Relatório Técnico: Identificação de Transações Financeiras Fraudulentas com Redes Neurais Artificiais

Gabriel Viana Dantas [*] Thiago Emanuell Vieira Moura [†]
gviana@discente.ufg.br thiagoemanuell@discente.ufg.br

Wendel Marques de Jesus Souza [‡]
wendelmjs@discente.ufg.br

1 Introdução

O objetivo deste artefato é, em linhas gerais, documentar todas as etapas que compõem a criação de uma solução para um problema do domínio do mercado financeiro. A equipe é formada por: Gabriel Dantas, engenheiro de conhecimento responsável pela implementação da solução; Thiago Moura, analista de dados, responsável por realizar a análise exploratória dos dados; e, Wendel Marques, gerente, responsável pela confecção deste relatório técnico e da apresentação. O trabalho é originado da disciplina Inteligência Artificial (2020.2), do Instituto de Informática (INF-UFG). Uma das suas principais finalidades é o alinhamento entre teoria e prática, por meio do oferecimento da oportunidade da aplicação dos conhecimentos adquiridos em um problema do mundo real.

A inteligência artificial é uma área da ciência da computação que está cada vez mais presente na vida das pessoas. Por exemplo, está em serviços de busca de informações (como o Google, Bing etc), em plataformas de *streamings*, em *marketplaces*, em televisões entre diversos outros lugares e dispositivos. Naturalmente, empresas de todos os ramos têm utilizado algoritmos de inteligência artificial para resolver problemas, otimizar processos e criar novas soluções. Nessa perspectiva, o foco deste trabalho é no setor financeiro, que sofre com um problema antigo e que nos últimos anos foi modernizado: a fraude financeira. Assim sendo, mais especificamente, este trabalho apresenta uma solução capaz de identificar transações financeiras fraudulentas utilizando redes neurais artificiais.

2 Descrição do problema

O exponencial crescimento do uso das tecnologias da informação e comunicação (TICs) propiciou diversas mudanças nos hábitos das pessoas. Entre essas mudanças, é possível citar como clientes de instituições financeiras passaram a realizar operações bancárias. Há alguns

[*]Instituto de Informática, UFG – Engenheiro de Conhecimento.
[†]Instituto de Informática, UFG – Analista de Dados.
[‡]Instituto de Informática, UFG – Gerente.

anos, realizar uma simples atividade como consultar o extrato de transações poderia demandar o deslocamento do cliente até um caixa eletrônico. Atualmente não só atividades simples como também as mais complexas podem ser realizadas por meio de um computador ou até mesmo um celular com acesso à *internet*. Estas e outras praticidades são alguns dos reflexos da alta adesão às TICs e também da democratização do acesso à *internet*.

A popularização da realização de operações bancárias por meio da *internet*, que antes só podiam ser realizadas em agências bancárias, inicialmente foi caracterizada pelo uso do termo *internetbanking*. Esse termo se refere ao ambiente bancário na *internet*, o qual é usado para referenciar, majoritariamente, ao acesso bancário por meio de um navegador de *internet*. Além disso, nos últimos anos surgiu o termo *mobilebanking*, que por sua vez refere-se ao acesso por meio de um aplicativo específico instalado em um dispositivo móvel, como *smartphones* e *tablets*.

De acordo com uma pesquisa realizada pela Febraban (Federação Brasileira dos Bancos), o *mobilebanking* é a forma de acesso mais utilizada no Brasil[5]. Em 2020, o órgão revelou que 63% das operações bancárias foram feitas por meios digitais. Nesse sentido, o *mobilebanking* não é novidade no Brasil. Empresas como Bradesco, que investe bilhões em processos de transformação digital, por exemplo, recentemente lançou o Next, um banco digital. Ainda de acordo com os estudos da Febraban, os investimentos em tecnologia feitos pelos bancos cresceram 48% em 2019 em relação ao ano anterior Febraban:19.

Em 2020, a digitalização dos serviços bancários foi impulsionada pela pandemia do novo coronavírus (Covid-19). Para se ter uma ideia, o aplicativo Caixa Tem, da Caixa Econômica Federal, possuía cerca de 1 milhão de downloads em 2019. Durante a pandemia, a aplicação foi utilizada pelos brasileiros para realizar movimentações do auxílio emergencial do Governo Federal, recurso pago a cidadãos brasileiros que se encontram em situação de vulnerabilidade social. Por meio dele, é possível consultar o saldo, realizar pagamentos de boletos e contas (água, luz e *internet*), realizar transferências e, inclusive, pagar diretamente nas máquinas de cartão através da leitura de um *QRCode*.

Outro fator que acelerou a transformação digital dos meios de pagamento no Brasil, foi o lançamento do Pix pelo Banco Central. O Pix é um sistema de pagamentos instantâneos que permite a conclusão de pagamentos e transferências em cerca de dez segundos. Criado para facilitar as transações financeiras existentes atualmente, um dos seus benefícios é a disponibilidade de 7 dias por semana, 24 horas por dia. Além disso, essa nova modalidade oferece suporte a pagamentos via *QRCode*.

Todavia, embora esse novo paradigma esteja intrinsecamente conectado com diversos benefícios, é acompanhado também de pontos negativos. A digitalização das instituições financeiras trouxe consigo maior exposição dos serviços bancários. Os clientes podem, praticamente, realizar operações de qualquer lugar e a qualquer momento. Dessa forma, aumentou-se também a exposição às fraudes. Fraude pode ser entendida como um crime que tem por objetivo obter vantagem sobre alguém. No contexto financeiro, algumas fraudes são: geração de boletos falsos, pirâmides financeiras, clonagem de cartão de crédito, falsos profissionais de investimentos entre outros golpes e modos de operação.

No contexto de contas digitais, é possível citar os casos de fraudes envolvendo o Pix e o Caixa Tem. Um levantamento da Febraban revelou que o roubo de dados por meio de aplicativos de mensagens instantâneas e falsas centrais de atendimentos estão entre as principais fraudes. Por causa da rapidez do sistema de pagamentos, os criminosos conseguem rapidamente movimentar o dinheiro sem que as vítimas percebam em um tempo hábil. No caso do Caixa Tem, também por meio de engenharia social aliada aos poucos recursos de segurança do aplicativo, criminosos conseguiram se apropriar indevidamente de recursos financeiros de alguns usuários.

Segundo uma publicação do Nilson Report[8], em 2016 o setor global de cartões sofreu uma perda de US$ 21,8 bilhões por fraude, o que corresponde a um aumento de 4,4% em relação ao ano anterior. Por causa desse crescente problema, empresas desse ramo têm investido cada vez mais na prevenção e detecção de fraudes. A Kroll [2], empresa mundial especializada em investigações corporativas, estimou que em 2020 possam ter sido gastos mais de US$ 150 bilhões no combate às fraudes. Esse esforço se faz necessário devido ao fato de que, à medida que as empresas desenvolvem novas soluções, os fraudadores aumentam a sofisticação e a complexidade dos esquemas de fraude.

