Sequencing Sampling Algorithms to Boost Performance of Classifiers on Imbalanced Data Sets

Gian F. S. Barbosa[1], **Péricles B. C. de Miranda**[1], **Rafael Ferreira Mello**[1], **Ricardo M. A. Silva**[2]

[1]Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Recife – PE – Brazil

[2]Centro de Informática – Universidade Federal de Pernambuco
Recife – PE – Brazil

{pericles.miranda}@ufrpe.br

Abstract. *Sampling techniques play an essential role in aiding classifiers that learn from imbalanced data sets, since these techniques return a more balanced version of the imbalanced data set. Under sampling reduces the number of examples of the majority class in order to balance the number of examples of each class, which may delete relevant examples. Over sampling generally use all the examples available in the minority class to synthesize new instances, which may include noisy examples or outliers. Thus, depending on the nature of the data, using an algorithm of one type or another may negatively impact classifier's performance. This paper proposes the 'Sequencing of sampling algorithms', which allows the composition of different sampling algorithms in a pipeline for data balancing. The results showed that the proposal was able to obtain statistically higher average accuracy and F_1 score when compared to traditional and hybrid sampling approaches in most of the selected imbalanced problems.*

Resumo. *As técnicas de amostragem desempenham um papel essencial na ajuda aos classificadores que aprendem com conjuntos de dados desbalanceados, uma vez que essas técnicas retornam uma versão mais balanceada do conjunto de dados desbalanceados. Under sampling reduz o número de exemplos da classe majoritária para equilibrar o número de exemplos de cada classe. Isso pode fazer com que exemplos relevantes sejam excluídos. Over sampling geralmente usa todos os exemplos disponíveis na classe minoritária para sintetizar novas instâncias, que podem incluir dados ruidosos ou outliers. Assim, dependendo da natureza dos dados, o uso de um algoritmo de um tipo ou outro pode impactar negativamente o desempenho do classificador. Este artigo propõe o 'Sequenciamento de Algoritmos de Amostragem', que permite a composição de diferentes algoritmos de amostragem em um pipeline, para o balanceamento de dados. Os resultados mostraram que a proposta foi capaz de obter acurácia e F_1 score médios estatisticamente maiores quando comparados aos das abordagens de amostragem tradicionais e híbridas, na maioria dos problemas desbalanceados selecionados.*

1. Introdução

Conjuntos de dados desbalanceados estão presentes em muitos domínios, como previsão de defeito de software (Wang and Yao 2013), detecção de câncer em imagens de mamografia (Woods et al. 1993) e previsão de manutenção em sistemas de energia elétrica (Ramentol et al. 2016). No entanto, classificadores em geral assumem que os dados apresentam uma distribuição equilibrada de exemplos. Deste modo, classificadores treinados em conjuntos de dados desbalanceados tendem a reconhecer mal os exemplos da classe minoritária impactando negativamente no seu desempenho (Van Hulse et al. 2007; He and Garcia 2009).

Algoritmos de amostragem são comumente usados para tratar o problema de desbalanceamento dos dados (He and Garcia 2009). Dado um conjunto de dados desbalanceados, os algoritmos de amostragem produzem um novo conjunto de dados em que a distribuição de exemplos em cada classe é mais equilibrada. Existem duas estratégias tradicionais para o balanceamento de dados: *Under sampling* e *Over sampling* (Haixiang et al. 2016). Algoritmos de *Under sampling* reduzem o número de exemplos da classe majoritária, objetivando balancear o número de exemplos de cada classe. Esse procedimento pode fazer com que exemplos relevantes sejam eliminados. Já os algoritmos de *Over sampling* geralmente usam todos os exemplos disponíveis na classe minoritária para sintetizar novas instâncias, podendo incluir dados ruidosos ou discrepantes. Deste modo, dependendo da natureza dos dados em que se estiver trabalhando, usar um algoritmo de um tipo *ou* de outro pode impactar negativamente no desempenho do classificador (de Morais and Vasconcelos 2019; Haixiang et al. 2016).

Diante desta problemática, alguns trabalhos passaram a criar técnicas híbridas, através da combinação de algoritmos de amostragem, objetivando agregar o melhor dos algoritmos de *over* e *under sampling* (de Morais and Vasconcelos 2019; Rivera 2017; Batista et al. 2004; Sáez et al. 2015). O trabalho realizado por (Batista et al. 2004), por exemplo, usou algoritmos de *under sampling*, *Edited Nearest Neighbors* (ENN) ou *Tomek Links*, como uma etapa de limpeza após o *over-sampling* com o algoritmo SMOTE. Estes trabalhos mostraram resultados promissores no balanceamento de dados, melhorando o desempenho de classificadores.

