

# Uma abordagem para a escolha do melhor método de seleção de instâncias usando meta-aprendizagem

Shayane de Oliveira Moura\*<sup>†</sup>, Marcelo Bassani de Freitas\*, Halisson A. C. Cardoso\* e George D. C. Cavalcanti\*

\*Centro de Informática, Universidade Federal de Pernambuco, Recife, Brasil

\*Email: {som2, mbf2, hacc, gdcc}@cin.ufpe.br

<sup>†</sup>Departamento de Informática, Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Ouricuri, Brasil

**Resumo**—Métodos de seleção de instâncias (MSIs) têm sido amplamente estudados e propostos. Entretanto, nenhum destes métodos obtém boa taxa de acerto em todas as bases de dados. Neste trabalho, é proposta uma arquitetura para selecionar o melhor MSI para uma dada base de dados. Estratégias de meta-aprendizagem são utilizadas para treinar um meta-classificador que aprende sobre o relacionamento entre a taxa de acerto de MSIs e a estrutura das bases de dados. O método proposto foi avaliado com bases de dados públicas e mostrou resultados melhores que abordagens tradicionais.

## I. INTRODUÇÃO

O objetivo de métodos de seleção de instâncias (MSIs) é reduzir o número de instâncias de uma base de dados. O conjunto resultante de instâncias deve apresentar a mesma ou até melhor taxa de classificação que aquela alcançada com o uso da base de dados completa. MSIs eliminam ruídos e fazem as bases de dados mais descritivas em relação aos problemas que representam, aumentando a eficiência dos processos de classificação.

A performance dos MSIs varia de acordo com as bases de dados nas quais eles são aplicados. Peculiaridades destas bases criam ambientes que são mais adequados a alguns métodos que outros. Os trabalhos de Reinartz [1], Kim e Oommen [2] e Czarnowski e Jedrzejowicz [3] corroboram que a falta de um MSI que constantemente se sobreponha aos demais direciona pesquisas recentes à busca de formas para definir o MSI mais adequado para cada base de dados.

A seleção do melhor MSI para uma base de dados foi estudada por Caises et al. [4]. Eles propuseram o método *Selective Combination for Instance Selection* (SCIS) para determinar o mais adequado MSI a ser aplicado às bases de dados de entrada. O SCIS usa regras empíricas construídas baseadas em medidas de caracterização de bases de dados. Embora este estudo tenha obtido bons resultados, este processo é limitado. Considerando que são feitas por humanos, regras empíricas são limitadas ao tamanho da amostra, esforço do especialista e duração da análise, que afetam diretamente a eficácia e eficiência do processo.

O problema de seleção de MSIs pode ser resolvido com meta-aprendizagem, que é uma abordagem mais formal e automatizada, como mostrado por Leyva et al. [5]. Meta-aprendizagem é um procedimento que busca encontrar padrões de comportamento em muitas bases de dados em relação à saída. Ela aprende de uma meta-base composta de medidas de caracterização que representam estruturas de bases de dados

(as meta-características) e da performance dos algoritmos (os rótulos) quando aplicados àquelas bases de dados.

Este artigo propõe uma arquitetura que usa meta-aprendizagem para selecionar o melhor MSI por base de dados. A arquitetura usa um meta-classificador para aprender como as estruturas das bases de dados podem influenciar na performance dos MSIs. O meta-classificador é treinado com uma meta-base que é composta por medidas de características extraídas de diferentes bases de dados. Um método de reamostragem é utilizado para aumentar o número de bases de dados disponíveis para meta-base.

Este artigo possui a seguinte organização: a Seção II descreve os MSIs selecionados; a Seção III mostra a arquitetura proposta que é composta pela construção da meta-base, o treinamento do meta-classificador e o procedimento de avaliação; as configurações experimentais usadas neste trabalho são detalhadas na Seção IV; os resultados são apresentados na Seção V; e a Seção VI conclui este artigo e sugere trabalhos futuros.

## II. MÉTODOS DE SELEÇÃO DE INSTÂNCIAS USADOS NESTE ARTIGO

Desde 1968, quando o primeiro MSI foi criado, *Condensed Nearest Neighbor* - CNN [6], mais de setenta outros métodos foram sugeridos por muitos autores, como podemos ver na taxonomia de MSIs apresentada por Garcia et al. [7]. Eles também mostram que os métodos estudados se distinguem por suas abordagens em relação à:

- tipo de seleção: métodos de condensação, edição e híbridos;
- direção de pesquisa: incremental, decremental, em lote, misto e fixo;
- avaliação da pesquisa: filtro e *wrapper*;
- outras propriedades: representação, função de distância e votação.