As duas principais formas de combate às fraudes no mercado financeiro são a prevenção e a detecção dessas atividades. A prevenção de fraudes é caracterizada pela antecipação à tentativa de fraude, que tem como objetivo a diminuição da quantidade de futuras ações que causem danos financeiros aos clientes e instituições. Já a detecção de fraudes ocorre durante a tentativa de fraude. Esta etapa tem o objetivo de mitigar a fraude. Nessa perspectiva, o objetivo deste trabalho é, portanto, apresentar uma solução eficaz e eficiente, capaz de detectar movimentações de recursos financeiros em contas digitais que possuam características de transações fraudulentas.

Assim sendo, o foco do sistema é a identificação de atividades suspeitas em operações de transferências e pagamentos virtuais realizados através de contas digitais. A sua implementação deve corroborar com a automatização do processo de identificar possíveis fraudes, com o menor número de alarmes falsos, também chamados de falsos positivos. Embora as equipes de análise de fraudes sejam extremamente capacitadas para identificar uma transação fraudulenta, é praticamente impossível monitorar todos os eventos. Entretanto, é recomendado que tais equipes trabalhem em conjunto com o sistema, uma vez que é importante mitigar também os falsos positivos, tendo em vista que o seu acontecimento pode trazer aborrecimentos para os clientes, resultando em danos à reputação da instituição. Portanto, considerando os fatores supracitados, é perceptível que faz-se necessário o uso de tecnologia de ponta no combate às transações fraudulentas.

3 Descrição da base de dados

O objetivo geral desta seção é a obtenção de *insights* sobre o conjunto de dados. Para tanto, foram utilizadas técnicas de visualização de dados e alguns cálculos estatísticos básicos. Essa etapa é chamada de análise exploratória de dados (AED). Por meio dela, é possível também encontrar problemas, padrões e relacionamentos que poderão apoiar a etapa de pré-processamento, que antecede a implementação da solução.

3.1 Estrutura do conjunto de dados

O conjunto de dados é uma base de dados sintético que foi gerada utilizando o simulador chamado *PaySim* [6]. O *PaySim* usa dados agregados do conjunto de dados privado para gerar um conjunto de dados sintético que se assemelha à operação normal de transações e injeta comportamento malicioso para avaliar posteriormente o desempenho dos métodos de detecção de fraude.

Os *logs* originais foram fornecidos por uma empresa multinacional, fornecedora do serviço financeiro móvel que atualmente opera em mais de 14 países em todo o mundo. O conjunto de dados sintéticos é reduzido em 1/4 do conjunto de dados original. A tabela 1 ilustra um exemplo de uma transação presente na base de dados.

Onde:

Tabela 1: Verificação de algumas linhas do conjunto de dados

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig
0	1	PAYMENT	9839.64	C123100681	170136.0
nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	
160296.36	M1979787155	0.0	0.0	0	

- *step* - mapeia uma unidade de tempo no mundo real. Neste caso, 1 etapa corresponde a 1 hora de tempo. Total de etapas 744 (simulação de 30 dias).
- *type* - pode ser CASH-IN, CASH-OUT, DEBIT, PAYMENT ou TRANSFER.
- *amount* - montante da transação em moeda local.
- *nameOrig* - cliente que iniciou a transação
- *oldbalanceOrg* - saldo inicial antes da transação
- *newbalanceOrig* - novo saldo após a transação
- *nameDest* - cliente que é o destinatário da transação
- *oldbalanceDest* - destinatário do saldo inicial antes da transação. Observe que não há informações para clientes que começam com M (Comerciantes). item *newbalanceDest* - novo destinatário do saldo após a transação. Observe que não há informações para clientes que começam com M (Comerciantes).
- $isFraud$ - São as transações feitas pelos agentes fraudulentos dentro da simulação. Neste conjunto de dados específico, o comportamento fraudulento dos agentes visa lucrar ao assumir o controle das contas dos clientes e tentar esvaziar os fundos transferindo para outra conta e retirando do sistema.
- $isFlaggedFraud$ - O modelo de negócios visa controlar as transferências em massa de uma conta para outra e sinaliza tentativas ilegais. Uma tentativa ilegal neste conjunto de dados é uma tentativa de transferir mais de 200.000 em uma única transação.

O conjunto de dados é composto por 6.362.620 linhas (transações) e 11 colunas (atributos). Há dados do tipo *int*, $float$ e $object$.

O número total de transações é igual a 6.362.620, dos quais 8213 representam transações fraudulentas e 6.354.407 representam movimentações não fraudulentas. O total de fraudes é equivalente a 13% de todas as transações. Este resultado mostra que é um conjunto de dados desequilibrado. Em outras palavras, faz necessário realizar o balanceamento na etapa de implementação.

Além disso, existem muitas discrepâncias nas variáveis quantitativas. Esta característica é apresentada no gráfico abaixo. Geralmente, *outliers* muito grandes são removidos porque há uma chance de que isso ocorra devido à gravação incorreta. Mas, neste caso, a gravação errada não é o caso, porque transações fraudulentas muito grandes tendem a acontecer. Portanto, não podemos remover os *outliers* e há necessidade de tratá-los antes da análise ou modelagem.

Tabela 2: Informações dos atributos

Index	Column	Dtype
0	step	int64
1	type	object
2	amount	float64
3	nameOrig	object
4	oldbalanceOrg	float64
5	newbalanceOrig	float64
6	nameDest	object
7	oldbalanceDest	float64
8	newbalanceDest	float64
9	isFraud	int64
10	isFlaggedFraud	int64

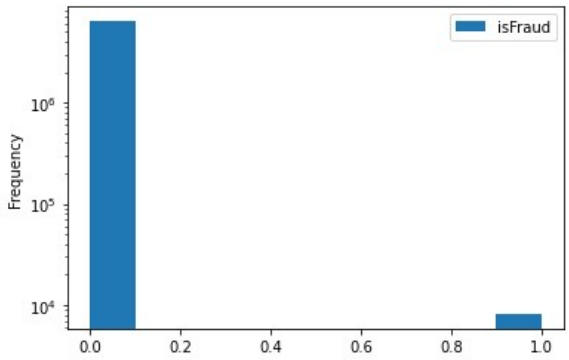

Figura 1: Histograma de isFraud.

Não só a quantidade total de movimentações mas também a discrepância nas variáveis quantitativas podem significar um problema. Isso porque, embora o modelo possa criar uma solução generalizada com uso de um conjunto de exemplos consideravelmente baixo, um cenário ideal seria um em que conjunto de exemplos fosse significativo e representativo. Como base nos análises, é possível afirmar que a base de dados não possui valores faltantes ou linhas duplicadas, o que dispensa esse tipo de limpeza.

Existem 5 tipos de transações definidas no conjunto de dados, a saber: *cash_in*, *cash_out*, *debit*, *payment* ou $transfer$.

