Revista Tecnologia e Sociedade
ISSN: 1984-3526

https://periodicos.utfpr.edu.br/rts

Novo algoritmo *ensemble* para detecção de fraude em transações de cartão de crédito

Daniel Henrique Miguel de Souza
daniel.henrique@aluno.ufabc.edu.br
Mestre em Engenharia de Produção
Universidade Federal do ABC,
Santo André, São Paulo.

Claudio José Bordin Júnior
claudio.bordin@ufabc.edu.br
Doutor em Engenharia Elétrica
Universidade Federal do ABC,
Santo André, São Paulo.

RESUMO

Transações fraudulentas em operações com cartão de crédito geram perdas financeiras expressivas, incentivando o desenvolvimento de algoritmos capazes de detectá-las. Nesta linha, propõem-se neste artigo novas técnicas de aprendizado de máquina para a solução de problemas de detecção binária com classes desbalanceadas, ou seja, para os quais uma das classes (e.g., ocorrência de uma fraude) é bem menos frequente que a outra. As técnicas propostas combinam classificadores de formulações distintas, treinados com conjuntos de dados obtidos através de diferentes formas de amostragem. A combinação de classificadores é realizada através de voto majoritário ou do novo esquema de voto singelo, que visa aumentar a taxa de detecção de fraude. Os algoritmos propostos tiveram os seus desempenhos avaliados através de simulações numéricas utilizando dados de transações financeiras reais. Os resultados das simulações indicaram que os novos algoritmos exibem métricas de detecção vantajosas em relação a técnicas do estado-da-arte.

PALAVRAS-CHAVE: Detecção de Fraudes. Aprendizado de Máquina. Aprendizado Combinado (*Ensemble Learning*). Inteligência Artificial.

INTRODUÇÃO

O cartão de crédito é um dos produtos financeiros mais visados pelos fraudadores dada a simplicidade em se desviarem altos valores, além da usual demora na descoberta de fraude (AWOYEMI *et al.,* 2017). O montante de fraudes neste tipo de operação atingiu US$ 32,39 bilhões no mundo em 2020 e apresenta tendências de crescimento (SAVVY, 2020). Segundo (SAVVY ,2020), houve em 2020 um impacto financeiro 3 vezes maior ao observado em 2011, que foi de US$ 9,84 bilhões. Além disso, é previsto que este valor chegue a US$ 40,63 bilhões em 2027, com uma proporção de casos de fraude de 5,68 centavos para cada 100 dólares movimentados (SAVVY, 2020). Diante disso, o desenvolvimento de modelos estatísticos para detecção de fraude é de extrema importância para mitigar o risco neste tipo de operação. Os modelos de detecção de fraude atuam na etapa de checagem de políticas de crédito para aprovação ou não de transações, reconhecendo padrões de comportamentos fraudulentos na operação em questão. Esta abordagem possibilita a estimação de possibilidade de fraude em tempo real (AWOYEMI *et al.,* 2017).

Atualmente, predomina-se o uso de técnicas de Aprendizado de Máquina (*Machine Learning* - ML) para detecção de fraude em operações com cartão de crédito, usualmente utilizando uma abordagem de classificação, com aprendizado supervisionado. Destacam-se para esta finalidade os modelos baseados em Árvores de Decisão (*Decision Trees* – DT) (XUAN, 2018), k-Vizinhos-mais-Próximos (*k-Nearest Neighbors* - kNN) (ITOO, 2021), Redes Bayesianas (AWOYEMI *et al.,* 2017) e Redes Neurais (*Neural Networks* - NN) (YANG; SHAMI, 2020).

Em (MAES, 2002), comparou-se o desempenho de Redes Bayesianas (*Bayesian Networks* - BN) e de NN na detecção de fraudes usando dados de transações reais ocorridas no Brasil. Neste estudo, as BN apresentaram um melhor resultado sob as principais métricas de desempenho de classificadores. Na referência (KANG, 2016), é demonstrada a viabilidade do uso de NN no mesmo problema. Em (SHEN *et al.,* 2007), por sua vez, compararam-se o desempenho de NN, DT e de classificadores baseados em Regressão Logística (Logistic Regression - LR) para o mesmo fim, concluindo-se que as NN levam a melhores resultados. Em (AWOYEMI *et al.,* 2017), por outro lado, utilizaram-se modelos Bayesianos Ingênuos (*Naïve Bayes* - NB), de LR e kNN, tendo-se observado um melhor desempenho para o kNN. Em (ITOO, 2021), compararam-se, sob condições distintas, os mesmos algoritmos utilizados em (AWOYEMI *et al.,* 2017), observando-se um melhor desempenho para a LR.

Um dos principais desafios na detecção de transações fraudulentas é a sua baixa frequência relativa comparada à de transações genuínas. O uso de métodos de ML tradicionais para problemas deste tipo, com classes desbalanceadas (AWOYEMI et al., 2017), geralmente produz desempenhos insatisfatórios. Três técnicas distintas são comumente empregadas para amenizar esse problema: na primeira, os dados são rebalanceados (AWOYEMI *et al.*, 2017) durante o pré-processamento, por exemplo, amostrando-se com reposição da classe minoritária até que a frequência relativa das classes se equilibre. A segunda possibilidade consiste em modificar os algoritmos de ML para acomodar o desbalanceamento de classes (LUO, 2019). Por fim, na terceira, o efeito das classes desbalanceadas é minimizado por meio da combinação de métodos (*ensembles*) de ML (SOHONY *et al.*, 2018).

