

Comparação de desempenho entre Máquina de Vetor de Suporte e Comitê de Redes Neurais Artificiais para Classificação de Spam

André Hermenegildo Costa Silva¹, Wilian Soares Lacerda¹, Bruno de Abreu Silva¹

¹Depto. de Ciência da Computação – Universidade Federal de Lavras (UFLA)
Caixa Postal 3037 – CEP 37.200-000 – Lavras – MG – Brazil

ahcsilva90@gmail.com, lacerda@dcc.ufla.br, bruno.abreu@dcc.ufla.br

Abstract. *The e-mail is one of the most popular communication tool. However, it is not uncommon to find undesired messages in our electronic mail boxes. These messages are known as spams. In a context where more and more messages are sent and received over the world, computational techniques to filter spams have increasingly importance. This paper aims to present two classifiers to filter e-mail messages, which is to identify whether a particular email is spam or not. It was used two machine learning techniques: Support Vector Machine (SVM) and Artificial Neural Network (ANN). For the ANN classifier, we used five different ANNs with Resilient Propagation (Rprop) learning algorithm (Backpropagation variation) and distinct architectures (layers) and settings (learning rate and number of iterations), forming a committee of networks. The two classifiers showed average rates of accuracy of 91.3% and 93.6% respectively. The SVM showed superior results compared to a single specific ANN (depending on architecture), but it was lower in the performance of a committee of ANNs. Furthermore, SVM is faster in training than the committee of ANN.*

Resumo. *O e-mail é uma das mais populares formas de comunicação. Porém, não é raro encontrar mensagens indesejadas nas caixas de entrada do correio eletrônico. Estas mensagens são conhecidas como spams. Em um contexto onde uma quantidade cada vez maior de mensagens são enviadas e recebidas por todo mundo, técnicas para filtrar automaticamente estas mensagens consideradas como spams são de grande importância. O objetivo deste artigo é apresentar dois classificadores para filtrar mensagens de e-mail indesejadas, isto é, identificar se determinado e-mail é ou não é um spam. Foram utilizadas duas técnicas de aprendizado de máquina: Máquina de Vetor de Suporte (SVM, do inglês Support Vector Machine) e Rede Neural Artificial (RNA). Para o segundo classificador, utilizou-se 5 (cinco) RNAs do tipo Multilayer Perceptron (MLP) com o algoritmo de aprendizagem Resilient Propagation - Rprop (variação do Backpropagation), com arquiteturas (camadas) e configurações (taxa de aprendizado e número de épocas) distintas, formando um comitê de redes. Os dois classificadores apresentaram taxas médias de acerto de 91,3% e 93,6%, respectivamente. Percebeu-se que a técnica de aprendizado SVM apresentou resultados superiores quando comparada a uma única RNA específica (dependendo da arquitetura), porém mostrou-se inferior quanto ao desempenho de um comitê de RNA. Além disso, SVM se mostrou mais rápida na realização do treinamento do que as RNAs.*

1. Introdução

Segundo [Baranauskas and Monard 2000], o objetivo do aprendizado de máquina é o desenvolvimento de técnicas computacionais sobre a aprendizagem bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. O aprendizado de máquina consiste, principalmente, na agregação do conhecimento baseado em experiências anteriores.

Existem inúmeras técnicas de aprendizado de máquina. As técnicas mais conhecidas são: Redes Neurais Artificiais (RNAs), Algoritmos Genéticos, Regra dos Vizinhos Mais Próximos (*K-Nearest Neighbor*), Árvores de Decisão, Máquina de Vetor de Suporte (SVM, do inglês *Support Vector Machine*), Sistemas *Fuzzy*, etc.

Conforme [Bishop 1995], reconhecimento de padrões abrange uma ampla gama de problemas que tem grande significado prático, como reconhecimento de voz, classificação de caracteres escritos à mão, detecção de erros em diagnósticos médicos, dentre outros.

Um problema que também pode ser resolvido por meio do uso de técnicas de reconhecimento de padrões é a classificação de mensagens de *e-mail* entre mensagens indesejadas (*spams*) e mensagens desejadas (*não-spams*). De acordo com [Kaur and Sharma 2016], cerca de 70% dos *e-mails* recebidos no ambiente empresarial são *spams*. Este tipo de mensagem indesejada prejudica a efetividade da comunicação, causando diversos problemas, como estouro da capacidade de armazenamento da caixa de entrada, armazenamento de dados desnecessários, sobrecarga da rede, além do tempo e energia gasto removendo e identificando este tipo de mensagem.

O alvo de estudo deste artigo é, através de reconhecimento de padrões, apresentar dois classificadores que sejam capazes de filtrar mensagens de *e-mail*, ou seja, identificar se determinado *e-mail* é ou não *spam*. Para construir estes classificadores, foram utilizadas duas técnicas de aprendizado de máquina: SVM e RNAs.

O restante deste artigo está organizado da seguinte maneira: na Seção 2, é apresentado o referencial teórico a respeito das técnicas de aprendizado SVM e RNAs; as estratégias adotadas para se alcançar os objetivos deste trabalho e os experimentos efetuados são descritos na Seção 3; na Seção 4, são relatados os resultados e o desempenho obtido pelos dois classificadores utilizados e; por fim, na Seção 5, são apresentadas as conclusões do presente trabalho.