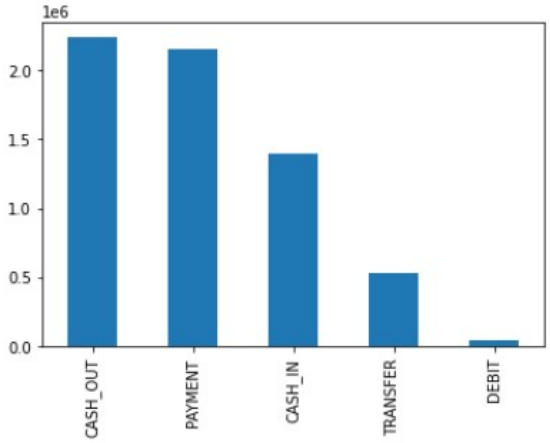

Figura 2: Tipos de transação.

O gráfico acima mostra que a maioria das transações são do tipo *payment* e *cash_out*. Apenas uma pequena fração é do tipo *debit*.

3.2 Transações fraudulentas e não fraudulentas

O número de transações fraudulentas e/ou marcadas com fraudes (*isFlaggedFraud*), separadas por tipos:

Tabela 3: Transações fraudulentas e não fraudulentas

type	isFraud	isFlaggerdFraud
CASH_IN	0	0
CASH_OUT	4116	0
DEBIT	0	0
PAYMENT	0	0
TRANSFER	4097	16

Isso significa que embora existam cinco tipos de transação, as fraudes acontecem apenas quando o tipo de transação é *transfer* ou *cash_out*.

Figura 3: Fraudes acontecem em *transfer* ou *cash_out*

3.3 Diferença entre os balanços antes e depois de cada transação

Há uma grande diferença entre a escala de *amount* de fraudes e de não fraudes, como evidenciado na figura 4.

Foi verificado que o número de transações fraudulentas com erros de saldos na conta de origem é igual a 57. Além disso, o número de transações fraudulentas com erros de saldos na conta de destino é igual a 5324.

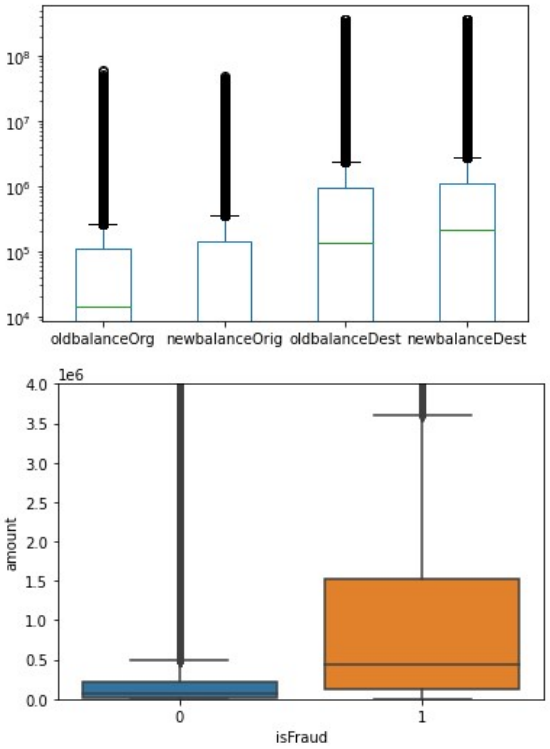

Figura 4: Diferença entre os balanços antes e depois de cada transação

Na tabela 4 são mostradas as porcentagens de transações totais com erros de saldos na conta de origem, divididas por tipo de transação. Já na tabela 5 são mostradas as porcentagens de transações totais com erros de saldos na conta de destino, divididas por tipo de transação. Onde:

- *errorbalanceOrig* = Reflete o erro entre o saldo inicial e final, na conta de origem.
- *errorbalanceDest* = Reflete o erro entre o saldo inicial e final, na conta de destino.

Tabela 4: Porcentagens de transações com *errorBalanceOrig* diferente de zero

type	Porcentagem
CASH_IN	1.000000
CASH_OUT	0.898052
DEBIT	0.329962
PAYMENT	0.594509
TRANSFER	0.958963

Tabela 5: Porcentagens de transações com *errorBalanceDest* diferente de zero

type	Porcentagem
CASH_IN	1.000000
CASH_OUT	0.417347
DEBIT	0.475743
PAYMENT	1.000000
TRANSFER	R 0.436039

3.4 Seleção de atributos

O atributo *isFlaggedFraud* indica se uma transação foi identificada como suspeita de fraude na simulação. Essa informação não existe se o modelo for generalizado para outras transações e o atributo é descartado do treinamento.

Alguns clientes são do tipo comerciante e possuem identificador com inicial M. Nesses casos a informação de saldo do destinatário não é utilizável, sendo sempre valor 0. Todas transações possuem clientes comerciantes se, e somente se, o tipo da transação é *payment*. Logo a informação de cliente comerciante está codificada nessa categoria do atributo *type* e podemos excluir os atributos *nameOrig* e *nameDest* do modelo.

Os atributos *oldbalanceOrg*, *newbalanceOrig*, o*ldbalanceDest* e *newbalanceDest* representam os valores de saldo do remetente e destinatário antes e depois da transação, fornecendo informação sobre o contexto dos participantes de uma transação. São utilizados no modelo.

Ao analisar o atributo *amount* (valor da transação), percebeu-se que é uma informação importante na identificação de fraudes. Sendo assim, deve fazer parte do conjunto de treinamento.

O *boxplot* dos valores das transações em escala logarítmica revela que a maioria se encontra em valores na escala de 10.000 e 100.000. Há a presença de *outliers* mas eles são mantidos no modelo para que possam aprender com transações de alto volume.

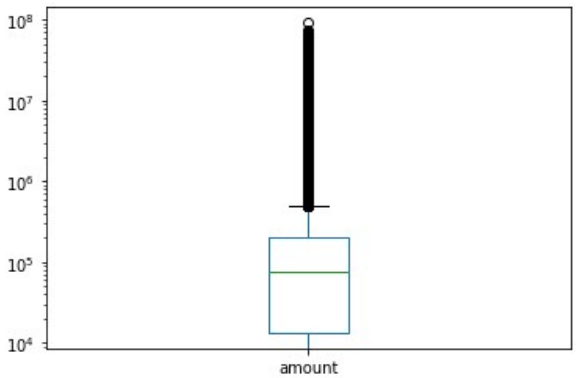

Figura 5: Diferença entre os balanços antes e depois de cada transação

A tabela 6 revela que fraudes têm no máximo um *amount* de 10.000.000,0.

Tabela 6: *Amount*

#	Transações fraudulentas	Transações não fraudulentas
Min	0.0	0.01
Max	10000000.0	92445516.64

É possível examinar o valor da transação e o saldo inicial do cliente para transações fraudulentas e não fraudulentas de saque e transferência separadamente. As medianas dessas variáveis são comparadas porque a média é enviesada por causa de *outliers*. Nos gráficos, podemos ver que essas variáveis são anormalmente altas para transações fraudulentas de saque em comparação com transações de saque não fraudulentas.

