Fraudulent Account Detection Using Hierarchical Classification

Andressa O. Souza[1], Mariana Mota[1], Helen C. S. C. Lima[2], Wellington Souza[3], Marcos Nicolau[3], Gladston Moreira[1], Eduardo J. S. Luz[1]

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

[2]Departamento de Computação e Sistemas – Universidade Federal de Ouro Preto (UFOP)
35931-088 – João Monlevade – MG – Brasil

[3]Departamento Antifraude – Gerencianet S.A
Ouro Preto – MG – Brasil.

{andressa.souza, mariana.regina}@aluno.ufop.edu.br

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

{helen, gladston, eduluz}@ufop.edu.br

Abstract. *Nowadays, we have been experiencing a paradigm change in the financial sector, with a high decrease in physical bank branches and an increase in online services. However, the easy opening of accounts provided by this change also increased fraud cases. This work presents the financial fraud detection problem under a new taxonomy and investigates hierarchical classification techniques for the task. The global hierarchical approach (CLUS-HMC), whereby the classifier considers the entire class hierarchy, resulted in better Recall values for fraudulent classes (33.31% for class E and 35.09% for class F), evidencing a promising research path.*

Resumo. *Hoje vivemos uma mudança de paradigma no setor financeiro, com forte redução das agências bancárias físicas e aumento de serviços online. Contudo, a facilidade de abertura de contas digitais propiciada por esta mudança de paradigma também tem levado a um aumento nos casos de fraude. Este trabalho apresenta o problema de detecção de fraude financeira sob uma nova taxonomia e, também, investiga técnicas de classificação hierárquica para a tarefa. A abordagem hierárquica global (CLUS-HMC), em que toda a hierarquia de classes é considerada pelo classificador, resultou em melhores valores de Recall para as classes fraudulentas (33.31% para classe E e 35.09% para classe F), indicando um caminho de pesquisa promissor.*

1. Introdução

Hoje vivemos uma mudança de paradigma no setor financeiro, com forte redução das agências bancárias físicas. As instituições financeiras estão se adaptando para funcionar de forma quase que exclusivamente *online*. Contudo, a facilidade de abertura de contas digitais também tem levado a um aumento nos casos de criação de contas utilizadas para fraude. Dessa forma, a identificação automática dessas contas é essencial para evitar maiores prejuízos. No Brasil, o art. 14 da Lei 8.078 de 1990 [1] garante que as instituições

[1]Lei nº 8.078, de 11 de setembro de 1990. Diário Oficial da República Federativa do Brasil.

financeiras devem se responsabilizar pela reparação dos danos sofridos pelo consumidor, independente da existência de culpa.

Abordagens utilizando aprendizado de máquina, principalmente aprendizagem supervisionada, vêm se tornando aliadas na prevenção e identificação de diferentes tipos de fraudes financeiras: transações fraudulentas de cartões de crédito, lavagem de dinheiro, fraude em seguros, fraude contábil fraude bancária, entre outros [Li et al. 2021, Niu et al. 2019, Roy et al. 2018, Dreżewski et al. 2015]. Os trabalhos que abordam fraudes em cartões de crédito são os mais populares, uma hipótese para isso é a existência de bases de dados públicas. Até onde sabemos, as bases de dados públicas disponíveis organizam os dados em duas categorias (fraude e não-fraude). Todavia, a depender da base de dados financeira, uma taxonomia pode emergir. De acordo com [Zheng and Zhao 2020], abordagens hierárquicas, em que a taxonomia do problema é considerada, favorecem a classificação em problemas fortemente desbalanceados. Motivados pelos achados reportados em [Zheng and Zhao 2020], este trabalho se propõem a investigar o problema de detecção de contas financeiras fraudulentas por meio de duas abordagens hierárquicas (classificação hierárquica local por nó pai e classificação hierárquica global). Técnicas de superamostragem também são investigadas no contexto de classificação hierárquica, visando estudar o impacto do desbalanceamento entre classes. Ainda, propõem-se aqui um novo conjunto de atributos baseados na Lei de Benford [Benford 1938].

Os experimentos foram efetuados em uma base de dados real, coletada junto a empresa brasileira do setor financeiro Gerencianet S.A., com contas categorizadas por analistas financeiros em seis classes distintas e fortemente desbalanceadas (A, B, C, D, E e F), além das superclasses de fraude e não-fraude. Resultados mostram que a abordagem de classificação hierárquica global proposta obteve os melhores valores de *Recall* para as classes fraudulentas (33.31% para classe E e 35.09% para a classe F). Todavia, o modelo baseado em classificação hierárquica local por nó pai, também proposto aqui, se mostrou mais eficaz para classes não fraudulentas e também alcançou melhor *F-score* para a classe F (39.21%). Os resultados experimentais indicam que o problema se torna mais desafiador quando se tem uma taxonomia envolvida, e que o uso de métodos baseados em classificação hierárquica pode ser uma alternativa viável para a modelagem do problema. Os atributos derivados da lei de Benford se mostraram importantes para a tomada de decisão dos classificadores.