2. Referencial Teórico

Conforme [Lorena and Carvalho 2007], a teoria da aprendizagem estatística estabelece uma série de princípios que devem ser seguidos na obtenção de classificadores com boa generalização, definida como a sua capacidade de prever corretamente a classe de novos dados do mesmo domínio em que o aprendizado ocorreu.

SVMs foram propostas no início da década de 90 como um método para a classificação de padrões e de regressão não-linear [Cortes and Vapnik 1995, Amorim 2007]. São fortemente baseadas na teoria de aprendizagem estatística, criada por [Vapnik 1995], de forma que uma SVM consiste em uma implementação aproximada do método de minimização do risco estrutural.

Para exemplificar Máquina de Vetor de Suporte, considere o problema da classificação binária. Para duas classes de dados bem separadas, a tarefa de aprendizado consiste em encontrar um *hiperplano direcionado*, ou seja, um hiperplano orientado que separe essas duas classes. Assim, instâncias da *classe 1* estariam de um lado deste hiperplano e instâncias da *classe 2* estariam do outro lado (ambas as classes com rótulos distintos), como pode ser visualizado na Figura 1. O hiperplano direcionado encontrado pela Máquina de Vetor de Suporte se encontra, então, entre as duas classes de dados, separando-as. Os pontos mais próximos em ambos os lados deste hiperplano de separação têm mais influência sobre a sua posição, por isso, são chamados *vetores de suporte* [Campbell and Ying 2011]. Estes pontos estão destacados (+ve e -ve) na Figura 1.

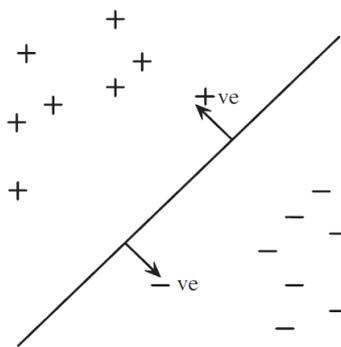


Figura 1. Classificação binária e hiperplano de separação [Campbell and Ying 2011].

A SVM normalmente obtém boa capacidade de generalização em problemas de classificação, se comparada com outras técnicas de aprendizado [?, Amorim 2007]. De acordo com [Bellman 1961], Máquina de Vetor de Suporte não sofre com o problema da *maldição da dimensionalidade*, ou seja, seus resultados podem ser melhores que técnicas de aprendizado que sofrem com esse problema, como é o caso de RNAs do tipo *Multilayer Perceptron* (MLP).

As RNAs são técnicas computacionais que têm inspiração nos neurônios biológicos e nos sistemas nervosos e que adquirem conhecimento através da experiência [Barreto 1997].

O primeiro modelo de neurônio foi proposto por [McCulloch and Pitts 1943]. No final da década de 1950, Rosenblatt deu prosseguimento às ideias de [McCulloch and Pitts 1943]. Ele criou uma genuína rede de múltiplos neurônios do tipo discriminadores lineares denominada *Perceptron* [Kovacs 1997]. Um dos modelos de RNAs mais utilizados é o MLP [Reed and Marks 1999], por ser um modelo que realiza aproximações de forma universal, além de ser capaz de representar qualquer função linear ou não linear.

O neurônio artificial de [McCulloch and Pitts 1943], mostrado na Figura 2, recebe *impulsos de entrada* onde estes são representados pelo vetor de entrada $X = [x_1, x_2, \dots, x_n]$. As *conexões sinápticas* estão representadas pelo vetor $W = [w_1, w_2, \dots, w_n]$. A saída u da *junção somadora* é o somatório de cada componente x_i do vetor de entrada X multipli-

cado pelo seu respectivo peso $w_i \in W$, incluindo, também, o valor do *bias*. Então, a saída u é fornecida a uma *função de ativação* que retorna o resultado de saída do neurônio. A saída do neurônio é representada por y .

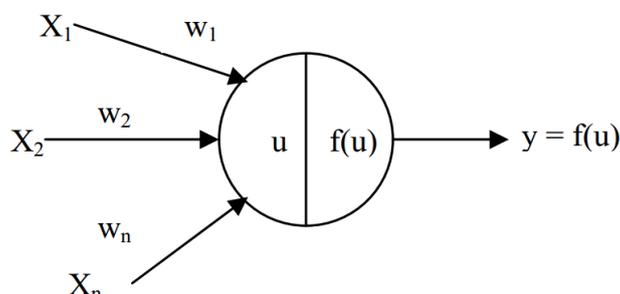


Figura 2. Representação do neurônio artificial de McCulloch e Pitts [Karrer et al. 2005].

É importante salientar que através da *função de ativação* que é determinado o comportamento, ou seja, a saída de um determinado neurônio artificial. Existem vários modelos de funções de ativação como, por exemplo: *Função Linear*, *Função Gaussiana*, *Função Logarítmica*, *Função Degrau*, *Função Sinal*, *Função Tangente Hiperbólica* e *Função Sigmoidal*.