A principal contribuição deste artigo é introduzir um novo método *ensemble* para a melhoria de desempenho em problemas de detecção de fraude com classes desbalanceadas. Especificamente, propõe-se a combinação de algoritmos de ML a partir do voto majoritário (COLETTA, 2016) e do método proposto de **voto singelo**. Além disso, o método proposto inova ao combinar classificadores treinados de maneiras distintas, considerando o balanceamento das classes pela sobreamostragem da classe minoritária ou pela subamostragem da classe majoritária (SOHONY *et al.*, 2018).

O texto a seguir está organizado da seguinte forma: na Seção **Algoritmos de Aprendizado de Máquina**, descreve-se brevemente o funcionamento dos classificadores implementados. Na Seção **Metodologias Clássicas de Amostragem**, por sua vez, são descritos alguns dos principais métodos de amostragem utilizados para balanceamento de classes. Em seguida, na Seção **Métodos Agregados**, detalha-se o funcionamento de esquemas de combinação de classificadores e, na Seção **Novo Algoritmo para Combinação de Classificadores,** o novo método de combinação proposto. Na Seção **Metodologia**, descrevem-se o pré-processamento dos dados, configurações e métricas de desempenho utilizadas. Na Seção **Resultados e Discussões**, são descritas as simulações realizadas para avaliar o desempenho do algoritmo proposto em diferentes cenários, comparando o mesmo com o estado da arte. Por fim, na Seção **Considerações Finais**, são apresentadas as considerações finais e propostas de continuidade deste estudo.

ALGORITMOS DE APRENDIZADO DE MÁQUINA

Neste trabalho, foram utilizados 6 classificadores, abrangendo suas formas clássicas, além destas com algumas modificações. Os modelos kNN, NB, LR, *Random Forest* (RF), *Gradient Boosted-Tree* (GBT) e *Multilayer Perceptron* (MLP) foram implementados individualmente e sob uma abordagem de classificadores agregados (SOHONY *et al.*, 2018).

O kNN é um algoritmo não paramétrico que parte da premissa de que em um conjunto de dados, as observações semelhantes estão próximas de acordo com alguma métrica de distância (ITOO, 2021). O objetivo do kNN é classificar um objeto a partir de um conjunto de pontos rotulados corretamente observados no conjunto de treinamento. Para isto, calcula-se uma medida de distância entre o novo objeto e cada observação do conjunto de treinamento, de modo a encontrar as $\boldsymbol{k}$ observações (vizinhos) mais próximos do objeto de interesse. Definidos os $\boldsymbol{k}$ vizinhos mais próximos, o rótulo da nova observação é definido pelo voto majoritário entre os rótulos das $\boldsymbol{k}$ observações mais próximas deste, gerando uma estimativa da classe.

O NB é um classificador *Bayesiano* que visa classificar um objeto a partir de regras probabilísticas, utilizando o teorema de Bayes. Para isso, existe a premissa de independência condicional entre os atributos dada a classe, ou seja, não são consideradas (se existirem) as dependências estatísticas entre as *features* utilizadas para treinar do modelo (MAES, 2002).

O modelo LR (YANG; SHAMI, 2020) é um método estatístico aplicável a cenários em que a variável resposta é binária. Este modelo visa gerar uma função que tenha como saída a probabilidade de um exemplo pertencer a uma classe,

baseado no comportamento das *features*. A transformação logística consiste em transformar a variável resposta em uma razão de probabilidades, que em sequência é convertida a variável com base logarítmica. Dada a não-linearidade desta operação, estimam-se os coeficientes deste modelo por meio da função de máxima verossimilhança (YANG; SHAMI, 2020).

As DTs (YANG; SHAMI, 2020) são modelos estatísticos em que se tem um fluxograma de informação em formato de árvore, com intuito de particionar os dados com algum critério, de modo a encontrar "nós" que melhor particionem os dados, buscando a menor árvore possível. Em DTs, utiliza-se a estratégia de "dividir para conquistar" (YANG; SHAMI, 2020), particionando os dados em sub-regiões (retângulos) até que todas as folhas sejam puras, ou até atingir um critério de parada. Para partição dos dados, aplica-se algum cálculo que define o "Ganho de Informação" que determinada variável traz para os dados em questão, selecionando para o teste o atributo com maior ganho de informação. Este algoritmo é extremamente utilizado, pois é base para vários algoritmos de ML mais robustos (YANG; SHAMI, 2020), como RF (XUAN, 2018) e DT com o uso de *boosting* (YANG; SHAMI, 2020), descritas a seguir.

Bagging (*Bootstrap Aggregating*) (YANG; SHAMI, 2020) é uma técnica para gerar múltiplos preditores a partir de amostras *bootstrap* (XUAN, 2018) do conjunto de treinamento, os quais são reunidos em um único preditor. A amostragem *bootstrap* consiste em construir amostras de tamanho $\boldsymbol{n}$, a partir de uma amostragem com reposição da amostra $\boldsymbol{N}$. Existem modelos baseados em DTs que utilizam o conceito de *Bagging Predictors*. Um deles é o *Bagging Classification Trees* (BCT), em que a partir de amostras *bootstrap*, são construídas $\boldsymbol{m}$ DTs utilizando todas as variáveis do conjunto de dados utilizado, sendo a classificação final de uma nova observação feita pelo voto majoritário entre as classificações das DTs construídas. Uma variação deste método é o RF (XUAN, 2018), se diferenciando do BCT na etapa de construção de cada DT, em que são utilizadas para cada classificador uma porção das variáveis disponíveis selecionadas aleatoriamente (XUAN, 2018), gerando assim preditores descorrelacionados.

Os métodos BCT e RF são baseados na construção paralela de $\boldsymbol{m}$ árvores de decisão e classificação final por voto majoritário. Outro método baseado em DT utilizado amplamente é o GBT (YANG; SHAMI, 2020). Este modelo utiliza da técnica de *Boosting* para construção do modelo, sendo um preditor construído sequencialmente a partir do resíduo do preditor anterior (YANG; SHAMI, 2020).