2. Problema de Fraude Financeira e a Classificação Hierárquica

Entre os diversos tipos de fraude financeira abordados na literatura, estão: fraude bancária, fraude de seguros, fraude em transações de cartões de crédito, lavagem de dinheiro, entre outras, e as técnicas de classificação tradicionais são vastamente citadas como possíveis soluções [Devi et al. 2019, Xuan et al. 2018, Li et al. 2021]. Em [Li et al. 2021], três algoritmos meta-heurísticos são utilizados para ajustar os parâmetros do classificador *Support-Vector Machines* (SVM) para a identificação de transações fraudulentas de cartões de crédito: *Cuckoo Search*(CS), *Genetic Algorithms* (GA) e *Particle Swarm Optimization* (PSO). Os dados utilizados foram coletados no departamento de aplicação da lei na China, são desbalanceados e uma acurácia de 91,56% foi reportada.

A utilização de um algoritmo *Random Forest* ponderado sensível ao custo para a

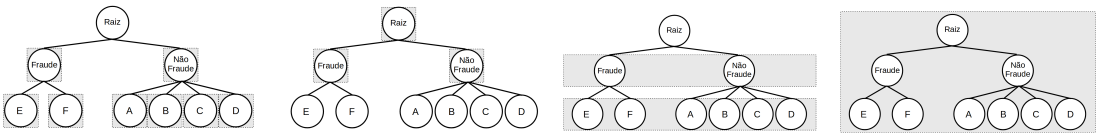
resolução do problema de fraudes em cartões de crédito é proposta em [Devi et al. 2019]. Para a realização do trabalho, foram utilizadas duas bases de dados públicas, com dados de aprovação de crédito: *German Credit Data* e *Australian Credit Approval*. A proposta foi comparada ao *Random Forest* padrão e ao *RF-based Imbalanced Data Cleaning and Classification* (RF-IDCC). O método proposto obteve um melhor resultado das métricas de *F-score*, *G-mean* e AUC em ambas as bases, sendo nessa ordem, 78.217, 80.230 e 0.6921, para a base *German Credit Data* e 76.815, 82.298 e 0.0.778, para a base *Australian Credit Approval*.

Em [Niu et al. 2019], utilizou-se seis técnicas de classificação para resolver o problema de fraudes em cartões de crédito: *Logistic Regression* (LR), *K-Nearest Neighbor (KNN)*, *Support-Vector Machines* (SVM), *Decision Tree* (DT), *Random Forest* (RF), *XGBoost* (XGB). A base de dados utilizada foi a *Credit Card Fraud Detection*, que é pública, desbalanceada e possui transações de cartão de crédito feitas em setembro de 2013 por titulares de cartões europeus. Dentro das abordagens supervisionadas, XGB e RF obtiveram o melhor desempenho com AUROC = 0,989 e AUROC = 0,988, respectivamente. Embora outras técnicas mais recentes, como as baseadas em redes profundas [Roy et al. 2018], já foram investigadas para o problema, as técnicas mais promissoras, ainda hoje, são baseadas em SVM, XGB e RF. É sabido que o desbalanceamento influencia fortemente este tipo de tarefa. Em [Shenvi et al. 2019], investigou-se técnicas para compensar o desbalanceamento e os experimentos mostraram ganhos, em especial, com técnicas de subamostragem.

Os trabalhos citados nesta seção abordam o problema de fraude financeira como um problema típico de classificação binária (classificação plana). Ainda, a maioria dos trabalhos publicados na literatura se propõem a investigar transações fraudulentas. Diferentemente, este trabalho apresenta o problema orientado a classificação de contas fraudulentas e investiga classificação hierárquica.

2.1. Classificação hierárquica

Nos problemas de classificação hierárquica, as classes a serem preditas são organizadas em uma hierarquia e representadas pelos nós, assim como em uma árvore ou um Grafo Acíclico Dirigido (DAG) [Freitas and de Carvalho 2007]. Entre os tipos de classificação hierárquica está a abordagem classificador local, que envolve os classificadores: local por nó, local por nó pai e local por nível; e a abordagem do classificador global. Além da abordagem de classificação tradicional, também chamada de plana, neste trabalho são exploradas as abordagens hierárquicas do classificador local por nó pai e do classificador global. A Figura 1 ilustra cada um dos tipos aplicado ao problema de fraude financeira. Na Figura 1, temos também os tipos de contas distribuídos em classes estruturadas em uma hierarquia baseada em árvore. Em um primeiro nível, as contas são categorizadas em duas classes - fraude e não-fraude - das quais chamamos de superclasses. Em um seguindo nível, as contas não fraudulentas são divididas em quatro classes (A, B, C e D), enquanto as contas fraudulentas são divididas em duas classes: E e F. Para lidar com a abordagem de classificação hierárquica, é essencial que a hierarquia seja considerada usando uma perspectiva de informação local. Em [Freitas and de Carvalho 2007], são citados três tipos de classificadores hierárquicos locais, considerando uma classe estruturada em hierarquia de árvore de rótulo único e predição obrigatória do nó folha: classificador local por nó, classificador local por nó pai e classificador local por nível.