Segundo [Gomes and Ludermir 2009], as funções de ativação mais utilizadas na prática são a *Função Sigmoidal* e a *Função Tangente Hiperbólica*. Optou-se por utilizar neste trabalho a *Função Sigmoidal (Sigmoid Function)* pois, conforme [Yegnanarayana 2004], ela apresenta vantagens sobre a *Função Tangente Hiperbólica*. Uma das vantagens, destacada por [Hsiao et al. 1998], é a capacidade de normalização, atuando como um escalonador para adaptar diferenças individuais nos padrões de entrada. O gráfico da *Função Sigmoidal* é mostrado na Figura 3 e sua representação matemática é apresentada a seguir:

$$f(x) = \frac{1}{1 + e^{-\lambda x}}$$

Geralmente, uma RNA é organizada em três grupos: *Camada de Entrada*, *Camadas Ocultas* e *Camada de Saída*. A Camada de Entrada é onde são apresentados os exemplos (ou padrões) necessários para que a RNA adquira conhecimento. Nas Camadas Ocultas (ou escondidas) é realizado todo processamento e extração de características dos exemplos (ou padrões) apresentados, por meio das conexões sinápticas. E, por fim, na Camada de Saída, a saída final da RNA é apresentada e, posteriormente, comparada com a desejada. Como exemplo, uma representação genérica de uma rede neural artificial pode ser visualizada na Figura 4.

Descrito por [Haykin 1998], a utilização de RNAs oferece várias propriedades úteis e capacidades, sendo que algumas delas devem ser mencionadas como:

- Não-linearidade: um neurônio artificial pode ser linear ou não;
- Mapeamento de Entrada-Saída: partindo do aprendizado supervisionado, envolve a modificação dos pesos sinápticos de uma RNA pela aplicação de um conjunto de amostras de treinamento rotulados (padrões ou exemplos);

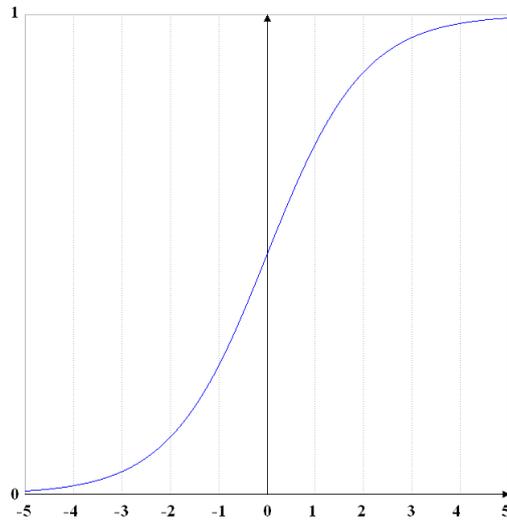


Figura 3. Gráfico da função sigmoideal.

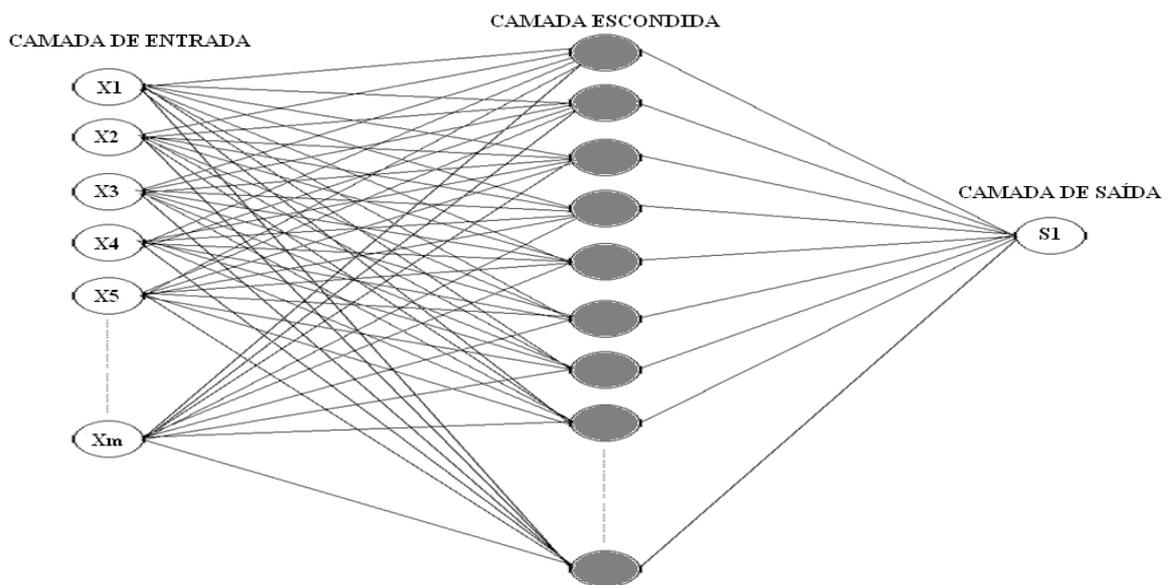


Figura 4. Arquitetura genérica de uma rede neural.

- Adaptabilidade: as RNAs têm uma capacidade inata de adaptar seus pesos sinápticos a modificações do meio ambiente.