O presente trabalho propõe o 'Sequenciamento de Algoritmos de Amostragem'. A abordagem proposta é inspirada nas técnicas híbridas, e permite a composição de diferentes algoritmos de amostragem em sequência. Dada uma base de dados desbalanceada como entrada, esta passa por um *pipeline* de algoritmos de amostragem que realizam pré- e pós-processamentos, para refinar o balanceamento, obedecendo a ordem em que estão posicionados no *pipeline*. Ao final do processo, é retornada uma base de dados balanceada. Vale salientar que, de acordo com nosso conhecimento, a abordagem proposta ainda não foi investigada no problema em questão.

A proposta foi comparada com seis diferentes abordagens de amostragem: quatro de *under sampling*, *Near Miss*, *ENN*, *Tomek Links* e *One Sided Selection* (OSS); 1 de *over sampling*, *SMOTE*; e uma abordagem híbrida, *SMOTE + Tomek Links*. No experimento foram considerados 10 problemas desbalanceados comumente utilizados na literatura. Os resultados mostraram que a proposta foi superior estatisticamente, em termos de acurácia e F_1 *score*, que a maioria das demais abordagens, em quase todos os problemas; indicando que o sequenciamento de algoritmos de amostragem é uma alternativa promissora que

pode trazer benefícios para o processo de balanceamento de dados.

Este artigo está dividido na seguinte forma: a Seção 2 apresenta trabalhos relacionados ao uso de algoritmos híbridos no balanceamento de dados; na Seção 3 apresentamos a proposta deste trabalho. Na Seção 4, é apresentada a metodologia experimental adotada; na Seção 5, encontra-se os resultados; e na Seção 6, a conclusão e trabalhos futuros.

2. Trabalhos Relacionados

Algoritmos de *under* e *over sampling* são comumente usados para o balanceamento de dados. O primeiro reduz o número de exemplos da classe majoritária, objetivando balancear o número de exemplos de cada classe. O problema é que esse procedimento pode fazer com que exemplos relevantes sejam eliminados. Já o segundo costuma usar os exemplos disponíveis na classe minoritária para produzir novas instâncias, podendo incluir dados ruidosos ou discrepantes. Seus pontos negativos motivaram o estudo de abordagens híbridas que tentam usufruir do melhor de cada estratégia.

Batista et al. (Batista et al. 2004) propôs o uso de *Edited Nearest Neighbors* (ENN) ou *Tomek Links* como uma etapa de limpeza após o *over sampling* com o SMOTE. O ENN remove os exemplos classificados erroneamente pelos seus vizinhos mais próximos ao número k, enquanto o *Tomek Links* remove os exemplos que pertencem a classes diferentes e são os vizinhos mais próximos uns dos outros. Os autores avaliaram os métodos propostos, SMOTE + ENN e SMOTE + Tomek Links, em 10 conjuntos de dados usando uma árvore de decisão C4.5 como classificador base e AUROC (do inglês, *Area under Receiver Operating Characteristic*) como a métrica de desempenho. Os resultados mostraram que os métodos híbridos obtiveram melhor desempenho que o SMOTE sozinho.

Saz et al. (Sáez et al. 2015) propôs o uso de um filtro de ruído baseado em conjunto iterativo para remover exemplos ruidosos após *over sampling* com SMOTE. O objetivo era remover exemplos ruidosos já presentes nos dados e também aqueles introduzidos pelo SMOTE. A experimentação empregada para avaliar o método proposto, SMOTE-IPF, compreendeu 9 conjuntos de dados, C4.5 Decision Tree como classificador base e AUROC como a métrica de desempenho. Os resultados do SMOTE-IPF superaram os SMOTE, SMOTE + ENN e SMOTE + Tomek links.

Recentemente, Rivera (Rivera 2017) propôs *Noise Reduction A Priori Synthetic OverSampling* (NRAS), que remove exemplos ruidosos da classe minoritária antes do *over-sampling*. O NRAS remove exemplos da classe minoritária que tem o número de vizinhos mais próximos pertencentes à classe minoritária abaixo de um limiar e sintetiza novos exemplos de forma semelhante ao SMOTE. No entanto, antes de calcular os vizinhos mais próximos, o NRAS inclui a probabilidade de pertencer à classe minoritária como um novo recurso no conjunto de treinamento. A experimentação conduzida para avaliar o NRAS incluiu 41 conjuntos de dados, 4 classificadores e 3 métricas de desempenho. O NRAS foi comparado a vários algoritmos de *over-sampling* e consistentemente alcançou uma classificação média mais alta para as métricas de revocação e média geométrica.