(a) Local por nó. (b) Local por nó pai. (c) Local por nível. (d) Global.

Figura 1. Diferentes tipos de classificação hierárquica.

O classificador local por abordagem de nó é o mais utilizado na literatura e consiste em treinar um classificador binário para cada nó da hierarquia de classes (exceto o nó raiz) [Freitas and de Carvalho 2007]. Durante a fase de teste, a saída de cada classificador binário será uma previsão indicando se um dado exemplo de teste pertence ou não à classe prevista do classificador. Na Figura 1 (a) temos um exemplo do classificador local por abordagem de nó, onde os retângulos correspondem a classificadores binários.

No classificador local por nó pai, para cada nó pai na hierarquia de classes, um classificador multiclasse é treinado para distinguir entre seus nós filhos [Freitas and de Carvalho 2007]. Na Figura 1 (b), considere a abordagem de previsão de classe de cima para baixo e suponha que o classificador de primeiro nível atribui a amostra à classe Fraude. O classificador de segundo nível, que foi treinado apenas com os filhos dessa classe, poderá então fazer a sua atribuição de classe apenas como classe E ou F, evitando assim o problema de fazer previsões inconsistentes e respeitando a restrições naturais de pertencimento à classe.

No classificador local por nível, um classificador multiclasse é treinado para cada nível da hierarquia de classes [Freitas and de Carvalho 2007]. Na Figura 1 (c), considere que os dois classificadores são treinados, um para cada nível de classe, para prever uma ou mais classes em seu nível de classe. A principal desvantagem dessa abordagem é a possível inconsistência de associação de classe, uma vez que ao treinar classificadores diferentes para cada nível da hierarquia, é possível ter saídas como Fraude no primeiro nível e classe A no segundo nível. Caso essa abordagem seja utilizada, é necessário um pós-processamento para corrigir a inconsistência de previsão [Freitas and de Carvalho 2007].

A abordagem de classificação hierárquica global ou *Big-Bang* consiste em um único modelo de classificação que considera toda a hierarquia de classes, de forma que a previsão pode ocorrer em qualquer nível de hierarquia [Borges et al. 2013]. A principal vantagem desta abordagem é que não há necessidade de treinar um grande número de classificadores e lidar com a inconsistência na previsão de classes. Sua principal desvantagem é o aumento da complexidade no desenvolvimento do modelo do classificador. Na Figura 1 (d), temos um exemplo de classificador global.

3. Metodologia

3.1. Definição do conjunto de dados

Os dados foram fornecidos pela instituição financeira brasileira Gerencianet S.A. e disponibilizados em forma de um banco de dados relacional. Movimentações financeiras de clientes foram registradas compreendendo o período de agosto de 2019 à outubro de 2021. Todas as informações foram devidamente anonimizadas e os respectivos valores normalizados a fim de se ocultar os valores reais.

Visando a criação de um conjunto de dados tabular, mais apropriado para aprendizagem supervisionada, foram extraídos atributos com base nas tabelas do banco de dados relacional fornecido. A Tabela 1 apresenta a distribuição, por categoria, dos atributos propostos. No total, foram gerados 185 atributos, resultando em um conjunto de dados tabular com dimensões 45.209 x 185, composto por atributos numéricos, booleanos e categóricos. Os atributos numéricos foram normalizados entre 0 e 1. Os categóricos foram convertidos em valores inteiros entre 0 e $n-1$, em que n representa o número de possíveis valores de atributo, técnica conhecida como *label encoder*. Os valores ausentes foram preenchidos com o valor fixo -1. Ressaltamos que este trabalho é orientado à classificação de contas bancárias e, que uma conta bancária é categorizada por múltiplas transações efetuadas por uma pessoa física (ou jurídica) em um determinado período de tempo.

A categoria de atributos chamada Lei de Benford foi proposta aqui com o intuito de se investigar atributos derivados na Lei proposta em [Benford 1938], que refere-se a distribuição de dígitos. Um conjunto de números satisfaz a lei de Benford se o primeiro dígito $d(d \in 1, ..., 9)$ ocorre com a seguinte probabilidade: $P(d) = \log_{10} \frac{(d+1)}{(d)}$. Foram gerados os seguintes atributos:

