

# Protótipos Autogerados para Tarefas de Classificação com Seleção Dinâmica de Classificadores

Alberto Manastarla\* e Leandro A. Silva†

Programa de Pós-Graduação em Engenharia Elétrica e Computação

Universidade Presbiteriana Mackenzie

São Paulo, Brasil

\*manastarla@hotmail.com, †leandroaugusto.silva@mackenzie.br

**Resumo**—Este artigo apresenta um novo *framework* para métodos baseados em Seleção Dinâmica de Classificadores DS (*Dynamic Selection*), que combina técnicas de seleção de protótipos e a utilização de metainformações para a tomada de decisão de um classificador dinâmico versátil, capaz de operar com diferentes algoritmos baseados em instâncias ou modelos, visando melhorar a precisão e eficiência na seleção dos classificadores mais competentes para determinar a classe de uma instância em análise na fase de generalização em problemas de classificação em diversos domínios de aplicação. O *framework* utiliza protótipos representativos para formar o conjunto de seleção dinâmico, permitindo que o algoritmo KNN defina regiões de competência com menor demanda computacional. Os resultados experimentais, baseados em 24 conjuntos de dados bem conhecidos da literatura, indicaram que o *framework* proposto, em particular, a variação SGP-DES-K baseada no algoritmo KNN, obteve um desempenho superior em comparação com outras técnicas de seleção dinâmicas e estáticas conhecidas na literatura, evidenciando a eficácia do *framework* proposto para lidar com diferentes desafios em problemas de classificação em diversos domínios de aplicação.

**Index Terms**—Seleção de Protótipos, Redução de Dados, Seleção Dinâmica de Classificadores, Seleção de *Ensemble*

## I. INTRODUÇÃO

Os sistemas de Aprendizado de Máquina são amplamente utilizados para resolver problemas complexos de classificação em diversos domínios de aplicação. Sistemas baseados em conjuntos, que combinam múltiplos classificadores base, mostram melhor desempenho em comparação com classificadores individuais [22]. A Seleção Dinâmica (DS) tem atraído atenção devido à sua capacidade de selecionar e combinar classificadores base específicos para cada instância de teste [6].

A DS é dividida em seleção dinâmica de um único classificador DCS (*Dynamic Classifier Selection*) e seleção dinâmica de um conjunto de classificadores DES (*Dynamic Ensemble Selection*). O DCS seleciona um único classificador, enquanto o DES seleciona um subconjunto de classificadores e combina suas saídas [6]. No entanto, as técnicas atuais de DS não

consideram adequadamente diferentes cenários, resultando em comportamento inesperado do sistema [21].

Neste artigo, propomos um *framework* que aplica técnicas de redução de dados e utiliza metainformações para avaliar os classificadores base mais competentes para dada instância em análise. A abordagem melhora significativamente a precisão e eficiência da fase de generalização, ao mesmo tempo em que reduz a complexidade computacional e lida com conjuntos de dados desbalanceados e com diferentes níveis de densidade ou esparsidade [21].

Nosso *framework* utiliza protótipos representativos para formar um conjunto dinâmico de seleção, permitindo que o algoritmo KNN (*K-Nearest Neighbors*) defina regiões de competência com menor demanda computacional [6]. O classificador dinâmico proposto pode operar com diferentes algoritmos de classificação para escolher os melhores classificadores do *pool* com base nas metainformações extraídas do conjunto dinâmico de seleção DSEL (*Dynamic Selection Dataset*) [6].

Em resumo, o *framework* proposto combina técnicas de seleção de protótipos e um classificador dinâmico versátil que toma suas decisões com base em metainformações provenientes dos protótipos supostamente mais representativos do conjunto de treinamento e que pode operar com diferentes algoritmos de classificação. Dessa forma, nossa abordagem busca melhorar, de forma significativa, a precisão e eficiência da fase de generalização, lidando com os desafios e questões de pesquisa em aberto no campo da seleção dinâmica de classificadores.

O restante deste artigo está organizado da seguinte forma: a seção 2 fornece uma breve explicação das técnicas de redução e Seleção Dinâmica de Classificadores, enquanto a seção 3 descreve o *framework* proposto. A metodologia e os resultados experimentais são elucidados nas seções 4 e 5. Na última seção, são apresentadas as conclusões.

Este estudo foi financiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código Financeiro 001.

## II. TÉCNICAS DE REDUÇÃO E SELEÇÃO DINÂMICA DE CLASSIFICADORES

Para definir o problema de seleção dinâmica de classificadores DS (*Dynamic Selection*) de forma matemática, considere os seguintes elementos:

$C = c_1, c_2, \dots, c_M$  é o conjunto de  $M$  classificadores base.  $\mathbf{x}_j$  é uma instância de teste com rótulo de classe desconhecido.  $\theta_j = \mathbf{x}_1, \dots, \mathbf{x}_k$  é a região de competência de  $\mathbf{x}_j$ , em que  $\mathbf{x}_k$  é uma instância pertencente a  $\theta_j$ .  $\delta_{i,j}$  é a competência estimada do classificador base  $c_i$  para a classificação de  $\mathbf{x}_j$ .

O objetivo do problema de seleção dinâmica de classificadores é encontrar o classificador  $c^* \in C$  com maior competência  $\delta_{i,j}$  para classificar a instância de teste  $\mathbf{x}_j$ , considerando a região de competência  $\theta_j$ , eq. 1:

$$c^* = \arg \max_{c_i \in C} \delta_{i,j}(\mathbf{x}_j, \theta_j, c_i) \quad (1)$$

A competência  $\delta_{i,j}$  pode ser estimada de várias formas, por exemplo, pela taxa de acerto local, pela distância entre a instância de teste e a fronteira de decisão ou pela diversidade entre os classificadores na região de competência. A escolha da medida de competência e a definição da região de competência variam de acordo com as técnicas e algoritmos específicos de DS.

No caso do objetivo do problema de redução de dados utilizando-se a seleção de protótipos, PS (*Prototype Selection*) é uma abordagem utilizada em aprendizado de máquina para reduzir o tamanho do conjunto de treinamento, mantendo ou melhorando a precisão do modelo de classificação. Essa redução é alcançada selecionando-se um subconjunto representativo de instâncias chamado conjunto de protótipos ( $P$ ), que é utilizado no lugar do conjunto de treinamento original ( $T$ ) durante o treinamento e a classificação de instâncias.