Os trabalhos apresentados anteriormente combinaram dois algoritmos de amostragem, e mostraram resultados promissores quando comparados à abordagens tradicionais. Porém, como trata-se de uma ideia recente, poucos trabalhos se debruçaram sobre a real

contribuição e impacto da combinação de algoritmos no balanceamento de dados. Diferentemente dos trabalhos anteriores, a abordagem proposta neste trabalho permite a composição de não apenas dois, mas de vários algoritmos de amostragem sendo executados em sequência. Além disso, este trabalho realiza uma análise da real contribuição e impacto da combinação de algoritmos. Mais detalhes sobre o sequenciamento de algoritmos de amostragem, proposto aqui, serão apresentados a seguir.

3. Sequenciamento de Algoritmos de Amostragem

O sequenciamento de algoritmos de amostragem trata-se da criação de um *pipeline* de execução de algoritmos de amostragem (ver Figura 1). Ao receber como entrada uma base de dados desbalanceada, cada algoritmo A_i é executado, obedecendo a ordem em que se encontra no *pipeline*, provendo a base retornada como entrada para o próximo algoritmo A_{i+1}. A_{i+1} será responsável por refinar o balanceamento feito pelo algoritmo anterior, e passar a base refinada para o próximo algoritmo da sequência, caso exista. Ao final do processo, é gerada uma base resultante balanceada.

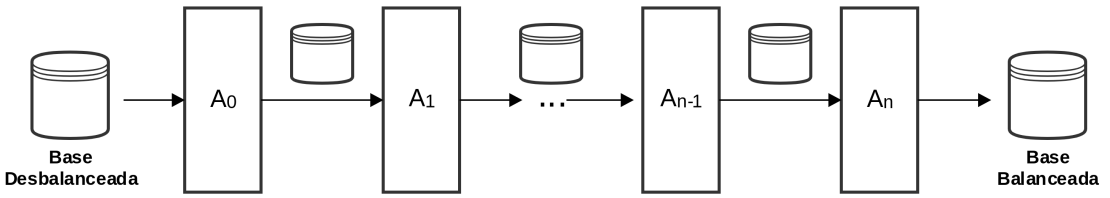

Figura 1. *Pipeline* de execução de algoritmos.

Neste trabalho, foram considerados 6 algoritmos de amostragem: 4 de *under sampling*: *NearMiss*, ENN, *Tomek Links*, *One Sided Selection*; 1 de *over sampling*: SMOTE; e 1 híbrido: SMOTE + *Tomek links*. Em termos práticos, o *pipeline* foi definido como um vetor de inteiros, de tamanho N, onde cada posição pode assumir valores no intervalo $[0, 6]$. Cada valor entre 1 e 6 representa um algoritmo de amostragem: (1) *NearMiss*, (2) ENN, (3) *Tomek Links*, (4) *One Sided Selection*, (5) SMOTE e (6) SMOTE + *Tomek links*. Caso uma determinada posição do vetor possua o valor 0, significa dizer que não há algoritmo de amostragem alocado naquela posição. A Figura 2 mostra um exemplo de *pipeline* considerando um vetor com $N = 7$. Neste exemplo, o *pipeline* possui 5 algoritmos de amostragem em sequência, pois 2 posições do vetor receberam o valor zero. Posto isso, se pode ver que o *pipeline* pode assumir diferentes configurações. Pode haver repetição de algoritmos de amostragem; se pode criar *pipelines* com 0 a N algoritmos; e posicionar os algoritmos como desejar. Para $N = 7$, o número total de combinações possíveis de sequências seria de 823.543.

A abordagem proposta é flexível, permitindo que o especialista possa construir sequências de execução arbitrárias e realizar uma variedade de análises quanto à contribuição do sequenciamento destes algoritmos. Além de propor o sequenciamento de algoritmos de amostragem, o presente trabalho se propõe a analisar o impacto da combinação de diferentes algoritmos de amostragem no balanceamento de dados. A seguir é apresentada a metodologia experimental adotada para avaliar e analisar o desempenho da proposta frente à outras abordagens comumente usadas na literatura.