- Um atributo numérico, para cada tipo de transação, que representa a diferença absoluta entre a proporção esperada pela Lei de Benford e a proporção obtida, para cada algarismo.
- Um atributo numérico, para cada tipo de transação, que representa a média das diferenças absolutas entre a proporção esperada pela Lei de Benford e a proporção obtida, para cada algarismo da transação financeira.
- Um atributo booleano, para cada tipo de transação, indicando se existem algarismos que não apareceram no primeiro digito de todas movimentações de um tipo de transação financeira, isto é, que tiveram proporção nula e não atenderam a distribuição de Benford. Definiu-se um limiar de 3 algarismos, de forma que se mais de 3 algarismos não apareceram no primeiro digito da distribuição da transação, o atributo recebeu o valor *False*. Do contrário, recebeu *True*.
- Um atributo booleano, para cada tipo de transação, indicando se existe algum algarismo teve proporção extrapolada em relação ao esperado e não atendeu a distribuição de Benford. Se algum algarismo extrapolou em 90% ou mais a proporção esperada no primeiro digito, o atributo recebeu o valor *False*. Do contrário, recebeu *True*.
- Um atributo booleano, para cada tipo de transação, indicando se a distribuição obtida teve algarismos com proporção nula ou com proporção extrapolada em relação ao esperado.

Na Tabela 2 temos a distribuição das transações utilizadas para a geração dos atributos Benford. Foi considerado todo o período dos dados coletados da base. Ressaltamos que uma conta é composta por diversas transações e que este trabalho visa classificação de contas.

Na Figura 2, pode-se verificar a distribuição da Lei de Benford obtida e esperada para um cliente legítimo e para um cliente classificado como fraude. A linha vermelha representa o valor esperado pela Lei de Benford para a distribuição de cada dígito e as colunas (verde) representam o valor obtido. Na Figura 2 (b), percebe-se a violação da Lei de Benford.

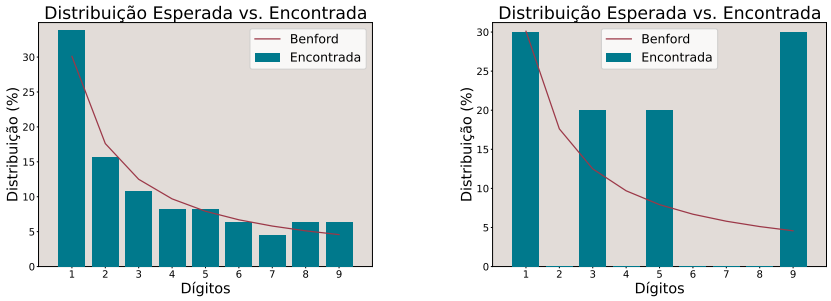
Tabela 1. Atributos por categoria

Categoria	Descrição
Conta (2 atributos)	Os atributos identificam se a conta pertence a uma pessoa jurídica ou física e o total de usuários da conta.
Usuário (6 atributos)	Os atributos identificam informações sobre o usuário da conta: data de aniversário, profissão informada, se a pessoa é exposta publicamente, se é estrangeira, o DDD do celular e o tempo gasto em ligação de atendimento da instituição financeira.
E-mail (6 atributos)	Os atributos identificam informações sobre o e-mail utilizado no cadastro: domínio, comprimento do e-mail, se o primeiro nome está presente no e-mail, se o e-mail inicia com um número, se o e-mail termina com um número e quantidade de números no e-mail.
TED (9 atributos)	Os atributos identificam: a quantidade de transações TED com a conta como origem, o valor total das transações TED com a conta como origem, a quantidade de transações TED com a conta como destino, o valor total das transações TED com a conta como destino, o horário mais cedo registrado por uma transação TED feita pela conta, o horário médio das transações TED feitas pela conta, o horário mais tarde registrado por uma transação TED feita pela conta e valor médio mensal recebido pela conta por TED.
Contas pagas (2 atributos)	Os atributos apresentam a quantidade de contas pagas e o valor total.
Recarga de celular (1 atributo)	O atributo indica se fez alguma recarga no celular.
Pix (12 atributos)	Os atributos identificam: a quantidade de transações Pix com a conta como origem, o valor total das transações Pix com a conta como origem, a quantidade de transações Pix com a conta como destino, o valor total das transações Pix com a conta como destino, o horário mais cedo registrado por uma transação Pix feita pela conta, o horário médio das transações Pix feitas pela conta, o horário mais tarde registrado por uma transação Pix feita pela conta e valor médio mensal enviado pela conta por Pix, o horário mais cedo registrado por uma transação Pix recebida pela conta, o horário médio das transações Pix recebidas pela conta, o horário mais tarde registrado pelo recebimento de uma transação Pix e valor médio mensal recebido pela conta por Pix.
Cobranças (13 atributos)	Os atributos identificam: o total de cobranças emitidas pela API da empresa, o total de cobranças feitas por carnê, o total de cobranças feitas por boleto, o total de cobranças feitas por cartão, o total de cobranças pagas, o total de cobranças não pagas, a quantidade de cobranças emitidas pela conta, a soma dos valores das cobranças emitidas pela conta, o total de cobranças com desconto, o horário mais cedo registrado de uma emissão de cobrança pela conta, o horário médio das emissões de cobrança pela conta, o horário mais tarde registrado de emissão de cobrança pela conta e valor médio mensal de emissões de cobrança pela conta.
Lei de Benford (80 atributos)	Os atributos identificam informações da Lei de Benford para cada tipo de transação financeira realizada e recebida (Pix, TED, cobranças).
Dias entre transações (54 atributos)	Os atributos identificam os valores mínimo, máximo e médio de dias entre as transações financeiras realizadas e recebidas (Pix, TED, cobranças), durante os três últimos meses que antecedem a última transação.