Neste artigo, utilizou-se o algoritmo de aprendizagem *Resilient Propagation* (Rprop) que é uma variação do *Backpropagation*. No algoritmo *Backpropagation*, a tarefa de encontrar a melhor configuração para a taxa de *momentum* (além da taxa de aprendizado) pode ser difícil. Essa tarefa não é necessária quando se utiliza o algoritmo Rprop, sendo, assim, um dos motivos para sua utilização.

De acordo com [Medeiros Jr. et al. 2007], o algoritmo Rprop possui um melhor desempenho quando comparado ao tradicional *Backpropagation*. Ainda, segundo [Riedmiller and Braun 1993], o algoritmo Rprop apresenta resultados de convergência superiores e mais rápidos que o algoritmo *Backpropagation*.

3. Metodologia

Para a construção de ambos os classificadores - SVM e comitê de RNAs - foi utilizada a linguagem de programação Java (software JDK¹ versão 1.6.0).

No desenvolvimento do classificador baseado em SVM, utilizou-se a biblioteca LIBSVM² (Versão 3.12). Na implementação do comitê de RNAs, optou-se pelo uso do *framework* NEUROPH³ (Versão 2.6) por apresentar uma biblioteca bastante robusta e com diversos recursos relevantes para o presente trabalho.

A implementação dos classificadores e a execução dos treinamentos e testes foram realizadas em um computador com processador Pentium Dual-Core 2.00 GHz (1 MB de Cache), 2 GB de memória RAM e 160 GB de HD. O sistema operacional utilizado foi o Windows 7 (versão 64 bits).

A base de dados utilizada neste trabalho é do *UCI (University of California - Irvine) Machine Learning Repository* [Frank and Asuncion 2010, Hopkins et al. 1999]. Esta base de dados é composta por 4601 instâncias, sendo 1813 *spams* (39,4%) e 2788 *não-spams* (60,6%).

Cada instância dessa base de dados possui 58 características, sendo 57 valores contínuos e 1 nominal. Este atributo nominal (atributo de número 58) está localizado na última coluna da base de dados e ele indica se o *e-mail* foi considerado *spam* (1) ou não (0). A maioria dos outros atributos indicam se uma determinada palavra ou caractere ocorre frequentemente no *e-mail*, conforme detalhado a seguir:

- Os atributos no intervalo de 1 a 48 possuem valores contínuos. Cada atributo presente neste intervalo representa a porcentagem da ocorrência de uma determinada palavra em um *email*;
- Os atributos no intervalo de 49 a 54 também possuem valores contínuos. Cada atributo que está contido neste intervalo indica a porcentagem da ocorrência de um determinado caractere em um *email*;

¹Oracle Technology Network for Java Developers, disponível em <http://www.oracle.com/technetwork/java>

²LIBSVM - A Library for Support Vector Machines (Chih-Chung Chang and Chih-Jen Lin), disponível em <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

³Java Neural Network Framework Neuroph, disponível em <http://neuroph.sourceforge.net/>

- Os atributos 55, 56 e 57 medem o comprimento de sequências de letras maiúsculas consecutivas (valores contínuos).

Logo, pode-se perceber que os 57 valores contínuos são atributos de entrada de uma máquina de aprendizado e o único valor nominal é o atributo de saída, que indica à máquina de aprendizado qual a saída desejada do classificador.

Para ambos os classificadores, realizou-se a normalização de todas as características, exceto a nominal, onde o valor de cada característica dentre todas as instâncias variam entre -1 e +1. A principal vantagem da normalização dos dados é evitar que atributos em intervalos numéricos maiores dominem aqueles que estão em intervalos numéricos menores, influenciando na capacidade de generalização dos classificadores descritos.

3.1. Máquina de Vetor de Suporte

Para configuração da Máquina de Vetor de Suporte, optou-se pela metodologia proposta por [Hsu et al. 2010]. A função *kernel* utilizada foi a RBF (*Radial Basis Function*), que possui dois parâmetros empíricos a serem definidos por um professor (aprendizado supervisionado), *custo* (C) e *gama* (γ). Segue abaixo a função matemática da RBF (sendo x o Vetor de Características - padrões e exemplos de entrada - e y a Saída Desejada):

$$K(x, y) = e^{-\gamma \|x-y\|^2}$$

Utilizou-se o recurso de validação cruzada (*cross-validation*). A tarefa principal desta técnica consiste em identificar os melhores valores de C e γ para que o classificador possa prever os dados com melhores taxas de acerto e adquirir melhor capacidade de generalização. Uma importante característica da validação cruzada é que ela tem a capacidade de prevenir o problema de *overfitting*, ou seja, previne que a SVM tenha uma capacidade reduzida de generalização, não classificando adequadamente dados desconhecidos (dados de teste).

O tipo de validação cruzada escolhido foi o *k-Fold Cross-Validation*. Neste tipo de validação cruzada, o conjunto de dados de treinamento é dividido em k subconjuntos, e o método de validação é repetido k vezes. Em cada iteração, um dos k subconjuntos é usado como teste e os outros $k - 1$ subconjuntos são unidos para formar um conjunto de treinamento [Polat et al. 2006]. Neste trabalho, o parâmetro k foi configurado com valor 4 (*4-Fold Cross-Validation*).