De forma matemática, podemos definir o problema de seleção de protótipos pela Eqs. 2 e 3, definidas como:

$$P^* = \arg \min_{P \subseteq TR} |P| \quad (2)$$

Sujeito a:

$$E(P) \leq E(T) \quad (3)$$

$P^*$  é o conjunto de protótipos ótimo,  $|P|$  é o tamanho do conjunto de protótipos,  $E(P)$  é o erro de classificação obtido ao se utilizar o conjunto de protótipos  $P$ , e  $E(T)$  é o erro de classificação obtido ao se utilizar o conjunto de treinamento original  $T$ . O objetivo é encontrar o conjunto de protótipos  $P^*$  de tamanho mínimo, de modo que o erro de classificação seja igual ou inferior ao erro obtido ao utilizar o conjunto de treinamento original  $T$ .

As técnicas de redução de instâncias no contexto de seleção dinâmica podem desempenhar dois objetivos importantes para melhorar a eficiência da estimativa da competência dos classificadores base para métodos DS: reduzir ou eliminar instâncias problemáticas para o processo de aprendizado ou predição dos métodos DS, ou seja, exigir menos poder computacional

e atuar para minimizar certas condições encontradas nos dados, a fim de diminuir a complexidade dos dados para atingir um desempenho superior tanto em termos de tempo de processamento quanto de precisão em diferentes problemas de classificação [20].

As técnicas de redução de instâncias se baseiam em duas abordagens diferentes: seleção de instâncias [13] e geração de protótipos [23]. A seleção de instâncias busca encontrar o melhor subconjunto de instâncias do conjunto de treinamento. Por outro lado, a geração de protótipos cria novas instâncias, chamadas protótipos, para representar o conjunto de treinamento original.

Os métodos de seleção de instâncias, também conhecidos como seleção de protótipos (PS) [13], mostraram sucesso ao enfrentar desafios de métodos de aprendizado baseados em instâncias, como complexidade computacional, requisitos de armazenamento e tolerância a ruídos. Os métodos de PS podem ser divididos em três grupos: métodos de condensação, edição e híbridos, cada um com suas próprias forças e fraquezas.

Na literatura, uma variedade de métodos visa selecionar os protótipos mais relevantes de um conjunto de treinamento. Essas propostas aparecem em diferentes áreas de aprendizado de máquina, como aprendizado desbalanceado, classificação de uma classe, regressão [2], big data [3] e aprendizado baseado em cluster [9].

O aprendizado de *ensemble* tem como objetivo superar a precisão de uma tarefa de classificação combinando classificadores. [21] mostrou que os métodos de PS podem melhorar a precisão das técnicas de seleção dinâmica de um conjunto de classificadores DES (*Dynamic Ensemble Selection*), em que os classificadores base são selecionados, em tempo real, de acordo com cada nova instância de teste a ser classificada.

Para obter mais informações ou revisões sobre métodos de seleção de protótipos, os artigos [13], [16] e [24] podem ser consultados. A seção a seguir descreve o *framework* proposto e sua integração com um método de seleção de protótipo híbrido para superar algumas das desvantagens vistas em métodos PS baseados em condensação ou edição.

## III. PROPOSTA *Self-Generating Prototype Dynamic Ensemble Selection*

O *framework* proposto, fig. 1, chamado *Self-Generating Prototype Dynamic Ensemble Selection* (SGP-DES), integra o algoritmo *Self-Generating Prototype Entropy* (SGPE) em um processo de seleção dinâmica de classificadores. Esse *framework* opera em dois modos: *Model-based* (baseado em modelos) e *Instance-based* (usando o conjunto de dados como modelo para predição). A estrutura é dividida em três fases: (1) geração, (2) treinamento e (3) generalização. As fases (1) e (2) são executadas offline, enquanto a fase de generalização ocorre em tempo de execução, conforme demonstrado na fig. 1.

Na fase de geração, um *pool* de classificadores base  $C$  é criado usando-se uma parte pré-definida do conjunto de treinamento original. O conjunto de seleção dinâmica (*DSEL*)

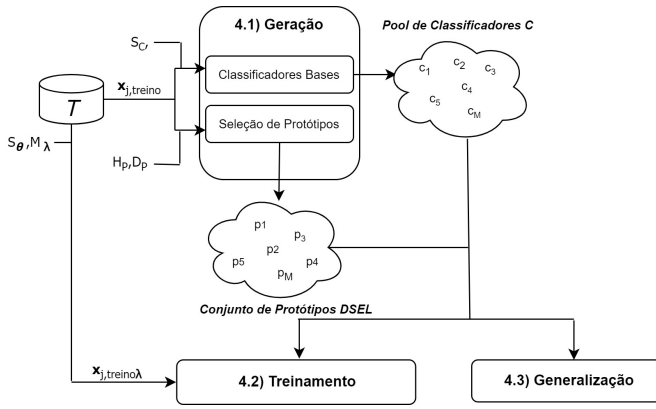


Figura 1. *framework* Proposto - Fase de geração do *pool* de classificadores base e seleção de protótipos para formação do conjunto de seleção dinâmica *DSEL*

é formado por meio do processo de seleção de protótipos com a ajuda do algoritmo SGPE. Esse método é responsável por calcular a menor distância entre um protótipo gerado e uma instância do conjunto de treinamento, retornando um conjunto reduzido de instâncias que preserva a distribuição de classes e propriedades do conjunto original.

Na fase de treinamento, metainformações são extraídas do conjunto *DSEL* e são utilizadas para formar o conjunto *DSEL\**, usado para otimizar o classificador dinâmico. Dependendo do modo escolhido, (*Model-based* ou *Instance-based*), o classificador dinâmico será treinado usando metainformações ou consultará o conjunto *DSEL\** em tempo de execução durante a fase de generalização.

A última etapa é a generalização, em que novas instâncias de teste são utilizadas no processo preditivo. Primeiramente, é definida a região de competência da instância de teste com base no conjunto *DSEL*. As metainformações da instância são extraídas em relação à região de competência local. Em seguida, dependendo do modo escolhido, as metainformações são usadas para selecionar os classificadores base mais competentes (*Model-based*) ou determinar os classificadores base mais competentes com base nas metainformações extraídas da instância em análise (*Instance-based*).