As contas do conjunto de dados foram rotuladas por especialistas do setor financeiro da instituição financeira parceira e estão distribuídas da seguinte forma: a classe A possui 37 instâncias; a classe B possui 323 instâncias; a classe C possui 3.397 instâncias; a classe

Tabela 2. Total de transações financeiras por classe.

	Cobranças	Pix Enviado	Pix Recebido	TED Enviado	TED Recebido
Classe A	32	24	18	32	10
Classe B	318	244	163	317	61
Classe C	3.240	2.331	1.432	2.990	682
Classe D	25.483	19.818	11.876	17.431	3.641
Classe E	2.109	934	488	1.260	133
Classe F	625	303	177	397	49



(a) Cliente classificado como legítimo.

(b) Cliente classificado como fraude.

Figura 2. Distribuições de proporção dos dígitos de Benford para os valores de uma transação de saída de clientes reais.

D possui 36.993 instâncias; a classe E possui 3.646 instâncias; e a classe F possui 813 instâncias. Sendo assim, o conjunto de dados é fortemente desbalanceado.

3.2. Balanceamento dos dados

Para contornar o problema de desbalanceamento entre as classes, os algoritmos de sobreamostragem foram avaliados: *Synthetic Minority Over-sampling TEchnique* (SMOTE)[Han et al. 2005] , *Borderline Synthetic Minority Over-sampling TEchnique* (Borderline SMOTE) [Han et al. 2005], *Conditional Tabular Generative Adversarial Network* (CTGAN) [Xu et al. 2019] e *Tabular Variational Auto-Encoder* (TVAE) [Xu et al. 2019].

3.3. Classificação

Neste trabalho, optou-se por usar o tradicional classificador *Random Forest* [Breiman 2001] para a abordagem de classificação plana e classificação hierárquica por nó pai. Para a abordagem de classificação hierárquica global, foi utilizado o algoritmo CLUS-HMC [Vens et al. 2008], implementado no *software* CLUS, também baseado em florestas de árvores aleatórias. O classificador *Random Forest* foi escolhido para análise por ser um dos mais populares para o problema de detecção de fraude financeira [Devi et al. 2019, Xuan et al. 2018]. Neste trabalho, o impacto do classificador é minimizado e o foco do trabalho é centrado na influência da taxonomia, técnicas para compensar o desbalanceamento entre classes e atributos derivados da lei de Benford.

3.3.1. CLUS-HMC

O CLUS-HMC [Vens et al. 2008] é baseado em estrutura de árvore de agrupamento preditivo (*predictive clustering tree* ou PCT). Esta estrutura visualiza uma árvore de decisão como uma hierarquia de *clusters*, de forma que o nó superior corresponde a um *cluster* contendo todos os dados, que são recursivamente particionados em *clusters* menores. Dessa forma, os PCTs são construídos de modo que cada divisão reduz ao máximo a variação.

O algoritmo recebe como entrada um conjunto de instâncias de treinamento. O *loop* principal procura o melhor teste de valor de atributo aceitável que pode ser colocado em um nó. Se tal teste pode ser encontrado, então o algoritmo cria um novo nó rotulado t^* e chama a si mesmo recursivamente para construir uma subárvore para cada subconjunto *(cluster)* na partição P^* induzida por t^* nas instâncias de treinamento. O melhor teste é o que reduz ao máximo a variância induzida no treinamento. Maximizar a redução de variância maximizará a homogeneidade do *cluster* e melhorará o desempenho preditivo. Se nenhum teste reduzir significativamente a variância, então o algoritmo cria uma folha e a rotula com um caso representativo das instâncias dadas. A distância utilizada pelo CLUS-HMC é a distância euclidiana ponderada.

3.4. Avaliação dos resultados

Para a avaliação dos modelos, foram utilizadas as métricas mais comuns para problemas de detecção de fraude: *Recall*, Precisão e *F-score*. O *Recall* (R) é dado por $R = \frac{V_P}{(V_P+F_N)}$, sendo V_P os verdadeiros positivos e F_N os falsos negativos. A Precisão (P) é dada por $P = \frac{V_P}{(V_P+F_P)}$, onde F_P representa os falsos positivos . O *F-score* (F) é dado por $2 \times \frac{(P \times R)}{(P+R)}$.

4. Experimentos

Os experimentos descritos aqui foram implementados por meio de bibliotecas em *Python* [2]. Para o ajuste dos hiperparâmetros dos classificadores *Random Forest* foi utilizado o método *GridSearchCV*, com o objetivo de se maximzar o *F-score* da classe F. Os modelos foram construídos utilizando os hiperparâmetros encontrados e foram treinados com validação cruzada de 10 *k-folds*, utilizando o conjunto de dados com e sem sobreamostragem. Todos os 185 atributos do conjunto de dados foram considerados durante os experimentos.