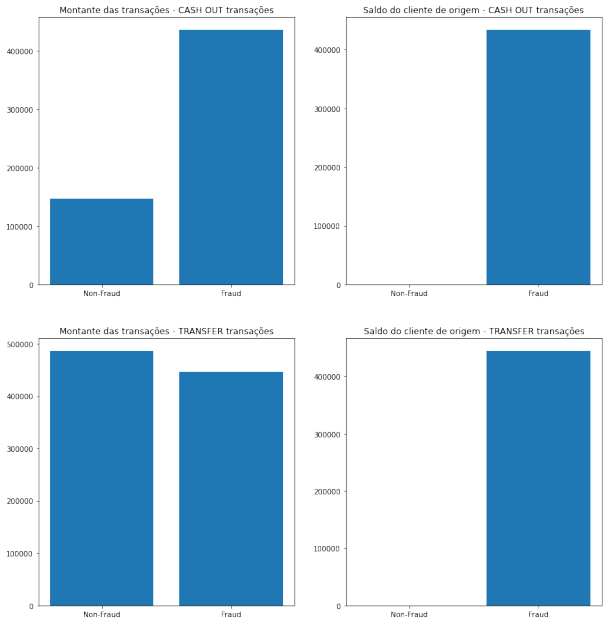

Figura 6: Montantes e saldos

Podemos examinar o saldo inicial do destinatário em busca de saques fraudulentos e não fraudulentos e transações de transferência separadamente. A partir dos gráficos, podemos ver que esta variável é muito menor para transações de transferência fraudulenta em comparação com as transações de transferência sem fraude, o que é o inverso do que observamos no saldo do cliente.

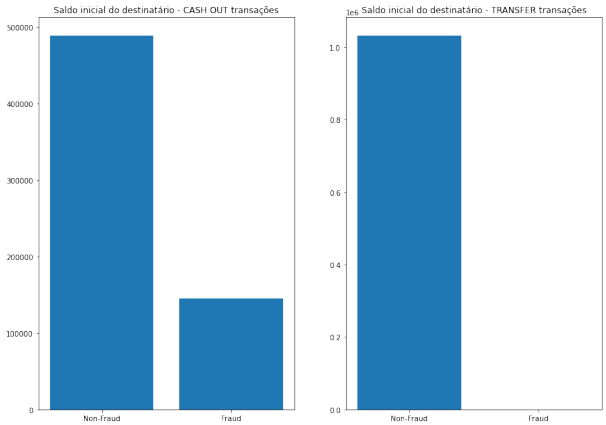

Figura 7: Saldos iniciais do destinatário

Com base nas análises, conclui-se que, para que seja possível utilizar a técnica de balanceamento escolhida, devemos converter o atributo categórico (*type*) em numérico.

4 Descrição da solução

A Inteligência Artificial (IA) é uma das áreas que mais cresceram nos últimos anos. Embora o termo "inteligência artificial"tenha nascido oficialmente em 1956, em uma conferência realizada nos Estados Unidos, só recentemente tem gerado um crescente interesse, isso porque diversas aplicações comerciais práticas têm surgido. Um sistema de IA é aquele cujo comportamento pode ser considerado inteligente. Tipicamente, consegue analisar grandes volumes de dados, identificar padrões e tendências, perceber o ambiente e, além disso, possui a habilidade de análise para para a tomada de decisão

O aprendizado de máquina é um ramo da IA na qual a criação dos sistemas é baseada na ideia de que algoritmos podem aprender com dados. A finalidade é formular predições de forma automática com velocidade e precisão com o mínimo de intervenção humana. Os métodos mais populares de aprendizado de máquina são: aprendizado de máquina não supervisionado e supervisionado. Os algoritmos de ambos os métodos podem ser treinados utilizando dados históricos sobre um determinado domínio. No primeiro caso, os dados não são rotulados, isto é, o algoritmo deve descobrir o que está sendo apresentado. Já no segundo caso, os algoritmos são treinados por meio de exemplos com rótulos.

Neste último método, o sistema recebe um conjunto de entradas junto com as saídas corretas correspondentes. Posteriormente, após analisar e comparar informações, consegue adquirir a capacidade de indicar corretamente quais as saídas corretas quando um novo conjunto de dados for utilizado como entrada. Esse tipo de sistema é usualmente empregado na resolução de problemas nos quais dados históricos podem prever eventos futuros. Um desses problemas é a detecção de transações financeiras fraudulentas. Por exemplo, um conjunto de dados nesse domínio poderiam ser rotulados com 1 para indicar que uma transação é possivelmente uma tentativa de fraude e com 0 para indicar uma transação não fraudulenta.

4.1 Modelo utilizado

Problemas complexos como a detecção de transações fraudulentas podem exigir o uso de sistemas mais sofisticados. Nesse sentido, será apresentada uma solução na qual é utilizado um tipo especial de aprendizado de máquina que simula o modelo do cérebro humano, a saber: redes neurais artificiais.

4.1.1 Redes neurais artificiais

Redes Neurais Artificiais (RNA) são sistemas computacionais distribuídos que apresentam um modelo matemático inspirado nas estruturas neurais biológicas presentes nos organismos inteligentes. Esses organismos têm em processo de aquisição do conhecimento a experiência como uma das principais fontes. Uma RNA possui unidades de processamento simples, densamente interconectadas. O cérebro de um mamífero pode chegar a ter bilhões de neurônios. Por sua vez, uma rede neural de porte consideravelmente grande pode chegar a ter milhares de unidades de processamento.

A unidade fundamental do sistema nervoso dos animais é a célula nervosa, o neurônio. Um neurônio é um conjunto extremamente complexo de células, que é capaz de transmitir impulsos nervosos a outros neurônios, bem como a células musculares e glandulares. É constituído por 3 principais componentes, que são:

- Os dendritos, prolongamentos responsáveis pela recepção de estímulos nervosos provenientes de outros neurônios ou do ambiente;

- O corpo celular, a região na qual está localizada o núcleo do neurônio; e,
- O axônio, prolongamentos responsáveis pela condução dos impulsos elétricos produzidos no corpo celular.

Os dendritos recebem estímulos e então os transmitem para o corpo celular (ou soma). O soma combina e processa as informações recebidas. Por fim, de acordo com alguns fatores, o corpo celular gera um novo impulso e o envia para o amônio. Um esquema de um neurônio simplificado pode ser visualizado na figura 8.