Baseados nesta análise, foram escolhidos MSIs que possuem abordagens distintas, tais como: ENN-CNN, IB3, DROP3, ICF e C-pruner.

Na família de métodos de condensação, CNN tem uma direção de pesquisa incremental. O método CNN consiste de três passos. No primeiro passo, uma instância de cada classe é selecionada aleatoriamente para formar uma nova base de dados. No segundo passo, esta nova base de dados é usada para treinar um classificador que classifica todas as instâncias da

base de dados original. Finalmente, as instâncias classificadas erroneamente são incluídas na nova base de dados a fim de garantir que instâncias desconhecidas e similares àquelas classificadas erroneamente serão corretamente classificadas. CNN é muito sensível a ruído por causa da inclusão das instâncias classificadas erroneamente. Apesar disso, ele ainda é o mais usado em trabalhos recentes, pois as mudanças propostas para o algoritmo são mais complexas, necessitam de elevado tempo para processamento e continuam sensíveis a ruídos. Outros métodos desta família são: *Reduced Nearest Neighbor* - RNN (decremental) [8], *Selective Nearest Neighbor* - SNN (decremental) [9], *Patterns by Ordered Projection* - POP (batch) [10] and *Minimal Consistent Set* - MCS (decremental) [11].

O método decremental *Edited Nearest Neighbor* - ENN [12] pertence à família dos métodos de edição. Ele remove as instâncias que têm classes diferentes de seus vizinhos. ENN é normalmente usado apenas como filtro de ruídos, devido a sua baixa taxa de redução. Métodos que tentaram melhorar o ENN como *Repeated ENN* - RENN (decremental) e *All-KNN* (em lote) [13] aumentaram a complexidade, o custo computacional e não obtiveram boas taxas de redução. Então, foi adotada a combinação proposta por Caisés et al. [4] combinando ENN com CNN, visto que possuem resultados complementares.

A maioria dos métodos propostos atualmente é da família de métodos híbridos. Esses métodos tentam combinar as vantagens das abordagens de condensação e edição para construir métodos que alcancem boas taxas de redução, sem perdas relevantes em suas precisões e com baixo custo computacional. Por esses motivos, essa família possui uma maior importância e, portanto, foram utilizados quatro métodos híbridos. Eles diferem na direção de pesquisa: o método incremental *Instance Based 3* - IB3 [14]; os métodos decrementais e filtros de avaliação de pesquisa: *Decremental Reduction Optimization Procedure 3* - DROP3 [15] e C-pruner [16]; e o método de processamento em lote *Iterative Caise Filtering* - ICF [17].

No método IB3, uma nova base de dados é criada com as instâncias da base de dados original cujas instâncias aceitáveis mais próximas têm classes diferentes das instâncias que estão sendo adicionadas. A aceitabilidade de instâncias é baseada em intervalo de confiança e precisão de classificação.

O DROP3 executa primeiramente o algoritmo ENN. Depois, ordena as instâncias de acordo com suas distâncias para a instância mais próxima que possui classe diferente. Em seguida, DROP3 remove toda instância  $x$  que não altera a classificação de  $\mathcal{A}(x)$  - o conjunto das instâncias que dependem de  $x$  na classificação dos  $K$  vizinhos mais próximos (KNN).

O objetivo de ICF é remover instâncias dos centros das classes e reter as instâncias das bordas das classes. O ICF usa o método ENN como um filtro de ruído no seu pré-processamento. Logo após, as instâncias são removidas usando o conceito de alcançabilidade e cobertura. A alcançabilidade fornece o número de instâncias que a instância  $x$  descreve. A cobertura fornece o número de instâncias que são corretamente classificadas por  $x$ . A instância  $x$  é removida se a sua alcançabilidade é maior que sua cobertura.

O método C-pruner também usa os conceitos de alcançabilidade e cobertura, mas sua cobertura considera ape-

nas as instâncias da mesma classe. Estes dois conceitos são usados para determinar se uma instância é ruidosa, supérflua ou crítica. As instâncias críticas afetam a classificação de outras instâncias, por isso, são as únicas retidas na base de dados final. As instâncias ruidosas e supérfluas são removidas.