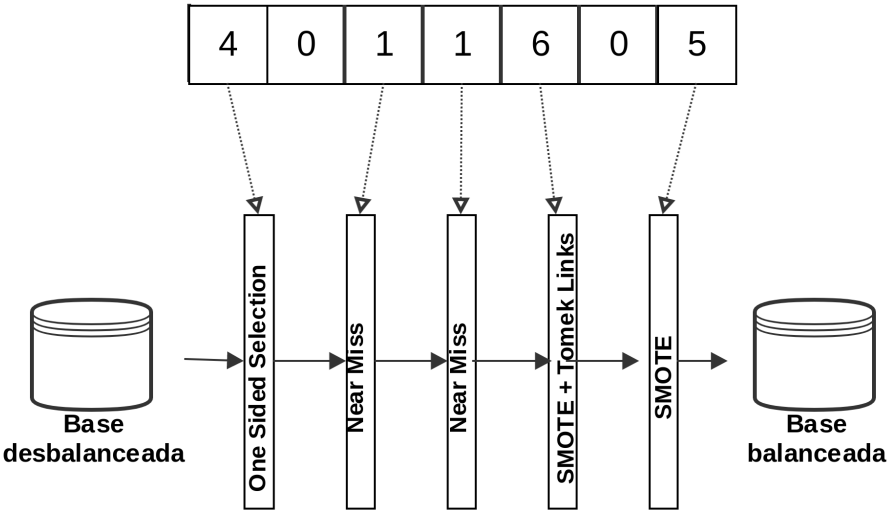

Figura 2. Exemplo de *pipeline* de execução de algoritmos, com $N = 7$.

4. Metodologia Experimental

Nesta seção, apresentamos a metodologia usada para avaliar e analisar a abordagem proposta. A abordagem proposta permite, dependendo do valor de N, a geração de uma grande variedade de sequências. Como não existe a possibilidade de analisar cada possível sequência nos experimentos, se decidiu criar e avaliar um determinado número de sequências aleatoriamente, usando-se um método computacional de busca aleatória (ver Algoritmo 1). Este método de busca é iterativo, e a cada iteração um conjunto de novas soluções $candidates$ é criado aleatoriamente seguindo uma distribuição uniforme. Neste trabalho, cada solução foi representada por um vetor de tamanho $N = 7$. Vale salientar que o conjunto de soluções criado pode conter soluções (sequências) de diferentes tamanhos (0 a N) e ordem de algoritmos. Assim como foi explicado na seção 3. Em seguida, cada solução pertencente a $candidates$ tem sua qualidade mensurada através da função $evaluate\_fitness$. Esta função executa o $candidate$ (sequência) corrente na base de dados desbalanceada ($imbalanced\_dataset$), gerando uma base de dados balanceada. Em seguida, os resultados médios de acurácia e F_1 *score* são obtidos pelo classificador ($classifier$), através do experimento de validação cruzada 10-*fold* executado 30 vezes, quando aplicado à base de dados balanceada gerada. Neste trabalho, foi adotada a Máquina de Vetor de Suporte (SVM) como algoritmo de classificação. A melhor solução encontrada ($best$) é atualizada a cada iteração, de modo que ao final da busca é retornada a melhor solução dentre todas avaliadas.

O método de busca aleatória foi executado em cada uma das bases de dados consideradas neste trabalho (ver na Seção 4.1). Isto foi feito com o intuito de encontrar uma solução promissora (sequência de algoritmos) para cada uma delas. O método de busca adotou 1000 iterações como critério de parada, e $pop\_size = 100$. Isso significa que, para cada base de dados, foram avaliadas 100.000 diferentes sequências de algoritmos de amostragem, sendo a melhor ($best$), dentre as avaliadas, retornada.

Neste trabalho, comparamos a nossa proposta com abordagens tradicionais (*under* ou *over sampling*) e híbridas. As abordagens tradicionais escolhidas foram: *Near Miss*, *ENN*, *SMOTE*, *Tomek Links* e *One Sided Selection* (OSS). A abordagem híbrida

Algoritmo 1: Método de busca aleatória adotado.