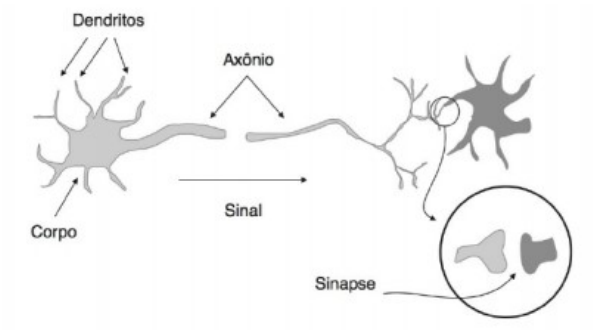

Figura 8: Neurônio biológico simplificado [4]

O conjunto de neurônios conectados uns aos outros formam a chamada rede neural. A sinapse é a região na qual dois neurônios entram em contato, cuja principal função é estabelecer uma comunicação, que se concretiza através da transmissão de impulsos nervosos entre eles. Estima-se que o cérebro humano possua uma quantidade de neurônios da ordem de 10 a 500 bilhões. Cada neurônio pode estar conectado a centenas ou milhares de outros neurônios. O resultado disso é a grande capacidade de processamento e armazenamento de informações.

Nessa perspectiva, é possível definir uma rede neural artificial como sendo uma estrutura complexa formada por várias unidades de processamento simples, os neurônios. Esses elementos de processamento, que são dispostos em uma ou mais camadas interligadas por um grande número de conexões, computam funções matemáticas. O comportamento inteligente de uma RNA origina-se das interações entre as unidades de processamento da rede. Essas conexões fazem o papel das sinapses biológicas e estão associadas a determinado peso. Os pesos codificam o conhecimento adquirido pela rede e podem assumir valores positivos ou negativos. No primeiro caso, a conexão tem comportamento excitatório, enquanto no segundo, inibitório; isto é, o sinal é transmitido ou não, respectivamente.

4.1.2 Arquitetura de uma rede neural artificial

O primeiro modelo computacional de um neurônio foi proposto em 1943 pelo neurocientista Warren McCullock e pelo matemático Walter Pitts. A sua operação pode ser resumida da seguinte forma:

- Sinais são apresentados à entrada;
- Cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade;
- É feita a soma ponderada dos sinais que produz um nível de atividade;

- Se este nível de atividade exceder um certo limite a unidade produz uma determinada resposta de saída.

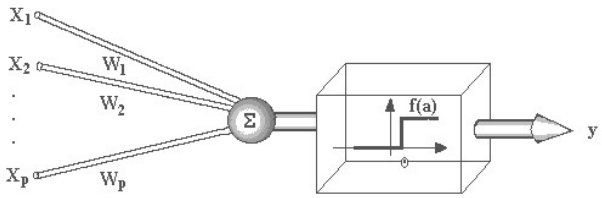

Figura 9: Esquema de unidade McCullock - Pitts [3]

Considerando a figura 9, suponha que:

- X1, X2, Xp e W1, W2, ..., Wp representam, respectivamente, p sinais e pesos de entrada;
- Exista um limitador t.
- Os sinais podem assumir valores booleanos (0 ou 1), enquanto os pesos podem assumir valores do conjunto dos números reais.

Neste modelo, a operação é dada por:

$$a = W1X1 + W2X2 + ... + WpXp$$

Ou seja, a saída da função f é a resposta do neurônio para a entrada. Neste exemplo, ela pode ser é dada por y = 1, se a >= t ou y = 0, se a < t. A saída de um neurônio é formada por meio da aplicação de uma função de ativação. Comumente são utilizadas as funções linear, limiar e sigmoidal. Seu objetivo é limitar a amplitude de saída do neurônio, isto é, transmitir o valor 0 (comportamento inibitório) ou 1 (comportamento excitatório).

Uma RNA, pode possuir uma ou mais camadas de neurônios. A rede apresentada na figura 10, que ilustra um exemplo de RNA com três camadas, recebe como entrada valores de dois atributos e gera dois valores em sua saída. Ela, e outras redes compostas por mais de uma camada, podem ser denominadas de rede multicamada. A última camada é chamada de camada de saída, enquanto as demais são denominadas camadas intermediárias ou ocultas.

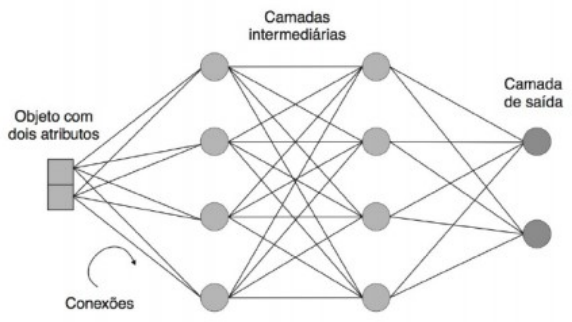

Figura 10: Exemplo de RNA multicamadas típica (FACELI et al, 2011, p. 111)[4]

As conexões entre os neurônios podem ser classificadas quanto ao seu grau de conectividade. Podendo ser completamente conectada, parcialmente conectada ou localmente conectada.

No primeiro e no segundo caso, os neurônios podem, respectivamente, estar conectados a todos os neurônios da camada anterior e seguinte ou a apenas alguns. Redes localmente conectadas são um tipo de rede parcialmente conectada em que os neurônios conectados entre si encontram-se em uma região bem definida.

Além disso, as RNAs podem apresentar conexões de retroalimentação (ou $feedback$). Geralmente, o sinal em uma rede neural flui das camadas de de entrada da rede em direção aos neurônios da camada de saída. Em redes com retropropagação, é possível que um neurônio receba em seus terminais de entrada a saída de um neurônio da mesma camada ou de uma camada posterior. RNAs sem conexões de retropapagação são denominas RNAs $feedforward$. A figura 11 ilustra exemplos desses tipos de rede.

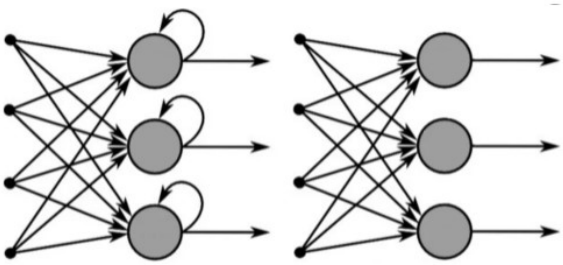

Figura 11: Redes $feedback$ e $feedforward$, respectivamente. [1]

4.1.3 Aprendizado de uma rede neural artificial

O processo de aprendizado resulta da execução iterativa de ajustes dos parâmetros. Esta etapa refere-se ao ajuste dos pesos associados às conexões da rede que melhora o desempenho do modelo. Assim sendo, essa etapa caracteriza o treinamento.

Na literatura existem diversos algoritmos de treinamento. Eles são divididos de acordo com os paradigmas de aprendizado supervisionado, não supervisionado ou por reforço (quando um agente externo avalia a resposta fornecida pela rede). Além disso, podem ser divididos em 4 agrupamentos, a saber: correção de erro, Hebbiano, competitivo e termodinâmico (ou Boltzmann).

4.1.4 Perceptron

Uma rede com todas as entradas conectadas diretamente com as saídas é chamada de rede neural de camada única ou rede perceptron. Redes que apresentam uma ou mais camadas intermediárias de neurônios e uma camada de saída são do tipo perceptron multicamadas multilayer perceptron).