Após a execução das k ($k = 4$) iterações, foram identificados os melhores parâmetros para C e γ . Então, C foi configurado com o valor 2^{17} e γ foi configurado com o valor 2^{-9} . Depois disto, foi realizado o treinamento e, posteriormente, o teste.

3.2. Comitê de Redes Neurais Artificiais

Proposto por [Haykin 1998, Oliveira et al. 2010], para o segundo classificador, foi implementado um comitê composto por 5 (cinco) RNAs do tipo MLP. Cada uma delas foi treinada com o algoritmo de aprendizagem Rprop. A função de ativação utilizada em cada neurônio do comitê foi a *função sigmoideal*.

A arquitetura de cada RNA presente no comitê foi configurada da seguinte maneira: uma camada de entrada com 57 neurônios correspondente a cada atributo dos dados; uma camada escondida, onde a quantidade de neurônios nessa camada é distinta

entre as 5 (cinco) redes presentes; e uma camada de saída com um neurônio, pois cada rede irá classificar dois estados possíveis, se determinado *e-mail* é (1) ou não (0) *spam*.

Quanto à taxa de aprendizado e número de épocas, foram definidos, também, valores distintos para cada RNA presente no comitê. Na Tabela 1, são exibidas essas informações e, também, o número de neurônios presentes na camada oculta de cada rede.

Tabela 1. Parâmetros de treinamento de cada RNA presente no comitê.

RNAs	Neurônios na Camada Oculta	Nº. de Épocas	Taxa de Aprendizado
RNA I	30	1000	0,75
RNA II	40	2000	0,28
RNA III	38	3500	0,001
RNA IV	45	3000	0,09
RNA V	50	5000	0,79

3.3. Divisão dos dados

A base de dados utilizada neste trabalho apresenta uma significativa discrepância entre a quantidade de *spam* e não-*spam*, sendo 1813 (39,4%) e 2788 (60,6%), respectivamente, de um total de 4601 instâncias.

Para verificar se a saída de cada classificador é coerente, o número de amostras no conjunto de treinamento foi definido de forma dependente ao conjunto de teste, ou seja, foi escolhido o mesmo número de *spams* e não-*spams* para o conjunto de teste e o restante das instâncias integraram o conjunto de treinamento. Então, para verificar a capacidade de generalização de cada classificador, foram criados 5 (cinco) grupos, cada grupo formado pelos conjuntos de treinamento e teste. O gráfico da divisão dos dados pode ser visualizado na Figura 5.

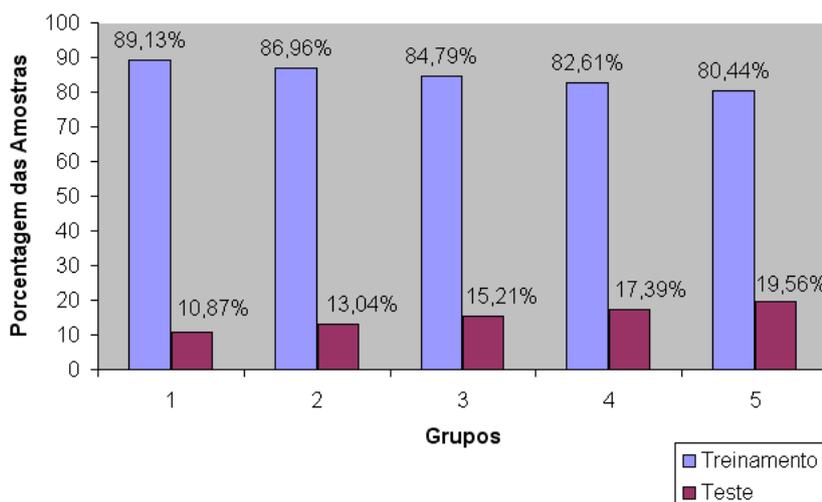


Figura 5. Gráfico da divisão dos dados.

Com essa forma de divisão dos dados foi possível verificar se o classificador é imparcial ou não quanto à classificação dos dados pois, pelo motivo de existir mais não-*spams* do que *spams*, o classificador poderia vir a classificar corretamente somente *e-mails*

não-spams e, conseqüentemente, não sendo capaz de reconhecer *e-mails* rotulados como *spam*.

Os dados dos conjuntos de treinamento foram embaralhados para impedir que instâncias de mesmo rótulo fossem apresentadas em grandes seqüências aos algoritmos de aprendizado de máquina, reduzindo, assim, as possibilidades de ocorrência de *overfitting*.

4. Resultados

Nesta seção, são apresentados os resultados obtidos por meio do uso de SVM e, posteriormente, os resultados com RNA são descritos. Por fim, são feitas algumas considerações importantes a respeito do tempo gasto no treinamento considerando as duas técnicas utilizadas neste trabalho.

4.1. Resultados obtidos com a técnica SVM

Utilizando a estratégia proposta na seção 3.3, dentre os 5 (cinco) grupos, o que obteve o melhor resultado foi o grupo 4 com 92,125% de acerto, classificando 737 das 800 amostras presentes no conjunto de teste deste grupo. O que obteve o pior resultado foi o grupo 2 com 90,5% de acerto, classificando 543 das 600 amostras presentes no conjunto de teste do referido grupo. Esses resultados e dos demais grupos podem ser observados na Tabela 2 e, também, na Figura 6.