As NN são modelos supostamente baseados no funcionamento cognitivo, utilizados, dentre outros fins, para reconhecer padrões em conjuntos de dados. Este tipo de modelo é composto por um conjunto de neurônios artificiais distribuídos em camadas, sendo cada neurônio um classificador (YANG; SHAMI, 2020). A rede MLP é composta por pelo menos 3 camadas, sendo estas uma camada de entrada, uma camada oculta e a camada de saída. Cada nó da rede (com exceção dos nós de entrada) utiliza uma função de ativação e um bias para transladar a função de ativação, caso necessário. O objetivo do MLP é encontrar os pesos das entradas que minimizem o erro do algoritmo, a partir de um algoritmo de retropropagação do erro (*Backpropagation*) (YANG; SHAMI, 2020).

Tendo em vista o desbalanceamento de classes neste trabalho, alguns autores propõem modificações em alguns modelos de ML que sofrem influência deste problema, trazendo uma possibilidade de ponderar as classes conforme sua

frequência em um conjunto de dados. Para a LR, (LUO, 2019) propõe a modificação da função de log-verossimilhança, incluindo pesos nas classes de acordo com sua frequência. LUO (2019) apresenta uma modificação análoga no RF, atribuindo às classes pesos inversamente proporcionais a frequência de ocorrência na classe, na etapa de voto majoritário. Esta ponderação pode ser estendida para outros métodos como o GBT (ALSHARKAWI, 2021) e o MLP (HUANG, 2016). Para o kNN, propõe-se aqui uma variação do método para cálculo dos pesos, em que ao contrário da ponderação pela distância, os pesos das classes de cada vizinho mais próximo assumem valores inversamente proporcionais à frequência de ocorrência da classe no conjunto de dados de treinamento.

METODOLOGIAS CLÁSSICAS DE AMOSTRAGEM

Como mencionado na **Introdução**, um dos grandes desafios na detecção de fraude crédito é a presença de desbalanceamento de classes nos conjuntos de dados utilizados para tal fim. Isto gera um viés nos modelos em rotular o objeto de interesse como não-fraude (transação genuína). Para resolver este problema, são propostos métodos para balanceamento das classes, visando obter uma proporção de 50% em ambas as classes no conjunto de dados de treinamento, e assim, eliminar este viés (SOHONY *et al.*, 2018).

As abordagens clássicas se baseiam na sobreamostragem da classe minoritária e subamostragem da classe majoritária (SOHONY *et al.*, 2018). Na subamostragem, realiza-se uma amostragem aleatória da classe majoritária sem reposição, até que se tenha a mesma proporção entre esta e a classe minoritária (SOHONY *et al.*, 2018). Já na sobreamostragem, utilizam-se técnicas para gerar dados a partir da classe minoritária de forma consistente. As mais conhecidas dentre estas são *Synthetic Minority Over-sampling Technique* (SMOTE) e *Random Over Sampling* (SOHONY *et al.*, 2018).

O método SMOTE é uma técnica de aumento de dados (*Data Augmentation*) (SOHONY *et al.*, 2020) para geração de novas instâncias a partir da combinação entre instâncias desta classe por meio do algoritmo kNN, construindo assim amostras sintéticas. A ideia é que se escolha aleatoriamente uma amostra da classe minoritária, e a partir deste exemplo, selecionam-se os k vizinhos mais próximos. Dentre estes, escolhe-se um vizinho aleatoriamente, e gera-se um novo exemplo a partir de uma combinação entre a amostra inicial e o vizinho selecionado. Isto é feita até que as classes estejam balanceadas (SOHONY *et al.*, 2020). Já na metodologia *Random Over Sampling*, as amostras da classe minoritária são replicadas aleatoriamente, gerando cópias destas instâncias (SOHONY *et al.*, 2018).

Quanto ao desempenho, (SOHONY *et al.,* 2018) mostra que as técnicas *Random Over Sampling* e SMOTE apresentam como vantagem um ganho de desempenho em relação à subamostragem da classe majoritária. Porém, (SOHONY *et al.,* 2018) defende que, ao utilizar subamostragem, o treinamento dos classificadores é expressivamente mais rápido.

Já (BAESENS *et al*., 2021), defende que não há um consenso quanto ao melhor desempenho entre o SMOTE e o *Random Over Sampling,* quando estes são comparados. (BAESENS *et al.,* 2021) mostra um desempenho ligeiramente melhor

de um método ou outro a depender das características dos conjuntos de dados utilizados, classificadores, dentre outros aspectos.

MÉTODOS AGREGADOS

Além da fragilidade de cada modelo de ML, a crescente complexidade dos problemas torna necessária a construção de métodos aprimorados para resolução destes. Uma proposta para melhoria de desempenho no processo de aprendizado de máquina é a abordagem combinada de modelos de ML, também conhecida como *ensembles*. Este tipo de metodologia visa fortalecer o poder preditivo e diminuir o viés de uma classe para o caso em questão (COLETTA, 2016).

Este tipo de metodologia é bem comum utilizando modelos da mesma classe. Têm-se por exemplo os métodos *ensembles* baseados em DT utilizados neste trabalho, em que são construídas DT paralelamente e a classificação é feita por meio de voto majoritário em relação à classificação de cada árvore (BCT e RF), ou têm-se DT construídas sequencialmente, a partir do resíduo da árvore anterior (GBT). Têm-se ainda as próprias NN, que consistem em conjuntos de neurônios artificiais combinados, visando a resolução de problemas de difícil abstração.