A rede neural artificial perceptron foi introduzida por Frank Rosenblatt. Esse modelo é inspirado nos trabalhos de McCulloch e Pitts, é um dos mais antigos e utiliza um único neurônio, classificando o resultado de forma linear. Ele é capaz de gerar saídas do tipo binário (0 ou 1, isto é, verdadeiro ou falso). Essa capacidade é ideal para a detecção de fraudes financeiras, por exemplo, tendo em vista que o objetivo é classificar se uma transação é fraudulenta (1) ou não (0).

Para se obter a classificação desejada, a principal etapa do treinamento supervisionado do modelo perceptron consiste em ajustar os pesos e os limiares. Após uma entrada ser apresentada à rede, uma saída é produzida. É calculada, então, a distância (também chamada de erro) entre a resposta atual (saída gerada), que pode ser representado por um estimador, e a desejada. Por meio dela os ajustes dos pesos das conexões são realizados, com o objetivo de minimizar o

erro. Esse processo é efetuado até que o erro seja satisfatoriamente pequeno. Ele representa a aprendizagem da rede. A figura 12 apresenta um esquema simplificado do treinamento de um perceptron.

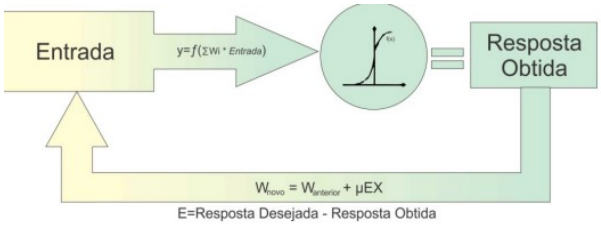

Figura 12: Esquema simplificado do treinamento de um perceptron. [7]

4.1.5 Avaliação da performance de uma rede neural artificial

Diz-se que o aprendizado ocorre quando a rede neural alcança um desempenho capaz de promover uma solução generalizada para uma classe de problemas. O resultado disso é a habilidade de resolver problemas do domínio em questão.

Uma das formas de avaliar a performance de uma RNA é o processo de dividir os dados em conjunto de treinamento e conjunto de testes. O conjunto de treinamento é usado para gerar o modelo e o conjunto de teste é usado para avaliar a capacidade de generalização. Em seguida, após o treinamento, uma matriz de confusão pode ser gerada. Ela contém quantidade de classificações corretas versus as classificações preditas para cada classe sobre um conjunto de exemplos. Em outras palavras, a matriz de confusão indica quais os erros e acertos do modelo.

A partir disso pode-se extrair, para cada classe, as variáveis: verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo.

- Verdadeiro positivo: ocorre quando no conjunto real, a classe que estamos buscando foi prevista corretamente.
- Falso positivo: ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente.
- Falso verdadeiro: ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente.
- Falso negativo: ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente.

Dessa forma é possível utilizar alguma métricas de avaliação como, por exemplo:

- Acurácia: porcentagem de elementos classificados corretamente (positivos ou negativos),
- Acurácia por classe: média das acurácias individuais para cada classe;
- Acurácia balanceada: média entre sensibilidade e especificidade;
- Precisão: que retorna, dentre os exemplos classificados como positivos pelo modelo, quantos eram realmente verdadeiros;
- Revocação ou *recall*: que define, dentre todas as situações de classe positiva (dos valores esperados), quantas foram classificadas como verdadeiras;
- F-Score: que representa a média ponderada de precisão e revocação.

4.2 Como o modelo foi instanciado

Na implementação do modelo foi gerado um classificador a partir do conjunto de treinamento. O classificador atribui às amostras a sua classe, com o objetivo de prever o valor esperado. Para avaliar a performance e corretude do modelo são feitos testes e geradas métricas. O classificador utilizado é uma rede neural perceptron, acíclica e de uma única camada. Para utilizar como base, foi treinada uma RNA com hiperparâmetros padrão.

Para encontrar uma configuração de rede melhor, foi utilizada a estratégia de busca exaustiva empírica até encontrar um resultado melhor que a RNA base. A busca em grade percorre um conjunto de parâmetros, treinando uma rede para cada combinação e utilizando a que produz a melhor métrica desejada. A métrica utilizada é a acurácia balanceada. Foram os parâmetros *eta*0, *random_state* e *warm_start*, em que:

- *eta*0 é a taxa de aprendizagem, a busca selecionou um valor de 0.5 ao invés do padrão 1;
- *random_state* é o estado inicial do gerador interno de números aleatórios, a busca selecionou o valor 70087306;
- *warm_start* permite que o treinamento utilize os atributos obtidos em treinamentos anteriores como a sua inicia inicialização, melhorando o desempenho da busca.

4.3 Descrição dos dados utilizados no modelo

De todos os atributos disponíveis na base de dados, não foram utilizados os seguintes: *nameDest*, *nameOrig*, $isFlaggedFraud$. Além dos atributos *newbalanceOrig*, *oldbalanceOrg*, *oldbalanceDest*, *newbalanceDest* e $isFraud$, foram gerados duas novas a partir dessas 4 primeiras. Conforme definido na seção 3.3, essas duas novas colunas são *errorbalanceOrig* e $errorbalanceDest$. A coluna *errorbalanceDest* foi gerada a partir da diferença entre *oldbalanceDest* e *newbalanceDest* e *errorbalanceOrig* a partir da diferença entre *newbalanceOrig* e *oldbalanceOrg*.

Do total dos dados 30% foram separados para teste e 70% divididos para a etapa de treinamento.

4.3.1 Normalização

Como visto anteriormente, há uma grande discrepância na escala dos valores numéricos. A solução para essa questão é a normalização, que é a re-escala dos dados numéricos dentro de uma faixa baseada em alguns critérios. O pacote preprocessing da Scikit-Learn fornece algumas funções de padronização. A técnica escolhida é denominada Standard Scaler, que não leva em consideração a forma da distribuição e transforma o dado para forma com média próxima de 0 e um desvio padrão próximo a 1

4.3.2 Balanceamento

O desbalanceamento das classes é tratado com técnica de *oversampling* por $SMOTE$ (Synthetic Minority Over-sampling Technique). A técnica gera novas amostras sintéticas da classe minoritária para aumentar a sua representatividade no conjunto de dados, sem duplicar amostras existentes. As amostras sintéticas são geradas por meio de interpolação no espaço das $features$ da classe minoritária, criando amostras similares às reais mas sem igualdade. O

oversampling feito equilibra as classes para uma proporção 1:1. A figura 13 mostra o resultado do balanceamento.

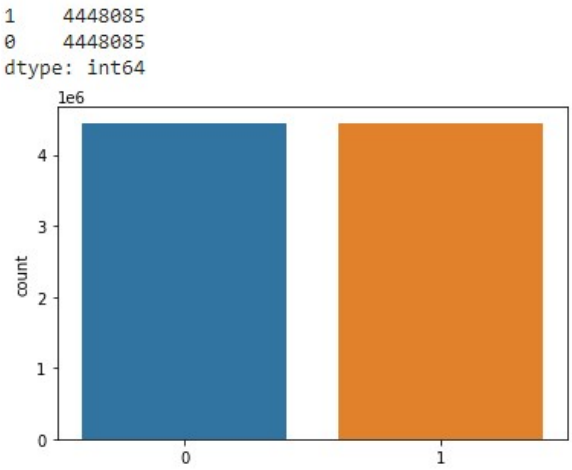

Figura 13: Resultado do *oversampling*.