Além da popularidade dos métodos mencionados, em Janowski e Grochowski [18] uma comparação entre algoritmos de seleção de instâncias mostra a performance relevante dos métodos escolhidos. Entre os métodos IB's, IB3 tem o melhor resultado. E em relação a métodos DROP's, DROP3 tem boas saídas com baixo custo. Os outros métodos têm performance semelhante àquelas do grupo a que pertencem, mas diferem conforme explicado anteriormente.

### III. MÉTODO PROPOSTO

A arquitetura proposta é composta por duas partes: Geração do Meta-classificador (Fig.1) e Avaliação do Meta-classificador (Fig.2).

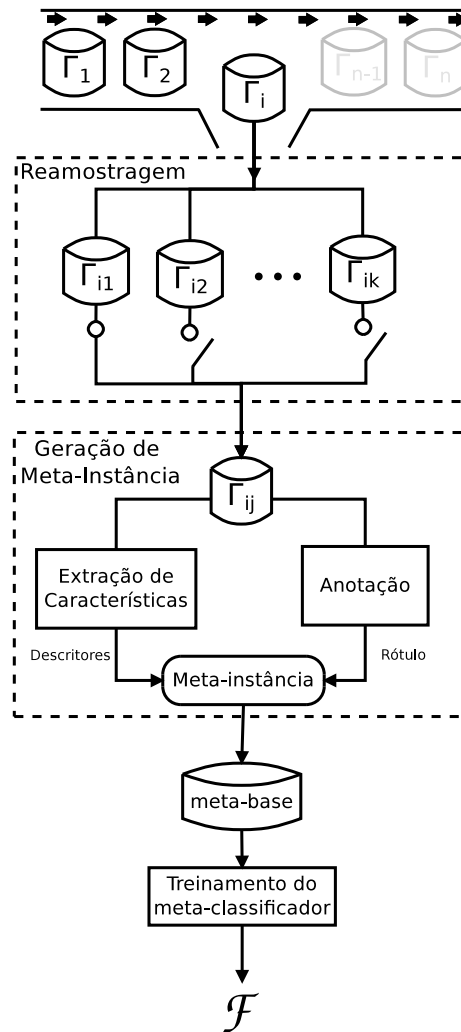


Fig. 1. Processo de construção da Meta-base e treinamento do meta-classificador. Cada base de dados é submetida à fase de reamostragem para aumentar o número de bases de dados disponíveis. Cada nova base de dados passa pelos processos de Extração de Características e Anotação para tornar-se uma meta-instância. A meta-base resultante é usada para treinar o meta-classificador  $\mathcal{F}$ .

A Geração do meta-classificador objetiva treinar o meta-classificador que é responsável por selecionar o “melhor” método de seleção de instâncias (MSI) dada uma base de dados. O meta-classificador aprende a partir de uma meta-base e a construção desta meta-base envolve dois processos: Extração de Características e Anotação.

Para cada base de dados produzida por um processo de reamostragem, seus descritores são obtidos (Extração de Características) e o “melhor” MSI é definido (Anotação). Este par (descritores, “melhor” MSI) define uma meta-instância que é incluída na meta-base. Como cada base de dados  $\Gamma_i$  gera apenas uma meta-instância, o método de reamostragem é usado para aumentar o número de bases de dados disponíveis e, portanto, a quantidade de dados disponíveis para o treinamento do meta-classificador.

Uma vez que a meta-base está completa, como descrito na Seção III-A, ela é usada para treinar o meta-classificador. O meta-classificador é avaliado usando uma base de dados de teste  $\Delta$  como mostrado na Fig.2.

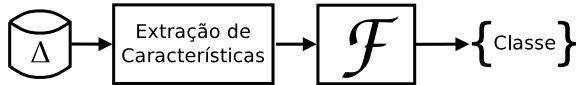


Fig. 2. Avaliação do Meta-classificador. Cada base de dados de teste tem suas características extraídas. Estas características são submetidas ao meta-classificador treinado  $\mathcal{F}$  para obter o MSI mais adequado para a base de dados de entrada  $\Delta$ .

#### A. Construção da meta-base

A meta-base requer duas fases para ser gerada: Reamostragem e Geração de Meta-instância, como mostrado na Fig.1.

