser considerados. Por ser facilmente paralelizável e compatível com aceleradores de hardware (GPUs e TPUs), a manutenção do modelo pode ser facilitada por meio de treinamento online (por mini-lotes) e transferência de aprendizagem. Portanto, é possível automatizar um *pipeline* para atualização constante (semanal) dos modelos de forma facilitada.

5. Conclusão

No atual mercado competitivo de empresas do setor financeiro, a previsão de desligamento de clientes é uma questão importante para reter clientes valiosos, otimizar o lucro, e fornecer ofertas ou serviços competitivos. Entre outros aspectos, este trabalho contribuiu com uma metodologia para a criação de um conjunto de dados baseado em séries temporais acumulativas. Além disso, foi apresentado uma abordagem de rotulagem baseada na redução de frequência de uso dos serviços, considerando o comportamento particular de cada cliente. Ainda neste estudo, um modelo de previsão de desligamento de clientes baseado em Redes Neurais Convolucionais Temporais foi avaliado em conjunto de dados de uma empresa do setor financeiro brasileiro. O modelo proposto obteve acurácia de 82.63%, sensibilidade de 61.5% e precisão de 41.58% e, superando outros métodos tradicionais empregados na literatura, *XG-Boost* e Floresta Aleatória.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, da FAPEMIG (APQ-01518-21), da Universidade Federal de Ouro Preto e da Gerencianet S.A.

Referências

Ahn, D., Lee, D., and Hosanagar, K. (2020). Interpretable deep learning approach to churn management. *Available at SSRN 3981160*.

Amin, A., Khan, C., Ali, I., and Anwar, S. (2014). Customer churn prediction in telecommunication industry: With and without counter-example. In *2014 European Network Intelligence Conference*, pages 134–137.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv:1803.01271*.

Caigny, A. D., Coussement, K., and Bock, K. W. D. (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.

Caigny, A. D., Coussement, K., Bock, K. W. D., and Lessmann, S. (2020). Incorporating textual information in customer churn prediction models based on a convolutional neural network. *International Journal of Forecasting*, 36(4):1563–1578.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, Z.-Y., Fan, Z.-P., and Sun, M. (2012). A hierarchical multiple kernel support vector machine for customer churn prediction using longitudinal behavioral data. *European Journal of Operational Research*, 223(2):461–472.

Chouiekh, A. et al. (2020). Deep convolutional neural networks for customer churn prediction analysis. *International Journal of Cognitive Informatics and Natural Intelligence (IJCINI)*, 14(1):1–16.

Coussement, K. and De Bock, K. W. (2013). Customer churn prediction in the online gambling industry: The beneficial effect of ensemble learning. *Journal of Business Research*, 66(9):1629–1636.

Devriendt, F., Berrevoets, J., and Verbeke, W. (2021). Why you should stop predicting customer churn and start using uplift models. *Information Sciences*, 548:497–515.

Gregory, B. (2018). Predicting customer churn: Extreme gradient boosting with temporal data. *arXiv:1802.03396*.

Hadden, J., Tiwari, A., Roy, R., and Ruta, D. (2007). Computer assisted customer churn management: State-of-the-art and future trends. *Computers & Operations Research*, 34(10):2902–2917.

Lima, E., Mues, C., and Baesens, B. (2009). Domain knowledge integration in data mining using decision tables: case studies in churn prediction. *Journal of the Operational Research Society*, 60(8):1096–1106.

Mena, C. G., De Caigny, A., Coussement, K., De Bock, K. W., and Lessmann, S. (2019). Churn prediction with sequential data and deep neural networks. a comparative analysis. *arXiv:1909.11114*.

Neslin, S. A., Gupta, S., Kamakura, W., Lu, J., and Mason, C. H. (2006). Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of Marketing Research*, 43(2):204–211.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Remy, P. (2020). Temporal convolutional networks for keras. `https://github.com/philipperemy/keras-tcn`.

Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2820–2828.

Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., and 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.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *arXiv:1706.03762*.

Verbeke, W., Martens, D., Mues, C., and Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Systems with Applications*, 38(3):2354–2364.

Zhong, J. and Li, W. (2019). Predicting customer churn in the telecommunication industry by analyzing phone call transcripts with convolutional neural networks. In *Proceedings of the 2019 3rd International Conference on Innovation in Artificial Intelligence*, pages 55–59.

Zhuang, Y. (2018). Research on e-commerce customer churn prediction based on improved value model and xg-boost algorithm. *Management Science and Engineering*, 12(3):51–56.