5 Resultados obtidos

A rede padrão obteve uma acurácia de 93,85%, com mais de 100.000 classificações incorretas. Um *recall* de 96,38% significa que próximo de 4% das fraudes não foram reconhecidas. Ao calcular a média do *recall* sobre as fraudes e não fraudes é obtida a acurácia balanceada de 95,12%, representando o quanto a rede evitou erros de forma geral. A matriz de confusão é apresentada na figura 14.

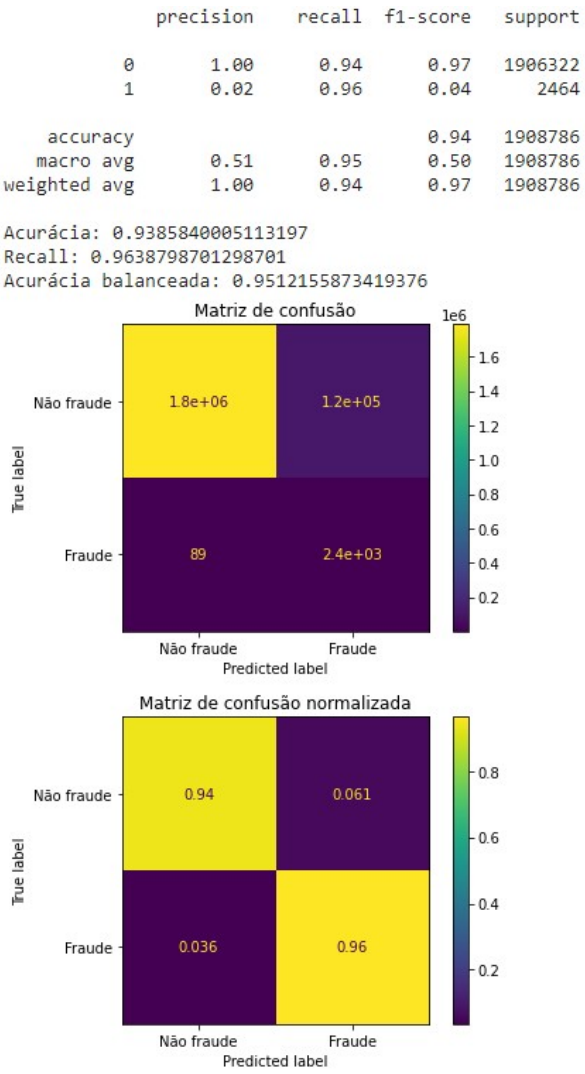

Figura 14: Matriz de confusão da uma rede perceptron padrão.

Em seguida, a busca exaustiva empírica foi realizada. A melhor rede encontrada apresenta os seguintes parâmetros e métricas (figura 15):

- eta0: 0.5;
- random_state: 70087306;
- warm_start: True.

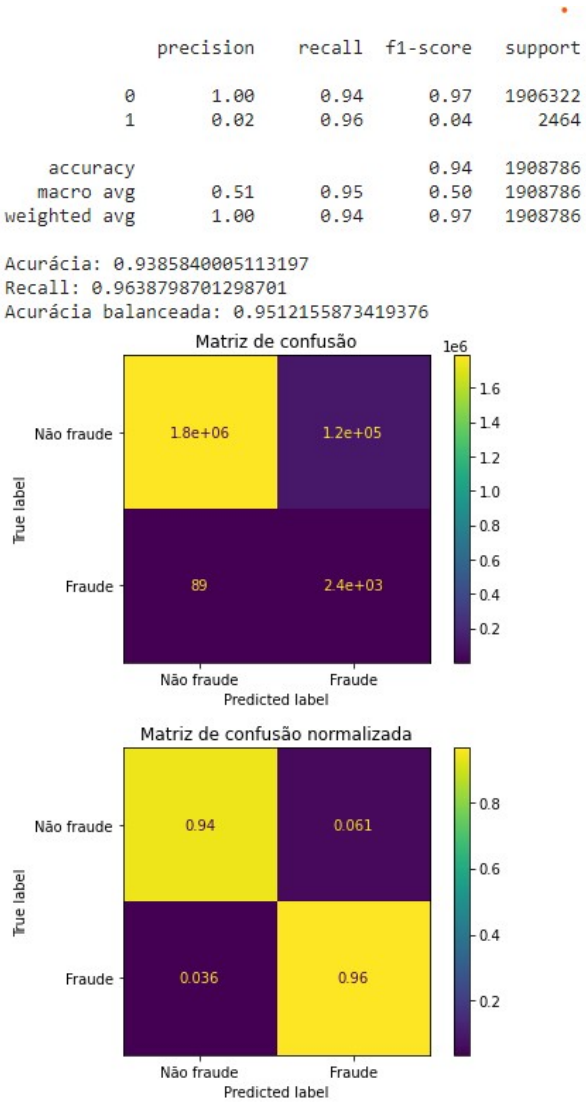

Figura 15: Matriz de confusão da melhor rede encontrada.

A rede otimizada obteve uma acurácia de 94,84%, com menos de 100.000 classificações incorretas. Um *recall* de 96,50% significa que 3,5% das fraudes não foram reconhecidas. Ao calcular a média do *recall* sobre as fraudes e não fraudes é obtida a acurácia balanceada de 95,67%, representando o quanto a rede evitou erros de forma geral.

6 Instruções de uso

Para que seja possível utilizar o modelo, antes de tudo deve-se criar uma conta do Google (*accounts.google.com*). Em seguida, se necessário, salve o arquivo *.ipynb* no Google Drive. Após isso, abra o arquivo com Google Colaboratory.

Algumas células de código dependem da execução de código de células anteriores. Células de código que são dependências de células posteriores possuem no início o comentário **#DEPENDÊNCIA**. Neste caso elas devem ser executadas com sucesso antes de prosseguir.

Siga os passos abaixo:

1. **Upload de arquivo** - Realize o *upload* da base de dados para o sistema de arquivos local.

O caminho é automaticamente configurado para o arquivo enviado. Execute a célula em questão e clique em **Escolher arquivos**. Em seguida selecione a base de dados.

2. **Montar o Google Drive localmente** - Monta o Google Drive no sistema de arquivos local. Utilize a célula de configuração para definir o caminho do arquivo da base de dados.

3. **Configurar o caminho da base de dados** - Neste passo é atribuído o caminho do arquivo no sistema de arquivos local a ser usado como base de dados.

4. **Seleção dos dados** - Todas conversões e limpeza são realizadas nesta etapa. Gera os conjuntos de dados para a implementação do modelo.