Segundo (COLETTA, 2016), a utilização de *ensembles* originados da agregação de classificadores parecidos leva a um erro de predição em massa quando o método em questão não é o mais adequado para a estimação do objeto em análise. Para minimizar este efeito, existe também a abordagem de combinação de modelos distintos, como por exemplo LR e métodos baseados em DT, NN, kNN e BN, dentre outras possibilidades. Para este fim, a classificação final é realizada por meio de algum método de agregação entre as classificações de cada método, sendo o mais utilizado a Combinação de Classificadores por meio de Voto Majoritário (CC-VM). O funcionamento da agregação de classificadores por meio de voto majoritário pode ser descrito pelo Algoritmo 1 (COLETTA, 2016).

Algoritmo 1: CC-VM

1. **for** $\boldsymbol{i = 1}$ **to** $\boldsymbol{n}$:

1.1. **for** $\boldsymbol{j = 1}$ **to** $\boldsymbol{m}$:

1.1.1. Classifique o objeto $\boldsymbol{i}$ utilizando o classificador $\boldsymbol{M_j}$, sendo o resultado definido como $\boldsymbol{y_{ij}}$

1.2. end for

1.3. $\boldsymbol{y_i}$ = Classe mais frequente entre todas as classificações para $\boldsymbol{i}$, ou seja, $\boldsymbol{y_i = MODA\{y_{i1}, y_{i2}, \cdots, y_{im}\}}$

2. **end for**

3. **return** $\boldsymbol{y = \{y_1, y_2, \cdots, y_n\}}$

Fonte: COLETTA (2016)

NOVO ALGORITMO PARA COMBINAÇÃO DE CLASSIFICADORES

Visando mitigar as fragilidades dos métodos de sobreamostragem e subamostragem descritos na Seção anterior, o algoritmo proposto combina diferentes classificadores, sendo estes treinados com balanceamento de classes a partir da técnica de sobreamostragem *Random Over Sampling* ou da técnica de subamostragem *Random Under Sampling*. Disto resultam *ensembles* híbridos, em que a classificação destes é feita ou Algoritmo CC-VM, ou pelo método CC-VS, sendo este último, descrito a seguir.

Combinação de classificadores por Voto Singelo (CC-VS)

Propõe-se nesta seção agregar as decisões dos modelos do *ensemble* por Voto Singelo, no qual a detecção de fraude por um ou mais modelos faz com que a classificação do *ensemble* assuma o rótulo fraude.

Empiricamente, verifica-se que o uso deste método reduz expressivamente o erro de classificação da classe minoritária, sem um aumento relevante no erro da classe maioritária. Vale ainda destacar que, para a detecção de fraude, como um falso negativo apresenta um impacto muito maior do que um falso positivo, se a consequência da melhora das classificações de fraude for um pequeno aumento de erro na detecção de transações genuínas, a utilização do método em questão ainda é válida.

Além disso, tendo em vista que este método dispensa a necessidade de avaliar a moda das decisões de cada classificador individual., e que, após uma detecção positiva por um único classificador, a execução dos demais algoritmos é interrompida, o Algoritmo CC-VS tem menor complexidade computacional do que o método CC-VM.

O funcionamento deste método pode ser representado pelo Algoritmo 2.

Algoritmo 2: CC-VS

1. **for** $\boldsymbol{i = 1}$ **to** $\boldsymbol{n}$:

1.1. **for** $\boldsymbol{j = 1}$ **to** $\boldsymbol{m}$:

1.1.1. Classifique o objeto $\boldsymbol{i}$ utilizando o classificador $\boldsymbol{M_j}$, sendo o resultado definido como $\boldsymbol{y_{ij}}$

1.2 **end for**

1.3. **if** $\sum_{i=1}^{m} \boldsymbol{y_{ij}} \geq \boldsymbol{1}$ **then** $\boldsymbol{y_i = 1}$

1.4. **else** $\boldsymbol{y_i = 0}$

1.5. **end if**

2. **end for**

3. **return** $\boldsymbol{y = \{y_1, y_2, \cdots, y_n\}}$

Fonte: Autoria Própria

O uso desta função pode ser utilizado em problemas de classes desbalanceadas no geral., trazendo um desempenho superior, ou pelo menos, equivalente ao clássico método CC-VM.

METODOLOGIA

Nesta seção, são apresentadas as etapas necessárias para realização das simulações para análise de desempenho dos algoritmos. Primeiramente, descrevem-se as metodologias utilizadas para seleção de variáveis, otimização de hiperparâmetros e treinamento dos classificadores implementados. Em seguida, são descritos os conjuntos de dados de fraudes utilizados para testes dos algoritmos em questão.

Pré-processamento dos dados

Como etapas de pré-processamento, destacam-se amostragem e separação dos conjuntos de dados em grupos. Inicialmente, o conjunto de dados de treinamento foi transformado em 2 diferentes conjuntos de dados balanceados, sendo o primeiro deles com a utilização de sobreamostragem da classe minoritária (AWOYEMI *et al.*, 2017), e o segundo com a utilização de subamostragem da classe majoritária (SOHONY *et al.,* 2018).

Para a seleção de variáveis, propõe-se um método híbrido a partir de uma combinação de métodos *filter* e *wrapper* (SHARDLOW, 2016). Tendo em vista o grande volume de dados das bases de fraude utilizadas, foram utilizados os métodos de *filter* Correlação de Pearson e *Mutual Information* (SHARDLOW, 2016), e o método *Wrapper Feature Importance* (KUHN *et al.,* 2013) sendo este avaliado a partir dos resultados de uma RF.

Para cada um destes métodos, é analisada a relevância de cada variável, e comparada com *benchmarks* da literatura específicas para a Correlação de Pearson (AKOGLU, 2018), *Mutual Information* (BRILLINGER, 2004) e *Feature Importance* (MENZE *et al.,* 2011). A partir da classificação de cada *feature* como relevante ou não-relevante, a escolha desta é feita por meio de voto majoritário entre as 3 classificações.