Na Fig. 1, os hiperparâmetros que o SGP-DES recebe são:  $S_C$ , que define o tamanho do *pool* de classificadores base (nesse caso, o número de perceptrons a ser treinado com base no conjunto de treinamento original); os hiperparâmetros  $H_P$  e  $D_P$  para controlarem o número de protótipos a ser gerado para compor o conjunto dinâmico de seleção (*DSEL*); e os hiperparâmetros  $S_\theta$  e  $M_\lambda$ , que controlam o tamanho da porção  $T$  para o processo de otimização do classificador dinâmico e o modo de operação do classificador dinâmico, que pode ser *Instance-based* ou *Model-based*.

No modo *Instance-based*, o KNN é utilizado para determinar os melhores classificadores de acordo com as metainformações extraídas do conjunto *DSEL* e a região de competência delimitada por instâncias oriundas do conjunto  $T$ , parametrizadas por  $S_\theta$ . Esse processo de delimitação de região

de competência, com base nessas instâncias de treinamento, é usado para extrair metainformações, conforme as tabelas I e II, que serão utilizadas durante o processo de predição na fase de generalização. De acordo com as instâncias em análise, é feita a definição da região de competência pertencente a essas instâncias, a extração de metainformação e, por fim, a avaliação do classificador dinâmico, levando em consideração essas entradas da instância em análise e o processo de treinamento realizado para selecionar os classificadores mais competentes do *pool* de classificadores com base nessas informações.

No modo *Instance-based*, o conjunto *DSEL* é utilizado como modelo para o classificador dinâmico, utilizando o KNN. No entanto, quando o modo escolhido é o *Model-based*, o classificador dinâmico é otimizado por meio do conjunto *DSEL*, gerando um modelo preditivo e sendo utilizado no processo de seleção dinâmica dos classificadores mais competentes na fase de generalização. Neste estudo, o *Random Forest* foi utilizado para esse propósito.

As etapas do SGP-DES estão no pseudocódigo 1, enquanto o processo de geração de protótipos para a formação do *DSEL* pode ser encontrado no pseudocódigo 2.

#### Algoritmo 1: SGP-DES: Fases

```

1 Fase 1: Fase_de_Geração( $T$ )
2 Gerar o conjunto de classificadores base  $C$  usando  $T$ .
3 Aplicar o algoritmo SGPE para formar o conjunto DSEL.
4 Retornar  $C$ , DSEL
5 Fase 2: Fase_de_Treinamento(DSEL,  $T_\lambda$ , modo)
6 Extrair metainformação do conjunto DSEL com base nas
  instâncias  $T_\lambda$  e armazena em DSEL*.
7 se modo == "Baseado_em_Modelo" então
8   | Treinar o classificador dinâmico usando metainformação.
9 fim
10 Retornar Classificador_Dinâmico
11 Fase 3: Generalização(InstânciaTeste, DSEL,
  ClassificadorDinâmico, modo)
12 Definir a região de competência local para a instância de
  teste,  $x_j$ .
13 Comparar a instância de teste com as  $k$  instâncias
  (protótipos) mais próximas do DSEL.
14 Formar a região de competência  $\theta_{j, teste}$ .
15 Extrair metainformação  $v_{i,j}$  com base em cada classificador
  base e armazená-la em  $\theta_j$ .
16 se modo == "Baseado_em_Instância" então
17   | Comparar cada metainformação com as  $k$ 
  metainformações mais próximas armazenadas no
  conjunto DSEL*.
18   | Determinar classificadores base competentes de acordo
  com a vizinhança da metainformação extraída.
19 senão
20   | Classificadores_Competentes  $\leftarrow$ 
  SelecionarClassificadoresCompetentes
  (ClassificadorDinâmico, MetaInformação)
21   |
22   |
23 fim
24 Realizar combinação dos resultados dos classificadores base
  selecionados por meio de votação majoritária ponderada.
25 Prever o rótulo de classe da instância de teste.
26 retorna Previsão_Instância_Teste
    
```

O algoritmo de seleção de protótipos (SGPE) é descrito abaixo:

Tabela I  
MEDIDAS DE COMPLEXIDADE DE DADOS: META-INFORMAÇÕES PARA A FORMAÇÃO DO CONJUNTO *DSEL\**.

Medida	Descrição	Domínio	Objeto	# de Atr.
$f_{alg}$	Score região de competência	Acurácia	$\theta_j$	1
$f_{cond}$	Score região de competência Condicional	Acurácia	$\theta_j$	1
$f_{p_j, c_i}$	Distância dos Protótipos em relação às fronteiras de decisão	Confiança	$d(p_j, c_i)$	1
$f_{dist, inst}$	Distância da Instância em Análise em relação às fronteiras de decisão	Confiança	$x_j$	1
$f_{geral, suporte}$	Suporte região de competência	Probabilístico	$S(\theta_j)$	1
$f_{cond, suporte}$	Suporte região de competência Condicional	Probabilístico	$S(\theta_j, w_j)$	1
$f_{dist, centroide}$	Distância entre Centroides da região de competência e Instância em Análise	Confiança	$d(x_j, C(\theta_j))$	1
$f_{centroide}$	Centroide da região de competência da Instância em Análise	Confiança	$C(\theta_j)$	K