5. **Implementação do modelo** - Etapa em que a rede neural é criada.

7 Conclusão

Considerando os resultados apresentados na seção 5, pode-se afirmar que a utilização de redes neurais artificiais na detecção de transações financeiras pode ser potencialmente benéfica. A matriz de confusão mostradas na figura 16 avaliam o desempenho da primeira rede implementada no projeto, a rede padrão. Enquanto que a matriz de confusão mostradas na figura 17 avaliam a performance da rede otimizada. Comparando-as, é possível perceber que, em geral, houve melhora. A quantidade de falso negativo reduziu, enquanto que a quantidade de verdadeiro positivo aumentou.

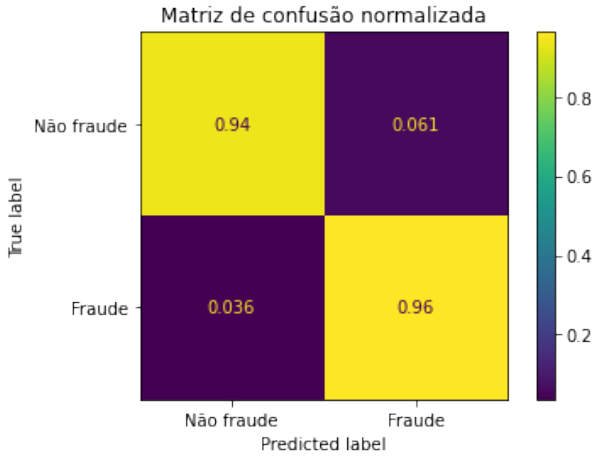

Figura 16: Matriz de confusão da rede padrão.

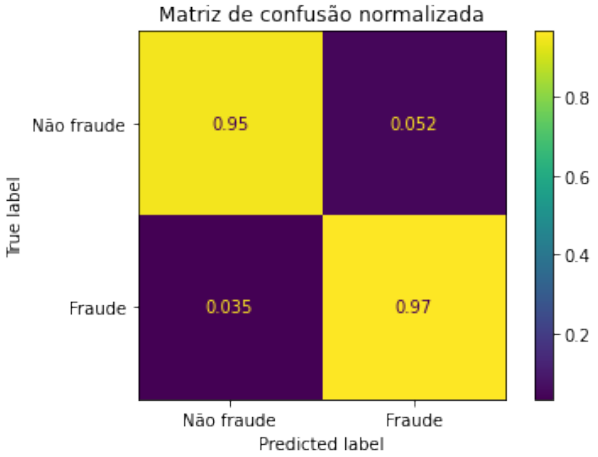

Figura 17: Matriz de confusão da rede otimizada.

Tabela 7: Métricas da rede otimizada

#	precision	recall	f1-score
0	1.00	0.95	0.97
1	0.02	0.97	0.05

Uma RNA, assim como o cérebro humano, aprende por experiência. Então, ao se analisar os resultados, deve ser considerado o fato de que o conjunto de dados pode não ser suficientemente grande, conforme explanado na seção 3.1. Assim sendo, medidas podem ser tomadas. Entre elas, pode-se cogitar a possibilidade de criar grupos de controle e obter apoio de análises manuais para aumentar a quantidade de exemplos, que podem ser utilizados em treinamentos.

Em grupos de controle, transações visivelmente fraudulentas seriam intencionalmente aprovadas. Além disso, parte das transações podem ser enviadas para análises manuais, as quais seriam realizadas pelas equipes de análise de fraudes. Essas práticas são realizadas por empresas como Konduto e Picpay. Elas auxiliam na medição de falsos positivos o que, consequentemente, resulta em modelos cada vez melhores.

7.0.1 RNA versus árvore de decisão

No trabalho anterior, foi utilizado a técnica de árvore de decisão para tentar resolver o mesmo problema do escopo deste trabalho. A seguir são apresentados os seus resultados. De um modo geral, a árvore de decisão, para esse conjunto de dados, obteve um desempenho melhor. Essa conclusão não significa que RNA não seja ideal para esse tipo de problema. Como foi visto anteriormente, o processo de treinamento da RNA melhorou seus resultados.

Tabela 8: Métricas da árvore de decisão

#	precision	recall	f1-score
0	1.00	1.00	1.00
1	0.72	0.70	0.71

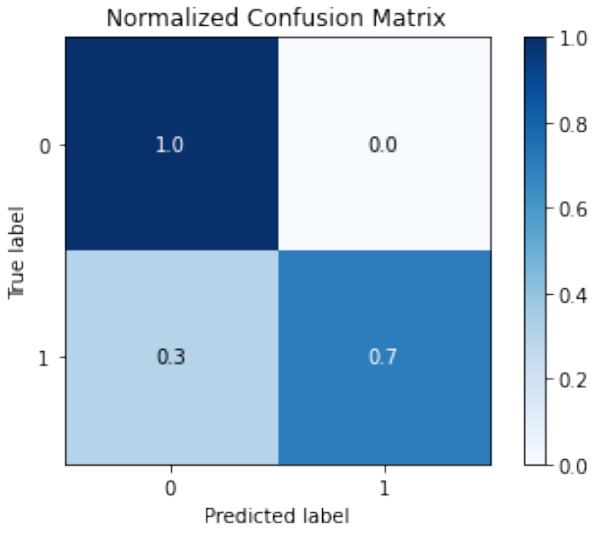

Figura 18: Matriz de confusão da árvore de decisão.

Referências

[1] Academy, D. S. **Deep Learning Book**. https://www.deeplearningbook.com.br/, último acesso em Abril de 2021, 2020.

[2] Barroso, F.; Cook, I.; Amirante, J. C.; Castillo, A. D.; Faulkner, J.; Harloff, G. E.; Romero, R.; Wheis, B. **Inteligência Artificial**. kroll.com/pt-br/publicacoes/global-fraud-risk-report-2019, último acesso em Abril de 2021, 2020.

[3] de L. F. de Carvalho, A. P. **Redes Neurais Artificiais**. https://sites.icmc.usp.br/andre, último acesso em Abril de 2021, 2009.

[4] Faceli, K.; Lorena, A. C.; Gama, J.; de Carvalho, A. C. P. L. F. **Inteligência artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro: LTC, 2011.

[5] Febraban. **Mobile banking é canal preferido dos brasileiros para pagamento de contas e transferências bancárias**. https://portal.febraban.org.br/noticia/3301/pt-br, último acesso em Abril de 2021, 2019.

[6] Lopez-Rojas, N. A.; Elmir, A.; Axelsson, S. **PaySim: A financial mobile money simulator for fraud detectionh**. https://www.kaggle.com/ntnu-testimon/paysim1, último acesso em Abril de 2021, 2017.

[7] Machado, V. P. **Inteligência Artificial**. http://www.uece.br/computacaoead, último acesso em Abril de 2021, 2020.

[8] Report, N. **Card Fraud Losses Reach**. https://nilsonreport.com/mention/812/1link/, último acesso em Abril de 2021, 2016.