O método de reamostragem cria novas bases de dados a partir das originais. Qualquer método de reamostragem pode ser utilizado. O propósito é aumentar o número de bases de dados disponíveis e, portanto, aumentar a quantidade de dados para o treinamento do meta-classificador, visto que cada base de dados  $\Gamma_i$  gera apenas uma meta-instância.

Os componentes da meta-instância (descritores e rótulo) são criados separadamente nos processos de Extração de Características e Anotação, respectivamente (Fig.1).

1) *Extração de Características*: As características são extraídas das bases de dados  $\Gamma_{ij}$  (Fig.1) e seus propósitos são dar informação sobre a estrutura das bases de dados e dar informações que influenciem na performance dos MSIs.

2) *Anotação*: Este processo é responsável por selecionar o “melhor” MSI para cada base de dados  $\Gamma_{ij}$ . Neste trabalho, o melhor método é aquele que alcança a maior média de precisão de classificação.

A precisão média para cada MSI é calculada da seguinte forma: (1) a base de dados é reduzida por N MSIs (Fig.3) e (2) os novos conjuntos de instâncias gerados  $\Omega_1$  são submetidos a M (Fig.3) classificadores ( $c_1, c_2, \dots, c_M$ ) para obter suas precisões de classificação. Depois de obtidas as médias das precisões, o MSI com maior média é o selecionado como rótulo. Este processo é ilustrado na Fig.3.

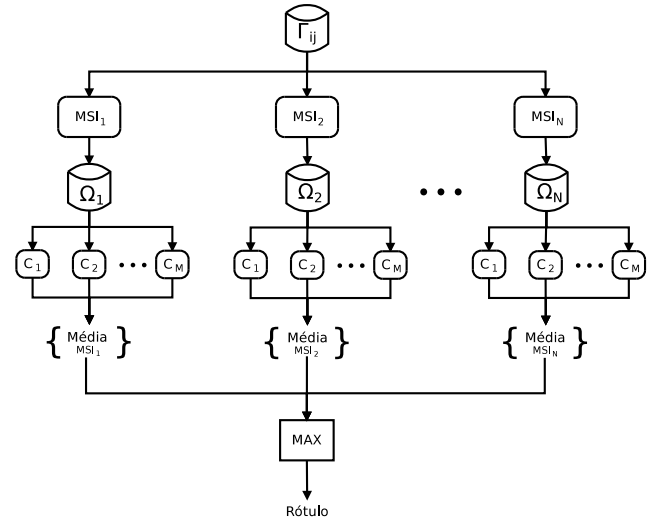


Fig. 3. Módulo de Anotação. Cada base de dados  $\Gamma_{ij}$  é reduzida por N MSIs e as precisões das bases de dados resultantes  $\Omega_i$  são calculadas como a média dos M classificadores  $c_m$ . O rótulo é o MSI com a maior média de precisão.

Para minimizar a influência dos classificadores sobre a decisão do melhor MSI, os classificadores envolvidos devem ser diversos e usar diferentes abordagens.

#### IV. CONFIGURAÇÃO DOS EXPERIMENTOS

Nos experimentos, são utilizadas 16 bases de dados, das quais 12 são do repositório *UCI Machine Learning*, 2 são do *LIBSVM*, 1 é da *mldata.org* e 1 é do artigo de Bernardó et al. [19] (Tabela I). Para cada base de dados, 50 novas bases são criadas usando o procedimento *Bagging* [20] (Reamostragem), como ilustrado na Fig.1, a fim de aumentar a quantidade de amostras para o treinamento do meta-classificador. Esse procedimento constrói um total de 800 bases de dados  $\Gamma_{i,j}$ , com  $i = 1, 2, \dots, 16$  e  $j = 1, 2, \dots, 50$  selecionando as instâncias aleatoriamente com reposição.

TABELA I. INFORMAÇÕES DAS BASES DE DADOS

Data set	# Instances	# Attributes	# Classes
Balance Scale	625	4	3
Banknotes	1372	4	2
Ecoli	336	7	8
Parkinsons	197	23	2
Sonar	208	60	2
Tic tac toe	958	9	2
Transfusion	748	4	2
Vehicle	946	18	4
Vertebral Column 2C	310	6	2
Vowel	528	10	11
Wall-following Sensor 2	5456	2	4
Wall-following Sensor 4	5456	4	4
Fourclass Scale	862	2	2
German	1000	24	2
Banana	138	2	2
Tao	1888	2	2

O Módulo de Extração de Características (Fig.1) extrai nove medidas. Tais medidas são as mesmas usadas por Caisés et al. [4]:

- **Cardinalidade média de conjuntos locais** mostra instâncias de bordas e se as bordas das classes são

regulares ou irregulares.

