

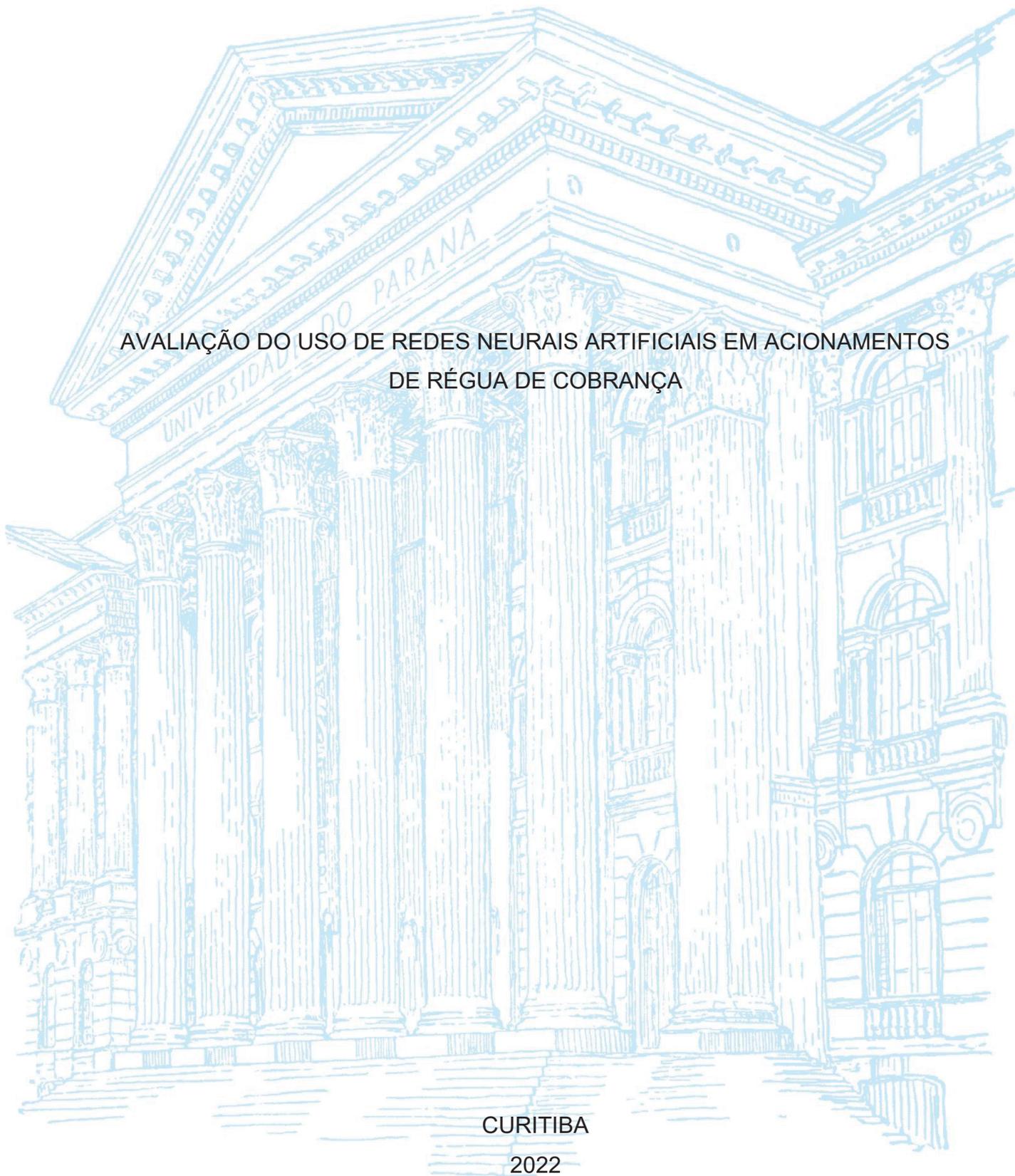
UNIVERSIDADE FEDERAL DO PARANÁ

RODRIGO VAZ DE JESUS

AVALIAÇÃO DO USO DE REDES NEURAIS ARTIFICIAIS EM ACIONAMENTOS  
DE RÉGUA DE COBRANÇA

CURITIBA

2022



RODRIGO VAZ DE JESUS

AVALIAÇÃO DO USO DE REDES NEURAIS ARTIFICIAIS EM ACIONAMENTOS  
DE RÉGUA DE COBRANÇA

Monografia apresentada ao curso de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada.

Orientador: Prof. Dr. Razer Anthom Nizer Rojas Montaña

CURITIBA

2022

## TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **RODRIGO VAZ DE JESUS** intitulada: **AVALIAÇÃO DO USO DE REDES NEURAIS ARTIFICIAIS EM ACIONAMENTOS DE RÉGUA DE COBRANÇA**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 07 de Julho de 2022.



RAZER ANTHOM NIZER ROJAS MONTAÑO  
Presidente da Banca Examinadora



JAIME WOJCIECHOWSKI  
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

# Avaliação do uso de Redes Neurais Artificiais em acionamentos de régua de cobrança

Rodrigo Vaz de Jesus  
Universidade Federal do Paraná UFPR  
Curitiba, Paraná, Brasil  
rodrigo.jesus@ufpr.br

Dr. Razer Anthon Nizer Rojas Montaña  
Universidade Federal do Paraná UFPR  
Curitiba, Paraná, Brasil  
razer@ufpr.br

**Resumo**—Em empresas prestadoras de serviços de Telecomunicações, a definição de uma estratégia de cobrança requer a avaliação de inúmeras variáveis, a fim de determinar qual o perfil do público inadimplente alvo e como alcançá-lo. Isso é necessário para que se atinja o melhor rendimento do processo de cobrança, isto é, recuperar a maior parte da dívida com o menor custo operacional possível. É um desafio para as empresas, pois o mercado está em constante transformação e exige a revisão recorrente dessas estratégias. O presente artigo avalia a eficiência das Redes Neurais Artificiais na cobrança de clientes inadimplentes de uma empresa de Telecomunicações. Uma base de indivíduos que receberam acionamentos de cobrança, já identificada com aqueles que efetuaram pagamento, foi submetida a modelos supervisionados de Redes Neurais Artificiais. O objetivo foi avaliar a capacidade da Inteligência Artificial prever o pagamento ou não desses clientes e assim definir a necessidade de um acionamento de cobrança. Ao final, foram comparadas as quantidades reais observadas de acionamentos para esse público com a quantidade predita de eventos pelo algoritmo. Observou-se que as Redes Neurais Artificiais foram capazes de efetuar a previsão de pagamento de clientes inadimplentes e possibilitariam a economia no custo da cobrança, sem prejudicar a eficiência desta. Também foi notado que a quantidade de amostras, bem como o conjunto de variáveis submetido ao algoritmo, interferiu de modo sensível no resultado das previsões.

**Palavras-chave**—*régua de cobrança, cobrança, inteligência artificial, redes neurais, inadimplência, recuperação de dívida*

**Abstract**—In companies providing telecommunications services, defining a collection strategy requires the evaluation of multiple variables, in order to determine the profile of defaulters and how reach them. This is necessary to reach the full potential of the collection process, that meaning, recovering as much of the debt as possible with the absolute lowest cost of operation. This is a challenge for companies, seeing that the market is in constant transformation and demands recurrent revisions of these strategies. This paper evaluates the efficiency of Artificial Neural Networks when collecting defaulter's in a telecommunication enterprise. A database containing individuals who have already been notified was submitted to supervised Artificial Neural Networks, with the objective being to foresee whether payments would be made or not, thus determining the need for action. Lastly actions taken by traditional means were compared to the predictions of payment made by the algorithm. It was observed that the Artificial Neural Networks were able to predict the payment of bad debt customers and would enable savings in the cost of collection without impairing its efficiency. It was also noted that the number of samples, as well as the set of variables submitted to the algorithm, significantly interfered in the prediction results.

**Keywords**—*collection, artificial intelligence, debt default, artificial neural network, debt recovery*

## I. INTRODUÇÃO

O Brasil atingiu no mês de dezembro de 2020 a marca de 61,4 milhões de pessoas inadimplentes, o que representava 38,6% da população adulta brasileira, uma queda de 3,1% em relação ao mesmo mês do ano anterior [1]. Naquele ano, teve o início e a evolução da pandemia da COVID-19 (*Corona Virus Disease 2019*) no Brasil, a qual impactou diretamente na economia do país. O primeiro semestre foi marcado por uma redução expressiva nos empregos com carteira assinada, seguido de uma reação na segunda metade do ano em decorrência das medidas e auxílios realizados pelo governo [2].

Historicamente, inúmeras crises impactaram de forma diferente no índice de inadimplência [3]. Este cenário demanda que as empresas revisem constantemente as estratégias de cobrança adotadas, a fim de aprimorar os processos de recuperação de dívida e acompanhar essas oscilações de comportamento do mercado.

Uma das técnicas utilizadas para recuperar a dívida de clientes inadimplentes é a régua de cobrança. Ela consiste em estabelecer um conjunto de regras que define como os clientes serão cobrados, em que momento e por qual meio, conforme a evolução do atraso da dívida [4].

Empresas que adotam a régua de cobrança precisam avaliar o comportamento dos seus clientes para então definir as melhores estratégias de abordagem na cobrança. Para essa análise, levam-se em conta, geralmente, o montante da dívida, dias em atraso, dados pessoais do cliente (como idade, sexo, região em que reside etc.), e o custo operacional que será empregado nesses acionamentos. Uma vez definidas essas estratégias, elas são aplicadas massivamente aos clientes inadimplentes.

O canal de comunicação com os clientes é um item importante a ser avaliado no momento de desenhar uma régua de cobrança. Isto porque atualmente existem diversos meios de contatá-los, como ligação telefônica, carta, e-mail, SMS etc., e cada um possui um custo operacional específico. Além disso, a efetividade de cada meio pode variar conforme o perfil do consumidor alvo.

Dois a cada dez domicílios brasileiros ainda não utilizavam a Internet em 2019, limitados por fatores como a falta de conhecimento dos moradores, idade avançada, renda incompatível com o valor do serviço e indisponibilidade técnica [5]. A Fig.1 demonstra a proporção de uso da Internet entre as regiões brasileiras e por grupos de idades.

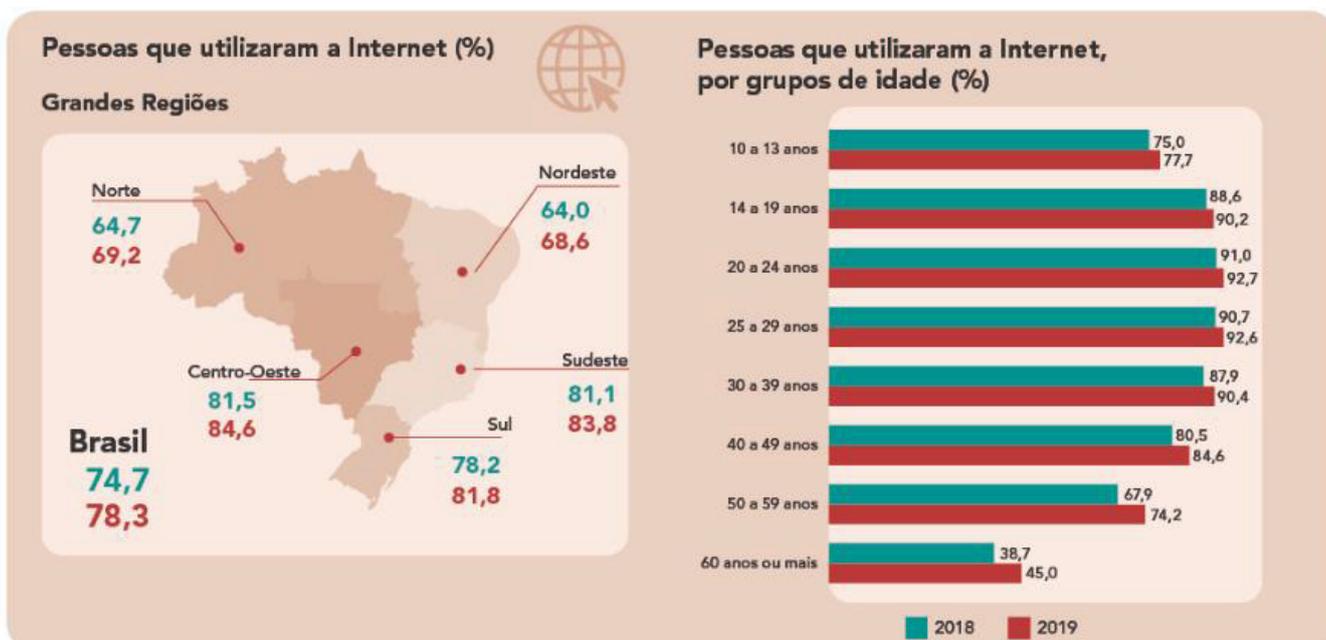


Fig. 1 Uso da Internet no Brasil por regiões e grupos de idade [5].

Estes dados demonstram que, apesar da Internet ser amplamente difundida, canais que a utilizam requerem uma avaliação prévia criteriosa para que sejam efetivos.

Complementar à uma definição ajustada dos meios de acionamentos, há também uma preocupação das empresas com a experiência do cliente. Uma régua de cobrança deve ter seu comportamento ponderado pelo perfil do cliente. Consumidores que frequentemente atrasam suas dívidas, efetuam maior volume de negociações ou até menos possuem dívida ativa de outros serviços, exigem uma quantidade maior de estímulos de cobrança na tentativa de recuperar esta dívida. Em contrapartida, um cliente que com bom histórico de pagamentos, mas que esqueceu de pagar uma fatura pontualmente, por exemplo, não necessita de uma cobrança tão agressiva.

Dada a quantidade de variáveis a serem consideradas para a definição de uma régua de cobrança, viu-se a oportunidade de se aplicar a Inteligência Artificial (IA) para desempenhar esta função. A IA automatiza tarefas complexas e pode ser usada em qualquer área de atividades intelectuais desenvolvidas por humanos, se tornando um campo de uso universal [7].

Um sistema com IA seria capaz de analisar todas essas variáveis e determinar se um cliente irá ou não pagar sua dívida após um determinado acionamento de cobrança? Esta foi a motivação deste trabalho. Uma máquina que fosse capaz de prever o comportamento do cliente em cobrança, possibilitaria o envio de acionamentos de cobrança somente nos casos e quando houvesse necessidade, sem comprometer a recuperação da dívida.

A hipótese é que um modelo de IA é capaz de definir a necessidade de enviar um evento da régua de cobrança a cada cliente, baseando-se na avaliação de dados estáticos momentâneos, sem o processamento recorrente dos dados históricos. Com isto seria possível reduzir os custos com acionamentos descabidos e manter a eficiência da cobrança. De forma resumida, o algoritmo seria capaz de indicar quais clientes precisam de um estímulo para pagamento e quais irão pagar sem a necessidade desse.

Nesta perspectiva, o presente estudo tem o objetivo de desenvolver um modelo supervisionado de Inteligência Artificial, capaz de indicar os clientes que requerem estímulos de cobrança, em seus estágios atuais de inadimplência, e avaliar se as previsões obtidas causariam a redução dos eventos de forma eficaz.

## II. FUNDAMENTAÇÃO TEÓRICA

### A. Conceitos de Cobrança

O conceito de inadimplência é comumente confundido com o de endividamento. Endividamento é qualquer dívida assumida pelo consumidor, mesmo que não esteja vencida [11]. Já a inadimplência trata-se da dívida em atraso decorrente do não pagamento de alguma obrigação financeira até a sua data de vencimento [12]. De forma representativa: quando um indivíduo realiza o financiamento de um bem, ele torna-se endividado até que ocorra a quitação total deste financiamento; se este mesmo indivíduo atrasa uma parcela deste financiamento, torna-se inadimplente.

Segundo a SERASA [13], no Brasil, as sete principais razões para uma pessoa tornar-se inadimplente são: desemprego, redução da renda familiar, compras realizadas para terceiros, ausência de educação financeira, falta de controle nos gastos, atraso de salário e enfermidades. A diversidade de fatores que influenciam este indicador só reforça que é necessário que as empresas revisem suas estratégias de cobrança frequentemente, para acompanhar e atender o momento do mercado. Uma dessas estratégias é o uso da régua de cobrança.