Tabela 2. Resultado de desempenho da SVM.

Grupos	Acurácia	Classificação
Grupo 1	91%	455/500
Grupo 2	90,5%	543/600
Grupo 3	91,43%	640/700
Grupo 4	92,125%	737/800
Grupo 5	91,44%	823/900

A partir das taxas de acurácia de cada grupo foram calculados a média aritmética, desvio padrão e coeficiente de variação chegando aos resultados de: aproximadamente 91,3%, 0,6 e 0,0066 (ou 0,66%), respectivamente. Logo, através do coeficiente de variação percebe-se que as taxas de acurácia variaram pouco entre um grupo e outro.

4.2. Resultados obtidos com o comitê de RNAs

Quanto ao comitê de RNAs, notou-se que cada rede presente no comitê demorou consideravelmente para concluir seu treinamento e teste e, conseqüentemente, despendendo tempo para ponderar a saída de cada uma dessas redes para a saída final do comitê.

Analogamente à SVM, também se utilizou a estratégia proposta na subseção 3.3. Então, dentre os 5 (cinco) grupos, o que obteve o melhor resultado foi o grupo 1 com 95% de acerto, classificando 475 das 500 amostras presentes no conjunto de teste deste grupo. O que obteve o pior resultado foi o grupo 5 com 92% de acerto, classificando 828 das 900 amostras presentes no conjunto de teste do referido grupo. Estes resultados e dos demais grupos podem ser observados na Tabela 3 e, também, na Figura 6.

A partir das taxas de acurácia de cada grupo foram calculados média aritmética, desvio padrão e coeficiente de variação chegando aos resultados: aproximadamente

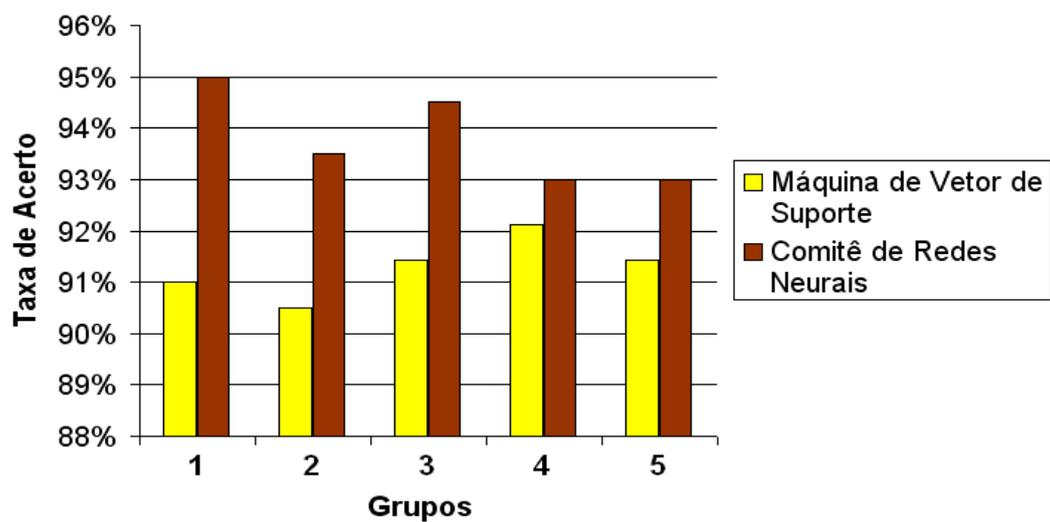


Figura 6. Acurácia das técnicas de aprendizado.

Tabela 3. Resultado de desempenho do comitê de RNAs.

Grupos	Acurácia	Classificação
Grupo I	95%	475/500
Grupo II	93,5%	561/600
Grupo III	94,5%	661/700
Grupo IV	93%	744/800
Grupo V	92%	828/900

93,6%, 1,194 e 0,0127 (ou 1,27%), respectivamente. Logo, através do coeficiente de variação percebe-se que as taxas de acurácia do segundo classificador (Comitê de Redes Neurais Artificiais) também variaram pouco entre um grupo e outro. Porém, nota-se que a taxa de variação desse classificador (1,27%) variou mais do que a taxa de variação do primeiro classificador (0,66%) (Máquina de Vetor de Suporte).

Conforme pode ser observado também na Figura 6, o comitê de Redes Neurais Artificiais apresentou taxas de acerto superiores às taxas da Máquina de Vetor de Suporte em todos os grupos considerados.

4.3. Tempo gasto no treinamento

Uma das grandes vantagens da SVM é a sua rápida convergência no treinamento, sendo mais rápida do que uma RNA [Ding and Dubchak 2001] e, conseqüentemente, um comitê de RNAs. Neste trabalho, a SVM se mostrou 2,5 vezes mais rápida do que uma única RNA e 12 vezes mais rápida do que um comitê de RNAs.

O tempo médio gasto, o desvio-padrão e o coeficiente de variação resultantes do treinamento da SVM foram, aproximadamente, 43,8 segundos; 2,05; e 0,05 (ou 5%), respectivamente.

Para o comitê de RNAs, o tempo médio gasto no treinamento foi de 9 minutos e 7 segundos, o desvio-padrão e o coeficiente de variação resultantes foram de 76,07 e 0,13 (ou 13%), respectivamente.

Percebe-se através do coeficiente de variação que o comitê de RNAs tem uma variação bem maior que a SVM em relação ao tempo gasto no treinamento. Esta variação pode ser visualizada na Figura 7. O tempo despendido pelos dois classificadores no treinamento de cada grupo e o tempo total está representado na Tabela 4 e Tabela 5.

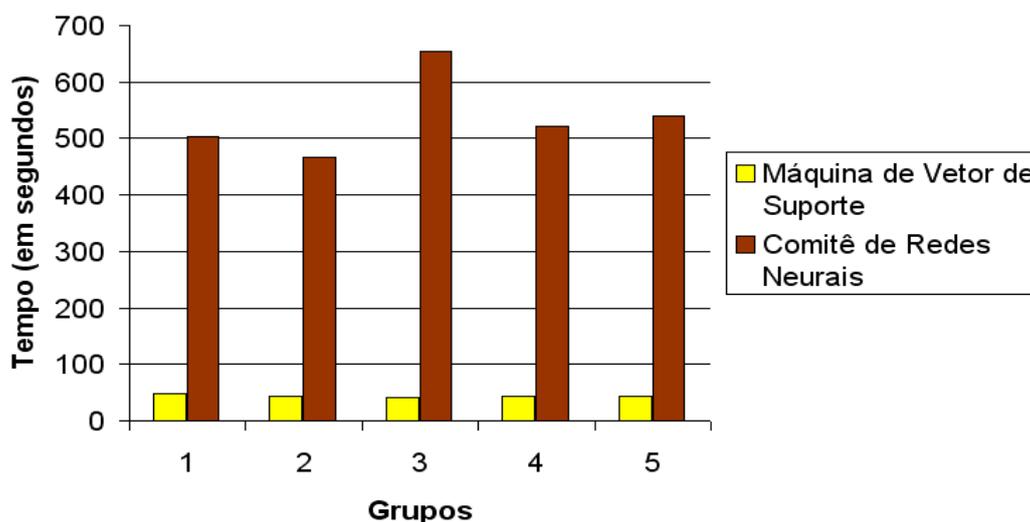


Figura 7. Gráfico do tempo despendido no treinamento.

5. Conclusões

A partir da análise dos resultados, percebe-se que o segundo classificador, implementado com um comitê de Redes Neurais Artificiais do tipo *Multilayer Perceptron* possui um desempenho superior ao classificador implementado com a técnica de aprendizado Máquina

Tabela 4. Tempo gasto no treinamento da SVM.

Grupos	Tempo gasto
Grupo I	0 min 47 s
Grupo II	0 min 44 s
Grupo III	0 min 42 s
Grupo IV	0 min 44 s
Grupo V	0 min 42 s
Total	3 min 39 s

Tabela 5. Tempo gasto no treinamento do comitê de RNAs.

Grupos	Tempo gasto
Grupo I	8 min 24 s
Grupo II	7 min 46 s
Grupo III	10 min 54 s
Grupo IV	8 min 42 s
Grupo V	9 min 01 s
Total	44 min 47 s

de Vetor de Suporte. No trabalho de [Assis 2006], são descritos casos em que ocorreram 100% de classificação correta de *e-mails spams* e *não-spams*. Assim, é possível deduzir que os resultados obtidos neste trabalho tanto por meio do primeiro classificador quanto por meio do segundo classificador não são tendenciosos (médias de 91,3% para o primeiro classificador e 93,6% para o segundo classificador).

Semelhante aos resultados do trabalho de [Drucker et al. 1999], a técnica SVM apresentou resultados excelentes. Além disso, essa técnica mostrou desempenho superior quando comparada a uma única RNA, dependendo da arquitetura e configuração desta última. Esta é a grande desvantagem em se utilizar RNAs, uma vez que é necessário definir valores de muitos parâmetros de forma empírica para garantir a convergência e capacidade de generalização. Ou seja, para se chegar ao melhor resultado é necessário testar diversas arquiteturas diferentes (variação do número de neurônios na camada oculta), realizar variações nas taxas de aprendizado, *momentum* e número de épocas de treinamento (iterações).

Ainda, neste trabalho, percebeu-se uma grande vantagem da SVM, que é sua velocidade na execução do treinamento quando comparada a uma RNA e, conseqüentemente, a um comitê de RNAs. É possível concluir, portanto, que a escolha entre o uso de SVM ou RNAs para a classificação de mensagens de *e-mail* entre *spams* e *não-spams* dependerá se o mais importante é obter um menor tempo de treinamento (neste caso, SVM foi mais adequada) ou se é mais importante obter maior taxa de acerto (neste caso, o comitê de RNAs apresentou resultados mais adequados).

Referências

Amorim, D. G. (2007). *Redes ART con categorias internas de geometria irregular*. Tese de Doutorado - Universidade de Santiago de Compostela - Departamento de Eletrônica e Computação. 112 p.