Configuração das simulações

No primeiro experimento, implementaram-se os algoritmos KNN, LR, RF, GBT, e MLP, sendo usadas tanto suas formulações clássicas, quando suas versões adaptadas para classes desbalanceadas, descritas na Seção **Algoritmos de Aprendizado de Máquina**. Para o MLP, cada simulação foi executada com as funções de otimização Adam (BOCK, 2019) e *lbfgs-Quasi-Newton* (BORHANI, 2020). Os modelos clássicos foram treinados e testados com conjuntos de dados balanceados, enquanto os adaptados receberam os conjuntos de dados originais. Todos os classificadores foram então ordenados pelo seu desempenho em ordem crescente, sendo esta ordenação realizada a partir das métricas de desempenho Sensitividade (sens), que representa porcentagem de acertos das transações fraudulentas, e *Balanced Classification Rate* (BCR), que representa a porcentagem de acerto médio entre ambas as classes, e, sendo selecionados então os 5 melhores classificadores treinados com sobreamostragem da classe minoritária, e

os 5 melhores classificadores treinados a partir da subamostragem da classe majoritária. Estas métricas são definidas a seguir, nas Equações (2) e (4).

Em seguida, foram implementadas as estratégias de *ensemble* CC-VM e CC-VS. Para o CC-VM, foram testados todos os *ensembles* possíveis contendo 3 e 5 modelos individuais, resultando assim em um total de 372 *ensembles* distintos. Vale destacar que, para evitar empates no voto majoritária, o número de métodos combinados para a estratégia CC-VM deve ser ímpar.

Para o CC-VS, por sua vez, foram testados todos os *ensembles* possíveis com 2, 3, 4 e 5 modelos individuais, totalizando 627 *ensembles*. Os resultados deste experimento são mostrados na Seção **Resultados e Discussões**.

Não foram testadas configurações com mais de 5 modelos, devido a restrições de complexidade computacional., considerando que os métodos propostos devem rodar em tempo real (THORNE *et al.,* 2017).

Os hiperparâmetros (YANG; SHAMI, 2020) de cada classificador foram determinados através de um procedimento de busca exaustiva da seguinte forma: foi selecionado um conjunto de valores de hiperparâmetros, e, em seguida, os métodos foram treinados e testados para cada combinação de valores possível, sendo selecionados os valores que maximizaram a métrica de desempenho F1 *Score* (F1), sendo esta definida como

$$F1 = \frac{T_P}{T_P + \frac{1}{2}(F_P + F_N)}, \tag{1}$$

usando *holdout* (YADAV; SHUKLA, 2016).

Para a validação dos hiperparâmetros, cada classificador foi treinado e testado via K-*Fold* (YADAV; SHUKLA, 2016) usando K = 10. Por este método, cada modelo foi treinado e testado 10 vezes; cada execução empregou 9/10 do conjunto de treinamento para treinamento e 1/10 para teste. Tomando como base o trabalho de (AWOYEMI *et al.,* 2017), para avaliação de desempenho, cada classificador foi avaliado pelas métricas sens, Specificidade (spec) e bcr, descritas por

$$sens = \frac{V_P}{V_P + F_N}, \tag{2}$$

$$spec = \frac{V_N}{V_N + F_P}, \tag{3}$$

$$bcr = \frac{1}{2}\left(\frac{V_P}{V_P + F_N} + \frac{V_N}{V_N + F_P}\right) = \frac{1}{2}(sens + spec), \tag{4}$$

em que **spec** representa a taxa de acertos das transações genuínas, e $\mathbf{V_P}$, $\mathbf{V_N}$, $\mathbf{F_P}$ e $\mathbf{F_N}$ representam a quantidade de verdadeiros positivos, verdadeiros negativos, falsos positivos e falsos negativos, respectivamente, sendo estes valores definidos conforme Tabela 1.

Tabela 1: Matriz de Confusão Binária

$Classe$	$predita\ C_+$	$predita\ C_-$
$verdadeira\ C_+$	V_P	F_N
$verdadeira\ C_-$	F_P	V_N

Fonte: MONARD (2003)

Conjuntos de dados de fraude

Para avaliar o desempenho dos algoritmos implentados, utilizaram-se dois conjuntos de dados: O primeiro deles consiste em transações de cartão de crédito realizadas na Europa em setembro de 2013, disponibilizados em (ULB, 2017). Trata-se de uma base de dados desbalanceada contendo 284807 transações, sendo destas, 284315 (99.83%) transações genuínas, e 492 (0.17%) fraudulentas. A base tem um total de 31 variáveis, sendo destas, uma variável contendo a data/hora da transação, uma variável com o rótulo da classe, com o valor 1 para fraude e 0 para não-fraude, e 29 variáveis explicativas. Este conjunto foi disponibilizado em duas partes, sendo uma delas de treino/teste, com 227485 transações (227455 transações genuínas e 390 fraudes) e a outra de validação do algoritmo, contendo 56962 observações (56860 transações genuínas e 102 fraudes).

Já o segundo (EDA, 2020), consiste em um conjunto de dados de uma instituição financeira Turca, disponibilizada pela empresa **Yapi Kredi Teknoloji**. Há neste conjunto um total de 62380 transações, sendo 61572 (98.70%) genuínas e 808 (1.30%) fraudes. A base tem um total de 26 variáveis, sendo destas, uma variável com data/hora da transação, uma variável com o rótulo da classe, com o valor 1 para fraudes e 0 para não-fraudes, e 24 variáveis explicativas. Estes dados foram disponibilizados em duas partes, sendo uma delas de treino/teste, com 31190 transações (30400 não-fraudes e 790 fraudes) e um conjunto de validação do algoritmo, contendo 31190 observações (31172 não-fraudes e 18 fraudes).