A régua de cobrança, ou simplesmente régua, funciona com base em um planejamento pré-estabelecido, que determina quando e como serão feitas as cobranças. A partir da avaliação do perfil dos clientes inadimplentes, define-se uma sequência de ações de cobrança a serem realizadas conforme o atraso da dívida evolui. A Fig. 2 representa o exemplo de uma régua de cobrança: um cliente com cinco dias de atraso receberá um evento de SMS; se permanecer inadimplente, ao atingir dez dias de atraso receberá uma carta,

com quinze dias um e-mail e assim sucessivamente para os demais eventos previstos.

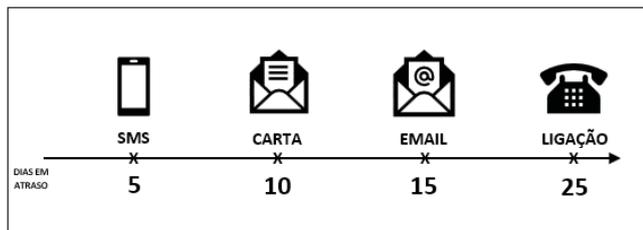


Fig. 2 Exemplo de régua de cobrança (Elaborado pelo autor).

Uma régua bem planejada garante que os acionamentos de cobrança ocorram no melhor momento e reflitam em uma recuperação de dívida satisfatória para empresa [4]. Como parte desse planejamento, as empresas avaliam o histórico de comportamento dos seus clientes e os segmentam em grupos com perfis semelhantes.

Outra preocupação na régua de cobrança é a experiência do cliente. Acionamentos excessivos ou uma abordagem invasiva podem gerar uma percepção negativa, acarretar em reclamações e eventuais processos judiciais. Nos últimos dez anos, assuntos relacionados à cobrança indevida estão no topo das reclamações nos órgãos de Proteção e Defesa ao Consumidor (PROCON) de todo o Brasil [6]. O desafio das empresas nesse aspecto é ajustar a intensidade da cobrança de forma proporcional ao risco de inadimplência do cliente, ou seja, o consumidor mais propenso a não pagar seus débitos requer uma quantidade maior de acionamentos, enquanto aquele com melhor perfil de pagamento, menor.

O custo operacional dos meios de acionamento empregados também é relevante no processo de criação de uma régua de cobrança. É necessário que se trace uma relação entre o total gasto com aquele canal e o retorno obtido (quantos clientes pagaram após receberem um estímulo por aquele meio). O cenário ideal é obter a maior recuperação de dívida com menor custo operacional possível.

Analisar todos esses dados e obter informações relevantes para a definição de uma estratégia é uma tarefa complexa, dada a volatilidade do mercado e ao volume de dados gerados pelas empresas diariamente. Isso requer o envolvimento de diversas equipes operacionais, desde especialistas no negócio para acompanhar e antecipar movimentos do mercado, até os analistas e desenvolvedores que irão de fato extrair e manipular dados para fomentar as decisões executivas.

### B. Fonte de dados

Em empresas que possuem seus sistemas informatizados, os dados podem ser obtidos através de relatórios predefinidos desses sistemas ou então consultas diretas em bancos de dados utilizando *Structured Query Language* (SQL). O SQL é uma linguagem padronizada para a definição e manipulação de dados em banco de dados relacional [14]. Este por sua vez, é um agrupamento lógico e estruturado dos dados representados por registros e tabelas, os quais se relacionam através de identificadores únicos respeitando regras de integridade predefinidas [15]. A obtenção de um conjunto de dados de estudo é o ponto de partida para a aplicação das técnicas de Inteligência Artificial.

### C. Inteligência Artificial

A IA surgiu da ideia de fazer com que uma máquina pensasse igual ao homem. O primeiro trabalho foi realizado

em 1943, quando Warren McCulloch e Walter Pitts propuseram o primeiro modelo de neurônios artificiais. Esse modelo foi sendo aprimorado ao longo dos anos até que em 1956, John McCarthy, Marvin Minsky, Claude Shannon, e Nathaniel Rochester, construíram uma máquina capaz de demonstrar teoremas matemáticos, consagrando nesse ano o nascimento da Inteligência Artificial [7]. Basicamente, consiste no uso de sistemas ou máquinas para reproduzir a inteligência humana na execução de tarefas. Para tal, tem como base seis disciplinas fundamentais, as quais exprimem as capacidades que um computador precisa ter para reproduzir os comportamentos humanos:

- 1) *Processamento de linguagem natural*: capacidade de compreender e responder estímulos de um idioma natural, de modo se comunicar com sucesso;
- 2) *Representação do conhecimento*: armazenar o conhecimento adquirido que possibilite utilizá-lo posteriormente;
- 3) *Raciocínio automatizado*: utilizar as informações armazenadas para responder a perguntas e obter novas conclusões;
- 4) *Aprendizado de máquina*: capacidade de aprender e se adaptar a novos padrões;
- 5) *Visão computacional*: habilidade de perceber objetos do mundo real;
- 6) *Robótica*: aptidão para manipulação de objetos e locomoção.

Na régua de cobrança, a necessidade é avaliar os dados de clientes para identificação de padrões de comportamento, para então, definir-se a estratégia de cobrança. Portanto, a disciplina de aprendizado de máquina é abordagem mais adequada para a introdução da IA na régua de cobrança.

#### a. Aprendizado de Máquina

Do inglês *Machine Learning*, tem por objetivo o desenvolvimento de técnicas computacionais sobre o aprendizado, bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um programa de aprendizado de máquina é capaz de tomar decisões baseando-se na experiência de resultados anteriores, bem como adaptar-se a partir deles. O conhecimento é gerado através da inferência indutiva, uma forma lógica de obter conclusões genéricas a partir de um conjunto específico de exemplos [16].

O aprendizado de máquinas subdivide-se em diversas abordagens, sendo as duas principais: o aprendizado não supervisionado, quando o problema apresentado não possui resultados conhecidos e o programa irá identificar padrões nos dados para a tomada de decisão; e o aprendizado supervisionado, quando se conhece o resultado esperado e este é utilizado para treinar o programa e posteriormente aplicá-lo para predição de novos dados [14].

No aprendizado de máquina supervisionado o algoritmo é treinado com uma amostra de dados que contém as variáveis relevantes para a análise – características ou atributos – com seus respectivos resultados já observados – as classes. O objetivo é construir um classificador que possa determinar corretamente a classe de novos dados ainda não identificados, ou seja, sem classes, a partir do conhecimento adquirido com os dados de treino. O problema é denominado como classificação quando os rótulos das classes são discretos e regressão quando são valores contínuos [16].

Já no aprendizado não supervisionado, o algoritmo recebe apenas os atributos dos dados, sem as classes, a partir dos quais busca identificar padrões e criar agrupamentos (ou *clusters*), que normalmente precisam ser avaliados posteriormente para determinar o que significam no contexto do problema em análise [14]. A Fig. 3 apresenta a hierarquia de aprendizado indutivo, com destaque aos nós que levam ao aprendizado supervisionado de classificação.

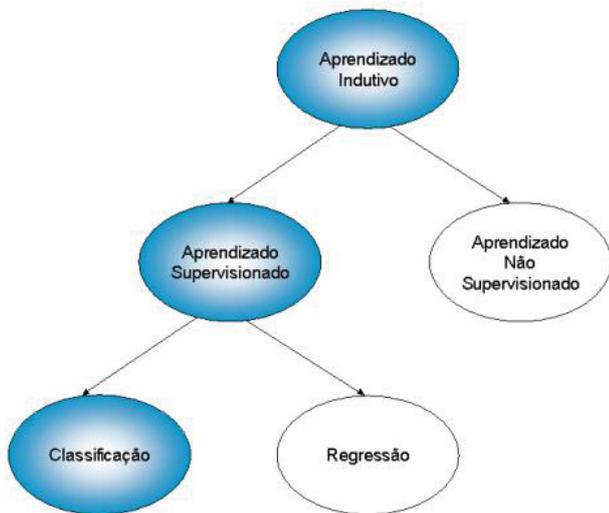


Fig. 3 A hierarquia do aprendizado [16].

Além dos tipos de aprendizado de máquina, MONARD [16] descreve os paradigmas que esta abordagem pode seguir, bem como exemplos de algoritmos para cada uma:

- *Simbólico*: utiliza representações simbólicas dos conceitos. Ex.: árvores de decisão;
- *Estatístico*: aplica modelos estatísticos para encontrar uma boa aproximação. Ex.: SVM;
- *Baseado em Exemplos*: classificam dados nunca vistos a partir da semelhança com exemplos armazenados. Ex.: K-NN;
- *Conexionista*: análoga ao sistema nervoso humano, baseia-se em uma rede de elementos interconectados. Ex.: Redes Neurais Artificiais (RNAs);
- *Genético*: derivado do modelo biológico de aprendizado, em que os elementos competem entre si e proliferam apenas os mais fortes. Ex.: Algoritmos Genéticos.

No modelo tradicional de régua de cobrança, os dados dos clientes são relacionados a fim de identificar padrões. O perfil de bom ou mal pagador, pode estar relacionado a região em que o cliente reside, idade ou ao tipo de serviço/produto que ele possui, por exemplo. A abordagem conexionista é a que mais se aproxima do problema da régua de cobrança no mundo real, uma vez ela relaciona os dados e estabelece conexões entre eles, de forma semelhante ao processo feito por um humano nessa tarefa.

#### b. Redes Neurais Artificiais

As RNAs são modelos computacionais análogos ao sistema nervoso central humano. Nela os dados são recepcionados por neurônios artificiais, que avaliam e

ponderam a informação através de funções predefinidas e repassam a outros neurônios. Esse processo se repete até que, ao final, o resultado é gerado pelo neurônio de saída [7]. No corpo humano, a habilidade de aprender é resultado da formação de novas conexões (sinapses) entre neurônios no cérebro.

A Fig.4 demonstra a representação de conexão de neurônios naturais e neurônios artificiais. Na imagem à esquerda, é mostrada a estrutura de dois neurônios naturais, em que é possível observar a sinapse entre eles. À direita, a representação de uma cadeia de neurônios artificiais,  $N_n$ , em que as setas representam as sinapses,  $u_n$  as informações de entrada e  $y_n$  as saídas.

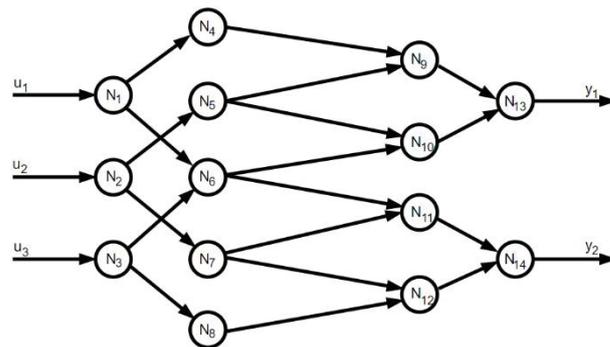
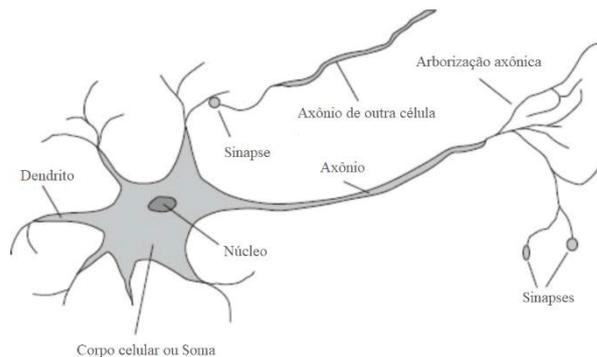


Fig. 4 Representação de conexão entre neurônios naturais (acima) e neurônios artificiais (abaixo) [7].

Um neurônio biológico é composto por corpo celular, dendritos, axônio e suas terminações. As informações de outros neurônios são receptadas pelos dendritos, através de sinais eletroquímicos, e conduzidas para o corpo celular. No corpo a informação é processada e um impulso elétrico é enviado através do axônio. Caso exista a conexão com outros neurônios, a informação é repassada através das sinapses [20].

O neurônio artificial segue uma estrutura análoga ao biológico. As informações (atributos dos dados a serem avaliados ou as saídas de outros neurônios) são receptadas pelas entradas, as quais recebem um peso. Uma soma ponderada desses pesos é realizada e atribuída a uma função matemática de ativação. Esta função limita a amplitude do sinal de saída do neurônio a algum valor finito e gera um valor de saída para o neurônio [7][21].

Do mesmo modo que ocorre em neurônios biológicos, o neurônio artificial possui um ou mais sinais de entrada e apenas um sinal de saída. Cada sinal de entrada possui um peso associado que representa o grau de importância que ela representa em relação àquele neurônio. Quanto mais estimulado um sinal de entrada, mais estimulado será o peso

correspondente, tornando o mais representativo para o neurônio. Vale ressaltar que na primeira iteração da rede neural artificial esses pesos possuem valores aleatórios, uma vez que são necessários valores iniciais para o início do processamento.

A função de entrada realiza o somatório ponderado dos sinais de entrada pelos seus respectivos pesos e *bias*. O *bias* (viés, em português) é uma constante utilizada para ajustar a saída da soma ponderada e dar mais flexibilidade à rede neural (como por exemplo, para que seja possível a propagação da saída nos casos em que a entrada tenha valor igual a zero). Em seguida, é aplicada a função de ativação sobre essa soma para se obter a saída [25].

A Fig. 5 demonstra um modelo matemático simples de um neurônio artificial, em que  $x_n$  são os sinais de entrada;  $w_n$  os pesos da ligação com este neurônio;  $b_k$  o *bias*;  $\varphi(\cdot)$  a função de ativação e  $y_k$  o sinal de saída.

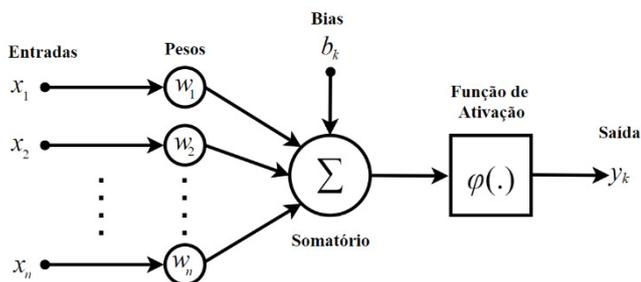


Fig. 5 Modelo matemático simples de um neurônio artificial (Adaptado de [24]).

Matematicamente o comportamento de um neurônio artificial pode ser representado pela Eq. 1.

$$y_k = F\left(\sum_{i=1}^n (x_i w_i) + b_k\right)$$

Eq. 1 Equação do comportamento de um neurônio artificial [24].

Em uma RNA os neurônios estão distribuídos em camadas, que podem ser de entrada, oculta ou de saída. As camadas de entrada e saída são as responsáveis pela passagem da informação para dentro e para fora da rede neural, respectivamente. Já nas camadas ocultas ocorrem o processamento da informação através das funções de ativação.

#### i. Funções de ativação

As funções de ativação são responsáveis pela transformação não linear do sinal de entrada, possibilitando captar as relações mais complexas entre os dados. Quando não temos a função de ativação, os pesos e bias simplesmente fazem uma transformação linear. As principais funções de ativação são: Com Limite, Linear, Sigmoide, Tangente Hiperbólica e ReLU [24].

1. Com Limite (ou *threshold*): define a saída 1 ou 0 de acordo com um limite estabelecido.

$$\varphi(x) = \begin{cases} 1, & \text{se } x \geq 0.5. \\ 0, & \text{c. c.} \end{cases}$$

Eq. 2 Equação da Função de Ativação Com Limite [24].

2. Linear: aplica apenas um fator de multiplicação ao valor que recebe.

$$\varphi(x) = ax$$

Eq. 3 Equação da Função Linear [24].

3. Sigmoide: define a saída apenas com números entre 0 e 1. Desejável em redes em que se precisar ter somente números positivos na saída.

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

Eq. 4 Equação da Função Sigmoide [24].

4. Tangente Hiperbólica (TanH): similar a sigmoide, porém com intervalo de saída entre -1 e 1.

$$\varphi(x) = \tanh(x)$$

Eq. 5 Equação da Função Tangente Hiperbólica [24].