- Assis, J. a. M. d. C. (2006). Detecção de e-mails spam utilizando redes neurais. Master's thesis, Universidade Federal de Itajubá (UNIFEI). Mestrado em Engenharia Elétrica, 112 p.
- Baranauskas, J. A. and Monard, M. C. (2000). Reviewing some machine learning concepts and methods. Technical Report 102, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos. 52 p.
- Barreto, J. M. (1997). Introdução as redes neurais artificiais. In *V Escola Reginal de Informática da SBC Regional Sul. Maringá, Paraná*, pages 41–71. Universidade de Maringá; Universidade Federal de Santa Catarina; Universidade Federal de Santa Maria.
- Bellman, R. E. (1961). *Adaptive control processes - A guided tour*. Princeton University Press, Princeton, New Jersey, USA. 255 p.
- Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, New York. 482 p.
- Campbell, C. and Ying, Y. (2011). *Learning with Support Vector Machines*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers. 96 p.
- Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. In *Machine Learning*, volume 20, pages 273–297. Kluwer Academic Publishers Hingham, MA, USA.
- Ding, C. H. Q. and Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics/Oxford Journals*, 17:349–358. NERSC Division, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA 94720, USA.
- Drucker, H., Wu, D., and Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5):1048–1054. AT&T Labs-Res., Red Bank, NJ.
- Frank, A. and Asuncion, A. (2010). UCI Repository of Machine Learning Databases. Disponível em <<http://archive.ics.uci.edu/ml>>, acessado em Janeiro/2012.
- Gomes, G. S. S. and Ludermir, T. B. (2009). Redes neurais artificiais com funções de ativação complemento log-log e probit para aproximar funções na presença de observações extremas. *Learning and Nonlinear Models. Revista da Sociedade Brasileira de Redes Neurais (SBRN)*, 6(2):142–153.
- Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2 edition, USA/New York. 842 p.
- Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (1999). UCI Repository of Machine Learning Databases. Disponível em <<http://archive.ics.uci.edu/ml/datasets/Spambase>>, acessado em Janeiro/2012.
- Hsiao, T., Lin, C., Zeng, M., and Chiang, H. K. (1998). The implementation of partial least squares with artificial neural network architecture. In *Proceedings of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 20(3), pages 1341–1343.

- Hsu, C. W., Chang, C. C., and Lin, C. J. (2010). A practical guide to support vector classification. Disponível em <<http://www.csie.ntu.edu.tw/~cjlin/>>, acessado em Janeiro/2012.
- Karrer, D., Cameira, R., Vasques, A., and Benzecry, M. (2005). Redes neurais artificiais: conceitos e aplicações. In *IX Profundão - Encontro de Engenharia de Produção da UFRJ*, Rio de Janeiro. 12 p.
- Kaur, H. and Sharma, A. (2016). Improved email spam classification method using integrated particle swarm optimization and decision tree. In *2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 516–521.
- Kovacs, Z. L. (1997). *Redes Neurais Artificiais: Fundamentos e Aplicações*. Collegium Cognition, 2a. edição, São Paulo/Brasil. 174 p.
- Lorena, A. C. and Carvalho, A. C. P. L. F. (2007). Uma introdução as Support Vector Machines. *Revista de Informática e Tecnologia Aplicada - RITA*, XIV(2):43–67.
- McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous. In *Bulletin of Mathematical Biophysics*, volume 5, pages 115–133. Springer New York, New York.
- Medeiros Jr., M. F., Santos, C. K. S., Oliveira, J. T., Pires, P. S. M., Melo, J. D., Neto, A. D. D., and Leitao, J. J. A. L. (2007). Influence of signal pre-processing in the efficiency of algorithms based on neural networks for disturbance classification. *Computational Intelligence in Image and Signal Processing*, pages 95–100.
- Oliveira, R. C., Acevedo, N. I. A., Neto, A. J. S., and Neto, L. B. (2010). Aplicação da técnica de máquina de comitê de redes neurais artificiais para a solução de problemas inversos em transferência radiativa. *Tendências em Matemática Aplicada e Computacional - TEMA*, 11(2):171–182.
- Polat, K., Kara, S., Latifoglu, F., and Günes, S. (2006). A novel approach to resource allocation mechanism in artificial immune recognition system: Fuzzy resource allocation mechanism and application to diagnosis of atherosclerosis disease. *Springer Berlin / Heidelberg. Artificial Immune Systems. Lecture Notes in Computer Science*, 4163:244–255. Dept. of Electrical & Electronics Engineering, Elcuk University, 42075 Konya, Turkey.
- Reed, R. D. and Marks, R. J. (1999). *Neural Smithing - Supervised Learning in Feed-forward Artificial Neural Networks*. MIT Press Cambridge Editora, MA, EUA. 352 p.
- Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning - the RPROP algorithm. In *Proceedings of the International Conference on Neural Networks*, pages 586–591, San Francisco, CA. IEEE.
- Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag New York Inc., New York. 333 p.
- Yegnanarayana, B. (2004). *Artificial Neural Networks*. Prentice-Hall of India, New Delhi/India. 476 p.