RESULTADOS E DISCUSSÕES

Para o experimento descrito na Seção **Metodologia**, visando melhorar a visualização dos resultados, as métricas de desempenho foram tabuladas, consolidando os resultados dos 5 melhores classificadores individuais com cada metodologia de amostragem utilizada, os três melhores *ensembles* usando o Algoritmo 1 (CC-VM), e os três melhores *ensembles* usando o Algoritmo 2 (CC-VS). As tabelas são classificadas primeiro pela métrica **bcr** e depois pela **sens.** A Tabela 2 descreve todas as siglas empregadas.

Para comparação, foram rodadas na mesma configuração: i) modelos individuais, ii) algoritmos baseados na estratégia de combinação CC-VS, algoritmos baseados na estratégia de combinação CC-VM, incluindo iv) o algoritmo CC-VM

introduzido em (PHUA, 2004), que agrega NB, MLP e DT simples, de forma análoga à de (BAGGA *et al.,* 2020), sendo este uma combinação apenas de classificadores treinados com sobreamostragem da classe minoritária (CC-VM-Over (PHUA, 2004)), ou sendo uma combinação de classificadores treinados com subamostragem da classe majoritária (CC-VM-Under (PHUA, 2004)). Como base, consideram-se métodos como de bom desempenho se suas taxas de classificação corretas forem de pelo menos 70% (MOEPYA *et al.,* 2014); sendo esta a referência para a análise a seguir.

Tabela 2: Siglas para Modelos Implementados

Modelo	**Sigla**
Naïve Bayes	NB
k-Nearest Neighbors	KNN
k-Nearest Neighbors Modificado	KNN-m
Regressão Logística	LR
Regressão Logística Modificada	LR-m
Random Forest	RF
Random Forest Modificado	RF-m
Gradient-Boosted Tree	GBT
Gradient Boosted-Tree Modificado	GBT-m
MLP com Otimizador Adam	MLP-A
MLP Modificado com Otimizador Adam	MLP-A-m
MLP com Otimizador lbfgs	MLP-l
MLP Modificado com Otimizador lbfgs	MLP-l-m
Árvore de Decisão	DT

Fonte: Autoria Própria

As Tabelas 3 e 4 listam, respectivamente, os resultados obtidos usando os conjuntos de validação das bases de dados europeias e turcas descritas na Seção **Metodologia**. Os números após os conjuntos CC-VM e CC-VS são os índices (arbitrários) das configurações de melhor desempenho descritas na Tabela 3. Já a Tabela 5, lista a configuração de cada *ensemble* apresentado nas Tabelas 3 e 4. Nas Tabelas 3, 4 e 5, o termo ***Under*** após a sigla do classificador indica que este foi treinado com subamostragem da classe majoritária, enquanto que, de forma análoga, o termo ***Over*** indica que o classificador foi treinado com sobreamostragem da classe minoritária.

Tabela 3: Desempenho de classificadores individuais e *ensembles* para o conjunto de dados de transações europeias (ULB, 2017).

Modelo	**bcr**	**sens**	**spec**
CC-VM-11	0.924	0.853	0.996
CC-VM-12	0.924	0.853	0.996

CC-VM-13	0.924	0.853	0.996
LR-Under	0.924	0.853	0.996
LR-m-Under	0.924	0.853	0.996
CC-VS-7	0.924	0.853	0.996
CC-VS-8	0.924	0.853	0.996
CC-VS-15	0.924	0.853	0.996
MLP-A-Under	0.920	0.853	0.987
MLP-m-Under	0.920	0.853	0.987
GBT-m-Under	0.920	0.873	0.967
KNN-Over	0.897	0.794	0.999
KNN-m-Over	0.897	0.794	0.999
LR-Over	0.892	0.784	0.999
LR-m-Over	0.892	0.784	0.999
MLP-l-Over	0.892	0.784	0.999
CC-VM-Under (PHUA, 2004)	0.698	0.951	0.445
CC-VM-Over (PHUA, 2004)	0.657	0.314	1.000

Fonte: Autoria Própria

Tabela 4: Desempenho de classificadores individuais e *ensembles* para o conjunto de dados de transações turcas (EDA, 2020).

Modelo	**bcr**	**sens**	**spec**
CC-VM-147	0.906	1.000	0.812
CC-VM-148	0.906	1.000	0.812
CC-VM-157	0.906	1.000	0.812
CC-VS-6	0.874	1.000	0.748
CC-VS-7	0.874	1.000	0.748
CC-VS-14	0.874	1.000	0.748
RF-Under	0.861	0.944	0.778
RF-m-Under	0.861	0.944	0.778
GBT-Under	0.858	1.000	0.716
GBT-m-Under	0.852	1.000	0.704
MLP-A-Over	0.840	0.778	0.902
MLP-A-m-Over	0.840	0.778	0.902
KNN-Under	0.838	0.944	0.731
CC-VM-Under (PHUA, 2004)	0.828	0.944	0.711

MLP-l-Over	0.786	0.889	0.683
CC-VM-Over (PHUA, 2004)	0.758	0.611	0.905
MLP-l-m-Over	0.720	0.722	0.719
LR-Over	0.663	0.500	0.826

Fonte: Autoria Própria

Tabela 5: Lista de Modelos presentes em cada Classificador Agregado (*Ensemble*)