5. ReLU (Unidade Linear Retificada): define todos os valores negativos como 0 na saída. Valores positivos permanecem com o valor inicial.

$$\varphi(x) = \max(0, x)$$

Eq. 6 Equação da Função ReLU [24].

#### ii. Arquitetura

A arquitetura de uma rede neural artificial é definida conforme a maneira como os neurônios artificiais são agrupados. A Fig. 6 demonstra uma arquitetura simples de uma RNA, em que há uma camada de entrada (os dados a serem avaliados), uma única camada central com os neurônios que irão processar esses dados (camada oculta) e por fim a camada de saída, onde serão gerados os resultados. Este modelo é chamado de *Perceptron* [7].

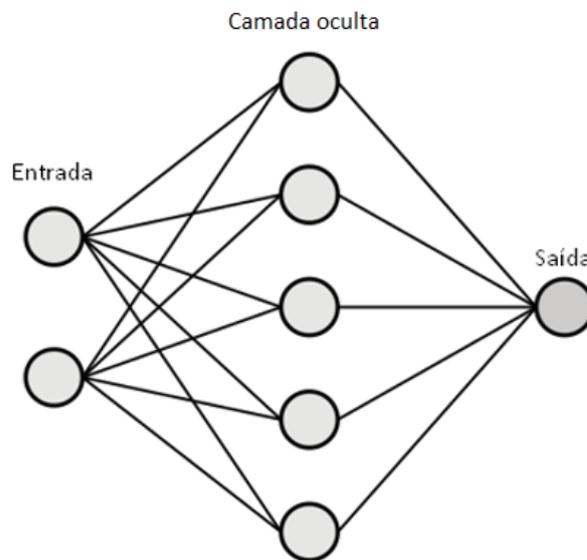


Fig. 6 Modelo Perceptron (Elaborado pelo autor).

As RNAs que utilizam apenas uma camada de neurônios têm a característica de mapear a entrada diretamente à saída, ou seja, é incapaz de formar uma representação interna. Isso impossibilita, por exemplo, que padrões de entrada com estrutura similares, que levem a saídas diferentes sejam identificados. Isso leva a uma outra arquitetura denominada *MultiLayer Perceptron* (MLP), em que há múltiplas camadas de neurônios (também chamadas de camadas ocultas),

possibilitando então a representação de dados não-lineares [24]. A Fig. 7 apresenta um modelo de MLP.

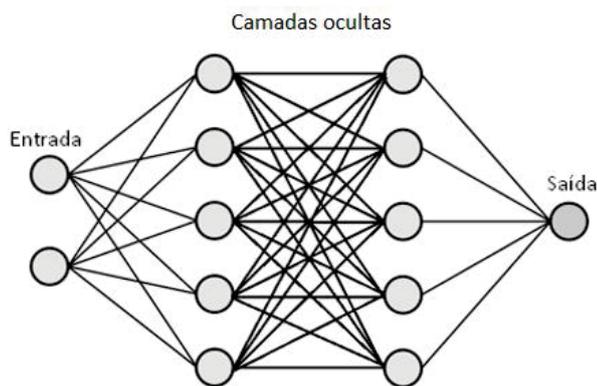


Fig. 7 Modelo MLP (Elaborado pelo autor).

Os dados em uma rede neural artificial são processados em uma ou mais iterações, ou seja, após os neurônios artificiais processá-los e atualizarem o valor dos seus respectivos pesos, pode ocorrer uma nova iteração em que os dados são novamente processados por esses neurônios, mas agora com os pesos da iteração anterior, e atualizam novamente os seus pesos. Esta iteração é denominada época [25]. O incremento de épocas em uma RNA possibilita que a rede fortaleça a relação entre os dados, porém, não implica em uma relação diretamente proporcional, já que em algum momento a RNA atingirá o máximo de seu aprendizado, naquelas condições, e não apresentará a evolução nos resultados, mesmo aumentando essas iterações.

### iii. Métricas de avaliação

As métricas de avaliação dizem respeito a forma como é medido o desempenho do algoritmo de Inteligência Artificial, isto é, avaliar se o resultado foi bom ou ruim. Em uma RNA de aprendizado supervisionado de classificação, uma das formas de se mensurar esse desempenho é a através de uma matriz de confusão. Esta permite comparar o número de classificações corretas versus as classificações previstas para cada classe [30]. A Tabela 1 apresenta a estrutura de uma matriz de confusão.

Matriz de confusão		Classe prevista	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Tabela 1 Matriz de Confusão [30].

A partir da matriz de confusão é possível calcular a acurácia que o modelo obteve. A acurácia é uma métrica que calcula o percentual de acertos, ou seja, a razão entre a quantidade de acertos e o total de entradas [29]. Utilizando como base a matriz de confusão, a fórmula da acurácia pode ser expressa conforme a Eq. 7.

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP}$$

Eq. 7 Fórmula da acurácia a partir da matriz de confusão (Elaborado pelo autor).

### D. Estado da Arte

A Inteligência Artificial vem sendo amplamente utilizada na área de cobrança nos últimos anos. Uma notícia veiculada pela NSC TOTAL [8] em junho de 2020, apresenta que a IA está sendo aplicada por empresas de os todos portes e que buscam principalmente reduzir custos e melhorar o atendimento ao cliente. A maioria dos estudos analisados na área são direcionados especificamente ao atendimento da cobrança através de *chatbots*.

Os *chatbots* são programas de computador especializados em simular e processar conversas humanas, sejam escritas ou faladas, de modo que uma pessoa interaja com dispositivos digitais como se estivesse se comunicando com uma pessoa real [9]. Esses programas utilizam técnicas de aprendizado de máquina aliadas ao processamento de linguagem natural (PLN). O PLN consiste em interpretar e reproduzir computacionalmente as expressões em linguagem natural humana, como fala e a escrita, por exemplo [10]. Na prática, são agentes virtuais capazes de serem empregados em qualquer área de negócio, facilitando o autoatendimento de clientes e minimizando a interação humana, principalmente em questões mais triviais.

A Fig. 8 demonstra um exemplo de *chatbot* em um aplicativo de mensagens instantâneas em que o objetivo é a venda de automóveis. O agente virtual agiliza o atendimento, fazendo questionamentos simples relacionados à motivação do contato do cliente. Assim, se e quando necessário, o cliente é direcionado ao atendimento humano mais adequado ao assunto que deseja tratar.

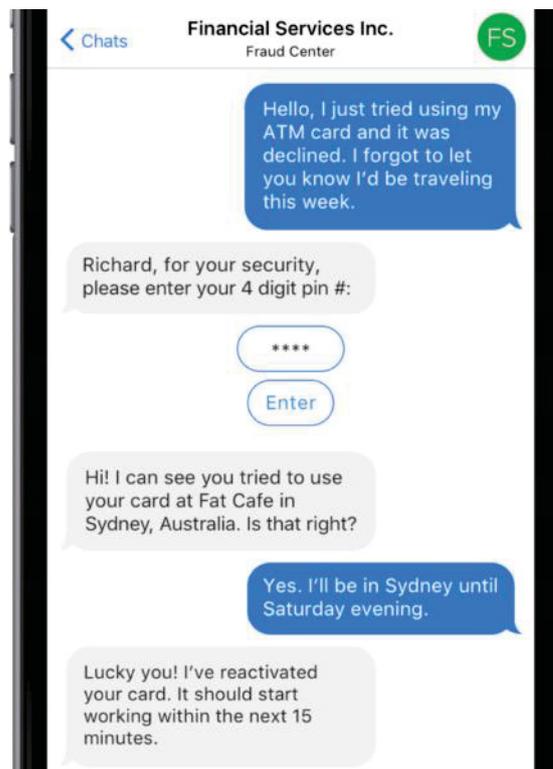


Fig. 8 Exemplo de *chatbot* [31].

Segundo o trabalho desenvolvido por SCHUNK [17], o uso de *chatbots* é amplamente utilizado pelas empresas como solução de automatização do canal de atendimento com o cliente. Evidenciou que a aplicação dessa tecnologia gera benefícios às empresas, principalmente ao que tange redução

de custos e aumento de produtividade. Um agente virtual possibilitou ao cliente um atendimento rápido e eficiente, principalmente de questões mais triviais e recorrentes, sem a necessidade da interação de um operador humano.

A pesquisa sobre a Inteligência Artificial aplicada na cobrança identificou também uma abordagem semelhante à proposta neste artigo. O trabalho de VAN DE GEER [18] aborda a aplicação do aprendizado de máquina para melhorar o desempenho dos acionamentos de cobrança via telefone. Conseguiu-se demonstrar que a técnica aplicada não só ampliou a capacidade de telefonemas diários, como aumentou a recuperação de dívida. Tais resultados foram alcançados através da aplicação da IA para definir as políticas de agendamentos telefônicos, levando em conta o perfil de inadimplência dos clientes.

### III. MATERIAIS E MÉTODOS

O projeto de desenvolvimento desse trabalho foi organizado em quatro etapas: entendimento do negócio, extração e higienização dos dados, definição do modelo e treino e avaliação do modelo e resultados. Na Fig. 9 é demonstrada a ordem de execução dessas etapas. Na sequência, o detalhamento de cada uma delas.



Fig. 9 Etapas de desenvolvimento do projeto (Elaborado pelo autor).

#### A. Definição do Objeto de Estudo

A primeira fase consistiu em definir o público a ser avaliado. Para este trabalho, os dados foram obtidos de uma empresa multinacional do ramo de Telecomunicações que utiliza a técnica de régua de cobrança para acionar os seus clientes inadimplentes. Nesta empresa, os clientes percorrem uma régua de cobrança baseada no atraso da dívida, em dias, onde eventos são disparados conforme a evolução deste atraso. Os canais são diversificados e são aplicados em marcos fixos da régua de cobrança, baseados na estratégia definida naquele momento pela empresa.

Ao avaliar o processo atual de cobrança dessa empresa, identificou-se que os eventos de cobrança seguem uma estratégia única para todos os clientes e foi baseada em um estudo de comportamento realizado no passado. Dessa forma, viu-se a possibilidade de clientes estarem recebendo acionamentos sem necessidade, bem como clientes mais propensos à inadimplência não terem o estímulo devido para que efetuem o pagamento. Definiu-se então como o objeto de

estudo uma massa de clientes desta empresa que sofreu acionamentos pela régua de cobrança.

Os dados obtidos foram de um mês específico, não divulgado, contendo as informações expressas na Tabela 2. Os critérios para a escolha das variáveis foram baseados nos fatores que influenciam os indicadores de inadimplência demonstrado pelo IBGE [5], com algumas restrições impostas pela empresa. Foi delimitado, por exemplo, apenas clientes de município de Curitiba, Paraná, haja visto que outros municípios possuíam categoria de produtos diferentes e que poderiam interferir nos modelos, e esta variável não foi disponibilizada para o experimento.

Campo	Descrição
CLIENTE	Identificação Genérica do Cliente
DIAS_PAGTO	Quantidade em dias da diferença entre o vencimento da dívida e a data de pagamento (410 - Sem pagamento)
SEXO	Sexo do cliente (M - Masculino, F - Feminino, Vazio - Não informado)
IDADE	Idade em anos do cliente
SALDO	Saldo total da dívida
TEMPO_BASE	Tempo em meses de relacionamento do cliente com a empresa
DAUTO	Forma de pagamento em débito automático (S - Sim, N - Não)
EVENTO1	Primeiro evento da régua com 3 dias de atraso
EVENTO2	Segundo evento da régua com 5 dias de atraso
EVENTO3	Terceiro evento da régua com 16 dias de atraso
EVENTO4	Quarto evento da régua com 23 dias de atraso
EVENTO5	Quinto evento da régua com 30 dias de atraso
PAGOU	Indicativo de pagamento da dívida (1 - Pagou, 0 - Não pagou)

Tabela 2 Descrição dos dados utilizados no estudo (Elaborado pelo autor).

#### B. Tratamento dos Dados

Na etapa seguinte – Tratamento dos Dados – os dados foram extraídos, higienizados e padronizados para a utilização em um algoritmo de aprendizado de máquina. A base foi extraída de um banco de dados Oracle através de uma consulta SQL e armazenada em um arquivo do tipo texto. Foram extraídos ao todo 882 registros. No processo de extração já foram realizados alguns filtros. A delimitação do município e a remoção de registros com dados faltantes foram tratadas na própria consulta SQL.

A Tabela 3 apresenta uma amostra dos dados coletados, em sua condição original, ou seja, ainda sem a higienização e padronização dos dados.

CLIENTE	DIAS_PAGTO	SEXO	IDADE	SALDO	TEMPO_BASE	DAUTO	EVENTO1	EVENTO2	EVENTO3	EVENTO4	EVENTO5	PAGOU
6171	18	F	45	105,64	34	N	1	1	1	0	0	1
651	410	F	34	99,99	2	N	1	1	1	1	1	0
3	410	F	35	22,66	1	N	1	1	1	1	1	0
6091	-6	F	23	68,04	33	N	0	0	0	0	0	1
3468	9	F	52	174,99	12	N	1	1	0	0	0	1
865	410	F	31	109,99	2	N	1	0	0	0	0	0
381	37	F	66	46,66	1	N	1	0	0	0	0	1
1407	8	F	31	101,74	4	N	1	0	0	0	0	1
1460	23	F	20	104,52	4	N	1	1	1	0	0	1

Tabela 3 Amostra dos dados coletados (Elaborado pelo autor).

O arquivo com os dados foi carregado no ambiente do Google Colab, plataforma escolhida para a orquestração dos experimentos, por ser gratuita e on-line. Esta ferramenta permite criar documentos interativos com trechos de códigos de programação, executar os códigos em tempo real e exibir os resultados das execuções [26]. A linguagem de programação nesta plataforma é o Python e o ambiente funciona a partir do próprio navegador de internet, sem a necessidade da instalação de componentes na máquina local.

Além disso, o Google Colab já possui a biblioteca TensorFlow instalada, a qual oferece recursos e ferramentas para o desenvolvimento de programas de aprendizado de máquina [19]. A Fig. 10 apresenta o ambiente de trabalho do Google Colab. Nela é possível ver, de cima para baixo, o bloco de texto, o código em Python e o resultado da execução deste código.



Fig. 10 Ambiente de trabalho do Google Colab (Elaborado pelo autor).

A primeira ação sobre os dados foi observar a correlação no mundo real existente entre os dados. Isso permitiu identificar um cenário indesejável para a rede neural artificial. A variável DIAS\_PAGTO possuía relação direta com a PAGOU (a classe desse conjunto de dados). Todos os registros com a variável DIAS\_PAGTO = “410” possuíam a classe com valor = “0”. Traduzindo para o problema do mundo real, clientes que não efetuaram pagamento, possuíam um valor padrão na variável DIAS\_PAGTO. Desta forma, se fosse utilizada pelo algoritmo, induziria o comportamento da

Outra condição observada foi que coluna CLIENTE possui um valor único para cada registro. Sendo assim, foi definida como o índice dessa base, com o objetivo apenas de identificar os registros, sem participar do processamento da RNA.

Para o uso das RNAs os dados precisam ser normalizados (transformados para valores entre 0 e 1). Caso a normalização não seja realizada, os dados de entrada terão um efeito adicional sobre o neurônio, levando a decisões erradas, recomenda [7]. A Eq. 8 apresenta a fórmula para a normalização, onde  $x$  é o valor a ser normalizado,  $[x_{max}, x_{min}]$  é a variação do valor de  $x$  e  $[d1, d2]$  limite ao qual o valor de  $x$  será reduzido.

$$y = \frac{(x - x_{min})(d2 - d1)}{x_{max} - x_{min}} + d1$$

Eq. 8 Fórmula da Normalização [7]

Expresso este comportamento da RNA, foi identificado no conjunto de dados que os atributos SEXO e DAUTO possuíam valores no formato de texto, com valores possíveis “F” ou “M” e “S” ou “N”, respectivamente. Para adequá-los ao formato ideal para a RNA, os valores foram substituídos por 1 e 0, desta forma:

- SEXO: F => 0 e M => 1;
- DAUTO: S => 1 e N => 0.