- **Pontos isolados** é o número de pontos que são mais próximos a instâncias de outras classes (inimigas) que a instâncias pertencentes a sua própria classe.
- **Dispersão Intergrupos (DP1 e DP2)** determina o grau de proximidade entre grupos.
- **Nível de ruído** informa o nível de ruído a partir da execução do ENN.
- **Desbalanceamento** é o grau de disparidade do número de instância por classes.
- **Número de atributos nominais** é usado para escolher MSIs que trabalham com atributos nominais e eliminar aqueles que possuem performance baixa na presença destes atributos.
- **Número de classes** influencia na performance de classificadores e na complexidade dos algoritmos de treino.
- **Regiões** fornece informações sobre a distribuição dos dados.

Para o Módulo de Anotação, são usados cinco MSIs ( $MSI_1, MSI_2, \dots, MSI_5$ ): C-pruner [16], DROP3 [15], IB3 [14], ICF [17] e ENN-CNN [4] e cinco classificadores: C4.5 (árvore de decisão) [21], *K Nearest Neighbors* - KNN (aprendizado lento) [22], *Sequential Minimal Optimization* - SMO (máquina de vetores de suporte) [23], *Multilayer Perceptron com Backpropagation* - MLP-BP (rede neural) [24] e PART (*Crisp rule*) [25]. Eles representam uma grande variabilidade de abordagens.

As distâncias foram trocadas de Euclidiana para *Heterogeneous Value Difference Metric* - HVDM [26] porque é uma distância heterogênea que trata tanto atributos numéricos como nominais.

Todas as medidas de características foram implementadas usando o *software* Matlab 2012b. Os experimentos para redução da base de dados e classificação foram executados no *software* Keel Tool 2.0. Qualquer classificador poderia ser utilizado para o treinamento do meta-classificador. Nesse trabalho foi escolhida uma Rede Neural (MPL-BP) do *software* Weka 3.6.10 com os seguintes parâmetros: 1 camada escondida com 5 neurônios, taxa de aprendizado de 0.2, momento de 0.1 e os outros parâmetros seguiram os valores padrões do *software*.

## V. RESULTADOS

A tabela II mostra a taxa de classificação correta (precisão) por classificador. A abordagem proposta é comparada com os cinco MSIs apresentados. Os valores são as médias de todas as 800 bases de dados  $\Gamma_{i,j}$ . Destes valores, estão em negrito aqueles que representam as maiores precisões alcançadas por cada classificador.

O método proposto se sobrepõe a todos os MSIs como mostrado na Tabela II e Fig.4. O teste de Friedman mostrou uma diferença significativa com nível de confiança de 90%.

A fim de mostrar onde estão estas diferenças, o teste Wilcoxon *signed-ranks test* foi executado com 90% de nível de confiança. Esse teste mostrou que o método proposto é estatisticamente superior a todos os MSIs analisados.

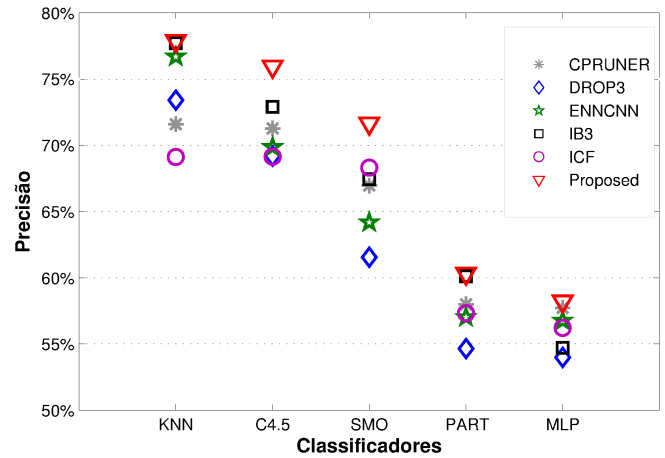


Fig. 4. Comparação de precisões. Esta figura mostra a média de precisões dos MSIs estudados e do método proposto para todas as 800 bases de dados, separados por classificador.