PARÂMETROS:
$imbalanced\_dataset$: Base de dados desbalanceada.
$classifier$: Algoritmo de classificação.
pop_size: O número de soluções avaliadas por iteração.
$best$: Melhor solução encontrada até o momento.
$best$ = None

```
while !stop_criterion do
    candidates = new_random_solutions(pop_size)
    for candidate in candidates do
        evaluate_fitness(candidate, classifier, imbalanced_dataset)
    end
    best = update_best(best, candidates)
end
return best
```

utilizada foi a *SMOTE + Tomek Links*. A comparação entre todas as abordagens foi realizada em termos da acurácia e F_1 *score* obtidos pela SVM através do experimento de validação cruzada 10-*fold*, executado 30 vezes. Como se pode ver, a mesma metodologia de avaliação adotada para a obtenção dos resultados da proposta (via método de busca aleatória), também foi seguida para cada uma das abordagens adversárias. A seguir, serão apresentadas as bases de dados usadas neste trabalho, configurações experimentais, e também cada sequência que foi encontrada, por base de dados, pelo método de busca aleatória.

4.1. Bases de Dados

Para os experimentos, foram utilizadas 10 bases de dados desbalanceadas provenientes do repositório UCI (Lichman 2013). A Tabela 1 resume todos os conjuntos de dados empregados, incluindo o número de exemplos, o número de atributos e a taxa de desequilíbrio (o número de exemplos majoritários dividido pelo número de exemplos minoritários).

4.2. Configurações

Neste trabalho, foi utilizada a SVM, como algoritmo de classificação, proveniente da biblioteca *Scikit learn* (Pedregosa et al. 2011). A SVM adotou a parametrização *default* da biblioteca, assumindo a seguinte parametrização: $C = 1,0$, $kernel$='rbf', $degree = 3,0$ e γ = 'auto'. Os algoritmos de amostragem usados neste trabalho são provenientes da biblioteca *imbalanced-learn* (Lemaître et al. 2017), e também foram configurados com seus parâmetros *default* (ver Tabela 2). Todas as execuções foram realizadas em um computador com um processador Intel Celeron J1800 com 1Mb de memória cache, velocidade de clock de 2,41 GHz e turbo de até 2,58 GHz, e 4 GB de RAM.

4.3. Sequências Retornadas

A Tabela 3 mostra cada sequência retornada (pelo método de busca aleatória) para cada um dos problemas desbalanceados. Como se pode ver, todas as sequências retornadas

Tabela 1. Descrição das bases de dados utilizadas no experimento. Cada base de dados é descrita por seu nome, o número de exemplos na classe majoritária (#Maj.), O número de exemplos na classe minoritária (#Min.) E a relação de desequilíbrio entre as classes (IR).

Nome	# Maj.	# Min.	IR
Blood Transfusion	353	118	2.99
column_2C	210	100	2.10
ecoli1	259	77	3.36
ecoli2	284	52	5.46
fourclass	555	307	1.81
glass1	137	76	1.80
pima	500	268	1.87
vehicle0	647	199	3.25
vehicle2	628	218	2.88
yeast	1028	425	2.42

Tabela 2. Parâmetros *default* de cada abordagem de amostragem.

Algoritmo	Parâmetros
ENN	$sampling\_strategy$='auto', $return\_indices$=False, $random\_state$=None, $n\_neighbors = 3$, $kind_sel$='all', $n\_jobs = 1$, $ratio$=None, $padding$='same'
NearMiss	$sampling\_strategy$='auto', $return\_indices$=False, $random\_state$=None, $n\_neighbors = 3$, $version = 1$, $n\_jobs = 1$, $ratio$=None, $n\_neighbors\_ver3 = 3$
OSS	$sampling\_strategy$='auto', $return\_indices$=False, $random\_state$=None, $n\_neighbors$=None, $n\_jobs = 1$, $ratio$=None, $n\_seed\_S = 1$
SMOTE	$sampling\_strategy$='auto', $random\_state$=None, $k\_neighbors$=5, $n\_jobs = 1$ $ratio$=None
SMOTE+Tomek	$sampling\_strategy$='auto', $random\_state$=None, $SMOTE$=None, $tomek$=None, $n\_jobs = 1$, $ratio$=None
TomekLinks	$sampling\_strategy$='auto', $return\_indices$=False, $random\_state$=None, $n\_jobs = 1$, $ratio$=None

possuem ao menos 3 algoritmos de amostragem, podendo ser um indício positivo de que sequências de algoritmos podem auxiliar no balanceamento de dados. Alguns pontos interessantes: dos 10 problemas desbalanceados envolvidos, 4 deles possuem sequências compostas apenas por algoritmos de *under sampling*; dentre as sequências retornadas, não há uma sequer composta apenas por algoritmos de *over sampling*; e para 60% dos problemas foram retornadas sequências heterogêneas, ou seja, compostas por algoritmos de *under*, *over sampling* ou híbrido.