Na abordagem de classificação plana, um classificador *Random Forest* foi construído e as técnicas de sobreamostragem foram utilizados de forma a igualar a quantidade de instâncias entre a classe majoritária D e as demais classes. Para a abordagem do classificador local por nó pai, três classificadores planos *Random Forest* foram construídos na fase de treinamento: um para distinguir as contas fraudulentas e as contas não fraudulentas; outro para distinguir, entre as contas fraudulentas, quais são pertencentes a classe E e quais são pertencentes a classe F; e outro para distinguir, entre as contas não fraudulentas, quais são pertencentes as classes A, B, C e D. Para o classificador do primeiro nível, os métodos de balanceamento foram utilizados de forma a igualar o número de instâncias entre as superclasses Fraude e Não Fraude. Para os classificadores do nível abaixo, os métodos de balanceamento foram utilizados de forma a igualar o número de instâncias entre as classes E e F para o nó fraude e para o classificador associado ao nó não-fraude, os métodos de balanceamento foram utilizados de forma a igualar o número de instâncias entre as

[2] *NumPy*, *Keras*, *Pytorch*, *Pandas*, *Sklearn* e *TensorFlow*

classes A, B, C e D. Na fase de teste, utilizou-se a abordagem de cima para baixo, ou seja, dada uma instância de teste a ser classificada, o sistema primeiro realiza a previsão do classificador associado à raiz nó (treinado para distinguir entre contas fraudulentas e contas não fraudulentas). Assim, se a classe fraude é atribuída à instância de teste na previsão do primeiro nível, o sistema realiza uma segunda previsão do classificador associado ao nó de classe fraude para atribuir a ele um de seus nós filhos (E ou F). Caso contrário, se classe não-fraude for atribuída à instância de teste na primeira predição, o sistema realizará uma segunda predição do classificador associado ao nó de classe não fraude para atribuir a ele um de seus nós filhos (A, B, C ou D). A abordagem de cima para baixo evita o problema de previsões inconsistentes, respeitando as restrições da hierarquia de classes.

Na abordagem do classificador global, apenas um classificador CLUS-HMC foi construído, considerando toda a hierarquia das classes. Nessa abordagem, os métodos de balanceamento de dados foram utilizados de forma a igualar o número de instâncias entre as classes não fraudulentas (A, B, C e D) das classes fraudulentas (E e F), de forma que o número de instâncias artificiais gerados para as classes E e F, foram definidos por $na_E = \frac{nA+nB+nC+nD}{2} - nE$ e $na_F = \frac{nA+nB+nC+nD}{2} - nF$, respectivamente, em que nX é o número de instâncias da classe X.

Para a utilização dos métodos de sobreamostragem foram considerados os seguintes hiperparâmetros: SMOTE (*sampling_strategy ='auto'*, *random_state=None*, *k_neighbors=5*, *n_jobs=None*); Borderline SMOTE (*m_neighbors=10*, *kind='borderline-1'*); CTGAN (*generator_dim= (258,128)*, *discriminator_dim = (128, 256)*, *cuda = False*, *verbose = True*) e TVAE (*compress_dims= (128,64)*, *decompress_dims = (64, 128)*, *cuda = False*).

5. Resultados e Discussão

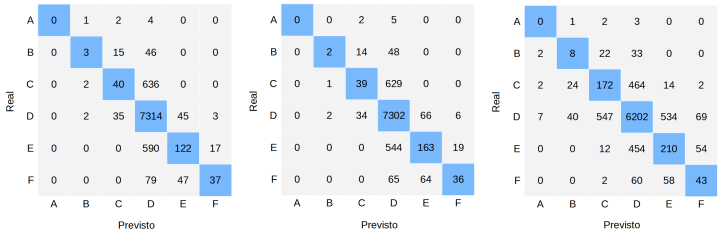
A Tabela 3 apresenta as médias e os desvios-padrão (em %) de *F-score* obtidos das abordagens de classificação plana (CP), classificação hierárquica local por nó pai (CHL) e classificação hierárquica global (CHG), após a validação cruzada de 10 *k-folds*. Os melhores valores obtidos por classe estão destacados em negrito.