Além desse cenário, também foram ajustadas as variáveis numéricas IDADE, SALDO e TEMPO\_BASE, uma vez que estas não possuíam valores dentro do intervalo entre 0 e 1. Neste caso, foi aplicada a normalização dos dados em cada uma dessas variáveis. Para finalizar a adequação dos dados, todo o conjunto de dados, inclusive a classe, foram convertidas para decimal (*float*). A Tabela 4 apresenta uma amostra do conjunto de dados após os ajustes realizados.

CLIENTE	SEXO	IDADE	SALDO	TEMPO_BASE	DAUTO	EVENTO1	EVENTO2	EVENTO3	EVENTO4	EVENTO5	PAGOU
6171	0.0	0.4891304347826087	0.1873381805284625	0.38202247191011235	0.0	1.0	1.0	1.0	0.0	0.0	1.0
651	0.0	0.3695652173913043	0.17731867352367442	0.02247191011235955	0.0	1.0	1.0	1.0	1.0	1.0	0.0
3	0.0	0.3804347826086957	0.04018442986345097	0.011235955056179775	0.0	1.0	1.0	1.0	1.0	1.0	0.0
6091	0.0	0.25	0.12065969143465155	0.3707865168539326	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3468	0.0	0.5652173913043478	0.3103209788969676	0.1348314606741573	0.0	1.0	1.0	0.0	0.0	0.0	1.0
865	0.0	0.33695652173913043	0.1950523142401135	0.02247191011235955	0.0	1.0	0.0	0.0	0.0	0.0	0.0
381	0.0	0.717391304347826	0.08274516758290477	0.011235955056179775	0.0	1.0	0.0	0.0	0.0	0.0	1.0
1407	0.0	0.33695652173913043	0.18042206064905125	0.0449438202247191	0.0	1.0	0.0	0.0	0.0	0.0	1.0
1460	0.0	0.21739130434782608	0.1853520127682213	0.0449438202247191	0.0	1.0	1.0	1.0	0.0	0.0	1.0
1783	0.0	0.2391304347826087	0.19499911331796418	0.056179775280898875	0.0	1.0	1.0	1.0	0.0	0.0	1.0
1292	0.0	0.45652173913043476	0.21276822131583617	0.0449438202247191	0.0	1.0	1.0	1.0	0.0	0.0	0.0
1419	0.0	0.6739130434782609	0.24591239581486077	0.0449438202247191	0.0	1.0	1.0	1.0	1.0	0.0	0.0
1952	0.0	0.6086956521739131	0.40782053555594966	0.06741573033707865	0.0	0.0	0.0	0.0	0.0	0.0	1.0
402	0.0	0.5869565217391305	0.22784181592480934	0.011235955056179775	0.0	1.0	0.0	0.0	0.0	0.0	0.0
6559	0.0	0.43478260869565216	0.2787728320624224	0.449438202247191	0.0	1.0	1.0	0.0	0.0	0.0	1.0

Tabela 4 Amostra dos dados após os ajustes e formatação (Elaborado pelo autor).

rede em entender que, sempre que um cliente tem o DIAS\_PAGTO = “410”, é um cliente que não vai pagar, diminuindo ou até mesmo desconsiderando a relevância dos demais atributos. Portanto, optou-se por remover esta variável do conjunto de dados.

### C. Definição, Teste e Avaliação do Modelo

Com os dados normalizados, a etapa seguinte foi a de definição, teste e avaliação do modelo de rede neural artificial. A primeira tarefa foi avaliar a frequência das classes no

conjunto de dados, para então determinar a proporção em que os dados seriam separados entre a massa de treino e teste.

A massa de treino são os dados que serão utilizados para treinar o modelo, ou seja, é o conjunto de atributos e as classes (resultado observado) que serão submetidos ao modelo para que esse gere as relações e crie o aprendizado. É importante dividir esses dados em uma proporção em que a RNA tenha dados suficientes para identificar corretamente a correlação entre os dados e as classes.

A Tabela 5 apresenta as frequências das classes no conjunto de dados. A proporção entre as classes 1 e 0 era de 69% e 31%, respectivamente.

PAGOU	FREQ.
1.0	612
0.0	270

Tabela 5 Frequência das classes no conjunto de dados (Elaborado pelo autor).

Considerando que a classe 1 representa aproximadamente 70% da base completa, optou-se por dividir a base em 90% para treino e 10% para teste (793 e 89 instâncias, respectivamente). Isso porque a quantidade de dados da base completa é relativamente baixa para o treinamento da rede. Então se priorizou um maior volume de dados para treino, a fim de garantir o melhor aprendizado do modelo. A separação

Conjunto de Dados	Tipo	Atributos
X0_train	Dados de treino	Todos, exceto PAGOU
X1_train	Dados de treino	SEXO, IDADE, SALDO, TEMPO_BASE e DAUTO
X2_train	Dados de treino	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO e EVENTO1
X3_train	Dados de treino	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1 e EVENTO2
X4_train	Dados de treino	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1, EVENTO2 e EVENTO3
X5_train	Dados de treino	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1, EVENTO2, EVENTO3 e EVENTO4
X0_test	Dados de teste	Todos, exceto PAGOU
X1_test	Dados de teste	SEXO, IDADE, SALDO, TEMPO_BASE e DAUTO
X2_test	Dados de teste	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO e EVENTO1
X3_test	Dados de teste	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1 e EVENTO2
X4_test	Dados de teste	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1, EVENTO2 e EVENTO3
X5_test	Dados de teste	SEXO, IDADE, SALDO, TEMPO_BASE, DAUTO, EVENTO1, EVENTO2, EVENTO3 e EVENTO4
Y_train	Classes de treino	PAGOU
Y_test	Classes de teste	PAGOU

Tabela 7 Composição dos dados de teste e treino (Elaborado pelo autor).

dos dados entre teste e treino foi realizada utilizando-se a função *train\_test\_split* da biblioteca *scikit-learn*.

A biblioteca *scikit-learn* traz uma série de ferramentas para análise de dados e aprendizado de máquina na linguagem Python. Dentre estas ferramentas, está a *train\_test\_split*, uma função que facilita a divisão do conjunto de dados em amostras de treino e teste. Com um único comando, a função realiza a divisão aleatória dos dados, retornando os conjuntos de treino e teste (conforme a proporção especificada para) e as respectivas classes em vetores separados [27].

A Tabela 6 apresenta como ficou a proporção das classes, nas bases de treino e teste. A proporção das classes manteve-se próxima à da base completa: 69% para a classe 1 e 31% para a classe 0 na base de treino; e 71% para a classe 1 e 29% para a classe 0 na base de teste.

Base de Treino		Base de Teste	
PAGOU	FREQ.	PAGOU	FREQ.
1.0	549	1.0	63
0.0	244	0.0	26

Tabela 6 Proporção das classes nas bases de treino e teste (Elaborado pelo autor).

Como o intuito do trabalho foi avaliar a necessidade ou não de enviar cada evento de uma régua de cobrança, foi necessário subdividir os dados de treino obtidos em frações, de modo que cada uma contivesse apenas os atributos anteriores àquele evento analisado. Por exemplo: para analisar a necessidade de envio do EVENTO1, foi necessário considerar apenas as variáveis SEXO, IDADE, SALDO, TEMPO\_BASE e DAUTO, já que no mundo real esses são os dados conhecidos até o momento de envio deste evento. Para o EVENTO2, foram considerados os dados do EVENTO1, mais o próprio atributo EVENTO1 e assim por diante.

A Tabela 7 ilustra como ficou a divisão destes dados de treino e teste.

Para a criação do modelo de RNA foi utilizada a API (*Application Programming Interface*, ou Interface de Programação de Aplicação, em português) Keras. É uma API de aprendizado de máquina especializada em RNAs, desenvolvida em Python e com integração com a plataforma Tensorflow. Ela possibilita o desenvolvimento de experimentos utilizando redes neurais artificiais de forma mais ágil e prática, através de instruções predefinidas destinadas à estruturação, treino e avaliação de modelos [28].

Seguindo a ordem lógica de execução de uma régua de cobrança, o primeiro conjunto de dados avaliado no modelo foi *X1\_train*, destinado a avaliar a necessidade de envio do EVENTO1. Para isso, inicialmente foi construído um modelo com uma camada oculta contendo 100 neurônios, uma camada de saída com 1 neurônio, função de ativação Sigmoide (já que a saída desejada será 0 ou 1) em ambas as camadas e uso de *bias* (já que existiam atributos com valor 0).

O primeiro modelo serviu como ponto de partida para ter as primeiras impressões sobre o comportamento da rede neural artificial sobre os dados. Os resultados gerados foram visualizados em uma matriz de confusão e, a partir desta, efetuou-se o cálculo da acurácia.

A partir da acurácia obtida no primeiro modelo, viu-se a necessidade de ajustar os parâmetros da RNA e reexecutá-lo novamente. Esse processo foi executado repetidas vezes, a fim de se obter o modelo que resultasse na melhor acurácia para os conjuntos de dados. Ao término dessas iterações, chegou-se ao modelo com melhor desempenho na predição dos dados.

A tarefa seguinte consistiu em comparar as classes previstas para cada um dos eventos com os resultados conhecidos. Foi adotada a seguinte premissa: os casos previstos com valor 1 (classe que indica pagamento do cliente), não haveria necessidade do envio daquele evento. Já os que foram previstos com a classe de valor 0, ou seja, não efetuariam pagamento, implicaria na necessidade de envio daquele evento para estímulo ao pagamento.

Desta forma, para cada evento, foram avaliados os seguintes cenários:

1. ENVIO DEVIDO: A quantidade de eventos que seriam enviados considerando a predição da classe 0 correta *versus* a quantidade de eventos enviados observada em que não houve pagamento. Este é o cenário em que o envio do evento foi devido;
2. ENVIO INDEVIDO: A quantidade de eventos que seriam enviados considerando a predição da classe 0 incorreta *versus* a quantidade de eventos enviados observada em que houve pagamento. Neste caso, os eventos foram indevidos, pois o cliente não necessitaria desse estímulo para pagar;
3. NÃO ENVIO DEVIDO: A quantidade de eventos que não seriam enviados considerando a predição da classe 1 correta *versus* a quantidade de eventos não enviados observada em que houve pagamento. Uma situação devida, dado que o pagamento ocorreria sem a necessidade do evento;
4. NÃO ENVIO INDEVIDO: A quantidade de eventos que não seriam enviados considerando a predição da classe 1 incorreta *versus* a quantidade de eventos não enviados observada em que não houve pagamento. É um cenário indevido, visto que um cliente em que precisava ser estimulado a pagar, não recebeu o evento.

A partir dos cenários acima, foi possível determinar se as predições do modelo de RNA seriam capazes de realizar os acionamentos de cobrança mais eficiente que o modelo tradicional, isto é, mais eventos classificados nos cenários 1 e 3 comparados ao observado na base analisada.

Feita a transposição desses cenários em uma matriz de confusão, obteve-se a representação demonstrada na Tabela 8.

MATRIZ DE CONFUSÃO			
OBSERVADO	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
	1	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
Classes		0	1
		PREDITO	

Tabela 8 Matriz de confusão com os cenários dos resultados (Elaborado pelo autor).

#### D. Ameaças à validade

Uma das ameaças à validade do trabalho foram as limitações de parâmetros disponíveis, como por exemplo, delimitação do município e ausência da variável que representa o tipo de serviço do cliente. Essa situação pode implicar em um viés indesejado no modelo, que consequentemente poderia apresentar uma performance insatisfatória quando submetido a casos reais de outros municípios.

Por outro lado, caso isso se concretizasse, seria um ponto de atenção sobre relevância desses parâmetros ausentes, o que poderia servir para uma reavaliação do modelo e possível trabalho futuro.

Outro fator que pode ter interferido no modelo obtido foi a quantidade de amostras disponibilizadas para o estudo. Uma amostra maior daria ao modelo mais flexibilidade ao estabelecer a relação entre os dados. Além disso, a desproporção entre as classes dificulta a generalização do problema pela rede neural artificial.

A premissa adotada de que um cliente em que foi previsto o não pagamento deve receber um evento é sensível. Isso porque a única forma concreta de avaliar essa condição seria submetendo esses clientes às predições da RNA em um ambiente real e avaliando os resultados posteriores.

Uma situação possível no mundo real, por exemplo, é o cliente que mesmo tendo recebido todos os acionamentos da régua de cobrança, não efetuou o pagamento da dívida. Esses casos precisariam de uma estratégia específica de acionamento, uma vez que o envio de todos os eventos acarretaria apenas em um custo desnecessário.

## IV. RESULTADOS E DISCUSSÕES

### A. Avaliação do EVENTO1

A primeira execução do modelo partiu de parâmetros arbitrários com a finalidade de ser o ponto de partida para o modelo, para posteriormente receber refinamentos até a obtenção dos resultados mais satisfatórios ao estudo.

A Fig. 12 apresenta a matriz de confusão com os resultados dessa primeira execução. No eixo horizontal estão as classes previstas pelo modelo, enquanto na vertical estão as classes observadas (conhecidas). Pôde-se observar que este primeiro modelo foi capaz de prever bem os clientes com a classe 1 (que efetuaram pagamento), mas teve dificuldade de identificar os clientes com a classe 0 (não pagaram).

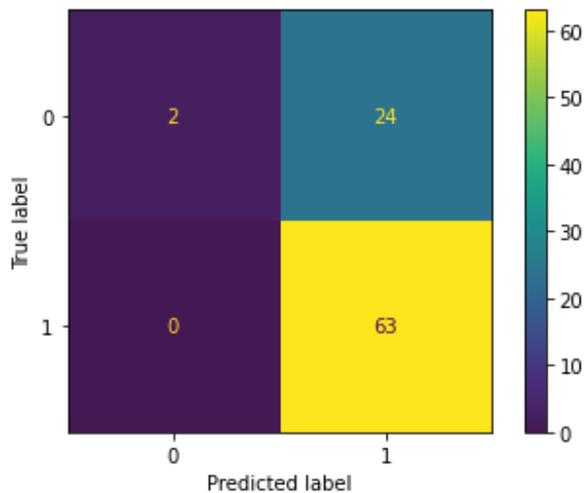


Fig. 11 Matriz de confusão da primeira execução do modelo (Elaborado pelo autor).

Outro comportamento avaliado foi a evolução da acurácia *versus* a perda do treino do modelo conforme a progressão das épocas. A Fig. 13 apresenta o gráfico dessas duas variáveis na primeira execução do modelo. Percebeu-se que a acurácia começou a apresentar melhora a partir da época 50, aproximadamente, sugerindo que um aumento na quantidade de épocas pudesse ampliar esta melhora.

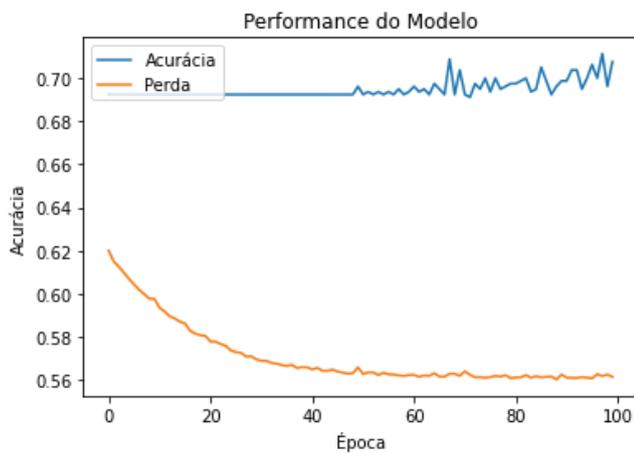


Fig. 12 Evolução da acurácia e perda no treino do modelo (Elaborado pelo autor).

A primeira alteração de parâmetro foi o aumento na quantidade de épocas, de 100 para 500. Esta alteração não gerou melhora significativa no resultado do treino. Em seguida, foi revertida a alteração da quantidade de épocas e alterada a função de ativação da Sigmoide para a Tangente Hiperbólica, o que também não apresentou melhora no modelo. Então foi incluída mais uma camada oculta contendo 100 neurônios, esta com a função de ativação Sigmoide, o que gerou um ganho de 0,01 na acurácia (1 ponto percentual) do treino. Embora o valor não seja expressivo, serviu de indicativo de melhora no modelo. Ao alterar a função de ativação da segunda camada oculta para Tangente Hiperbólica, apresentou novo ganho na acurácia.