Na seção 5, cada uma das sequências retornadas (na Tabela 3) terá seus resultados de acurácia e F_1 *score* comparados com os das 6 abordagens adversárias mencionadas na seção 4.

Tabela 3. Sequência utilizada para cada problema desbalanceado.

Problema	Sequência	# Algs.
collumn_2C	NearMiss → TomekLinks → TomekLinks → OSS → NearMiss	5
ecoli1	NearMiss → NearMiss → SMOTE+Tomek → OSS → NearMiss → NearMiss → TomekLinks	7
ecoli2	OSS → OSS → TomekLinks → OSS → OSS	5
fourclass	NearMiss → TomekLinks → ENN	3
glass1	ENN → OSS → ENN → TomekLinks → SMOTE → ENN → TomekLinks	7
pima	SMOTE+Tomek → SMOTE → NearMiss → OSS	4
transfusion	NearMiss → ENN → OSS → SMOTE+Tomek → TomekLinks	5
vehicle0	SMOTE+Tomek → SMOTE → SMOTE+Tomek → NearMiss → OSS	5
vehicle2	SMOTE → OSS → NearMiss	3
yeast	TomekLinks → OSS → OSS → ENN → OSS	5

5. Resultados

Nesta seção, comparamos os resultados obtidos pela proposta com os das abordagens tradicionais (*under* ou *over sampling*) e híbridas, em termos de acurácia e F_1 *score*.

A Tabela 4 mostra a média e desvio padrão da acurácia obtida por cada abordagem quando aplicadas a cada base de dados desbalanceada. Como se pode ver, as sequências obtidas pela proposta (via busca aleatória) atingiram acurácia média superior em relação a todas as demais, em todos os problemas. Com o intuito de ter uma comprovação estatística desta superioridade, foram realizados testes estatísticos como sugerido em (García et al. 2010). Primeiro, a hipótese nula é de que não há diferença entre os valores médios das sete abordagens. A hipótese alternativa é que existe pelo menos uma diferença entre os valores médios. Em segundo lugar, foi conduzido um teste não paramétrico chamado *Friedman Aligned-Ranks*, e a hipótese nula foi rejeitada, com um $p - value = 2,2 \times 10^{-16}$. Este resultado comprova que há diferença estatísticas entre as médias. Finalmente, uma vez que a hipótese nula foi rejeitada pelo teste de Friedman, um teste *post-hoc* foi realizado para identificar quais diferenças são significativas. Usamos o procedimento de *Finner* com correção do $p - value$ (já que múltiplas comparações estão sendo feitas) e definimos a abordagem proposta como o algoritmo de controle e o comparamos com as outras seis abordagens: *Near Miss*, *ENN*, *SMOTE*, *Tomek Links*, *OSS* e *SMOTE+Tomek*.

Os valores em negrito são os melhores valores, estatisticamente. Como se pode ver, os resultados obtidos pela proposta superaram estatisticamente a grande maioria das abordagens, não sendo superados por nenhuma delas. É possível identificar similaridade estatística dos resultados obtidos pela proposta com três abordagens, *NearMiss*, *ENN* e *TomekLinks*, em apenas três problemas de classificação, *vehicle0*, *yeast* e *ecoli02*, respectivamente.

A mesma análise, feita para a acurácia, também foi realizada com relação ao F_1 *score* (ver Tabela 5). Como se pode ver, a abordagem proposta também conseguiu resultados expressivos com relação ao F_1 *score*. Nos problemas *column_2C* e *transfusion*,

Tabela 4. Média e desvio padrão da acurácia obtidos pelas abordagens em cada base de dados desbalanceada.