Modelo	**Sigla**
CC-VM-11	RF-Under, GBT-Under, LR-Over
CC-VM-12	RF-Under, GBT-Under, MLP-A-Over
CC-VM-13	RF-Under, GBT-Under, MLP-A-m-Over
CC-VM-147	RF-Under, RF-M-Under, MLP-A -Under, LR-Over, MLP-A-Over
CC-VM-148	RF-Under, RF-M-Under, MLP-A -Under, LR-Over, MLP-A-m-Over
CC-VM-157	RF-Under, RF-M-Under, MLP-A-m-Under, LR-Over, MLP-A-Over
CC-VS-6	RF-Under, MLP-A
CC-VS-7	RF-Under, MLP-A-m
CC-VS-8	RF-Under, MLP-l
CC-VS-14	RF-m-Under, MLP-A
CC-VS-15	RF-m-Under, MLP-A-m
CC-VM-Over (PHUA, 2004)	DT-Over, NB-Over, MLP-A-Over
CC-VM-Under (PHUA, 2004)	DT-Under, NB-Under, MLP-A-Under

Fonte: Autoria Própria

Na Tabela 3, pode-se verificar que os classificadores baseados em LR, NN e GBT, sendo estes treinados com subamostragem da classe majoritária, apresentaram um desempenho médio de mais de 90% entre ambas as classes, com um acerto de 85% das fraudes, e entre 98% e 99% das transações genuínas. Os demais classificadores individuais apresentados mostraram um acerto de 89% médio entre ambas as classes, com um acerto das fraudes, variando entre 78% e 79%, e com um acerto de 99% de não-fraudes.

Já na Tabela 4, os melhores classificadores individuais mostraram um acerto médio que varia entre 83% e 86%, destacando o fato de que os classificadores treinados com subamostragem da classe majoritária apresentaram um acerto de mais de 90% das fraudes, e de cerca de 70% das não-fraudes, enquanto os classificadores treinados com sobreamostragem da classe minoritária apresentaram um acerto de fraude variando entre 72% e 88%, e algo entre 68% e 90% das transações genuínas. Aqui, há boa performance de classificadores baseados em DT, NN, LR e k-NN.

Primeiramente, percebe-se que não há um consenso em relação ao classificador individual que apresentou melhor desempenho, considerando que cada método apresentou melhor performance para um conjunto de dados específico. Com base nisso, percebe-se que, para ambos os conjuntos de dados, os *ensembles* propostos apresentaram desempenho no mínimo equivalente ao melhor classificador individual., e em alguns casos, superando o melhor método em até 6 pontos percentuais (p.p).

Já, comparando o algoritmo CC-VS com o estado da arte (CC-VM), nota-se que, para o primeiro conjunto de dados (UBL, 2017), ambos apresentaram resultados equivalentes, com métricas de desempenho idênticas. Já para o segundo conjunto de dados (EDA, 2020), conforme Tabela 4, o acerto das fraudes se manteve idêntico com 100% de acerto, havendo apenas um decréscimo no acerto de não-fraudes em 5 p.p. Porém, vale destacar que como um $\boldsymbol{F_N}$ apresenta um impacto muito maior do que um $\boldsymbol{F_P}$, este não é um fator crítico.

Por fim, a combinação de classificadores sendo estes treinados com diversas técnicas de amostragem superaram o estado da arte também na detecção de fraude. Com base no *ensemble* de referência descrito por (PHUA, 2004), para ambos os conjuntos de dados, a metodologia proposta superou o *ensemble* de (PHUA, 2004). Para o conjunto de dados de (ULB, 2017), o *ensemble* de (PHUA, 2004) não atingiu um acerto de 70% de ambas as classes em nenhuma de suas variações. Já no conjunto de dados de (EDA, 2020), pode-se observar na Tabela 4 que apenas o *ensemble* contendo classificadores treinados com subamostragem da classe majoritária (CC-VM-Under) (PHUA, 2004) obtiveram um acerto de mais de 70% em ambas as classes, e ainda assim, os *ensembles* propostos neste trabalho mostraram um acerto de 6 p.p a mais das fraudes, e de 3 p.p a 10 p.p a mais de acerto de não-fraudes.

CONSIDERAÇÕES FINAIS

Neste trabalho, propõe-se uma nova metodologia para enfrentar o problema de detecção de fraudes em transações com cartão de crédito, comtemplando a combinação de classificadores treinados com diferentes técnicas de balanceamento de classes, sendo agregadas pelo método de voto majoritário e pelo algoritmo proposto CC-VS. Foram realizadas simulações usando bancos de dados com transações reais de cartão de crédito, nas quais os métodos propostos foram comparados com o estado da arte. A partir dos resultados, observou-se que os *ensembles* considerando classificadores com metodologias distintas de amostragem têm desempenho equivalente ou melhor do que os modelos individuais de melhor desempenho. Também se observou que a estratégia de agregação CC-VS proposta apresentou desempenho equivalente ao método CC-VM, com menor complexidade computacional.

Acreditamos, portanto, que os algoritmos propostos são alternativas viáveis para apresentar esquemas de detecção de fraudes.

Novel Ensemble Algorithm for Fraud Detection in Credit Card Transactions

ABSTRACT

Fraudulent credit card transactions generate significant financial losses, encouraging the development of algorithms capable of detecting them. To that aim, this article proposes new machine learning techniques for solving binary detection problems with unbalanced classes, i.e., problems for which one of the classes (e.g., the occurrence of fraud) is much less frequent than the other. The proposed techniques combine classifiers of different formulations, trained with data sets obtained via different sampling schemes. The combination of classifiers is performed through majority voting or the new single voting method, which aims to increase the fraud detection rate. The proposed algorithms had their performances evaluated via numerical simulations using data from real financial transactions. The simulation results indicated that the new algorithms outperform state-of-the-art techniques in terms of detection metrics.