Tabela II  
EQUAÇÕES, DEFINIÇÕES DAS VARIÁVEIS E MEDIDAS

Medida	Equação	Definição das variáveis
$f_{alg}$	$f_{alg} = \frac{1}{K} \sum_{k=1}^K P(w_l   x_k \in w_l, c_i)$	$K$ : número de instâncias na região de competência $w_l$ : classe atribuída à instância $x_k$ pelo classificador $c_i$ $x_k$ : k-ésima instância na região de competência $c_i$ : i-ésimo classificador base
$f_{cond}$	$f_{cond} = \frac{\sum_{x_k \in w_l} P(w_l   x_k, c_i)}{\sum_{k=1}^K P(w_l   x_k, c_i)}$	$w_l$ : classe atribuída à instância $x_k$ pelo classificador $c_i$ $x_k$ : k-ésima instância na região de competência $c_i$ : i-ésimo classificador base
$f_{med, c_i}$	$d_{med}(c_i) = \frac{1}{ \theta_i } \sum_{p_j \in \theta_i} d(p_j, H_i)$	$\theta_i$ : região de competência do classificador $c_i$ $p_j$ : j-ésimo protótipo na região de competência $\theta_i$ $H_i$ : fronteira de decisão do classificador $c_i$
$f_{x_j, c_i}$	$d(x_j, c_i) = \frac{c_i(x_j)}{\ w_i\ }$	$x_j$ : j-ésima instância em análise $c_i$ : i-ésimo classificador base $w_i$ : vetor normal ao hiperplano de separação do classificador $c_i$
$f_{centroide}$	$C_\theta = \frac{1}{ \theta } \sum_{p_j \in \theta} p_j$	$\theta$ : região de competência $p_j$ : j-ésimo protótipo na região de competência $\theta$
$f_{dist, centroide}$	$dist(\bar{p}\theta, x_j) = \sqrt{\sum_i^n i = 1^n (\bar{p}\theta_i - x_{ji})^2}$	$\bar{p}\theta$ : vetor médio dos protótipos da região de competência $\theta$ $x_j$ : j-ésima instância em análise $n$ : número de atributos (ou características) das instâncias
$f_{cond, suporte}$	$ws_{x_j} = \arg, \max_{w_l} P(w_l   x_j, c_i)$	$ws_{x_j}$ : classe de suporte das instâncias $x_j$ para a região de competência $\theta$ $w_l$ : possíveis classes de saída $x_j$ : j-ésima instância em análise $c_i$ : i-ésimo classificador base
$f_{geral, suporte}$	$ws_{\theta_j} = \arg, \max_{w_l} P(w_l   \theta_j, c_i)$	$ws_{\theta_j}$ : suporte da região local de competência $\theta$ $w_l$ : possíveis classes de saída $\theta_j$ : j-ésima instância em análise $c_i$ : i-ésimo classificador base

As definições das variáveis utilizadas no pseudocódigo são:

- TR: Conjunto de treinamento.
- modo: Modo de operação do algoritmo, que pode ser “Model-based” ou outro.
- C: Variável para armazenar o resultado da fase de geração.
- *DSEL*: Conjunto de dados de seleção dinâmica, gerado na fase de geração.
- Classificador\_dinâmico: O classificador dinâmico treinado na fase de treinamento.
- Instância\_teste: Instância de teste para a qual a função de generalização deve fazer uma previsão.
- Região\_competência: Região de competência, que é o subconjunto de *DSEL* relevante para a instância de teste.
- Metainformações: Informações adicionais extraídas da

instância de teste e da região de competência.

- Classificadores\_competentes: Classificadores competentes selecionados com base no modo de operação e nas metainformações.
- Predição: Resultado da votação entre os classificadores competentes para a instância de teste.
- $T$ : Conjunto de treinamento.
- $J_p$ : Janelamento da média móvel.
- $D_p$ : Entropia desvio padrão.
- $H_p$ : Limite de entropia.
- $S$ : Conjunto de subconjuntos.
- $E$ : Entropias.
- $S_L$ : Maior subconjunto de  $S$ .
- $S_1$  e  $S_2$ : Subconjuntos resultantes da divisão de  $S_L$ .
- $E_i$ : Entropia do subconjunto  $S_L$ .

**Algoritmo 2: Self-Generating Prototype Entropy**


---

**Entrada:**  $T, J_p, D_p, H_p$   
**Saída:** Conjunto reduzido  $R$

```

1 Inicializar  $S, E$ 
2  $S = T$ 
3 enquanto gerando protótipos faça
4   Selecionar o maior subconjunto  $S_L$ 
5   Dividir  $S_L$  em  $S_1$  e  $S_2$ 
6   Calcular a entropia  $E_i$  de  $S_L$ 
7   Atualizar  $E$  e  $S$ 
8   Atualizar a média móvel simples (MMS) de acordo com
   janelamento  $J_p$  para  $E$ 
9   se desvio padrão de  $E \leq D_p$  então
10    Parar
11  fim
12 fim
13 Calcular os centroides de todos os subconjuntos em  $S$ 
14 para cada  $S_i$  em  $S$  faça
15   se  $Entropia(S_i) \leq H_p$  então
16    Selecionar instâncias reais próximas aos centroides
17   fim
18 fim
19 retorna Conjunto reduzido  $R$ 

```

---

- MMS: Média móvel simples.
- $R$ : Conjunto reduzido de instâncias selecionadas.

Em resumo, o *framework* SGP-DES proposto utiliza o algoritmo SGPE em conjunto com a seleção dinâmica de classificadores para oferecer uma abordagem eficiente e flexível na redução do conjunto de treinamento e na classificação de novas instâncias. O método é baseado em processos de autogeração de protótipos, avaliação da entropia durante o processo de autogeração e extração de metainformações para otimizar um classificador dinâmico na seleção dos classificadores base mais competentes para uma instância em análise, proporcionando uma solução abrangente e adaptável às diferentes necessidades do problema de classificação.

## IV. METODOLOGIA EXPERIMENTAL

A avaliação experimental da abordagem proposta envolve 24 conjuntos de dados conhecidos do repositório KEEL [11]. As características dos conjuntos de dados, incluindo o número de instâncias, número de atributos e número de classes, são especificadas na Tabela III. Os conjuntos de dados selecionados exibem características variadas para demonstrar a aplicabilidade geral e robustez da abordagem proposta.

Para cada conjunto de dados, os experimentos foram conduzidos usando-se vinte repetições. Para cada repetição, os conjuntos de dados foram divididos usando-se o método *holdout* [14], com 50% para treinamento, 25% para o conjunto de seleção dinâmica, *DSEL*, e 25% para o conjunto de teste,  $G$ . As divisões foram realizadas mantendo-se as probabilidades prévias de cada classe. Para o SGP-DES proposto, 25% dos dados de treinamento foram usados no processo de treinamento do classificador dinâmico.

Para os problemas de classificação de duas classes, o *pool* de classificadores foi composto por 50 Perceptrons gerados,

Tabela III

DESCRIBÇÃO DOS CONJUNTOS DE DADOS EXPERIMENTAIS.

#	Nome	#Ex.	#Atr.	#Cl.
1	Appendicitis	106	7	2
2	Banana	5300	2	2
3	Bupa	345	6	2
4	Contraceptive	1473	9	3
5	Glass	214	10	7
6	Haberman	306	3	2
7	Hayes_Roth	160	4	3
8	Heart	270	13	2
9	Ionosphere	351	33	2
10	Led7digit	500	7	10
11	Marking	6876	13	9
12	Monk2	432	6	2
13	Movement_libras	360	90	15
14	Phoneme	5404	5	2
15	Pima	768	8	2
16	Satimage	6435	36	7
17	Segment	2310	19	7
18	Sonar	208	60	2
19	Specfheart	267	44	2
20	Titanic	2201	3	2
21	Vowel	900	13	11
22	wine	178	13	3
23	Winequality_Red	1599	11	11
24	Winequality_White	4898	11	11

usando-se a técnica de Bagging. Para os problemas multiclases, o *pool* de classificadores foi composto por 50 Perceptrons multiclases. O uso de classificadores lineares Perceptron foi motivado pelos resultados relatados nos artigos [1], [5], [7], [18], que mostram ser possível resolver problemas complexos de classificação não lineares apenas com classificadores lineares [4], mostrando que o *framework* SGP-DES pode resolver problemas de classificação não linear com limites de decisão complexos, usando-se apenas alguns classificadores lineares. Os valores dos hiperparâmetros,  $S_\theta$ ,  $M$ ,  $S$ ,  $H$  e  $D$ , foram fixados em 25,  $[I|M]$ , 0,5, 0,1 e 0,1, respectivamente. Esses hiperparâmetros foram selecionados empiricamente com base em experimentos realizados e no artigo [19].

Acurácias e *ranks* médios foram reportados, com os melhores resultados em negrito. O teste de Friedman [12] foi usado para comparar as técnicas em diversos conjuntos de dados [10]. O teste *post-hoc* de Bonferroni-Dunn foi conduzido para comparações em pares entre os *ranks* de cada técnica [10]. Desempenhos significativamente diferentes foram identificados quando a diferença em *ranks* médios era maior que a diferença crítica.

A abordagem proposta foi comparada com os seguintes métodos: OLA (*Overall Local Accuracy*) [15], um método de seleção dinâmica que estima a precisão local de cada classificador no conjunto *DSEL* usando instâncias vizinhas da instância em análise e selecionando o classificador com a maior precisão local; LCA (*Local Class Accuracy*) [25], que estima a precisão local de cada classificador, considerando a classe da instância em análise e selecionando o classificador com a maior precisão local para essa classe; KNORA-E (*K-Nearest Oracle-Eliminate*) [17], um método de seleção dinâmica baseado em oráculo que seleciona um subconjunto de classificadores, que classificou corretamente as instâncias vizinhas da instância em análise, combinando os votos dos

classificadores selecionados; KNORA-U (*K-Nearest Oracle-Union*) [17], similar ao KNORA-E, mas considera a união dos classificadores que classificaram corretamente as instâncias vizinhas da instância em análise, combinando os votos dos classificadores selecionados; e o método de *ensemble* estático *Random Forest* empregando um número de árvores entre 25 e 200, em intervalos de 25, buscando o melhor desempenho no conjunto de validação e o 1-NN (KNN).

Mais informações desses métodos DS podem ser consultados no trabalho de revisão [8].

## V. RESULTADOS EXPERIMENTAIS

Nesta seção, apresentamos os resultados obtidos a partir dos experimentos realizados para comparar o desempenho do SGP-DES-K com outros métodos de classificação dinâmica e estática, incluindo *Instance-based* (1-NN), *Model-based* (RF), DCS, DES e *ensemble* estático (*Random Forest*). Os resultados estão resumidos na Tabela IV.

Ao analisar a tabela de resultados IV, é possível observar que o SGP-DES-K apresenta um desempenho geral superior aos demais métodos em termos de acurácia média. Especificamente, o SGP-DES-K obteve um valor médio de acurácia de 82,33%, com um desvio padrão de 5,49, enquanto os outros métodos apresentaram valores de acurácia média menores. Porém, o SGP-DES, nessas duas variações, apresenta um desvio padrão maior, indica maior variações dos resultados obtidos em comparação com os demais métodos.

Na tabela V, os resultados do teste de Friedman rejeitaram a hipótese nula de que não há diferença entre os métodos em termos de desempenho sobre a acurácia, com um valor de  $p$  igual a  $4,58e-22$ . Além disso, o teste de *post-hoc* de Bonferroni-Dunn, realizado sobre os resultados de acurácia e utilizando a variação SGP-DES-K como controle, indicou que o desempenho do SGP-DES-K é estatisticamente superior em relação a outros métodos avaliados na maioria dos casos. No entanto, existem algumas exceções, como o método KNORA-E e a outra variação do *framework* proposto, SGP-DES-R, que apresentaram desempenhos equivalentes ao SGP-DES-K em alguns conjunto de dados. Isso pode ser observado na fig. 2, na qual o SGP-DES-K, SGP-DES-R e o KNORA-E estão ligados pelo traço em negrito dentro da diferença crítica ajustada para o teste de *post-hoc* de Bonferroni-Dunn em 1,902 e pela análise de *Win-tie-loss* da tabela IV.

Ademais, a análise do teste de Wilcoxon Signed Rank, apresentada na última linha da tabela IV, indica que o SGP-DES-K supera estatisticamente a maioria dos outros métodos em termos de desempenho, considerando um  $\alpha = 0,05$ . Os métodos que alcançaram um desempenho equivalente ao SGP-DES-K são marcados com “~”, aqueles com desempenho estatisticamente superior são marcados com um “+”, e aqueles com desempenho inferior são marcados com um “-”. Neste estudo, utilizando esses conjuntos de dados, todos os métodos e a variação SGP-DES-R apresentaram um desempenho inferior, sustentado por um valor de  $p$  inferior a 0,05.

Os valores de  $p$  não ajustados e os valores ajustados de  $p$  ( $p_{Bonf}$ ), na tabela V, mostram que as hipóteses nulas de

equivalência de desempenho entre o SGP-DES-K e os métodos LCA, KNN, KNORA-U, *Random Forest* e OLA são rejeitadas, uma vez que seus valores ajustados de  $p$  são menores ou iguais a 0,007143.

Por outro lado, os métodos KNORA-E e SGP-DES-R apresentam valores ajustados de  $p$  maiores que 0,007143, o que significa que não há evidências suficientes para rejeitar a hipótese nula de equivalência de desempenho entre o SGP-DES-K e esses dois métodos. Os resultados obtidos pela aplicação do teste de Bonferroni-Dunn complementam a análise, reforçando que o SGP-DES-K é um método promissor com desempenho superior em comparação aos demais métodos avaliados, exceto para os métodos KNORA-E e SGP-DES-R, em que a diferença de desempenho não é estatisticamente significativa.

Em resumo, a análise dos resultados experimentais indica que a variação do SGP-DES, utilizando o modo *instance-base* (KNN) na operação do classificador dinâmico sobre os dados de metainformação, demonstrou ser mais promissora para a seleção dinâmica de classificadores, utilizando protótipos gerados pelos algoritmos SGPE. Entretanto, é importante considerar que o desempenho do SGP-DES-K pode variar de acordo com as características dos conjuntos de dados, e, em alguns casos, outros métodos de classificação ou outras variações do SGP-DES podem apresentar desempenho equivalente ou superior.

## VI. CONCLUSÕES

A variação SGP-DES-K, baseada no algoritmo KNN para seleção dinâmica dos classificadores mais competentes do *pool* de classificadores, obteve 82,33% de acurácia, com um desvio padrão de 5,49. Em comparação, o segundo melhor método, SGP-DES-R, apresentou uma acurácia média de 80,26%, com um desvio padrão de 5,94. Além disso, o SGP-DES-K obteve o menor *rank* médio entre todos os métodos comparados, com 2,479 e um desvio padrão de 1,933.

Tanto o teste de Wilcoxon Signed Rank quanto o teste de Friedman, com aplicação do procedimento de Bonferroni-Dunn, mostraram que o SGP-DES-K apresentou desempenho estatisticamente superior em relação a todos os outros métodos. No caso do teste de Bonferroni-Dunn, os valores de  $p$  ajustados ficaram abaixo do nível de significância de 0,05 para a maioria dos métodos comparados. Esses resultados fornecem evidências sólidas de que o SGP-DES-K é uma variação promissora do SGP-DES para a seleção dinâmica de classificadores e pode ser considerado uma abordagem eficaz para lidar com diferentes cenários de aplicação. Essa afirmação pode ser confirmada por diversos estudos que investigaram as vantagens do método KNN em comparação com métodos de seleção ou geração de protótipos, conforme descrito inicialmente neste estudo. Em termos de desempenho individual em cada conjunto de dados, o SGP-DES, nas duas variações testadas, obteve a maior acurácia em quinze dos 24 conjuntos de dados, destacando-se a variação SGP-DES-K, com treze melhores acurácias.

Tabela IV

COMPARAÇÃO ENTRE OS MODOS DO CLASSIFICADOR DINÂMICO EM CONJUNTOS DE DADOS PÚBLICOS: *Instance-based* (1-NN), *Model-based* (RF) E OUTROS MÉTODOS DE DCS, DES, *ensemble* ESTÁTICO (*Random Forest*) E O KNN. NA ÚLTIMA LINHA DA TABELA, O TESTE DE *Wilcoxon Signed Rank* É REALIZADO. AS VARIAÇÕES DO SGP-DES COM DESEMPENHO EQUIVALENTE AO SGP-DES-K SÃO MARCADAS COM “~”; AQUELAS COM DESEMPENHO ESTATISTICAMENTE SUPERIOR SÃO MARCADAS COM UM “+”, E AQUELAS COM DESEMPENHO INFERIOR SÃO MARCADAS COM UM “-”.

#	Conjuntos de Dados	SGP-DES-K	SGP-DES-R	OLA	LCA	KNORA-E	KNORA-U	RF	KNN
1	appendicitis	88,66 (4,31)	87,71 (6,44)	91,81 (4,88)	88,89 (6,99)	<b>92,82</b> (7,12)	91,9 (5,67)	84,98 (6,75)	83,07 (6,98)
2	banana	69,05 (8,34)	68,1 (15,33)	65,67 (4,04)	64,79 (3,7)	65,65 (4,15)	65,67 (4,08)	65,19 (0,18)	<b>77,61</b> (6,27)
3	bupa	<b>90,14</b> (10,06)	79,42 (3,46)	62,34 (8,78)	61,72 (7,9)	62,34 (5,17)	63,89 (9,36)	76,52 (4,1)	58,26 (5,16)
4	contraceptive	<b>80,67</b> (2,15)	76,98 (9,04)	53,54 (3,11)	50,48 (1,44)	51,94 (1,72)	53,02 (2,69)	73,49 (2,72)	43,45 (1,73)
5	glass	70,72 (17,16)	<b>75,57</b> (6,79)	54,61 (8,73)	50,05 (4,04)	57,53 (10,5)	53,52 (2,72)	60,06 (8,75)	57,6 (6,3)
6	haberman	<b>90,45</b> (4,19)	86,26 (5,54)	80,41 (5,19)	77,63 (3,15)	80,42 (3,61)	79,39 (4,09)	73,86 (1,01)	62,4 (3,85)
7	hayes-roth	<b>85,63</b> (11,18)	81,88 (8,09)	63,53 (5,85)	47,48 (10,22)	78,24 (7,2)	56,84 (13,44)	51,25 (9,53)	59,38 (5,85)
8	heart	<b>91,11</b> (4,42)	86,67 (5,14)	85,6 (5,46)	83,62 (6,6)	84,41 (3,84)	86 (5,34)	82,96 (6,98)	76,3 (4,79)
9	ionosphere	<b>92,04</b> (3,78)	88,06 (5,39)	89,33 (2,5)	84,75 (4,08)	89,33 (2,88)	89,03 (3,31)	87,48 (2,83)	83,47 (2,19)
10	led7digit	75,8 (3,7)	77 (2,92)	77,04 (3,39)	76,61 (3,85)	<b>78,75</b> (3,29)	77,04 (3,61)	70 (2,92)	70,8 (4,44)
11	marketing	<b>44,32</b> (2,93)	43,93 (2,88)	33,33 (1,58)	33,36 (1,64)	33,3 (1,32)	33,87 (1,49)	41,38 (1,35)	27,78 (1,64)
12	monk-2	<b>93,51</b> (3,83)	89,35 (4,06)	89,17 (2,69)	85,46 (4,25)	89,41 (3,8)	85,94 (4,14)	86,24 (3,04)	79,16 (1,87)
13	movement_libras	65 (3,17)	60,83 (3,01)	58,55 (3,34)	52,61 (5,78)	66,28 (4,46)	58,55 (3,75)	64,17 (1,16)	<b>76,67</b> (4,1)
14	phoneme	84,43 (2,36)	<b>91,48</b> (4,62)	81,5 (1,33)	81,16 (1,21)	81,28 (1,46)	80,92 (1,62)	80,08 (0,67)	79,92 (2,7)
15	pima	<b>90,37</b> (3,93)	88,42 (8,94)	79,69 (2,92)	77,05 (3,74)	80,52 (3,7)	79,69 (3,97)	83,3 (4,56)	67,58 (1,85)
16	satimage	87,18 (4,51)	85,98 (2,47)	87,93 (1,23)	87,44 (0,83)	87,81 (0,95)	87,21 (0,99)	85,41 (0,94)	<b>88,3</b> (0,56)
17	segment	94,07 (1,92)	91,86 (2,56)	94,96 (2,1)	94,31 (1,93)	94,63 (1,89)	93,94 (1,89)	94,78 (1,62)	<b>94,98</b> (0,79)
18	sonar	94,05 (4,2)	87,98 (4,09)	95,75 (5,48)	86,73 (13,21)	<b>97,4</b> (5,66)	94,99 (4,48)	96,53 (4,07)	87,26 (4,1)
19	spectfheart	<b>94,63</b> (5,99)	87,01 (10,14)	81,66 (3,34)	84,54 (0,22)	83,31 (2,22)	84,94 (1,68)	89,01 (0,22)	70,57 (6,61)
20	titanic	<b>88,96</b> (6,85)	88,69 (5,58)	83,91 (1,43)	82,6 (1,85)	83,57 (1,93)	83,33 (1,86)	77,6 (1,71)	78,83 (1,39)
21	vowel	64,44 (7,2)	65,86 (4,23)	62,26 (5,23)	56,2 (2,83)	67,77 (4,58)	55,55 (4,69)	66,67 (5,97)	<b>89,39</b> (3,35)
22	wine	96,43 (2,5)	96,43 (2,5)	94,62 (4,94)	95,79 (3,47)	96,44 (3,25)	<b>97,04</b> (2,94)	<b>97,04</b> (2,08)	95,24 (3,36)
23	winequality-red	<b>75,61</b> (6,62)	74,67 (8,75)	60,89 (3,28)	60,02 (3,39)	61,23 (3,75)	60,76 (3,37)	70,59 (3,7)	50,84 (3,26)
24	winequality-white	<b>68,8</b> (6,45)	66,11 (10,53)	52,56 (1,75)	52,67 (1,75)	52,76 (0,79)	53,59 (1,29)	60,28 (0,81)	40,67 (1,86)
	Acurácia Média	<b>82,33</b> (5,49)	80,26 (5,94)	74,19 (3,86)	71,49 (4,09)	75,71 (3,72)	73,60 (3,85)	75,79 (3,24)	70,81 (3,54)
	Rank Médio	<b>2,479</b> (1,933)	3,3542 (1,895)	4,5 (1,689)	6,333 (1,247)	3,708 (1,701)	4,812 (1,586)	4,729 (2,212)	6,083 (2,782)
	Win-tie-loss	n/a	4-3-17	5-0-19	1-4-19	7-1-16	4-2-18	4-0-20	5-0-19
	Wilcoxon-p	n/a	-(0,010041)	-(0,000172)	-(0,000067)	-(0,002576)	-(0,000193)	-(0,000096)	-(0,001757)

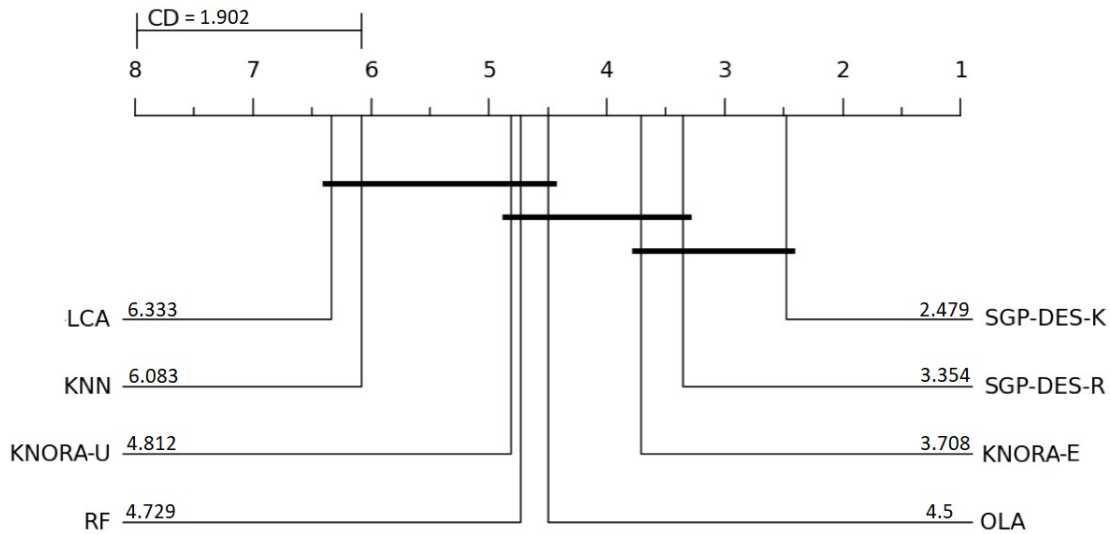


Figura 2. Representação gráfica da classificação média para cada técnica DES nos 24 conjuntos de dados. Para cada técnica, os números na linha principal representam sua classificação média. A diferença crítica (CD) foi calculada usando-se o teste *post-hoc* de Bonferroni-Dunn. Técnicas sem diferença estatística são conectadas por linhas adicionais.