### A. Resultados

Considerando que o objetivo do método proposto é escolher o MSI mais adequado para uma dada base de dados, as saídas do meta-classificador foram analisadas por base. Para cada base é possível calcular a média das precisões dos classificadores da Tabela II e atribuir uma posição (1º, 2º, 3º, 4º ou 5º) para cada MSI. Foi verificado com que frequência o meta-classificador escolhe o 1º, 2º, 3º, 4º ou o 5º melhor MSI para cada base. Esses resultados são mostrados na Fig.5.

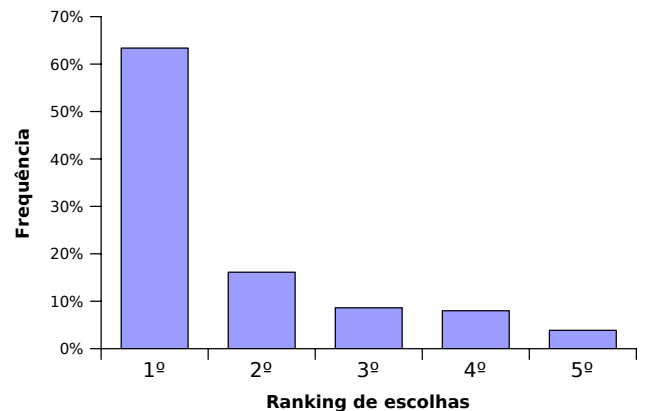


Fig. 5. Taxas de ranking do método proposto. Os MSIs foram ordenados em ranking pela média das precisões dos classificadores para cada base, a fim de mostrar com que frequência o método proposto escolhe os melhores MSIs.

O método proposto escolhe o melhor MSI para 508 bases de dados (63,5%), como mostrado na Fig.5. O segundo melhor MSI é selecionado para 128 bases de dados (16,0%). E apenas para 164 bases de dados (20,5%), o método proposto não seleciona um dos dois melhores MSIs.

## VI. CONCLUSÃO

Neste artigo foi proposto um método para selecionar o melhor MSI para uma dada base de dados baseado em meta-aprendizagem. Também foi proposto o uso de reamostragem para aumentar o número de meta-instâncias e melhorar a

TABELA II. MÉDIAS DE PRECISÃO PARA CADA MSI AVALIADOS POR CINCO CLASSIFICADORES.

	KNN	C4.5	SMO	PART	MLP
<b>CPRUNER</b>	71,59% ± 0,1866	71,27% ± 0,1671	66,96% ± 0,1768	58,00% ± 0,1684	57,73% ± 0,1726
<b>DROP3</b>	73,40% ± 0,1955	69,18% ± 0,1399	61,54% ± 0,1891	54,65% ± 0,1840	53,99% ± 0,1827
<b>ENN-CNN</b>	76,68% ± 0,1819	69,85% ± 0,1620	64,18% ± 0,1835	57,03% ± 0,1910	56,75% ± 0,1797
<b>IB3</b>	77,71% ± 0,1590	72,90% ± 0,1360	67,45% ± 0,1914	60,12% ± 0,1887	54,70% ± 0,1838
<b>ICF</b>	69,11% ± 0,2021	69,14% ± 0,1344	68,31% ± 0,1946	57,32% ± 0,1818	56,22% ± 0,1807
<b>Método Proposto</b>	<b>77,87% ± 0,1284</b>	<b>75,93% ± 0,1096</b>	<b>71,64% ± 0,1565</b>	<b>60,32% ± 0,1721</b>	<b>58,21% ± 0,1891</b>

performance do meta-classificador. As bases de dados reamostradas foram utilizadas na construção da meta-base. As medidas de caracterização de dados foram usadas como descritores e a maior precisão média de classificação foi usada para determinar o melhor MSI e, conseqüentemente, o rótulo. Os descritores e rótulos compõem as meta-instâncias inclusas na meta-base que foi usada para treinar um meta-classificador responsável por selecionar o melhor MSI.

Os resultados obtidos mostraram que o método proposto estatisticamente se sobrepõe a todos os MSIs estudados (C-pruner, DROP3, IB3, ICF e ENN-CNN). Logo, a abordagem proposta é adequada para ser usada automaticamente na fase de pré-processamento das bases de dados, visto que eficientemente seleciona um dos dois melhores MSIs em 79,5% dos casos.