Após esses ajustes, observou-se que o gráfico de acurácia do treino indicava tendência a melhorar conforme o aumento de épocas. Efetuou-se o incremento gradativo dessas épocas e, de fato, o modelo começou a apresentar a acurácia cada vez maior.

Todavia, ao avaliar a matriz de confusão das previsões sobre a base de testes, notou-se que a acurácia teve piora. Isso denota um comportamento chamado *overfitting*, que é basicamente quando o modelo aprende muito bem os dados de treino, mas não generaliza o aprendizado, tendo dificuldade em efetuar previsão para dados novos.

Na Tabela 9 são apresentados os primeiros experimentos e ajustes do modelo, que acarretaram nesse *overfitting*. Pode-se observar um ganho significativo na acurácia do modelo a medida em que a quantidade de épocas é ampliada.

Neurônios	Épocas	Função de Ativação	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.7150	0.6966
100	500	sigmoid	0.7163	0.7078
100	100	tanh	0.7188	0.6966
100/100	100	tanh/sigmoid	0.7276	0.6516
100/100	100	sigmoid/sigmoid	0.7163	0.6292
100/100	100	tanh/tanh	0.7314	0.6404
100/100	500	tanh/tanh	0.7377	0.6629
100/100	2000	tanh/tanh	0.7844	0.5842
100/100	5000	tanh/tanh	0.8298	0.6404
100/100	10000	tanh/tanh	0.9294	0.5842
100/100	20000	tanh/tanh	0.9836	0.6742

Tabela 9 Primeiros ajustes do modelo (Elaborado pelo autor).

Quando foi analisado o comportamento do treino considerando 20.000 épocas, foi possível visualizar esse comportamento. A Fig. 14 demonstra graficamente este cenário.

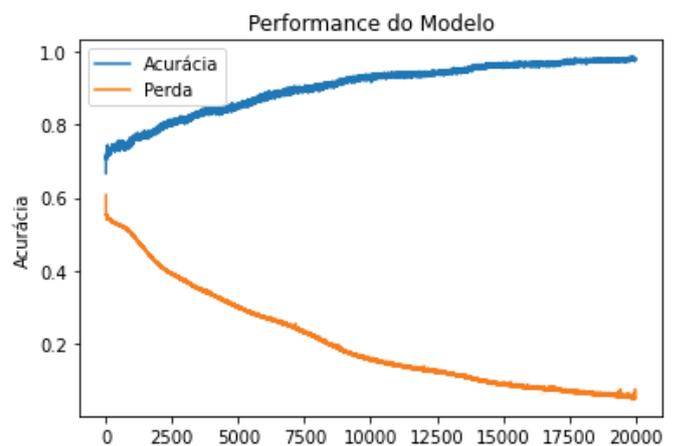


Fig. 13 Desempenho do modelo com 20 mil épocas (Elaborado pelo autor).

Nesse treinamento, a RNA atingiu a marca de 98,36% de acurácia. Entretanto, quando os dados de teste foram submetidos a esse mesmo modelo, a acurácia reduziu para 67,42% (69,66% do modelo com os parâmetros iniciais). A Fig. 15 apresenta a matriz de confusão gerada pelo modelo após avaliar os dados de testes.

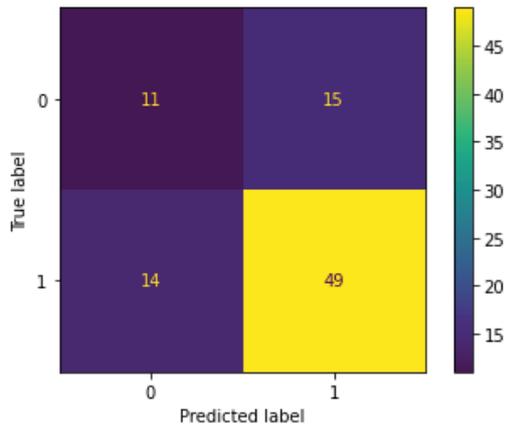


Fig. 14 Matriz de confusão dos dados de teste (Elaborado pelo autor).

Em decorrência deste comportamento, constatou-se que apenas o aumento das épocas da rede neural não apresentou resultado satisfatório. Portanto, prosseguiu-se com os ajustes de outros da parâmetros do modelo a fim de contornar o problema de *overfitting*.

A estratégia seguinte consistiu em adicionar uma camada de *dropout* na RNA. Esta camada tem a finalidade de remover neurônios da rede aleatoriamente, forçando os demais neurônios remanescentes a recalcularem seus pesos e *bias* [24]. Na prática, essa técnica força o modelo a se tornar mais robusto para a perda de qualquer evidência individual, tornando-o capaz de avaliar melhor novos dados na rede.

A primeira execução com apenas uma camada de neurônios e *dropout* resultou novamente em um aprendizado ineficiente. A rede neural atingiu 70,79% de acurácia na base de testes, porém identificou todas as instancias com a classe 1. Embora a acurácia seja relativamente alta, esse resultado foi considerado ruim, pois mostrou que a RNA não foi capaz de identificar nenhuma instância com a classe 0. A Fig. 16 apresenta a matriz de confusão dos dados de testes com esse novo modelo.

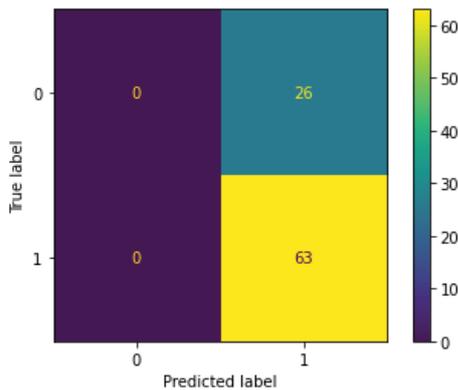


Fig. 15 Matriz de confusão dos dados de testes com *dropout* (Elaborado pelo autor).

As tentativas seguintes consistiram em avaliar o comportamento da rede após modificações nas camadas de neurônios. A primeira alteração foi a inclusão de uma nova camada de neurônios após a camada de *dropout* existente. Essa alteração surtiu efeito positivo no modelo. Além de aumentar a acurácia, tornou a rede neural mais genérica e ela foi capaz de prever corretamente instâncias de ambas as classes. A partir desse ponto, foram realizados ajustes

sucessivos nos parâmetros da rede novamente a fim de se obter um ganho na acurácia, sem comprometer a generalização conquistada no modelo.

Na Tabela 10 são apresentados os principais experimentos realizados na rede os respectivos resultados obtidos.

Neurônios	Épocas	Função de Ativação	Dropout	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.2	0.7087	0.6966
100/100	100	sigmoid	0.2	0.7125	0.7303
100/100	500	sigmoid	0.2	0.7427	0.7078
100/100	500	tanh	0.2	0.7377	0.6853
100/100	500	sigmoid	0.1	0.7339	0.6741
100/100	500	sigmoid	0.3	0.7238	0.7078
100/200	500	sigmoid	0.3	0.7390	0.6853
100/100/20	500	sigmoid	0.2	0.7377	0.6853
20/100/20	500	sigmoid	0.2	0.7289	0.6853
100/100	2000	sigmoid	0.2	0.7364	0.6853
100/100/100	100	sigmoid	0.2/0.2	0.7390	0.7415
100/100/100	500	sigmoid	0.2/0.2	0.7289	0.6853
100/100/100/20	100	sigmoid	0.2/0.2	0.7163	0.7415
100/100/100/100	100	sigmoid	0.2/0.2	0.7087	0.6404
20/100/100/20	100	sigmoid	0.2/0.2	0.7226	0.7528
10/100/100/20	100	sigmoid	0.2/0.2	0.7087	0.7415
50/100/100/20	100	sigmoid	0.2/0.2	0.7175	0.6741
20/100/100/10	100	sigmoid	0.2/0.2	0.7037	0.6629
20/100/100/20	500	sigmoid	0.2/0.2	0.7251	0.7078
20/100/200/100/20	100	sigmoid	0.2/0.2	0.7125	0.6853
20/100/200/20	100	sigmoid	0.2/0.2	0.7163	0.6853
20/200/200/20	100	sigmoid	0.2/0.2	0.7150	0.7303
20/500/500/20	100	sigmoid	0.2/0.2	0.7339	0.7415
20/500/500/20	100	sigmoid	0.2/0.3	0.7201	0.7303
20/500/500/20	100	sigmoid	0.2/0.1	0.7112	0.7303
20/1000/1000/20	100	sigmoid	0.2/0.2	0.6948	0.6853
20/100/100/20	100	sigmoid	0.2/0.2	0.7112	0.6741
20/100/100/20	100	sigmoid	0.2/0.2	0.7264	0.6741
20/100/100/100/100/20	100	sigmoid	0.2/0.2	0.7150	0.7415
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.1/0.2	0.7188	0.7078
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7150	0.7528
20/100/100/100/100/20	100	sigmoid	0.2/0.2/0.2	0.7049	0.6853
20/100/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7163	0.7303

Tabela 10 Experimentos realizados utilizando *dropout* para o EVENTO1 (Elaborado pelo autor).

Após a realização dos experimentos, o melhor resultado (maior acurácia na base de teste) foi obtido por duas configurações distintas, ambas com 100 épocas de para o treino:

1. Primeira configuração (na ordem):
  - a. Camada oculta com 20 neurônios, função de ativação Sigmoide e utilização de *bias*;
  - b. Camada oculta com 100 neurônios e função de ativação Sigmoide;
  - c. Camada com 20% de *dropout*;
  - d. Camada oculta com 100 neurônios e função de ativação Sigmoide;
  - e. Camada com 20% de *dropout*;
  - f. Camada oculta com 20 neurônios e função de ativação Sigmoide;
  - g. Camada de saída com 1 neurônios e função de ativação Sigmoide.
2. Segunda configuração (na ordem):
  - a. Camada oculta com 20 neurônios, função de ativação Sigmoide e utilização de *bias*;
  - b. Camada oculta com 100 neurônios e função de ativação Sigmoide;
  - c. Camada com 20% de *dropout*;

- d. Camada oculta com 100 neurônios e função de ativação Sigmoide;
- e. Camada com 10% de *dropout*;
- f. Camada oculta com 100 neurônios e função de ativação Sigmoide;
- g. Camada com 20% de *dropout*;
- h. Camada oculta com 100 neurônios e função de ativação Sigmoide;
- i. Camada oculta com 20 neurônios e função de ativação Sigmoide;
- j. Camada de saída com 1 neurônios e função de ativação Sigmoide.

As duas configurações atingiram a acurácia de 71,5% e 75,28% nas bases de treino e teste, respectivamente, o melhor desempenho obtido para este conjunto de dados. A Fig. 17 demonstra a matriz de confusão gerada pela predição da base de testes. Observou-se que a rede conseguiu melhorar a acurácia, sem comprometer a generalização.

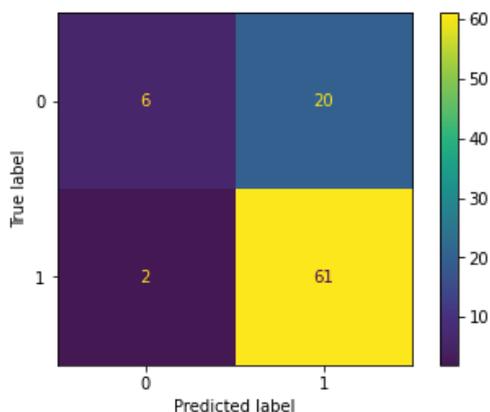


Fig. 16 Matriz de confusão do melhor resultado obtido para o EVENTO1 (Elaborado pelo autor).

Como duas configurações distintas resultaram na mesma acurácia para a base de testes, optou-se por seguir o trabalho com a primeira configuração supracitada, já que foi capaz de obter o mesmo resultado com uma rede menos complexa, ou seja, com menor quantidade de camadas e neurônios.

Com o modelo definido, o próximo passo foi comparar as predições com o observado para o primeiro conjunto de dados, referente ao EVENTO1. Na Tabela 11 é possível observar como os dados foram organizados para efetuar essa análise. Conforme o cenário, cada registro foi classificado em “ENVIO DEVIDO”, “ENVIO INDEVIDO”, “NÃO ENVIO DEVIDO” ou “NÃO ENVIO INDEVIDO”.

CLIENTE	PAGOU_OBSERVADO	PAGOU_PREDITO	EVENTO_OBSERVADO	EVENTO_PREDITO	RESULTADO_OBSERVADO	RESULTADO_PREDITO
1346	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
4237	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
2259	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
312	1	0	1	1	ENVIO INDEVIDO	ENVIO INDEVIDO
1557	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
6295	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1708	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
2968	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
3468	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
3453	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7695	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO

1474	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
1576	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6003	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8004	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5648	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7226	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
3819	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
3574	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6341	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6117	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
420	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2554	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6589	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
829	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
2464	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
5305	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8127	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
4074	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
1765	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
4125	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
307	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
6996	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8318	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
3900	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2110	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO

Tabela 11 Amostra dos dados preditos versus observados para o EVENTO1 (Elaborado pelo autor).

A partir das classes observadas do conjunto de dados de teste e suas respectivas predições, foi possível avaliar a efetividade do modelo de RNA para o EVENTO1. A Tabela 12 apresenta estes dados e a partir dela é possível identificar dois comportamentos (à esquerda os cenários observados e à direita os cenários baseados na predição da RNA):

1. Para clientes que efetuaram pagamento (classe = 1), a RNA reduziu os envios indevidos em aproximadamente 95% em relação ao praticado pela régua de cobrança tradicional;
2. Em contrapartida, para os clientes que deveriam receber o evento, a RNA apresentou uma piora de 73%, ou seja, ela deixou de disparar o evento para um cliente que deveria recebê-lo.

Observado		Predito		
Clientes que pagaram (observado)				
Instâncias	%	Instâncias	%	
ENVIO INDEVIDO	43	68,25%	2	3,17%
NÃO ENVIO DEVIDO	20	31,75%	61	96,83%
<b>Total</b>	<b>63</b>	<b>100,00%</b>	<b>63</b>	<b>100,00%</b>
Clientes que não pagaram (observado)				
Instâncias	%	Instâncias	%	
ENVIO DEVIDO	22	84,62%	6	23,08%
NÃO ENVIO INDEVIDO	4	15,38%	20	76,92%
<b>Total</b>	<b>26</b>	<b>100,00%</b>	<b>26</b>	<b>100,00%</b>

Tabela 12 Resultados consolidados da predição do EVENTO1 (Elaborado pelo autor).

Para o EVENTO1 isoladamente, portanto, a RNA não apresentou um resultado satisfatório, uma vez que ela obteve uma grande economia de disparos do evento em detrimento da eficiência de cobrança.

## B. Avaliação do EVENTO2

Para a análise do EVENTO2 foram replicados os mesmos procedimentos utilizados para o EVENTO1. Foi executada uma primeira bateria de avaliações sem o uso de *dropout* e o comportamento de tendência a *overfitting* se repetiu para este evento. A Tabela 13 apresenta as tentativas realizadas. Neste caso, o aumento das épocas até 2000 foi o suficiente para detectar o comportamento de tendência a *overfitting* da rede.

Neurônios	Épocas	Função de Ativação	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.7188	0.7415
100	500	sigmoid	0.7201	0.7415
100	100	tanh	0.7201	0.7415
100/100	100	tanh/sigmoid	0.7264	0.6966
100/100	100	sigmoid/sigmoid	0.7112	0.7078
100/100	100	tanh/tanh	0.7327	0.7303
100/100	500	tanh/tanh	0.7566	0.7078
100/100	2000	tanh/tanh	0.8298	0.7078

Tabela 13 Resultados para os modelos sem *dropout* para o EVENTO2 (Elaborado pelo autor).

Quando foram adicionadas as camadas de dropout na tentativa de inibir o *overfitting*, a RNA não gerou o efeito esperado. Mesmo trabalhando a quantidade de camadas, número de neurônios e a quantidade de épocas, os modelos persistiram em reproduzir resultados semelhantes. Novamente, os modelos conseguiram identificar bem as instâncias da classe 1, mas tiveram dificuldade em classificar os casos com a classe 0.