Problema	Algoritmos						
	Proposta	NearMiss	ENN	SMOTE	TomekLinks	OSS	SMOTE+Tomek
collumn_2C	**0.80**(±0.00)	0.73(±0.09)	0.77(±0.05)	0.75(±0.07)	0.74(±0.06)	0.74(±0.07)	0.76(±0.05)
ecoli1	**0.93**(±0.01)	0.89(±0.00)	0.86(±0.00)	0.86(±0.01)	0.89(±0.00)	0.88(±0.00)	0.86(±0.00)
ecoli2	**0.96**(±0.01)	0.76(±0.00)	0.92(±0.00)	0.89(±0.00)	**0.95**(±0.00)	0.93(±0.01)	0.89(±0.00)
fourclass	**0.75**(±0.00)	0.69(±0.06)	0.66(±0.09)	0.65(±0.08)	0.68(±0.07)	0.67(±0.08)	0.67(±0.07)
glass1	**0.64**(±0.03)	0.41(±0.04)	0.53(±0.10)	0.53(±0.10)	0.55(±0.11)	0.55(±0.11)	0.58(±0.08)
pima	**0.66**(±0.00)	0.55(±0.11)	0.62(±0.08)	0.60(±0.11)	0.61(±0.10)	0.62(±0.08)	0.58(±0.08)
transfusion	**0.74**(±0.00)	0.56(±0.02)	0.56(±0.02)	0.62(±0.01)	0.65(±0.01)	0.63(±0.01)	0.64(±0.01)
vehicle0	**0.95**(±0.00)	**0.94**(±0.03)	0.92(±0.01)	0.93(±0.02)	0.92(±0.04)	0.92(±0.05)	0.93(±0.03)
vehicle2	**0.96**(±0.00)	0.83(±0.07)	0.91(±0.07)	0.93(±0.05)	0.92(±0.06)	0.92(±0.011)	0.91(±0.06)
yeast	**0.94**(±0.00)	0.79(±0.00)	**0.93**(±0.00)	0.83(±0.00)	0.91(±0.00)	0.91(±0.00)	0.83(±0.01)

os resultados da proposta foram superiores aos de todas as demais abordagens. Nos problemas *ecoli1*, *ecoli2* e *glass1* a proposta foi superada estatisticamente por apenas um algoritmo de amostragem, mas ainda assim, conseguiu atingir valores acima da média dos demais algoritmos. Nos outros 5 problemas de classificação, a proposta empatou estatisticamente em primeiro lugar com o *ENN* para o *fourclass*, com o *SMOTE+Tomek* para o *pima*, com o *NearMiss* para *vehicle0*, com o *SMOTE* e *OSS* para o *vehicle2*, e com o *ENN* para o *yeast*.

Embora a proposta tenha alcançado resultados superiores em relação às demais abordagens, é importante salientar que dentre todas as sequências possíveis (823.543 combinações), considerando $N = 7$, apenas 12.41% delas (100.000) foram avaliadas através de um método de busca aleatória. Isso significa que com a utilização de um algoritmo de otimização inteligente (por exemplo, Algoritmo Genético), talvez seja possível encontrar sequências capazes de alcançar resultados ainda melhores.

Tabela 5. Média e desvio padrão do F_1 *score* obtidos pelas abordagens em cada base de dados desbalanceada.

Problema	Algoritmos						
	Proposta	NearMiss	ENN	SMOTE	TomekLinks	OSS	SMOTE+Tomek
collumn_2C	**0.66**(±0.02)	0.56(±0.02)	0.63(±0.03)	0.56(±0.03)	0.53(±0.02)	0.61(±0.02)	0.58(±0.03)
ecoli1	0.72(±0.01)	**0.75**(±0.00)	0.73(±0.00)	0.72(±0.02)	0.75(±0.00)	0.74(±0.00)	0.72(±0.01)
ecoli2	0.79(±0.06)	0.54(±0.00)	0.78(±0.00)	0.72(±0.00)	**0.83**(±0.00)	0.79(±0.04)	0.72(±0.01)
fourclass	**0.50**(±0.01)	0.47(±0.02)	**0.50**(±0.02)	0.45(±0.02)	0.46(±0.02)	**0.53**(±0.02)	0.48(±0.02)
glass1	0.30(±0.03)	0.32(±0.02)	0.31(±0.02)	**0.37**(±0.02)	0.24(±0.02)	0.19(±0.02)	0.27(±0.02)
pima	**0.42**(±0.02)	0.38(±0.02)	0.39(±0.02)	0.41(±0.02)	0.29(±0.02)	0.24(±0.02)	**0.42**(±0.02)
transfusion	**0.40**(±0.02)	0.16(±0.02)	0.24(±0.02)	0.17(±0.02)	0.15(±0.02)	0.19(±0.02)	0.21(±0.03)
vehicle0	**0.88**(±0.03)	**0.88**(±0.06)	0.85(±0.02)	0.87(±0.07)	0.85(±0.07)	0.85(±0.08)	0.86(±0.06)
vehicle2	**0.88**(±0.06)	0.72(±0.14)	0.82(±0.10)	**0.88**(±0.09)	0.85(±0.10)	**0.88**(±0.09)	0.85(±0.10)
yeast	**0.65**(±0.01)	0.41(±0.00)	**0.65**(±0.00)	0.47(±0.01)	0.27(±0.00)	0.29(±0.02)	0.47(±0.02)

Um ponto importante que vale ser destacado é o desempenho da abordagem híbrida *SMOTE+Tomek* tanto em acurácia quanto em F_1 *score*. Embora seja uma abordagem que combina dois algoritmos de amostragem, seus resultados foram aquém do esperado, sendo superados por algoritmos tradicionais de amostragem. Isto mostra que nem todas as combinações de algoritmos são capazes de realizar um bom balanceamento de dados, sendo importante avaliar outros tipos de combinações de algoritmos (com di-

ferentes tamanhos e ordens). Por fim, os resultados de acurácia e F_1 *score* alcançados pela abordagem proposta mostraram que o sequenciamento de algoritmos de amostragem é uma alternativa promissora e, que pode trazer benefícios para o processo de balanceamento de dados.

6. Conclusão e Trabalhos Futuros

Neste trabalho, é proposto o 'Sequenciamento de Algoritmos de Amostragem' com o intuito de evitar certas limitações encontradas por abordagens tradicionais. Inspirada em algoritmos de amostragem híbridos, a abordagem proposta permite a composição de diferentes algoritmos de amostragem em sequência, retornando ao final do *pipeline* uma base de dados balanceada. Os resultados mostraram que a proposta foi capaz de obter resultados médios de acurácia e F_1 *score* superiores estatisticamente quando comparados aos das abordagens tradicionais e híbridas, na maioria dos problemas desbalanceados. Isto mostra que o sequenciamento de algoritmos de amostragem é uma alternativa promissora, trazendo benefícios para o processo de balanceamento de dados.

Como trabalho futuro, pretendemos utilizar algoritmos genéticos na busca por sequências mais otimizadas. Além disso, iremos incluir no processo de otimização mais algoritmos de amostragem (tradicionais e híbridos), e mais métricas como *AUROC*, Média Geométrica, precisão e revocação. Também faz parte do escopo estudar sobre o impacto que a quantidade de algoritmos contidos na sequência e a ordem em que se encontram têm no balanceamento de dados. Por fim, ampliaremos o número de bases de dados desbalanceadas afim de validar a proposta em diferentes domínios.

Referências

[Batista et al. 2004] Batista, G. E., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29.

[de Morais and Vasconcelos 2019] de Morais, R. F. and Vasconcelos, G. C. (2019). Boosting the performance of over-sampling algorithms through under-sampling the minority class. *Neurocomputing*, 343:3–18.

[García et al. 2010] García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064.

[Haixiang et al. 2016] Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., and Bing, G. (2016). Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*.

[He and Garcia 2009] He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Trans. on Knowl. and Data Eng.*, 21(9):1263–1284.

[Lemaître et al. 2017] Lemaître, G., Nogueira, F., and Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

[Lichman 2013] Lichman, M. (2013). UCI machine learning repository.

[Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

[Ramentol et al. 2016] Ramentol, E., Gondres, I., Lajes, S., Bello, R., Caballero, Y., Cornelis, C., and Herrera, F. (2016). Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: The smote-frst-2t algorithm. *Engineering Applications of Artificial Intelligence*, 48:134–139.

[Rivera 2017] Rivera, W. A. (2017). Noise reduction a priori synthetic over-sampling for class imbalanced data sets. *Information Sciences*, 408:146–161.

[Sáez et al. 2015] Sáez, J. A., Luengo, J., Stefanowski, J., and Herrera, F. (2015). Smote–ipf: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Information Sciences*, 291:184–203.

[Van Hulse et al. 2007] Van Hulse, J., Khoshgoftaar, T. M., and Napolitano, A. (2007). Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942. ACM.

[Wang and Yao 2013] Wang, S. and Yao, X. (2013). Using class imbalance learning for software defect prediction. *IEEE Transactions on Reliability*, 62(2):434–443.

[Woods et al. 1993] Woods, K. S., DOSS, C. C., BOWYER, K. W., SOLKA, J. L., PRIEBE, C. E., and KEGELMEYER JR, W. P. (1993). Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(06):1417–1436.