Trabalhos futuros incluem análise de outras medidas de caracterização de dados e suas contribuições para o processo de classificação; considerar a porcentagem de redução para cada MSI no módulo de Anotação e seus relacionamentos com a precisão; utilizar outros métodos de reamostragem, como *Random Subspace*; analisar o melhor algoritmo de classificação para o meta-classificador; e comparar os resultados obtidos com outros trabalhos.

#### AGRADECIMENTOS

Este trabalho foi financiado parcialmente pelo IF Sertão-PE, Governo do Estado de Pernambuco e pelas agências governamentais brasileiras CNPq e CAPES.

#### REFERÊNCIAS

- [1] T. Reinartz, "A unifying view on instance selection," *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 191–210, 2002.
- [2] S.-W. Kim and B. J. Oommen, "A brief taxonomy and ranking of creative prototype reduction schemes," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 232–244, 2003.
- [3] I. Czarnowski and P. Jedrzejowicz, "Instance reduction approach to machine learning and multi-database mining," in *Proceedings of the Scientific Session organized during XXI Fall Meeting of the Polish Information Processing Society, Informatica, ANNALES Universitatis Mariae Curie-Skłodowska, Lublin*. Citeseer, 2006, pp. 60–71.
- [4] Y. Caíses, A. González, E. Leyva, and R. Pérez, "Combining instance selection methods based on data characterization: An approach to increase their effectiveness," *Information Sciences*, vol. 181, no. 20, pp. 4780–4798, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0020025511002878>
- [5] E. Leyva, A. González, and R. Pérez, "Knowledge-based instance selection: A compromise between efficiency and versatility," *Knowledge-Based Systems*, vol. 47, pp. 65–76, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0950705113001172>
- [6] P. Hart, "The condensed nearest neighbor rule (corresp.)," *Information Theory, IEEE Transactions on*, vol. 14, no. 3, pp. 515–516, 1968.
- [7] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 3, pp. 417–435, 2012.
- [8] G. Gates, "The reduced nearest neighbor rule (corresp.)," *Information Theory, IEEE Transactions on*, vol. 18, no. 3, pp. 431–433, 1972.
- [9] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour, "An algorithm for a selective nearest neighbor decision rule (corresp.)," *Information Theory, IEEE Transactions on*, vol. 21, no. 6, pp. 665–669, 1975.
- [10] J. C. Riquelme, J. S. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition*, vol. 36, no. 4, pp. 1009–1018, 2003.
- [11] B. Dasarathy, "Minimal consistent set (mcs) identification for optimal nearest neighbor decision systems design," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 24, no. 3, pp. 511–517, 1994.
- [12] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-2, no. 3, pp. 408–421, 1972.
- [13] "An experiment with the edited nearest-neighbor rule," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-6, no. 6, pp. 448–452, 1976.
- [14] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Machine learning*, vol. 6, no. 1, pp. 37–66, 1991.
- [15] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine learning*, vol. 38, no. 3, pp. 257–286, 2000.
- [16] K.-P. Zhao, S.-G. Zhou, J.-H. Guan, and A.-Y. Zhou, "C-pruner: An improved instance pruning algorithm," in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 1. IEEE, 2003, pp. 94–99.
- [17] H. Brighton and C. Mellish, "Advances in instance selection for instance-based learning algorithms," *Data mining and knowledge discovery*, vol. 6, no. 2, pp. 153–172, 2002.
- [18] N. Jankowski and M. Grochowski, "Comparison of instances selection algorithms i. algorithms survey," in *Artificial Intelligence and Soft Computing - ICAISC 2004*, ser. Lecture Notes in Computer Science, L. Rutkowski, J. Siekmann, R. Tadeusiewicz, and L. Zadeh, Eds. Springer Berlin Heidelberg, 2004, vol. 3070, pp. 598–603.
- [19] E. Bernadó, X. Llorca, and J. M. Garrell, "Xcs and gale: A comparative study of two learning classifier systems on data mining," in *Advances in learning classifier systems*. Springer, 2002, pp. 115–132.
- [20] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [21] J. R. Quinlan, *C4.5: programs for machine learning*. Morgan Kaufmann, 1993, vol. 1.
- [22] T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.
- [23] J. Platt et al., "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.
- [24] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representation by back propagation," *Parallel distributed processing: exploration in the microstructure of cognition*, vol. 1, 1986.
- [25] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," 1998.
- [26] D. R. Wilson and T. R. Martinez, "Improved heterogeneous distance functions," *arXiv preprint cs/9701101*, 1997.