Dos experimentos realizados com os ajustes de parâmetros, seis apresentaram a melhor acurácia para os dados de teste: 74,15%. A Tabela 14 demonstra os principais parâmetros testados para a RNA.

Neurônios	Épocas	Função de Ativação	Dropout	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.2	0.7074	0.7078
100/100	100	sigmoid	0.2	0.7037	0.7078
100/100	500	sigmoid	0.2	0.7390	0.7303
100/100	500	tanh	0.2	0.7465	0.7078
100/100	500	sigmoid	0.1	0.7364	0.7303
100/100	500	sigmoid	0.3	0.7352	0.7303
100/200	500	sigmoid	0.3	0.7276	0.7303
100/100/20	500	sigmoid	0.2	0.7390	0.7303
20/100/20	500	sigmoid	0.2	0.7264	0.7191
100/100	2000	sigmoid	0.2	0.7528	0.7415
100/100	3000	sigmoid	0.2	0.7465	0.7415
100/100/100	100	sigmoid	0.2/0.2	0.7301	0.6853
100/100/100	500	sigmoid	0.2/0.2	0.7415	0.7303
100/100/100/20	100	sigmoid	0.2/0.2	0.7251	0.7078
100/100/100/100	100	sigmoid	0.2/0.2	0.7314	0.7078
20/100/100/20	100	sigmoid	0.2/0.2	0.7188	0.7303
10/100/100/20	100	sigmoid	0.2/0.2	0.6999	0.7303
50/100/100/20	100	sigmoid	0.2/0.2	0.7339	0.7191
20/100/100/10	100	sigmoid	0.2/0.2	0.7352	0.7303
20/100/100/20	500	sigmoid	0.2/0.2	0.7226	0.7191
20/100/200/100/20	100	sigmoid	0.2/0.2	0.7125	0.6853
20/100/200/20	100	sigmoid	0.2/0.2	0.7125	0.7078
20/200/200/20	100	sigmoid	0.2/0.2	0.7163	0.6966
20/500/500/20	100	sigmoid	0.2/0.2	0.7390	0.7078
20/500/500/20	100	sigmoid	0.2/0.3	0.7314	0.7191
20/500/500/20	100	sigmoid	0.2/0.1	0.7465	0.7078
20/1000/1000/20	100	sigmoid	0.2/0.2	0.7100	0.7415
20/100/100/100/100/20	100	sigmoid	0.2/0.2	0.7276	0.7191
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.1/0.2	0.7264	0.7303
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7314	0.7303
20/100/100/100/100/20	100	sigmoid	0.2/0.2/0.2	0.7125	0.7078
20/100/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7213	0.7078

Tabela 14 Experimentos realizados utilizando *dropout* para o EVENTO2 (Elaborado pelo autor).

Entre os modelos que apresentam o melhor resultado, optou-se pela utilização do modelo que teve maior número de acertos para a classe 0 (classe que o algoritmo teve maior dificuldade). Este modelo foi treinado com 3.000 (três mil) épocas e utilizou a seguinte configuração, na ordem:

- Camada oculta com 100 neurônios, função de ativação Sigmoid e utilização de *bias*;
- Camada oculta com 100 neurônios e função de ativação Sigmoid;
- Camada com 20% de *dropout*;

d. Camada de saída com 1 neurônios e função de ativação Sigmoid.

Na Fig. 18 é possível observar a matriz de confusão do modelo com o melhor resultado.

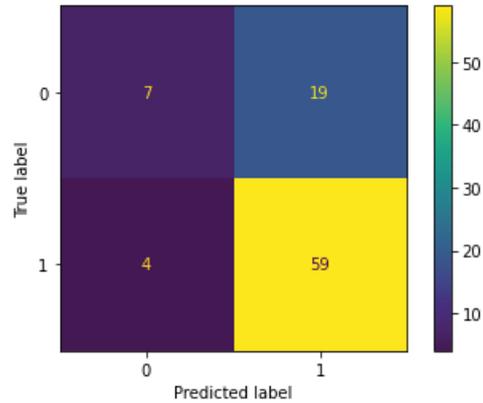


Fig. 17 Matriz de confusão do melhor resultado obtido para o EVENTO2 (Elaborado pelo autor).

Os resultados desse modelo foram tabulados para a comparação entre os dados observados e preditos para o EVENTO2. Do mesmo modo que foi realizado com para o EVENTO1, as instâncias foram classificadas conforme o cenário em que se enquadraram.

A Tabela 15 exibe uma amostra desses dados já classificados.

CLIENTE	PAGOU_OBSERVADO	PAGOU_PREDITO	EVENTO_OBSERVADO	EVENTO_PREDITO	RESULTADO_OBSERVADO	RESULTADO_PREDITO
1346	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
4237	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2259	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
312	1	0	1	1	ENVIO INDEVIDO	ENVIO INDEVIDO
1557	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
6293	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1708	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3468	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
3453	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7695	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
1474	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1576	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6003	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8004	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5648	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7226	1	1	1	0	ENVIO INDEVIDO	ENVIO INDEVIDO
3819	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3574	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6341	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6117	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
420	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2554	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6589	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
829	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
2464	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5305	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8127	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
4074	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
1765	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
4125	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
307	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
6996	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8318	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
3900	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2110	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
483	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
272	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2224	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2299	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
4252	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
2915	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
186	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO

Tabela 15 Amostra dos dados preditos *versus* observados para o EVENTO2 (Elaborado pelo autor).

Na Tabela 16 contém a consolidação dessas classificações (à esquerda os cenários observados e à direita os cenários baseados na predição da RNA).

Observado			Predito		
Clientes que pagaram (observado)					
Instâncias	%		Instâncias	%	
NÃO ENVIO DEVIDO	34	53,97%	ENVIO INDEVIDO	2	3,17%
ENVIO INDEVIDO	29	46,03%	NÃO ENVIO DEVIDO	61	96,83%
<b>Total</b>	<b>63</b>	<b>100,00%</b>	<b>Total</b>	<b>63</b>	<b>100,00%</b>
Clientes que não pagaram (observado)					
Instâncias	%		Instâncias	%	
ENVIO DEVIDO	16	61,54%	ENVIO DEVIDO	5	19,23%
NÃO ENVIO INDEVIDO	10	38,46%	NÃO ENVIO INDEVIDO	21	80,77%
<b>Total</b>	<b>26</b>	<b>100,00%</b>	<b>Total</b>	<b>26</b>	<b>100,00%</b>

Tabela 16 Resultados consolidados da predição do EVENTO2 (Elaborado pelo autor).

O desempenho da rede neural artificial para o EVENTO2 foi semelhante ao observado para o EVENTO1. O algoritmo foi capaz de reduzir em 93% os envios indevidos do evento, mas em contrapartida deixou de enviar 81% dos casos em que havia necessidade do evento, uma piora de 69% em relação ao observado pela régua de cobrança tradicional.

### C. Avaliação do EVENTO3

Na sequência, iniciou-se os treinos e ajustes da rede neural artificial para o EVENTO3. De modo semelhante aos modelos anteriores, o intuito nesta etapa foi criar um modelo que fosse capaz de prever se um cliente iria efetuar o pagamento, considerando as variáveis existentes, mais o recebimento ou não observado dos eventos anteriores. Os primeiros modelos treinados também não consideraram *dropout*, visto que houve o incremento de um novo atributo no conjunto de dados e, então, não era possível antever o comportamento da rede neste cenário.

Os resultados dos primeiros experimentos são apresentados na Tabela 17. Embora tenha sido adicionada uma nova variável ao conjunto de dados, o comportamento foi semelhante ao observado nos eventos anteriores: obteve-se melhora na acurácia durante os treinos, mas isso não implicou na melhora na predição dos dados de teste.

Neurônios	Épocas	Função de Ativação	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.7163	0.7303
100	500	sigmoid	0.7150	0.7415
100	1000	sigmoid	0.7427	0.6853
100	100	tanh	0.7213	0.7303
100/100	100	tanh/sigmoid	0.7301	0.6966
100/100	100	sigmoid/sigmoid	0.7125	0.7078
100/100	100	tanh/tanh	0.7352	0.7078
100/100	500	tanh/tanh	0.7516	0.7078
100/100	2000	tanh/tanh	0.8777	0.6404

Tabela 17 Resultados para os modelos sem *dropout* para o EVENTO3 (Elaborado pelo autor).

Nos treinos seguintes foi considerada uma ou mais camadas de *dropout*, similar ao praticado para os eventos anteriores. Os modelos treinados apresentaram comportamento semelhante ao observado para o EVENTO2, embora alguns ajustes tenham surtido efeito no conjunto de dados de treino, não generalizou o problema suficientemente para que a mesma melhora nos dados de teste. Também foram realizadas tentativas com números de épocas mais expressivos, porém os resultados começaram novamente a apresentar tendência ao *overfitting*, mesmo variando a estratégia com *dropout*. A seguir, na Tabela 18 são demonstrados os principais experimentos realizados para o EVENTO3 utilizando *dropout*.

Neurônios	Épocas	Função de Ativação	Dropout	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.2	0.7137	0.7078
100/100	100	sigmoid	0.2	0.7251	0.7303
100/100	500	sigmoid	0.2	0.7478	0.7303
100/100	1000	sigmoid	0.2	0.7415	0.7415
100/100	2000	sigmoid	0.2	0.7402	0.7528
100/100	5000	sigmoid	0.2	0.7566	0.7415
100/100	2000	tanh	0.2	0.7705	0.7191
20/100/20	100	sigmoid	0.2	0.7100	0.6966
20/100/20	500	sigmoid	0.2	0.7415	0.7191
20/100/20	1000	sigmoid	0.2	0.7453	0.7303
20/100/20	2000	sigmoid	0.2	0.7415	0.7191
100/100	2000	sigmoid	0.1	0.7453	0.7528
100/100	2000	sigmoid	0.3	0.7402	0.7528
100/100/100	2000	sigmoid	0.2	0.7516	0.7191
100/100/100	100	sigmoid	0.2/0.2	0.7201	0.7191
100/100/100	500	sigmoid	0.2/0.2	0.7377	0.7303
100/100/100	1000	sigmoid	0.2/0.2	0.7491	0.7191
100/100/100	20000	sigmoid	0.2/0.2	0.8802	0.6741
10/100/100/20	100	sigmoid	0.2/0.2	0.7137	0.7191
50/100/100/20	100	sigmoid	0.2/0.2	0.7062	0.7191
20/100/100/10	100	sigmoid	0.2/0.2	0.7150	0.7191
20/100/100/20	500	sigmoid	0.2/0.2	0.7415	0.7303
20/100/200/100/20	100	sigmoid	0.2/0.2	0.7188	0.7079
20/100/200/20	100	sigmoid	0.2/0.2	0.7327	0.6966
20/200/200/20	100	sigmoid	0.2/0.2	0.7201	0.7303
20/500/500/20	100	sigmoid	0.2/0.2	0.7251	0.6854
20/500/500/20	100	sigmoid	0.2/0.3	0.7213	0.7191
20/500/500/20	100	sigmoid	0.2/0.1	0.7226	0.7191
20/1000/1000/20	100	sigmoid	0.2/0.2	0.6923	0.7079
20/100/100/100/100/20	100	sigmoid	0.2/0.2	0.7238	0.7191
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.1/0.2	0.7100	0.7100
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7125	0.7125
20/100/100/100/100/20	100	sigmoid	0.2/0.2/0.2	0.7163	0.7191
20/100/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.7175	0.7303

Tabela 18 Experimentos realizados utilizando *dropout* para o EVENTO3 (Elaborado pelo autor).

A partir desses experimentos, três modelos obtiveram a melhor acurácia sobre os dados de teste, 75,28%. Todos com parametrização idênticas, exceto pela taxa de *dropout* utilizada (10%, 20% e 30%), e com treino realizado em 2.000 (duas mil) épocas. A seguir os parâmetros desses modelos:

- Camada oculta com 100 neurônios, função de ativação Sigmoide e utilização de *bias*;
- Camada oculta com 100 neurônios e função de ativação Sigmoide;
- Camada com (10%, 20% ou 30%) de *dropout*;
- Camada de saída com 1 neurônios e função de ativação Sigmoide.

Na Fig. 19 é apresentada a matriz de confusão gerada por esse modelo.

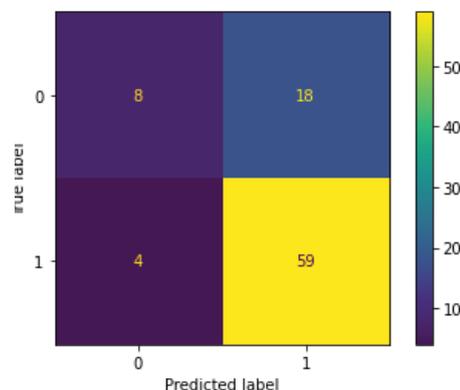


Fig. 18 Matriz de confusão do melhor resultado obtido para o EVENTO3 (Elaborado pelo autor).

As previsões geradas pelo modelo com melhor desempenho foram comparadas com os dados observado para o EVENTO3. A partir de então, os dados foram tabulados e classificados conforme os cenários em que se enquadraram, a fim de avaliar o desempenho do modelo frente ao problema do mundo real. A Tabela 19 apresenta uma amostra dos dados tabulados e já classificados do EVENTO3.

CLIENTE	PAGOU_OBSERVADO	PAGOU_PREDITO	EVENTO_OBSERVADO	EVENTO_PREDITO	RESULTADO_OBSERVADO	RESULTADO_PREDITO
1346	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
4237	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2259	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
312	1	0	0	1	NÃO ENVIO DEVIDO	ENVIO INDEVIDO
1557	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
6299	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
1708	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
2968	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3468	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3453	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7695	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1474	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1576	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
6003	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8004	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
5648	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7226	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3819	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3574	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6341	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6117	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
420	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2554	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6589	0	1	1	0	ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
829	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
2464	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5305	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8127	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO

Tabela 19 Amostra dos dados preditos versus observados para o EVENTO3 (Elaborado pelo autor).

Esses dados foram agrupados e comparados, conforme apresenta a Tabela 20 a seguir. À esquerda, os cenários observados e à direita os cenários baseados na predição da RNA.

Observado		Predito	
Clientes que pagaram (observado)			
<b>Instâncias</b>	<b>%</b>	<b>Instâncias</b>	<b>%</b>
ENVIO INDEVIDO	14 22,22%	ENVIO INDEVIDO	4 6,35%
NÃO ENVIO DEVIDO	49 77,78%	NÃO ENVIO DEVIDO	59 93,65%
<b>Total</b>	<b>63 100,00%</b>	<b>Total</b>	<b>63 100,00%</b>
Clientes que não pagaram (observado)			
<b>Instâncias</b>	<b>%</b>	<b>Instâncias</b>	<b>%</b>
ENVIO DEVIDO	20 76,92%	ENVIO DEVIDO	8 30,77%
NÃO ENVIO INDEVIDO	6 23,08%	NÃO ENVIO INDEVIDO	18 69,23%
<b>Total</b>	<b>26 100,00%</b>	<b>Total</b>	<b>26 100,00%</b>

Tabela 20 Resultados consolidados da predição do EVENTO3 (Elaborado pelo autor).

O comportamento se repetiu para o EVENTO3: a rede neural artificial apresentou uma melhora em relação a não disparar eventos desnecessários, contudo continuou a pecar em não enviar eventos nos casos em que havia a necessidade. Embora os cenários sejam semelhantes em relação aos eventos anteriores, notou-se uma pequena atenuação dessas diferenças. Para o EVENTO3, a redução dos envios indevidos foi de 71% e dos envios devidos de 60%.

#### D. Avaliação do EVENTO4

A avaliação seguinte foi sob os dados do EVENTO4, ou seja, todos os atributos das instâncias, mais as informações observadas para os eventos anteriores. Foi aplicada a mesma estratégia de avaliação e ajuste dos modelos, iniciando-se pelas execuções sem *dropout*. Na Tabela 21 são apresentados os resultados desta primeira rodada de experimentos para o EVENTO4.