KEYWORDS: Fraud Detection. Machine Learning. Ensemble Learning. Artificial Intelligence

REFERÊNCIAS

AKOGLU, Haldun. User's guide to correlation coefficients. **Turkish journal of emergency medicine**, v. 18, n. 3, p. 91-93, 2018.

ALSHARKAWI, Adham et al. Poverty Classification Using Machine Learning: **The Case of Jordan. Sustainability,** v. 13, n. 3, p. 1412, 2021.

AWOYEMI, John O.; ADETUNMBI, Adebayo O.; OLUWADARE, Samuel A. Credit card fraud detection using machine learning techniques: A comparative analysis. In: **2017 international conference on computing networking and informatics (ICCNI)**. IEEE, 2017. p. 1-9.

BAESENS, Bart et al. robROSE: A robust approach for dealing with imbalanced data in fraud detection. **Statistical Methods & Applications**, v. 30, n. 3, p. 841-861, 2021

BAGGA, Siddhant et al. Credit card fraud detection using pipeling and ensemble learning. **Procedia Computer Science**, v. 173, p. 104-112, 2020.

BOCK, Sebastian; WEIÿ, Martin. A proof of local convergence for the Adam optimizer. **In: 2019 International Joint Conference on Neural Networks (IJCNN)**. IEEE, 2019. p. 1-8.

BORHANI, Mostafa. Multi-label Log-Loss function using L-BFGS for document categorization. **Engineering Applications of Artificial Intelligence**, v. 91, p. 103623, 2020

BRILLINGER, David R. Some data analyses using mutual information. **Brazilian Journal of Probability and Statistics**, p. 163-182, 2004.

COLETTA, Luiz Fernando Sommaggio. **Abordagens para combinar classificadores e agrupadores em problemas de classificação**, 2016. Tese de Doutorado. Universidade de São Paulo.

DA SILVA, Nadia FF; HRUSCHKA, Eduardo R.; HRUSCHKA JR, Estevam R. Tweet sentiment analysis with classifier ensembles. **Decision support systems**, v. 66, p. 170-179, 2014.

EDA. **Credit Card Fraud Detection**, 2020. Disponível em: <https://data.edincubator.eu/organization/yapi-kredi-teknoloji-anonim-sirketi>. Acesso em 02/03/2022.

HUANG, Chen et al. Learning deep representation for imbalanced classification. **In: Proceedings of the IEEE conference on computer vision and pattern recognition**, 2016. p. 5375-5384

ITOO, Fayaz et al. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. **International Journal of Information Technology**, v. 13, n. 4, p. 1503-1511, 2021.

LUO, Hanwu et al. Logistic regression and random forest for effective imbalanced classification. In: **2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)**. IEEE, 2019. p. 916-917.

MAES, Sam et al. Credit card fraud detection using Bayesian and neural networks. In: **Proceedings of the 1st international naiso congress on neuro fuzzy technologies**. 2002.

MENZE, Bjoern H. et al. On oblique random forests. **In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. Springer, BeLRin, Heidelberg, 2011. p. 453-469.

MOEPYA, Stephen O.; AKHOURY, Sharat S.; NELWAMONDO, Fulufhelo V. Applying cost-sensitive classification for financial fraud detection under high class-imbalance. In: **2014 IEEE international conference on data mining workshop**. IEEE, 2014. p. 183-192.

MONARD, Maria Carolina; BARANAUSKAS, José Augusto. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

PHUA, Clifton; ALAHAKOON, Damminda; LEE, Vincent. Minority report in fraud detection: classification of skewed data. **Acm sigkdd explorations newsletter**, v. 6, n. 1, p. 50-59, 2004.

SAVVY, Merchant. Global Payment Fraud Statistics, **Trends & Forecasts**, 2020

SHARDLOW, Matthew. An analysis of feature selection techniques. **The University of Manchester**, v. 1, n. 2016, p. 1-7, 2016.

SHEN, Aihua; TONG, Rencheng; DENG, Yaochen. Application of classification models on credit card fraud detection. In: **2007 International conference on service systems and service management**. IEEE, 2007. p. 1-4.

SOHONY, Ishan; PRATAP, Rameshwar; NAMBIAR, Ullas. Ensemble learning for credit card fraud detection. In: **Proceedings of the ACM India Joint International Conference on Data Science and Management of Data**. 2018. p. 289-294

THORNE, James et al. Fake news stance detection using stacked ensemble of classifiers. In: **Proceedings of the 2017 EMNLP workshop: natural language processing meets journalism**. 2017. p. 80-83.

ULB. **Credit Card Fraud Detection - Anonymized credit card transactions labeled as fraudulent or genuine**, 2017. Disponível em: <https://www.kaggle.com/mlg-ulb/creditcardfraud/discussion>. Acesso em 02/03/2022.

XUAN, Shiyang et al. Random forest for credit card fraud detection. In: **2018 IEEE 15th international conference on networking, sensing and control (ICNSC)**. IEEE, 2018. p. 1-6.

YADAV, Sanjay; SHUKLA, Sanyam. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In: **2016 IEEE 6th International conference on advanced computing (IACC)**. IEEE, 2016. p. 78-83.

YANG, Li; SHAMI, Abdallah. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, v. 415, p. 295-316, 2020.

Recebido: 20/06/2022
Aprovado: 06/03/2023
DOI: 10.3895/rts.v19n56.15628

Como citar:

DE SOUZA, D. M. H.; BORDIN JR. C. J. Novo algoritmo ensemble para detecção de fraude em transações de cartão de crédito. **Rev. Tecnol. Soc**., Curitiba, v. 19, n. 56, p.128-145, abr./jun., 2023. Disponível em: https://periodicos.utfpr.edu.br/rts/article/view/15628. Acesso em: XXX.

Correspondência: