

SISTEMAS DE CLASSIFICA  O DE CONTATOS DE SONAR PASSIVO COM REJEI  O A CLASSES DESCONHECIDAS

GEANCARLO G. M. DA ROCHA*, JO  O BAPTISTA DE OLIVEIRA E SOUZA FILHO*[†]

** Programa de P s-Gradua  o em Engenharia El trica (PPEEL)
Centro Federal de Educa  o Tecnol gica Celso Suckow da Fonseca (CEFET/RJ)
Av. Maracan  229, Bloco E, Sala 205
Rio de Janeiro, RJ, Brasil*

*[†] Departamento de Engenharia Eletr nica e de Computa  o (DEL) – Escola Polit cnica (POLI)
Universidade Federal do Rio de Janeiro (UFRJ)
Av. Athos da Silveira Ramos 149, Bloco H, sala 219 (20)
Rio de Janeiro, RJ, Brasil*

Emails: geancarlo.rocha@gmail.com, jbfilho@poli.ufrj.br

Resumo— O monitoramento de sistemas de sonar passivo   usualmente realizado por operadores treinados com o aux lio de sistemas de apoio   decis o. Neste trabalho apresentaremos uma metodologia de classifica  o de contatos para a qual se realiza a integra  o de um subsistema de classifica  o baseado em redes neurais com outro para a detec  o de classes desconhecidas atrav s de um m todo de *k*-vizinhos mais pr ximos. Nossos experimentos utilizaram dados reais obtidos em raia ac stica e mostraram que a metodologia proposta   acurada, obtendo taxas de reconhecimento de classes conhecidas e desconhecidas de 92,87% e 60,32%, no pior caso; e de 94,35% e 79,87%, para o melhor caso.   tamb m mostrado que o sistema proposto apresenta um baixo custo computacional em fase de opera  o, possuindo a capacidade de opera  o em tempo real.

Palavras-chave— Sonar passivo, sistema de apoio   decis o, detec  o de novidades, reconhecimento de padr es

Abstract— The monitoring of passive sonar systems is usually performed by trained operators assisted by decision support systems. In this paper we propose a methodology for the classification of contacts for which we integrate a neural network classification subsystem to a unknown classes detection subsystem based on a *k*-nearest neighbors method. Our experiments used real data obtained in an acoustic lane and showed that our methodology is accurate, achieving a known/unknown class recognition rates of 92,87%/60,32%, respectively, in the worst-case scenario, and of 94,35%/79,87% in the best case. Additionally, we show that the proposed method has a low computational cost, thus can operate in real-time.

Keywords— Passive sonar, decision support system, novelty detection, pattern recognition.

1 Introdu  o

Uma das principais aplica  es militares do sonar passivo   a detec  o de embarca  es que, em geral,   realizada por operadores treinados, atrav s de equipamentos para a an lise sonora ou visual (atrav s do espectrograma) dos sinais captados, identificando-se a presen a de embarca  es por tons ou linhas espectrais em frequ ncias espec ficas.

Infelizmente, esta tarefa   dif cil em ambientes n o-controlados. H  diversas fontes de ru do, tais como animais e aqueles produzido pela pr pria embarca  o. Adicionalmente, efeitos de propaga  o ac stica, tais como o efeito *doppler*, bem como a propaga  o do sinal de interesse por m ltiplos caminhos em  guas rasas contribuem para uma degrada  o do sinal captado.

Para auxiliar o operador, m todos de processamento de sinais e de reconhecimento de padr es podem ser utilizados para o aux lio   detec  o e   identifica  o de embarca  es. Naturalmente,   poss vel que uma embarca  o detectada n o fa a parte do conjunto de assinaturas ac sticas conhecidas pelo operador de sonar ou pelos sistemas de

apoio   decis o dos quais ele se utiliza. No contexto de sistemas automatizados, caso sejam empregados sistemas de classifica  o sem a capacidade do reconhecimento de novidades, tais casos seriam classificados de forma incorreta.

Este trabalho prop e a identifica  o de novidades em sinais de sonar passivo segundo um procedimento de duas etapas. Na primeira, um sistema de classifica  o escolhe a classe mais prov vel para o sinal captado dentre aquelas conhecidas. Em sequ ncia, verifica-se a ader ncia desse sinal a um conjunto de sinais caracter sticos da classe selecionada, atrav s de uma t cnica de detec  o de novidades. Utilizando uma base de dados que cont m 8 classes de embarca  es fornecida pelo Instituto de Pesquisas da Marinha (IPqM), realizamos experimentos onde o compromisso entre a classifica  o correta de embarca  es conhecidas e a detec  o de embarca  es desconhecidas   avaliado, e comparamos nossa proposta   trabalhos anteriores (Guedes and Souza Filho, 2014). Adicionalmente, apresentaremos medi  es do tempo de execu  o da metodologia proposta, demonstrando a possibilidade de opera  o em tempo real do sistema proposto.

2 Trabalhos relacionados

O problema da detecção de novas classes de embarcações através de sinais de sonar passivo já foi investigado por outros autores. Em (Soares Filho, 2001), tem-se a classificação e a detecção de novidades utilizando redes *multilayer perceptron* (MLP) e redes ART-2 (*adaptive-resonance theory*). Em (Souza Filho, 2007) foram utilizadas redes neurais MLP “especialistas”, isto é, treinadas para discernir uma classe conhecida das outras restantes. Mais recentemente, (Guedes and Souza Filho, 2014) propôs a identificação de novidades por meio da modelagem dos sinais das classes conhecidas através da análise de componentes principais (PCA) e de *kernel* (KPCA), seguindo a motivação intuitiva de que os sinais das classes desconhecidas devem apresentar um erro de reconstrução mais elevado do que aqueles provenientes de classes conhecidas (Hoffmann, 2007). Dentre as abordagens propostas, destacou-se aquela referida como detectores “especializados”, pela qual os dados de cada classe são modelados individualmente.

3 Metodologia proposta

Similar ao detector especializado de (Guedes and Souza Filho, 2014), propomos o uso de modelos de detecção individuais para cada uma das classes conhecidas, porém considerando um grupo de técnicas de detecção de novidades mais amplo (vide Subseção 3.2), e com a arbitração de qual detector utilizar sendo realizada pelo sistema de classificação. Referiremo-nos a esta proposta como detecção de novidades pós-classificação. A Figura 1 ilustra esta abordagem, cuja operação pode ser resumida nos seguintes passos:

1. É recebido um vetor de características proveniente do sistema de pré-processamento de sinais (vide Seção 4).
2. O sistema de classificação determina qual é a classe mais provável dentre as N conhecidas.
3. Seleciona-se o escore produzido pelo detector corresponde à classe arbitrada pelo classificador, o qual é comparado a um limiar determinado previamente.
4. Se o escore for superior a este limiar, o sinal é classificado como desconhecido ou anormal (+1), e a classificação prévia é descartada. Se for inferior, a classificação prévia é mantida, e o sinal é considerado normal (-1).

Este sistema apresenta algumas características interessantes do ponto de vista operacional e de desempenho. Durante sua operação, faz-se necessária somente a avaliação do escore de apenas um detector, reduzindo-se o custo computacional

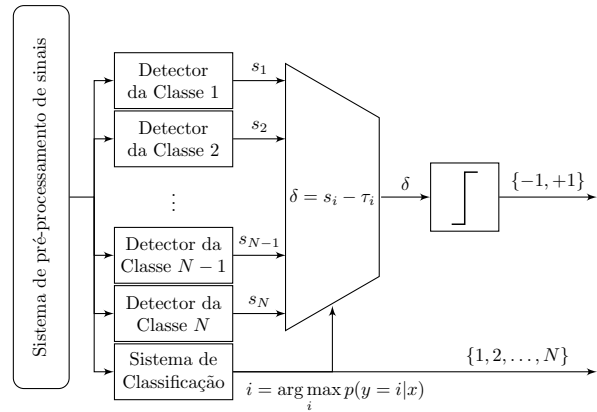


Figura 1: Ilustração da estrutura do detector de novidades pós-classificação.

e o atraso (*delay*) da decisão. É possível uma fácil integração do sistema proposto com sistemas de classificação baseados em técnicas robustas e de eficiência comprovada, tais como as redes MLP (Soares Filho, 2001)(Souza Filho, 2007) ou curvas principais (Fernandez, 2005). Além disso, por conta de sua arquitetura, temos um desacoplamento das fases de classificação e detecção, logo ambas podem ser otimizadas de forma independente, tal como será discutido na Subseção 3.3.

3.1 Sistema de classificação: Rede Neural MLP

Em outros trabalhos (Souza Filho, 2007)(Soares Filho, 2001), as MLPs mostraram resultados expressivos na classificação de sinais semelhantes àqueles envolvidos neste trabalho e, por isso, foram escolhidas para a demonstração do desempenho da metodologia de detecção pós-classificação. Foram consideradas redes de apenas uma camada escondida para a classificação multiclasse, conforme a representação a seguir:

$$\mathbf{x} \xrightarrow{A_1} \mathbf{a}_1 \xrightarrow{\text{ReLU}} \mathbf{z} \xrightarrow{\text{dropout}} \tilde{\mathbf{z}} \xrightarrow{A_2} \mathbf{a}_2 \xrightarrow{\text{softmax}} \hat{\mathbf{y}} \quad (1)$$

onde A_1 e A_2 representam transformações afins, bem como foram utilizadas funções de ativação retificadoras lineares (ReLU) (Glorot et al., 2011).

Para a regularização e o combate ao *overfitting* no treinamento das redes, utilizamos a parada antecipada (*early-stopping*), e a técnica de *dropout* (Srivastava et al., 2014) na camada escondida. Consideramos como função de custo a entropia cruzada, e a minimizamos através do método *rmsprop* (Tieleman and Hinton, 2012). Tendo em vista a arquitetura de rede e o método de treinamento escolhidos, restou a definição por validação cruzada de dois hiperparâmetros: a dimensão da camada escondida N_H ($\mathbf{z}, \tilde{\mathbf{z}} \in \mathbb{R}^{N_H}$), e a probabilidade $p \in [0, 1]$ de desativação das unidades considerada no processo de *dropout*.

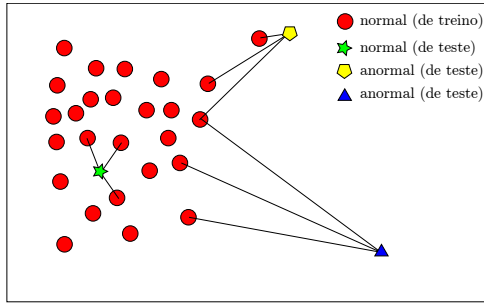


Figura 2: Ilustração do método k -NN.

3.2 Detector de novidades: k -NN

Diferentes modelos ou técnicas de detecção de novidades (Pimentel et al., 2014) podem ser implementadas nos detectores ilustrados na Figura 1. Na detecção de novidades, temos como objetivo a criação de funções de “escore” ($f: \mathbb{R}^D \rightarrow \mathbb{R}$), capazes de atribuir uma pontuação que indique o quão dissimilar um vetor de características arbitrário ($\mathbf{x}_* \in \mathbb{R}^D$) é em relação a um conjunto de dados conhecido ($\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N \equiv \mathbf{X} \in \mathbb{R}^{D \times N}$). Portanto, valores de escore altos indicam uma maior dissimilaridade perante aquilo que é considerado como conhecido. Assim, é possível definir um limiar τ de forma que se $f(\mathbf{x}_*) > \tau$, o vetor de características é apontado como anormal ou novidade. Claramente, a variação deste limiar fornece uma curva similar à curva ROC (*receiver operating characteristic*) (Fawcett, 2006), relacionando o compromisso entre a detecção de novidades e o reconhecimento das classes conhecidas. Semelhante a problemas de classificação binária, podemos considerar a área sob esta curva (AUC) como uma medida de desempenho.

Neste trabalho focamos nossa atenção em uma técnica que consiste no cálculo da mediana da distância euclidiana entre o vetor de características sob avaliação e os k -vizinhos mais próximos (k -nearest neighbors, k -NN) pertencentes ao conjunto de dados conhecidos, conforme ilustrado pela Figura 2. Neste exemplo, temos $k = 3$, e os eventos vermelhos (círculos) representam os dados da classe conhecida. Intuitivamente, o caso do evento de teste verde (estrela), para o qual este valor é baixo, é tido como similar à classe conhecida. O uso da mediana visa combater a presença de *outliers* na vizinhança de um evento de teste, como se tem no caso do evento amarelo (pentágono).

O pseudocódigo expresso no Algoritmo 1 apresenta uma implementação possível para este detector. Primeiro, é calculada a distância entre o vetor de características sob avaliação ($\mathbf{x}_*$) e os dados pertencentes à classe normal ($\mathbf{x}_i$). Após a determinação do vetor de distâncias $\mathbf{d}$, a função **select** seleciona as k menores distâncias, produzindo uma lista s , cuja mediana é calculada, fornecendo o valor do escore. É importante notar que as

funções **select** e **median** podem ser implementadas por um algoritmo de complexidade linear em pior caso, tal como o **introsselect** (Musser, 1997).

Algoritmo 1 Função de escore baseada em k -NN.

```

1: função ( $\mathbf{x}_* \in \mathbb{R}^D$ ,  $\mathbf{X} \in \mathbb{R}^{D \times N}$ ,  $k \in \mathbb{N}_+$ )
2:   para  $i=1:N$  faça                                 $\triangleright \mathcal{O}(ND)$ 
3:      $d_i \leftarrow \|\mathbf{x}_* - \mathbf{x}_i\|^2$                      $\triangleright \mathcal{O}(D)$ 
4:    $\mathbf{s} \leftarrow \text{select}(\mathbf{d}, k)$                          $\triangleright \mathcal{O}(N)$ 
5:    $r_* \leftarrow \text{median}(\mathbf{s})$                            $\triangleright \mathcal{O}(k)$ 
6:   retornar  $r_*$ 

```

3.3 Hiperparâmetros e validação cruzada

Para a sintonia dos hiperparâmetros do sistema proposto, foi considerada uma otimização independente do classificador e do detector de novidades, segundo as seguintes figuras de mérito: a eficiência média de classificação e a área sob a curva ROC (AUC), utilizando-se da técnica de k -fold aninhado (Japkowicz and Shah, 2011).

A seleção dos hiperparâmetros de cada detector (k no caso da técnica de detecção proposta) foi realizada de forma individual por classe. Assim, num cenário hipotético de 4 classes de embarcações conhecidas, rotuladas como A, B, C e D, para o ajuste do detector da classe A, uma das medidas de desempenho consistiria na média das áreas sob as diferentes curvas ROC associadas as três situações binárias de classificação da classe A *versus* as outras classes conhecidas, isto é:

$$\text{AUC}(A) = \frac{1}{3} \sum_{L \in \{B, C, D\}} \text{AUC}(A \times L) \quad (2)$$

Mais especificamente, devido ao uso do k -fold para a seleção dos detectores, adotamos a média obtida para os diferentes *folds*, dada por:

$$\overline{\text{AUC}}(A) = \frac{1}{k} \sum_{i=1}^k \text{AUC}_i(A) \quad (3)$$

Medidas análogas são utilizadas para a sintonia dos detectores das outras classes conhecidas (B, C e D).

Após a escolha dos detectores de cada uma das classes conhecidas, utilizamos os dados de validação para definirmos os limiares τ apropriados para cada taxa de reconhecimento (ou retenção) das classes conhecidas. Por exemplo, para uma taxa de retenção de 95% da classe A, foi calculado o 95º percentil dos escores para seus dados de validação. Dado o uso do k -fold, foi considerada a média dos limiares obtidos para cada *fold*.

4 Pré-processamento e base de dados

A base de dados utilizada para as simulações foi fornecida pelo Instituto de Pesquisas da Marinha

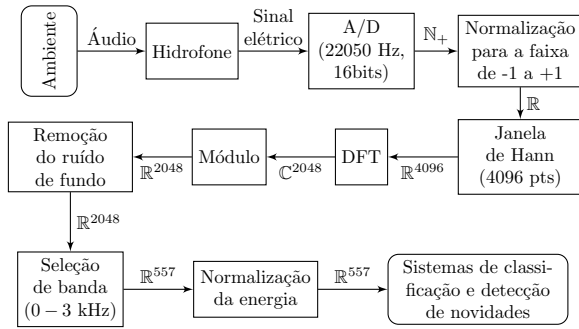


Figura 3: Ilustração da cadeia de pré-processamento utilizada.

(IPqM). Ela é constituída pelo ruído irradiado por 25 navios, pertencentes a 8 classes diferentes de embarcações, durante 263 corridas realizadas na raia acústica da Marinha do Brasil, localizada em Arraial do Cabo. Durante cada corrida, uma embarcação percorreu a raia, e suas condições operacionais foram mantidas constantes. O ruído emitido pelas embarcações foi captado por um hidrofone localizado junto ao fundo do assoalho submarino da raia acústica e processado de acordo com a cadeia de pré-processamento proposta em (Souza Filho, 2007), ilustrada na Figura 3. Seu objetivo é extrair o espectro na faixa de frequência de 0 a aproximadamente 3 kHz, que está diretamente associada ao ruído produzido pelo maquinário em operação nas embarcações (Souza Filho, 2007).

Como resultado desta cadeia, os sistemas de classificação e detecção de novidades recebem, aproximadamente, a cada 186 ms, um vetor de características de norma unitária com 557 dimensões. O total de janelas resultantes é de 2432, 3432, 4797, 3072, 7075, 2934, 2143 e 3392 para as classes de embarcações rotuladas de A a H.

5 Experimentos e resultados

Os experimentos conduzidos tiveram como objetivo demonstrar a eficácia do detector pós-classificação proposto, bem como avaliar o custo computacional envolvido, visando demonstrar a viabilidade de sua operação em tempo real.

O desempenho do método proposto foi comparado ao de detectores especializados (Guedes and Souza Filho, 2014), e, além da técnica de detecção por k -NN, também consideramos as técnicas PCA e KPCA, conforme o trabalho supracitado. Estes experimentos foram conduzidos em um esquema de validação cruzada de 10-fold aninhado (*nested k-fold cross-validation*) (Japkowicz and Shah, 2011), isto é, com dois níveis de k -fold sendo realizados. Para cada *fold* do nível superior (ou externo), foram obtidas estimativas de desempenho de teste de sistemas ajustados por um procedimento de 10-fold inferior (ou interno).

No ajuste dos hiperparâmetros das redes MLP do sistemas de classificação para a topologia proposta, consideramos redes neurais com 5, 10, 15, 25, 50, 75 ou 100 unidades na camada escondida, treinando-as sem e com o uso da técnica de *dropout*¹. Após realizar a escolha dos melhores hiperparâmetros pelo valor médio das acurácias médias dos 10 *folds* internos, utilizamos as respectivas 10 redes geradas para formar um *ensemble* com decisão por maioria de votos para a classificação dos dados de teste pertencentes ao correspondente *fold* externo.

Os valores considerados para a dimensão do subespaço nos detectores PCA e KPCA foram escolhidos de forma a se obter 10, 15, 20, ..., 90, 95 ou 99% de variância explicada. Para o detector KPCA, utilizamos o *kernel* RBF, com 0,01; 0,1; 0,5 ou 1 quanto aos valores de precisão possíveis. O número de vizinhos (k) do detector k -NN pôde assumir os valores 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 40 ou 50. Para cada modelo de detector (PCA, KPCA ou k -NN), escolhemos os melhores hiperparâmetros, conforme procedimento descrito na Subseção 3.3, e estimamos os limiares correspondentes a pontos de operação com reconhecimento de 5, 10, 15, ..., 90 e 95% dos dados de classes conhecidas.

Foram criadas 6 “situações”, onde supomos certas classes de nossa base de dados como conhecidas, enquanto que as restantes foram supostas desconhecidas. Estas situações consideramos os seguintes conjuntos de classes conhecidas: BC, BCE, BCEH, BCEFH, ABCEFH, ABCEFGH. Para a avaliação dos sistemas criados, consideramos as probabilidades de acerto de suas predições, levantadas empiricamente. Para os dados das classes supostamente conhecidas, consideramos uma predição correta quando o classificador aponta sua classe corretamente, e o detector de novidades correspondente não o aponta como desconhecido. Já para os dados das classes desconhecidas, basta que o detector da classe apontada pelo classificador o identifique como uma novidade.

A Figura 4 apresenta curvas comparando o desempenho das metodologias citadas para diferentes pontos de operação. As curvas dos sistemas com detectores especializados são apresentadas em linha pontilhada, enquanto que aquelas relativas a detectores pós-classificação em linha cheia. Ao se comparar as duas metodologias, verifica-se que a pós-classificação é sempre igual ou superior a de detectores especializados para cada um dos modelos considerados. Em geral, com detectores k -NN, a diferença é pequena, porém os detectores PCA e KPCA especializados apresentam uma baixa taxa de acerto das classes conhecidas. Isto é justificado pelo critério de decisão por erro mínimo do detector especializado suben-

¹Neste caso, com um hiperparâmetro de probabilidade igual a 0,5.

tender uma etapa de classificação por parte do detector escolhido, apesar de ele não ser necessariamente apropriado para realizar esta tarefa. A Figura 5 mostra a acurácia média de classificação das *ensembles* de MLPs utilizada no detector pós-classificação e dos classificadores implícitos aos sistemas baseados em detectores especializados. Fato importante é que os detectores *k*-NN também resultam em uma classificação de bom desempenho e, por este motivo, os sistemas com detectores especializados baseados neste modelo também apresentaram bons resultados. Ainda assim, os detectores pós-classificação são mais interessantes, pois os detectores especializados demandam a avaliação de escores para todas as classes conhecidas, um procedimento de escalabilidade ruim se considerarmos um número elevado de classes.

Por fim, com o objetivo de demonstrar a capacidade de operação da metodologia proposta em tempo real, apresentamos os resultados de uma avaliação do custo computacional dos detectores *k*-NN e das redes MLP na Figura 6. As curvas apresentam o tempo médio de 1000 execuções para uma janela espectral (557 dimensões) dos métodos em um computador pessoal com processador Intel Core i7 950 3.06 GHz, memória DDR3 1066, utilizando 1 núcleo. As implementações foram codificadas em Fortran 95, compiladas com o compilador gfortran 4.8.4 e utilizaram a biblioteca OpenBLAS (Xianyi et al., 2012). Para o método *k*-NN, considerou-se um conjunto de dados conhecidos de 2048 amostras, mas os resultados para outros quantitativos podem ser extrapolados linearmente. Verifica-se que, independente de *k*, sua execução leva aproximadamente 2ms. Já para as redes MLP, são apresentadas diferentes curvas com um número de classes na saída de 1, 10, 50 ou 100, e verifica-se que a diferença é mínima para esta faixa, atingindo, em pior caso, aproximadamente 0,7 ms para uma rede com 500 unidades na camada escondida (5x maior do que aquela considerada neste trabalho). Considerando que o sistema de pré-processamento envolvido produz uma janela espectral para a classificação com período de aproximadamente 186 ms, e que a metodologia proposta envolve a avaliação de um *ensemble* com 10 MLPs e um detector *k*-NN, podemos estimar, de forma aproximada, o tempo total de processamento como $10 \times 0,7 + 2 = 9$ ms, sugerindo a possibilidade de operação em tempo real do sistema proposto para a plataforma de implementação considerada.

6 Conclusões

A metodologia de detecção de novidades proposta neste trabalho se mostrou melhor, tanto do ponto de vista de acurácia quanto de esforço computacional e escalabilidade, do que a alternativa explorada em trabalhos anteriores. O método *k*-NN,

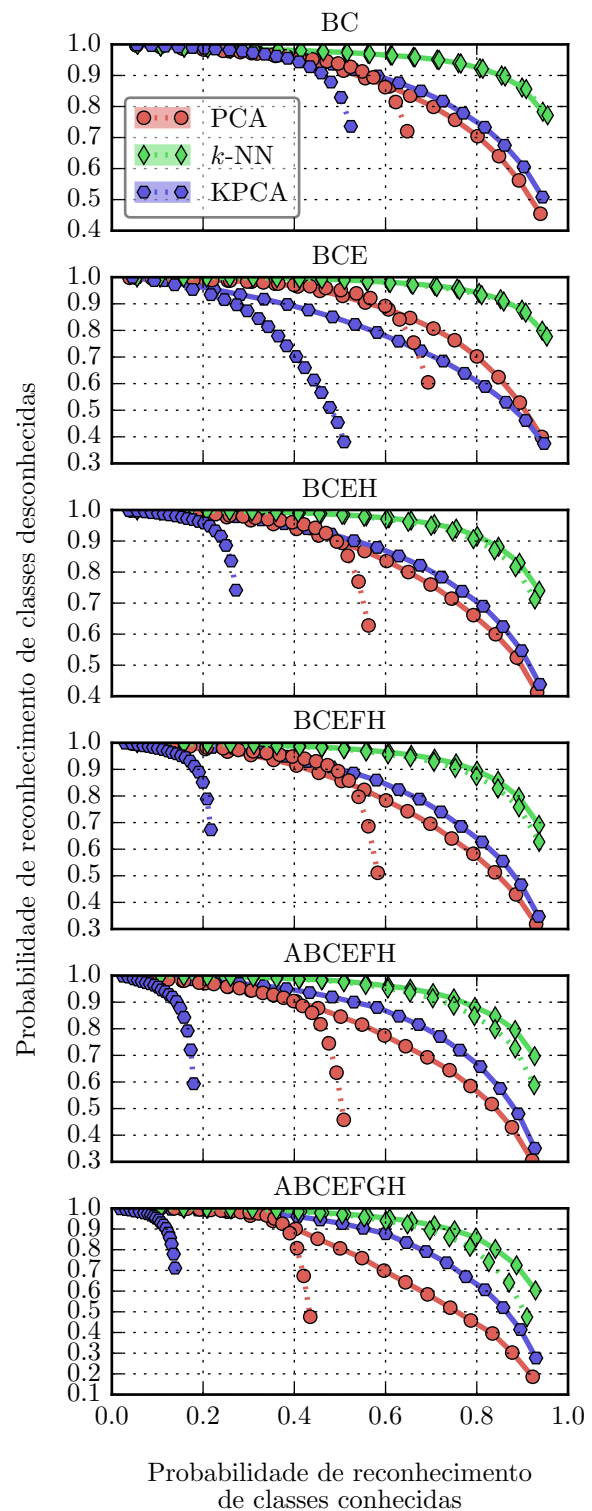


Figura 4: Curvas características dos sistemas avaliados (vide texto). Linhas pontilhadas representam sistemas com detectores especializados, enquanto que linhas cheias representam sistemas com detectores pós-classificação.

proposto inicialmente para a construção de detectores pós-classificação, também se mostrou efetivo para o desenvolvimento de detectores especializados. Através da medição do tempo de execução

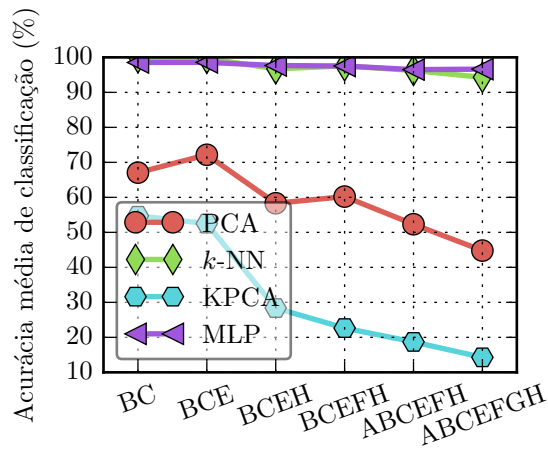


Figura 5: Acurácia média de classificação para as técnicas de detecção de novidades e ensemble MLP.

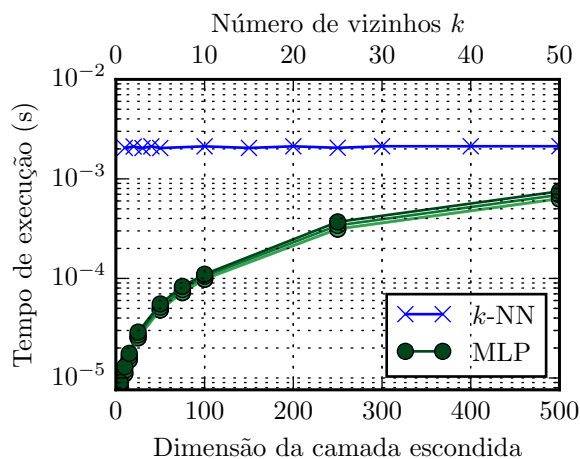


Figura 6: Tempo médio de execução do detector k -NN e de classificadores MLPs.

dos componentes individuais do sistema proposto, identificamos que este pode operar em tempo real. Como trabalhos futuros pretendemos investigar o uso de técnicas com custo computacional inferior ao k -NN, bem como avaliar o sistema proposto em situações mais críticas, como, por exemplo, para a identificação de navios específicos.

Agradecimentos

Os autores agradecem ao CNPq, à CAPES e à FAPERJ pelo apoio financeiro e infraestrutural. Ao IPqM, agradecemos pela cessão da base de dados e cooperação técnica.

Referências

Fawcett, T. (2006). An introduction to roc analysis, *Pattern recognition letters* **27**(8): 861–874.

Fernandez, H. L. (2005). *Classificação de navios baseada em curvas principais*, Master's thesis, COPPE/UFRJ.

Glorot, X., Bordes, A. and Bengio, Y. (2011). Deep sparse rectifier neural networks, *International Conference on Artificial Intelligence and Statistics*, pp. 315–323.

Guedes, R. M. and Souza Filho, J. B. d. O. e. (2014). Identificação de classes desconhecidas em sinais de sonar passivo através de componentes principais.

Hoffmann, H. (2007). Kernel pca for novelty detection, *Pattern Recognition* **40**(3): 863–874.

Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*, Cambridge University Press.

Musser, D. R. (1997). Introspective sorting and selection algorithms, *Softw., Pract. Exper.* **27**(8): 983–993.

Pimentel, M. A., Clifton, D. A., Clifton, L. and Tarassenko, L. (2014). A review of novelty detection, *Signal Processing* **99**: 215–249.

Soares Filho, W. (2001). Classificação do ruído irradiado por navios usando redes neurais, Tese de Doutorado, COPPE/UFRJ.

Souza Filho, J. B. O. (2007). Classificação neural de sinais de sonar passivo, Tese de Doutorado, COPPE/UFRJ.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting, *The Journal of Machine Learning Research* **15**(1): 1929–1958.

Tieleman, T. and Hinton, G. (2012). Lecture 6.5-rmsprop, *COURSERA: Neural networks for machine learning*.

Xianyi, Z., Qian, W. and Yunquan, Z. (2012). Model-driven level 3 blas performance optimization on loongson 3a processor, *Parallel and Distributed Systems (ICPADS), 2012 IEEE 18th International Conference on*, IEEE, pp. 684–691.