Tabela V

VALOR  $p$  AJUSTADO,  $\alpha = 0,05$  (FRIEDMAN) E BONFERRONI-DUNN

<b>i</b>	<b>Método</b>	<b>não ajustado <math>p</math></b>	<b><math>p_{Bonf}</math></b>
1	LCA	$\leq 0,000001$	$\leq 0,000001$
2	KNN	$\leq 0,000001$	0,000002
3	KNORA-U	0,000967	0,006772
4	<i>Random Forest</i>	0,001463	0,010239
5	OLA	0,004265	0,029853
6	KNORA-E	0,082157	0,575101
7	SGP-DES-R	0,215925	1,511475

É importante observarmos que, embora o SGP-DES-K tenha apresentado um desempenho geral superior, não foi o melhor método em todos os conjuntos de dados. Isso sugere que a seleção da variação do classificador dinâmico mais adequado pode depender das características específicas do problema em questão. No entanto, o desempenho consistente do SGP-DES-K, em uma ampla variedade de conjuntos de dados, demonstra sua aplicabilidade em diversos cenários.

Em conclusão, os resultados experimentais apresentados neste artigo indicam que o SGP-DES-K é um método de classificação dinâmica eficaz, capaz de superar outros métodos de DCS, DES e *ensemble* estático, como o *Random Forest*, em termos de acurácia média e *rank* médio. Além disso, o SGP-DES-K demonstrou ser um método promissor para lidar com diversos problemas de classificação em diferentes domínios.

Estudos futuros podem abordar a aplicação do SGP-DES-K em cenários como conjuntos de dados desbalanceados, onde classes minoritárias são sub-representadas e podem ser críticas para a análise. É importante também avaliar a complexidade computacional do SGP-DES-K em grandes conjuntos de dados e aplicações práticas na busca de otimizações em sua eficiência e robustez em variados contextos.

## REFERÊNCIAS

- Alpaydin, E., Jordan, M.I.: Local linear perceptrons for classification. *IEEE Transactions on Neural Networks* **7**(3), 788–794 (1996)
- Arnaiz-González, Á., Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C.: Instance selection of linear complexity for big data. *Knowledge-Based Systems* **107**, 83–95 (2016)
- Arnaiz-González, Á., Díez-Pastor, J.F., Rodríguez, J.J., García-Osorio, C.: Instance selection of linear complexity for big data. *Knowledge-Based Systems* **107**, 83–95 (2016)
- Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: A deep analysis of the meta-des framework for dynamic selection of ensemble of classifiers. *arXiv preprint arXiv:1509.00825* (2015)
- Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: Meta-des. oracle: Meta-learning and feature selection for dynamic ensemble selection. *Information fusion* **38**, 84–103 (2017)
- Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: Dynamic classifier selection: Recent advances and perspectives. *Information Fusion* **41**, 195–216 (2018)
- Cruz, R.M., Sabourin, R., Cavalcanti, G.D., Ren, T.I.: Meta-des: A dynamic ensemble selection framework using meta-learning. *Pattern recognition* **48**(5), 1925–1935 (2015)
- Cruz, R.M., Zakane, H.H., Sabourin, R., Cavalcanti, G.D.: Dynamic ensemble selection vs k-nn: why and when dynamic selection obtains higher classification performance? In: 2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA). pp. 1–6. *IEEE* (2017)
- Czarnowski, I.: Cluster-based instance selection for machine classification. *Knowledge and Information Systems* **30**, 113–133 (2012)
- Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research* **7**, 1–30 (2006)
- Derrac, J., Garcia, S., Sanchez, L., Herrera, F.: Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult. Valued Logic Soft Comput* **17** (2015)
- Friedman, J.H., Rafsky, L.C.: Multivariate generalizations of the wald-wolfowitz and smirnov two-sample tests. *The Annals of Statistics* pp. 697–717 (1979)
- García, S., Derrac, J., Cano, J., Herrera, F.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE transactions on pattern analysis and machine intelligence* **34**(3), 417–435 (2012)
- Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer (2009)
- Ho, T.K., Hull, J.J., Srikari, S.N.: Decision combination in multiple classifier systems. *IEEE transactions on pattern analysis and machine intelligence* **16**(1), 66–75 (1994)
- Kim, S.W., Oommen, B.J.: A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Analysis & Applications* **6**(3), 232–244 (2003)
- Ko, A.H., Sabourin, R., Britto Jr, A.S.: From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition* **41**(5), 1718–1731 (2008)
- Kuncheva, L.I., Rodriguez, J.J.: Classifier ensembles with a random linear oracle. *IEEE Transactions on Knowledge and Data Engineering* **19**(4), 500–508 (2007)
- Manastarla, A., Silva, L.A.: A self-generating prototype method based on information entropy used for condensing data in classification tasks. In: *International Conference on Intelligent Data Engineering and Automated Learning*. pp. 195–207. Springer (2019)
- Oliveira, D., Cavalcanti, G., Sabourin, R.: Dynamic classifier selection: Recent advances and perspectives. *Information Fusion* **41**, 195–216 (2018)
- Oliveira, D.V., Cavalcanti, G.D., Sabourin, R.: Online pruning of base classifiers for dynamic ensemble selection. *Pattern Recognition* **72**, 44–58 (2017)
- Polikar, R.: Ensemble based systems in decision making. *IEEE Circuits and systems magazine* **6**(3), 21–45 (2006)
- Triguero, I., Derrac, J., Garcia, S., Herrera, F.: A taxonomy and experimental study on prototype generation for nearest neighbor classification. A taxonomy and experimental study on prototype generation for nearest neighbor classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* **42**, 86–100 (2012). <https://doi.org/10.1109/TSMCC.2010.2103939>
- Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. *Machine learning* **38**(3), 257–286 (2000)
- Woods, K., Kegelmeyer, W.P., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. *IEEE transactions on pattern analysis and machine intelligence* **19**(4), 405–410 (1997)