Neurônios	Épocas	Função de Ativação	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.7327	0.8315
100	500	sigmoid	0.7327	0.8315
100	1000	sigmoid	0.7516	0.8427
100	2000	sigmoid	0.7617	0.7978
100	1000	tanh	0.7667	0.7753
100/100	1000	sigmoid	0.7667	0.8090
20/100	1000	sigmoid	0.7491	0.8315
20/100	300	sigmoid	0.7415	0.8315
10/100	1000	sigmoid	0.7604	0.8090
100/20	1000	sigmoid	0.7629	0.7978
20/100/20	1000	sigmoid	0.7604	0.7865

Tabela 21 Resultados para os modelos sem *dropout* para o EVENTO4 (Elaborado pelo autor).

O incremento de mais uma variável no conjunto de dados para a avaliação do EVENTO4, gerou uma melhora perceptível nos experimentos de maneira geral, quando comparados com os realizados para os eventos anteriores. Isso indicou que o acréscimo de variáveis referente aos eventos passados auxiliou a rede neural artificial a desenvolver relações mais robustas e, conseqüentemente, prever melhor as classes na massa de dados de teste.

A primeira leva de experimentos sem a utilização de *dropout* já apresentou o melhor conjunto de parâmetros para o processamento dos dados do EVENTO4. O modelo conseguiu atingir 84,27% de acurácia sobre os dados de teste. Embora o mesmo resultado tenha sido atingido por outro modelo utilizando *dropout*, novamente optou-se pelo primeiro dado que possui uma configuração mais simples e com melhor desempenho. Abaixo, na Tabela 22, seguem os experimentos realizados com a utilização do *dropout*.

Neurônios	Épocas	Função de Ativação	Dropout	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.2	0.7201	0.8202
100	200	sigmoid	0.2	0.7301	0.8315
100	500	sigmoid	0.2	0.7377	0.8427
100	1000	sigmoid	0.2	0.7440	0.8090
100/100	100	sigmoid	0.2	0.7238	0.8315
100/100	500	sigmoid	0.2	0.7554	0.8202
20/100	500	sigmoid	0.2	0.7478	0.8202
20/100/100	500	sigmoid	0.2	0.7453	0.8315
20/100/20	500	sigmoid	0.2	0.7491	0.8315
100/200/100	500	sigmoid	0.2	0.7453	0.8090
20/100/20	100	sigmoid	0.2	0.7440	0.8202
20/100/20	1000	sigmoid	0.2	0.7680	0.8090
20/100/20	2000	sigmoid	0.2	0.7642	0.7978
100/100	2000	sigmoid	0.1	0.7642	0.7978
100/100	2000	sigmoid	0.3	0.7680	0.7865
100/100/100	2000	sigmoid	0.2	0.7705	0.8315
100/100/100	100	sigmoid	0.2/0.2	0.7188	0.8090
100/100/100	500	sigmoid	0.2/0.2	0.7629	0.7978
100/100/100	1000	sigmoid	0.2/0.2	0.7680	0.7978
10/100/100/20	100	sigmoid	0.2/0.2	0.7201	0.8090
50/100/100/20	100	sigmoid	0.2/0.2	0.7238	0.7978
20/100/100/10	100	sigmoid	0.2/0.2	0.7201	0.8202
20/100/100/20	100	sigmoid	0.2/0.2	0.7226	0.8090
20/100/100/20	500	sigmoid	0.2/0.2	0.7604	0.8090
20/100/200/100/20	100	sigmoid	0.2/0.2	0.7125	0.7978
20/100/200/100/20	500	sigmoid	0.2/0.2	0.7478	0.7753
20/100/200/20	100	sigmoid	0.2/0.2	0.7264	0.7978
20/200/200/20	100	sigmoid	0.2/0.2	0.7201	0.7978
20/200/200/20	200	sigmoid	0.2/0.2	0.7402	0.8202
20/200/200/20	300	sigmoid	0.2/0.2	0.7541	0.8202
20/500/500/20	100	sigmoid	0.2/0.2	0.7100	0.7753
20/1000/1000/20	100	sigmoid	0.2/0.2	0.7137	0.7753
20/100/100/100/100/20	100	sigmoid	0.2/0.2	0.6999	0.7865
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.1/0.2	0.6974	0.7865
20/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.6999	0.7753
20/100/100/100/100/20	100	sigmoid	0.2/0.2/0.2	0.7314	0.7978
20/100/100/100/100/100/20	100	sigmoid	0.2/0.1/0.2	0.6898	0.7865

Tabela 22 Experimentos realizados utilizando *dropout* para o EVENTO4 (Elaborado pelo autor).

O modelo escolhido para avaliação do EVENTO4 foi treinado em 1.000 (mil) épocas, sem *dropout*, com a seguinte configuração, na ordem:

- Camada oculta com 100 neurônios, função de ativação Sigmoide e utilização de *bias*;
- Camada de saída com 1 neurônios e função de ativação Sigmoide.

A Fig. 20 apresenta a matriz de confusão gerada pelo modelo que apresentou a melhor acurácia durante os experimentos e que foi selecionado para a condução das análises seguintes.

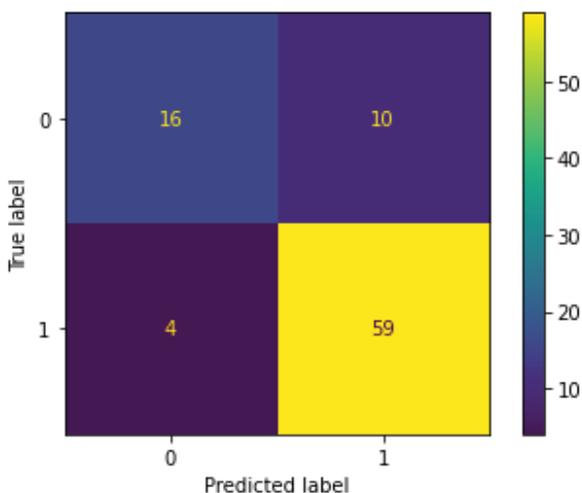


Fig. 19 Matriz de confusão do melhor resultado obtido para o EVENTO4 (Elaborado pelo autor).

A partir das previsões realizadas pelo modelo selecionado, os resultados foram organizados com os dados observados para o EVENTO4 e classificados conforme o cenário em que se enquadravam. Na Tabela 23 é exibida uma amostra dos dados já tabulados e classificados.

CLIENTE	PAGOU_OBSERVADO	PAGOU_PREDITO	EVENTO_OBSERVADO	EVENTO_PREDITO	RESULTADO_OBSERVADO	RESULTADO_PREDITO
1346	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
4237	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2259	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
312	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1557	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
6293	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
1708	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2968	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3468	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3453	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7695	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1474	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1576	1	0	0	1	NÃO ENVIO DEVIDO	ENVIO INDEVIDO
6003	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
8004	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
5648	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7226	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3819	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3574	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6341	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6117	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
420	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2554	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6589	0	1	1	0	ENVIO DEVIDO	NÃO ENVIO INDEVIDO
829	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2464	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5305	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
8127	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO

Tabela 23 Amostra dos dados preditos *versus* observados para o EVENTO4 (Elaborado pelo autor).

Em seguida, esses dados foram consolidados para que se fosse avaliada a eficiência da RNA em relação aos eventos de régua observados no conjunto de dados. Abaixo, Tabela 24, o resumo dos cenários identificados, após a comparação entre as classes preditas e observadas. À esquerda os cenários

observados e à direita os cenários baseados na predição da RNA.

Observado		Predito	
Clientes que pagaram (observado)			
	Instâncias	%	
ENVIO INDEVIDO	5	7,94%	
NÃO ENVIO DEVIDO	58	92,06%	
<b>Total</b>	<b>63</b>	<b>100,00%</b>	
Clientes que não pagaram (observado)			
	Instâncias	%	
ENVIO DEVIDO	19	73,08%	
NÃO ENVIO INDEVIDO	7	26,92%	
<b>Total</b>	<b>26</b>	<b>100,00%</b>	

Tabela 24 Resultados consolidados da predição do EVENTO4 (Elaborado pelo autor).

Para o EVENTO4 o comportamento da rede neural artificial mudou de forma acentuada em relação aos eventos anteriores. Notou-se que as previsões ficaram muito próximas do real observado, melhorando o cenário de não envios devidos (economia no disparo do evento) em apenas 1 caso (isso representa 20%) e uma piora dos não envios indevidos (casos em que deveria acionar o cliente, mas não o foi) de 16% (3 casos).

#### E. Avaliação do EVENTO5

Por fim, foi efetuada a avaliação do EVENTO5, último evento do conjunto de dados, para o qual foram considerados todos os eventos observados na base de origem da análise. Os experimentos seguiram a mesma ordem lógica do que foi executado para os demais eventos. Primeiramente, foram avaliados os algoritmos mais simples, com menos camadas e sem o recurso de *dropout*. A Tabela 25 demonstra os resultados desses experimentos.

Neurônios	Épocas	Função de Ativação	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.7957	0.8652
100	500	sigmoid	0.8020	0.8652
100	1000	sigmoid	0.8033	0.8764
100	2000	sigmoid	0.8260	0.8539
100/100	1000	sigmoid	0.8235	0.8539
20/100	1000	sigmoid	0.8159	0.8764
20/100	300	sigmoid	0.7982	0.8652
10/100	1000	sigmoid	0.8159	0.8764
100/20	1000	sigmoid	0.8323	0.8652
20/100/20	1000	sigmoid	0.8235	0.8764

Tabela 25 Resultados para os modelos sem *dropout* para o EVENTO5 (Elaborado pelo autor).

A inclusão de um novo atributo para a avaliação do EVENTO5 acarretou um comportamento indesejado da rede. Observou-se que o aprendizado ocorreu de forma muito rápida, em poucas épocas, e não foi capaz de progredir. Além disso, a taxa de perda começou a aumentar gradativamente conforme a evolução das épocas de treinamento. Isso demonstrou que a rede teve dificuldade de se adaptar e criar relações mais robustas entre os dados.

Na Fig. 21 é apresentado o gráfico da evolução da acurácia e da perda com o passar das épocas. Este gráfico foi gerado a partir do modelo com apenas uma camada de 100 neurônios e executado em 1.000 épocas, onde obteve a acurácia de 87,64%, o qual demonstra o comportamento explicado

anteriormente. Além disso, observou-se que a rede obteve um aprendizado rápido nas primeiras 50 épocas e permaneceu estável a partir de então.

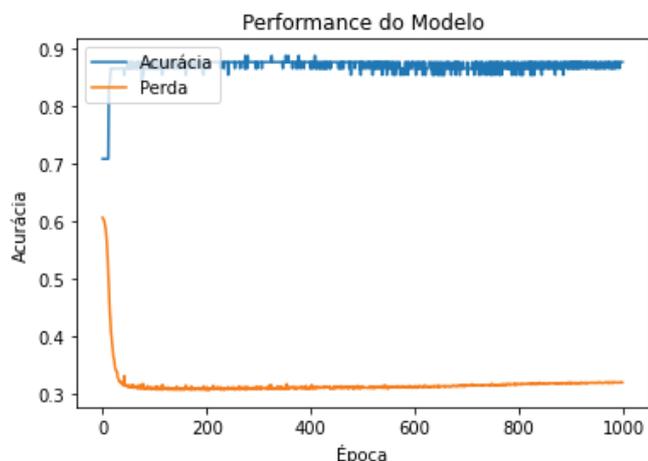


Fig. 20 Evolução da acurácia e perda durante o treinamento (Elaborado pelo autor).

Na tentativa de melhorar o desempenho do modelo, foram realizados novos experimentos utilizando o *dropout* e trabalhando os valores dos demais parâmetros. A Tabela 26 apresenta uma amostra dos parâmetros testados no modelo e as respectivas acurácias atingidas (nas bases de treino e teste).

Neurônios	Épocas	Função de Ativação	Dropout	Acurácia Treino	Acurácia Teste
100	100	sigmoid	0.2	0.8020	0.8652
100	200	sigmoid	0.2	0.8108	0.8652
100	500	sigmoid	0.2	0.8146	0.8652
100	1000	sigmoid	0.2	0.8033	0.8764
100	5000	sigmoid	0.2	0.8008	0.8764
100	10000	sigmoid	0.3	0.8247	0.8876
100/100	100	sigmoid	0.2	0.7982	0.8652
100/100	500	sigmoid	0.2	0.8134	0.8652
100/100	500	sigmoid	0.5	0.8096	0.8764
20/100	500	sigmoid	0.5	0.8083	0.8764
20/100/100	500	sigmoid	0.5	0.8247	0.8764
20/100/20	500	sigmoid	0.5	0.8121	0.8764
20/100/20	1000	sigmoid	0.5	0.8348	0.8764
20/100/20	5000	sigmoid	0.5	0.8272	0.8652
100/100	2000	sigmoid	0.1	0.8209	0.8764
100/100	2000	sigmoid	0.3	0.8235	0.8989
100/100	3000	sigmoid	0.3	0.8247	0.8876
100/100	10000	sigmoid	0.3	0.8474	0.8764
100/100/100	2000	sigmoid	0.2	0.8323	0.8764
100/100/100	100	sigmoid	0.2/0.2	0.8058	0.8652
100/100/100	500	sigmoid	0.2/0.2	0.8159	0.8652
100/100/100	1000	sigmoid	0.2/0.2	0.8764	0.8764
10/100/100/20	100	sigmoid	0.2/0.2	0.8083	0.8764
50/100/100/20	100	sigmoid	0.2/0.2	0.8008	0.8764
20/100/100/10	100	sigmoid	0.2/0.2	0.8020	0.8539
20/100/100/20	100	sigmoid	0.2/0.2	0.8071	0.8764
20/100/100/20	500	sigmoid	0.2/0.2	0.8096	0.8764
20/100/200/100/20	500	sigmoid	0.2/0.2	0.8247	0.8652
20/100/200/20	500	sigmoid	0.2/0.2	0.8247	0.8764
20/200/200/20	500	sigmoid	0.2/0.2	0.8134	0.8764
20/500/500/20	500	sigmoid	0.2/0.2	0.8108	0.8764
20/1000/1000/20	500	sigmoid	0.2/0.2	0.8235	0.8539

Tabela 26 Experimentos realizados utilizando *dropout* para o EVENTO5 (Elaborado pelo autor).

Os novos experimentos possibilitaram o ajuste de uma única configuração que alcançou a maior acurácia de todos os experimentos. Uma rede composta por duas camadas ocultas com 100 neurônios cada, uma camada de *dropout* com taxa de 30% e uma camada de saída um neurônio, utilizando a função de ativação Sigmoide e o uso de *bias*, atingiu a acurácia de 89,89% na base de testes.

A rede neural artificial conseguiu desenvolver uma relação otimizada entre os dados, o que possibilitou a predição satisfatória para ambas as classes do conjunto de dados. Isso sugeriu que o aumento de variáveis (todos os eventos da base de dados observados) foi positivo para o aprendizado da rede, já que, à medida que novos eventos foram incluídos para treino nos modelos, a quantidade de predições corretas também foram aumentando.

Na Fig. 22 é demonstrada a matriz de confusão do modelo que atingiu a maior acurácia nos testes para o EVENTO5.

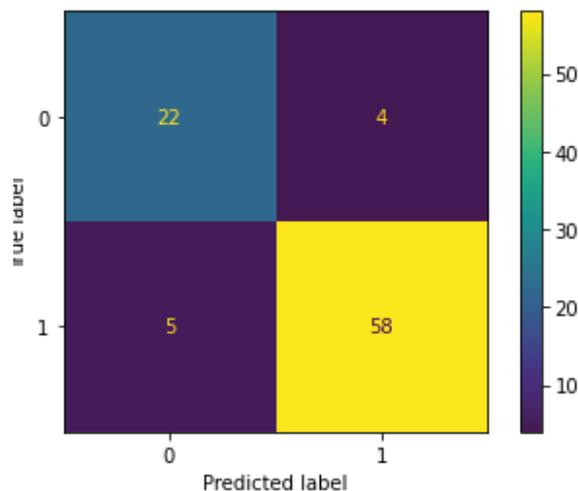


Fig. 21 Matriz de confusão do melhor resultado obtido para o EVENTO5 (Elaborado pelo autor).

Os resultados gerados pelo modelo foram exportados e planilhados para a classificação das instâncias conforme o cenário em que se enquadraram. A Tabela 27 apresenta uma amostra dos dados após a tabulação das predições, eventos observados e classificação de cada registro.