Foram calculados também as médias e os desvios-padrão (em %) de *Recall* e Precisão obtidos das abordagens de classificação plana (CP), classificação hierárquica local por nó pai (CHL) e classificação hierárquica global (CHG). Para o *Recall*, a abordagem hierárquica global forneceu os melhores resultados para as classes fraudulentas, sendo $33.31\% \pm 01.75\%$ para classe E, associado ao Borderline SMOTE e $35.09\% \pm 04.20\%$ para classe F, associado ao SMOTE. Para a abordagem de classificação plana, os melhores valores de *Recall* obtidos pelas classes fraudulentas foram de $31.34\% \pm 01.43\%$ para a classe E e $29.94\% \pm 03.64\%$ para a classe F, ambos obtidos em associação com o método SMOTE. Para a abordagem de classificação hierárquica por nó pai, os melhores valores de *Recall* foram de $29.98\% \pm 01.32\%$, para a classe E e $32.58\% \pm 04.15\%$ para a classe F, ambos associados ao SMOTE. Os melhores valores de Precisão foram obtidos por meio da abordagem de classificação plana associada ao CTGAN. Para a abordagem de classificação hierárquica local, o melhor valor obtido para a classe E foi de $59.38\% \pm 03.91\%$ e para a classe F foi de $70.70\% \pm 06.09\%$, ambos associados ao TVAE. Já para a abordagem de classificação hierárquica local, os melhores valores foram obtidos com a associação do CTGAN, sendo $26.27\% \pm 01.39\%$ para a classe E e $27.83\% \pm 03.30\%$ para a classe F.

Tabela 3. Média em (%) dos resultados de *F-score* (F) para as abordagens de classificação plana (CP), classificação hierárquica local por nó pai (CHL) e classificação hierárquica global (CHG) após validação cruzada de 10 k-folds

	A	B	C	D	E	F
CP	00.00 ± 00.00	09.07 ± 04.61	10.46 ± 01.13	91.04 ± 00.09	25.91 ± 01.29	33.32 ± 03.52
CP + SMOTE	00.00 ± 00.00	14.81 ± 05.93	34.93 ± 01.27	88.61 ± 00.18	**34.67 ± 01.48**	38.81 ± 03.60
CP + Borderline SMOTE	00.00 ± 00.00	08.25 ± 02.60	34.44 ± 01.57	88.97 ± 00.22	34.12 ± 01.53	33.08 ± 03.64
CP + CTGAN	00.00 ± 00.00	11.68 ± 05.80	14.25 ± 01.99	90.56 ± 00.08	13.65 ± 01.39	20.79 ± 03.52
CP + TVAE	00.00 ± 00.00	00.00 ± 00.00	16.78 ± 02.07	90.38 ± 00.11	10.03 ± 01.52	14.23 ± 04.07
CHL	00.00 ± 00.00	00.00 ± 00.00	10.16 ± 01.44	**91.25 ± 00.15**	31.91 ± 01.63	31.51 ± 05.11
CHL + SMOTE	00.00 ± 00.00	10.87 ± 05.04	**36.84 ± 00.93**	88.18 ± 00.20	34.17 ± 01.29	38.55 ± 03.06
CHL + Borderline SMOTE	00.00 ± 00.00	00.00 ± 00.00	36.40 ± 01.05	88.92 ± 00.23	34.50 ± 01.19	**39.21 ± 02.68**
CHL + CTGAN	00.00 ± 00.00	11.58 ± 04.15	14.16 ± 01.25	90.89 ± 00.17	19.94 ± 01.54	22.52 ± 03.29
CHL + TVAE	00.00 ± 00.00	00.00 ± 00.00	16.17 ± 02.13	90.84 ± 00.19	19.51 ± 01.58	23.65 ± 03.14
CHG	00.00 ± 00.00	11.61 ± 03.64	24.00 ± 01.21	84.87 ± 00.32	27.11 ± 01.40	25.87 ± 03.91
CHG + SMOTE	00.00 ± 00.00	14.33 ± 02.99	23.24 ± 01.31	83.45 ± 00.34	27.28 ± 01.29	28.12 ± 03.00
CHG + Borderline SMOTE	00.00 ± 00.00	12.07 ± 03.89	23.93 ± 01.28	83.76 ± 00.18	27.48 ± 01.14	26.50 ± 02.06
CHG + CTGAN	00.00 ± 00.00	14.05 ± 02.59	23.23 ± 01.23	84.98 ± 00.42	27.59 ± 01.38	28.41 ± 03.08
CHG + TVAE	00.00 ± 00.00	**15.89 ± 04.65**	22.98 ± 01.30	84.91 ± 00.37	27.45 ± 01.72	24.54 ± 03.49

Na Figura 3 temos as matrizes de confusão com os valores médios obtidos das diferentes abordagens de classificação após a validação cruzada de 10 *k-folds*.



(a) CP. (b) CHL. (c) CHG.

Figura 3. Matrizes de confusão das diferentes abordagens de classificação. Cada instância classificada corresponde a uma conta na instituição financeira. Média dos resultados das partições de teste.

Para análise das instâncias fraudulentas que foram erroneamente classificadas, foi utilizado o método SP-LIME [Ribeiro et al. 2016], que permite selecionar um conjunto de instâncias representativas para abordar a confiança no modelo, por meio de otimização submodular. Para o problema abordado aqui, o SP-LIME selecionou 30 instâncias representativas das classes de fraude (rotulada como E ou F). Na Figura 4 temos a explicação de uma instância da classe F classificada como pertencente a classe E, nela o atributo 1 relacionado a lei de Benford é uma evidência contra a classe E.

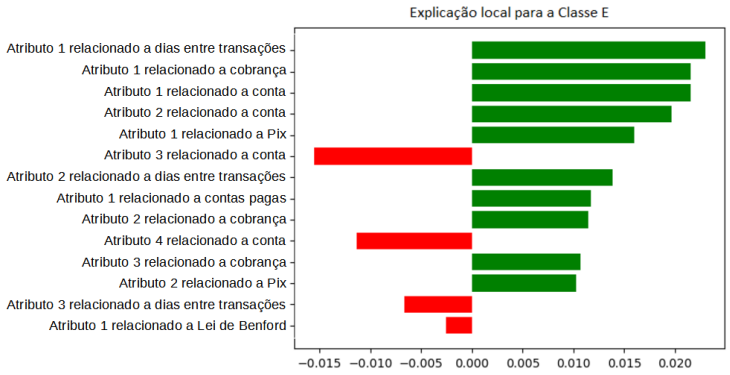


Figura 4. Explicação de uma instância da classe F classificada como pertencente a classe E.

Pelas matrizes da Figura 3, percebe-se grande confusão entre as classes E e F com a classe majoritária. As instâncias representativas das classes E e F classificadas incorretamente como D foram explicadas por meio de atributos das categorias: *Dias entre transações, Cobranças, Pix, TED e Conta*. Já as instâncias da classe E incorretamente classificadas como F e as instâncias da classe F incorretamente classificadas como E, foram explicadas pelos atributos das categorias: *Dias entre transações, Cobranças, Pix, TED, Conta, Contas pagas e Lei de Benford*.

6. Conclusão

Neste artigo abordou-se o problema de detecção de fraudes financeiras em uma taxonomia hierárquica. O procedimento experimental foi realizado em uma base de dados real, contendo 45.209 contas de usuários e 96.618 transações financeiras. Três abordagens de classificação foram propostas e comparadas: classificação plana, classificação hierárquica local por nó pai e hierárquica global. Quatro técnicas de sobreamostragem foram explorados para reduzir o desbalanceamento das classes: SMOTE, Borderline SMOTE, CTGAN e TVAE. A abordagem hierárquica global alcançou os melhores resultados de *Recall* para as classes fraudulentas (33.31% para classe E e 35.09% para classe F). Já a abordagem de classificação hierárquica local por nó pai obteve o melhor *F-score* para a classe F (sendo de 39.21%) associada ao emprego de Borderline SMOTE. Os resultados obtidos indicam que a abordagem hierárquica pode ser promissora para o problema e que atributos derivados da Lei de Benford podem auxiliar na detecção de instâncias fraudulentas. Como extensão do estudo, pretende-se investigar o problema de classificação de contas fraudulentas sob a perspectiva hierárquica com modelos de redes de convolução, utilizando-se outra forma de representação dos dados.

Agradecimentos

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

Referências

Benford, F. (1938). The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572.

Borges, H. B., Silla, C. N., and Nievola, J. C. (2013). An evaluation of global-model hierarchical classification algorithms for hierarchical classification problems with single path of labels. *Computers & Mathematics with Applications*, 66(10):1991–2002. ICNC-FSKD 2012.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Devi, D., Biswas, S. K., and Purkayastha, B. (2019). A cost-sensitive weighted random forest technique for credit card fraud detection. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6.

Dreżewski, R., Sepielak, J., and Filipkowski, W. (2015). The application of social network analysis algorithms in a system supporting money laundering detection. *Information Sciences*, 295:18–32.

Freitas, A. and de Carvalho, A. (2007). A tutorial on hierarchical classification with applications in bioinformatics. *Research and Trends in Data Mining Technologies and Applications*.

Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer.

Li, C., Ding, N., Zhai, Y., and Dong, H. (2021). Comparative study on credit card fraud detection based on different support vector machines. *Intelligent Data Analysis*, 25:105–119.

Niu, X., Wang, L., and Yang, X. (2019). A comparison study of credit card fraud detection: Supervised versus unsupervised.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Roy, A., Sun, J., Mahoney, R., Alonzi, L., Adams, S., and Beling, P. (2018). Deep learning detecting fraud in credit card transactions. In *2018 Systems and Information Engineering Design Symposium (SIEDS)*, pages 129–134.

Shenvi, P., Samant, N., Kumar, S., and Kulkarni, V. (2019). Credit Card Fraud Detection using Deep Learning. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–5.

Vens, C., Struyf, J., Schietgat, L., Džeroski, S., and Blockeel, H. (2008). Decision trees for hierarchical multi-label classification. *Machine Learning*, 73:185–214.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. In *NeurIPS*.

Xuan, S., Liu, G., Li, Z., Zheng, L., Wang, S., and Jiang, C. (2018). Random forest for credit card fraud detection. In *2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC)*, pages 1–6.

Zheng, W. and Zhao, H. (2020). Cost-sensitive hierarchical classification for imbalance classes. *Applied Intelligence*, 50(8):2328–2338.