CLIENTE	PAGOU_OBSERVADO	PAGOU_PREDITO	EVENTO_OBSERVADO	EVENTO_PREDITO	RESULTADO_OBSERVADO	RESULTADO_PREDITO
1346	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
4237	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2259	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
312	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1557	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
6293	1	0	0	1	NÃO ENVIO DEVIDO	ENVIO INDEVIDO
1708	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2968	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3468	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3453	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7695	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
1474	1	0	0	1	NÃO ENVIO DEVIDO	ENVIO INDEVIDO
1576	1	1	0	0	NÃO ENVIO DEVIDO	ENVIO DEVIDO
6003	1	0	0	1	NÃO ENVIO DEVIDO	ENVIO INDEVIDO
8004	1	1	1	0	ENVIO INDEVIDO	NÃO ENVIO DEVIDO
5648	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
7226	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
3819	0	1	0	0	NÃO ENVIO INDEVIDO	NÃO ENVIO INDEVIDO
3574	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6341	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6117	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
420	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
2554	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
6589	0	0	1	1	ENVIO DEVIDO	ENVIO DEVIDO
829	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
2464	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
5305	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO
8127	1	1	0	0	NÃO ENVIO DEVIDO	NÃO ENVIO DEVIDO

Tabela 27 Amostra dos dados preditos versus observados para o EVENTO5 (Elaborado pelo autor).

Na Tabela 28, é apresentado o comparativo consolidado das predições do EVENTO5. À esquerda os cenários observados e à direita os cenários baseados na predição da RNA.

Cientes que pagaram (observado)

	Instâncias	%		Instâncias	%
ENVIO INDEVIDO	1	1,59%	ENVIO INDEVIDO	5	7,94%
NÃO ENVIO DEVIDO	62	98,41%	NÃO ENVIO DEVIDO	58	92,06%
<b>Total</b>	<b>63</b>	<b>100,00%</b>	<b>Total</b>	<b>63</b>	<b>100,00%</b>

Cientes que não pagaram (observado)

	Instâncias	%		Instâncias	%
ENVIO DEVIDO	18	69,23%	ENVIO DEVIDO	22	84,62%
NÃO ENVIO INDEVIDO	8	30,77%	NÃO ENVIO INDEVIDO	4	15,38%
<b>Total</b>	<b>26</b>	<b>100,00%</b>	<b>Total</b>	<b>26</b>	<b>100,00%</b>

Tabela 28 Resultados consolidados da predição do EVENTO5 (Elaborado pelo autor).

As predições do EVENTO5 trouxeram um resultado inédito para o trabalho. Foi o único evento em que a RNA apresentou piora nos envios indevidos, ou seja, sugeriu o envio de evento desnecessariamente, e melhorou o envio nos casos em que havia a necessidade de tal. Foi o oposto do observado nos experimentos dos eventos anteriores.

Como a quantidade total de instancias da base de testes é pequena, a oscilação de apenas um registro tem uma representatividade percentual muito alta. Em se tratando dos envios indevidos, a rede neural artificial sugeriu o envio de 4 eventos desnecessários a mais que o observado (o que representa 400% a mais). Na sugestão de envios devidos, foram 4 acertos a mais que o observado (que representam 22% a mais).

Observando o EVENTO5 isoladamente, foi o caso em que a RNA não geraria economia poupando eventos desnecessários, pelo contrário, acabaria gerando um custo adicional com este evento, mas em contrapartida melhoraria a eficiência da cobrança, enviando o evento a mais clientes em que haveria necessidade.

#### F. Consolidação dos resultados

Cada evento avaliado individualmente apresentou resultados peculiares quando foram submetidos a rede neural artificial. Nos primeiros eventos, os modelos sugeriram uma melhora na inibição de envios desnecessários aos clientes, mas ao custo de comprometer a cobrança de clientes que deveriam receber os acionamentos. Nos eventos finais, essa relação se inverteu.

Então, agrupou-se o resultado de todos os eventos, considerando todos os acertos e erros de cada evento, comparados aos eventos conhecidos da régua de cobrança tradicional. Na Tabela 29 são demonstrados o somatório de todos os cenários gerados em todos os eventos. À esquerda os observados e à direita os baseados na predição da RNA.

Cientes que pagaram (observado)

	Instâncias	%		Instâncias	%
ENVIO INDEVIDO	92	29,21%	ENVIO INDEVIDO	17	5,40%
NÃO ENVIO DEVIDO	223	70,79%	NÃO ENVIO DEVIDO	298	94,60%
<b>Total Geral</b>	<b>315</b>	<b>100,00%</b>	<b>Total Geral</b>	<b>315</b>	<b>100,00%</b>

Cientes que não pagaram (observado)

	Instâncias	%		Instâncias	%
ENVIO DEVIDO	95	73,08%	ENVIO DEVIDO	57	43,85%
NÃO ENVIO INDEVIDO	35	26,92%	NÃO ENVIO INDEVIDO	73	56,15%
<b>Total Geral</b>	<b>130</b>	<b>100,00%</b>	<b>Total Geral</b>	<b>130</b>	<b>100,00%</b>

Tabela 29 Resultados consolidados de todos os eventos (Elaborado pelo autor).

De um modo geral, a rede neural artificial demonstrou que seria capaz de reduzir os custos operacionais inibindo os envios desnecessários. Gerou uma redução de 82% nos envios indevidos, partindo da premissa que clientes em que há a predição de que ocorrerá um pagamento, não há necessidade de envio daquele evento.

Por outro lado, apresentou dificuldade em melhorar a eficiência da cobrança, já que deixou de enviar eventos em situações em que o estímulo seria necessário. Apresentou uma piora de 40% nos eventos que sugeriu não enviar, mas que deveriam ser enviados.

Ao transpor esses resultados ao cenário de cobrança do mundo real, mesmo que a rede não tenha atingido um equilíbrio satisfatório entre a redução de envios de eventos desnecessários sem comprometer a eficiência da cobrança, há a possibilidade de o ganho financeiro gerado pela economia de eventos não disparados compensar, ou até mesmo superar, o saldo devedor que seria recuperado do público que deixou de ser estimulado.

## V. CONSIDERAÇÕES FINAIS

Concluiu-se então que a Inteligência Artificial seria capaz de aprimorar a execução de uma régua de cobrança. Com a aplicação correta e ajustada conforme o evento da régua de cobrança, ela mostrou ser eficiente em aprimorar o processo de recuperação da dívida de clientes inadimplentes, possibilitando a redução de custos operacionais.

O melhor resultado da Rede Neural Artificial foi obtido quando foram avaliadas todas as variáveis disponíveis na base de dados, ou seja, considerando os dados conhecidos do envio dos quatro eventos anteriores ao que foi predito. Este modelo atingiu 89,89% de acurácia e mostrou boa capacidade de classificar entre as duas classes do conjunto. Porém, para este mesmo público, a régua de cobrança tradicional também alcançou uma eficácia elevada, tornando o resultado da RNA relativamente sutil ao real observado. Além disso, o resultado deste modelo quando avaliado de forma independente, demonstrou que haveria melhora na cobrança dos clientes, mas com algum acréscimo no custo operacional com envios desnecessários.

Quando analisados todos os melhores resultados para cada evento de cobrança agrupados – o que na prática seria avaliar todos os eventos que um cliente recebeu pela régua de cobrança – notou-se que a Rede Neural Artificial acabou sendo menos conservadora em relação ao observado, ou seja, considerou mais clientes como possíveis pagadores e consequentemente inibiria o envio daqueles eventos. Por conta disso, ela apresentaria grande economia na cobrança por evitar o envio desnecessários de eventos, mas ao custo de não acionar parte de clientes que deveriam receber aquele estímulo.

A principal dificuldade encontrada no conjunto de dados trabalhado foi o volume total de registros muito baixo, associado à uma distribuição de classes desproporcional. Isso dificultou a generalização dos problemas, principalmente para os primeiros eventos em que a quantidade de atributos avaliada era menor. Além disso, um volume baixo de instâncias gerou oscilações abruptas no resultado, já que a representatividade fica mais alta à medida que a quantidade total é mais baixa.

Outro fator que pode ter contribuído de forma negativa para o trabalho foi a escassez de variáveis. O universo de onde

partiram os dados obtidos possui dezenas de outras variáveis, mas que não foram disponibilizadas para este estudo. Uma quantidade maior de variáveis permitiria um pré-processamento mais robusto, a fim de identificar quais delas eram de fato relevantes para uma rede neural artificial e assim aprimorar o aprendizado do modelo.

Por outro lado, mesmo diante das dificuldades impostas pela escassez de dados e variáveis, foi possível constatar que a Inteligência Artificial foi capaz de reproduzir resultados satisfatórios. Em um problema no mundo real, isso implica que teria grande versatilidade na sua utilização, como por exemplo, avaliar o comportamento de clientes novos na base de uma empresa. Por vezes, é necessário tomar alguma decisão com esse público, mas como ainda não há um histórico de relacionamento, a disponibilidade de dados e informações é reduzida.

Um trabalho futuro a ser realizado é a obtenção de um conjunto com maior quantidade de dados e mais variáveis relacionadas ao cliente. Além disso, uma outra abordagem seria avaliar cada evento individualmente com as mesmas variáveis, ou seja, sem considerar os eventos anteriores observados para as instâncias. Essa análise possibilitaria avaliar qual evento possui maior efetividade na cobrança do público analisado.

## VI. REFERÊNCIAS

- [1] Serasa Experian (Serasa). Sala de Imprensa. Notícias: Inadimplência no Brasil cai pela primeira vez em quatro anos e encerra 2020 com 61,4 milhões de pessoas, revela Serasa Experian. Disponível em: <https://www.serasaexperian.com.br/sala-de-imprensa/noticias/inadimplencia-no-brasil-cai-pela-primeira-vez-em-quatro-anos-e-encerra-2020-com-614-milhoes-de-pessoas-revela-serasa-experian/>. Acesso em: 27 mai. 2021.
- [2] Fagundes, A. et al. Valor. O Globo. Marcas da Pandemia. Disponível em: <https://valor.globo.com/coronavirus/a-economia-na-pandemia/>. Acesso em: 27 mai. 2021.
- [3] Vasconcelos, E.. Jornal Contábil. Qual o impacto da pandemia sobre a inadimplência do Brasil?. Disponível em: <https://www.jornalcontabil.com.br/impacto-da-pandemia-sobre-a-inadimplencia-do-brasil/>. Acesso em: 27 mai. 2021.
- [4] Cobre Facil. Gestão de Cobrança. Como criar uma régua de cobrança automatizada e evitar inadimplências. Disponível em: <https://www.cobrefacil.com.br/blog/regua-de-cobranca-automatizada>. Acesso em: 05 jun. 2021.
- [5] Instituto Brasileiro De Geografia E Estatística (IBGE). IBGE Educa. Matérias Especiais: Uso de Internet, Televisão e Celular no Brasil. Disponível em: <https://educa.ibge.gov.br/jovens/materias-especiais/20787-uso-de-internet-televisao-e-celular-no-brasil.html>. Acesso em: 27 mai. 2021.
- [6] O Globo. Economia. Defesa do Consumidor. Brasil registra mais de um milhão de reclamações de cobrança indevida em 2019. Disponível em: <https://oglobo.globo.com/economia/defesa-do-consumidor/brasil-registra-mais-de-um-milhao-de-reclamacoes-de-cobranca-indevida-em-2019-24305851>. Acesso em: 27 mai. 2021.
- [7] Russel, S. J.; Norvig, P. Artificial intelligence. 2nd. Rio de Janeiro: Elsevier, 2004.
- [8] NSC Total. Especiais Publicitários. Tecnologia para o Empreendedor. Microempreendedores podem utilizar robôs para cobrança de clientes. Disponível em: <https://www.nscetotal.com.br/noticias/microempreendedores-podem-utilizar-robos-para-cobranca-de-clientes>. Acesso em: 08 jun. 2021.
- [9] Oracle. Oracle Digital Assistant. O que É um Chatbot?. Disponível em: <https://www.oracle.com/br/chatbots/what-is-a-chatbot/>. Acesso em: 08 jun. 2021.
- [10] Covington, M. NLP for Prolog Programmers, Prentice-Hall, 1994.
- [11] Serviço de Proteção ao Crédito (SPC). O Conceito Do Endividamento e as Consequências da Inadimplência. Fevereiro, 2016.
- [12] Nu Pagamentos S.A. São Paulo. O que é inadimplência e o que significa estar inadimplente? Disponível em: <https://blog.nubank.com.br/o-que-e-inadimplencia/>. Acesso em: 28 mai.2021.
- [13] Serasa Experian (Serasa). Estudos e Pesquisas. Conheça as 7 principais causas de inadimplência no Brasil hoje. Disponível em: <https://www.serasaexperian.com.br/conteudos/estudos-e-pesquisas/conheca-as-7-principais-causas-de-inadimplencia-no-brasil-hoje/>. Acesso em: 27 mai. 2021.
- [14] IBM. Documentation. Disponível em: <https://www.ibm.com/docs/en/i/7.1?topic=concepts-structured-query-language>. Acesso em: 04 jun. 2021.
- [15] ORACLE. Banco de Dados. O que É um Banco de Dados Relacional?. Disponível em: <https://www.oracle.com/br/database/what-is-a-relational-database/>. Acesso em: 04 jun. 2021.
- [16] Monard, Maria Carolina, Baranauskas, José Augusto. Conceitos Sobre Aprendizado de Máquina. Sistemas Inteligentes Fundamentos e Aplicações. 1 ed. Barueri-SP: Manole Ltda, 2003.
- [17] Schunk, L. M. O Uso De Inteligência Artificial Por Meio De Chatbots No Processo De Atendimento Ao Cliente: Um Estudo Sobre Seus Benefícios. 2020. 136 f. Dissertação (Mestrado em Administração de Empresas) - Escola de Administração de Empresas de São Paulo, Fundação Getulio Vargas, São Paulo (SP), 2020. Disponível em: <https://bibliotecadigital.fgv.br/dspace/handle/10438/29943>. Acesso em 09 jun. 2021.
- [18] van de Geer, R.; Wang, Q.; Bhulai, S. Data-driven Consumer Debt Collection via Machine Learning and Approximate Dynamic Programming.. 2018. 32 f. Data-Driven Consumer Debt Collection via Machine Learning and Approximate Dynamic Programming. Amsterdã, Países Baixos, 2018. Disponível em: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3250755](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3250755). Acesso em: 09 jun. 2021.
- [19] TensorFlow. Documentação. Disponível em: <https://www.tensorflow.org/>. Acesso em: 09 jun. 2021.
- [20] Moreira, C., Revista de Ciência Elementar, 1(01):0006, Lisboa. 2003.
- [21] Rumelhart, D. E.; McClelland, J. L. Parallel distributed processing: explorations in the microstructure of cognition. Volume1: foundations. MITPress, 1986.
- [22] McCulloch, W. ; Pitts, W.. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biology, 5:115–133. 1943.
- [23] SAS Insights. Disponível em : [https://www.sas.com/pt\\_br/insights/analytics/](https://www.sas.com/pt_br/insights/analytics/). Acesso em: 28 jun. 2021.
- [24] M. Mitchell, Tom. Machine Learning. 1ª edição. McGraw-Hill, 1997.
- [25] Tafner, Malcon; Xerez, Marcos; Rodrigues, Ison. Redes Neurais Artificiais: Introdução e Princípios de Neurocomputação. EKO, 1996.
- [26] Google Colab. Documentação. Disponível em: <https://colab.research.google.com/>. Acesso em 28 jun. 2021.
- [27] scikit-learn. Documentação. Disponível em: <https://scikit-learn.org/stable/modules/classes.html>. Acesso em: 09 jun. 2021.
- [28] Keras. Documentação. Disponível em: <https://keras.io/about/>. Acesso em 28 jun 2021.
- [29] Rezende, S.O..Sistemas Inteligentes, Manole, 2003.
- [30] Ferrari, D. G.; De Castro Silva, L. N. Introdução a mineração de dados. Saraiva Educação S.A., 2017.
- [31] Chatbots Magazine. Chatbots, AI, NLP, Facebook Messenger, Slack, Telegram, and more. Disponível em: <https://chatbotmagazine.com/>. Acesso em: 20 mai 2022.