CLASSIFICAÇÃO DE MENSAGENS DE *SPAM* E *PHISHING* BASEADA NA INTELIGÊNCIA DE ENXAMES

Carine Geltrudes Webber, Vinícius Crestani e Maria de Fátima Webber do Prado Lima
Centro de Computação e Tecnologia da Informação - Universidade de Caxias do Sul

RESUMO

Este artigo apresenta um sistema inspirado na inteligência coletiva das abelhas capaz de classificar mensagens de e-mail como *spam*, *phishing* ou mensagens seguras. O algoritmo utilizado neste trabalho é o Artificial Bee Colony (Karaboga, 2005). Ele baseia-se no comportamento natural das abelhas forrageadoras, responsáveis por encontrar fontes de alimentos e passar informações através de uma dança para outras que aguardam na colmeia. Neste processo as abelhas espectadoras escolhem a melhor fonte e assim toda a colmeia se beneficia dos melhores resultados globais. No experimento realizado foram utilizados datasets públicos de mensagens legítimas, de *spam* e de *phishing*. Os resultados obtidos através da comparação com outros algoritmos podem ser considerados positivos, já que em alguns casos a taxa de acerto foi superior as obtidas por outros algoritmos.

PALAVRAS-CHAVE

Inteligência de enxames, classificação de dados, aprendizagem automática, inteligência artificial.

1. INTRODUÇÃO

Classificação de dados é uma forma de análise que utiliza modelos para definir a classe a qual pertence cada informação. Esses modelos são conhecidos como classificadores e têm a função de prever a categoria de uma informação (Han, 2012). Por exemplo, as técnicas de classificação são usadas na categorização de células cancerígenas, na realização de diagnósticos na medicina e na detecção de fraudes em mensagens de e-mail.

O processo de classificação de dados é usualmente dividido em duas etapas. A primeira fase do processo é a etapa de treinamento e a segunda é a etapa de testes. Na etapa de treinamento um classificador é construído através de um conjunto de exemplos. O conjunto de treinamento é formado por dados associados a uma classe. Por ser conhecida a classe a que pertence cada dado da amostra, esse processo é conhecido como aprendizagem supervisionada. A segunda etapa é conhecida como teste, onde é determinado se o nível de acerto do modelo é aceitável. Caso seja, o modelo é utilizado para classificar novos dados (Han, 2012).

O problema da classificação é um problema importante da computação para o qual não existe uma solução ótima. Diversas técnicas conhecidas, tais como as árvores de decisão, redes *bayesianas*, rede neurais, entre outras, são aplicadas para se tentar chegar próximo da melhor solução. O fato de nenhuma técnica produzir uma solução ótima evidencia a importância do estudo de novas técnicas.

A técnica de classificação a ser abordada neste trabalho é o algoritmo ABC (Artificial Bee Colony). Esse é um algoritmo baseado no comportamento das abelhas forrageadoras, proposto por Karaboga (2005) para resolver problemas de otimização numérica (Kumbhar, 2011). Além da utilização na otimização de soluções, existem outros trabalhos que demostram com sucesso sua utilização em *clustering* (Karaboga, 2009), *clustering* difuso (Karaboga, 2010) e classificação (Shukran, 2011).

Este artigo apresenta uma ferramenta de classificação de mensagens de e-mails baseado no algoritmo ABC (Karaboga, 2005). A classificação de mensagens de e-mail é útil para evitar que mensagens indesejadas ou fraudulentas cheguem as caixas postais dos usuários. Dados do mês de fevereiro de 2012 apresentados no site do Kaspersky Lab, demostram que aproximadamente 80% dos e-mails que circulam na internet são e-mails indesejados (*spam*) ou tentativas de golpes e fraudes (*phishing*) (Kaspersky Lab, 2012). Partindo da metáfora dos algoritmos baseados em comportamentos de enxames, o objetivo deste trabalho é buscar uma solução que classifique mensagens de e-mail como sendo *spams*, *phishing* ou mensagens seguras. Sendo assim, este artigo está organizado em 5 seções. A seção 2 introduz o algoritmo ABC. A seção 3 descreve

alguns trabalhos relacionados. A seção 4 aborda a implementação realizada. Finalmente, a seção 5 apresenta os resultados obtidos nos experimentos realizados e em comparação a outros métodos.

2. ALGORITMO ABC

O *Artificial Bee Colony* (ABC) é um algoritmo baseado na Inteligência de Enxames proposto por Karaboga em 2005. Na natureza, abelhas realizam a coleta de néctar em grandes áreas. Dentro da colmeia há abelhas com o papel designado de encontrar as flores que possuem a maior qualidade e quantidade de néctar. As abelhas comunicam-se umas com as outras através de danças que informam às demais abelhas sobre a direção, distância e a qualidade e o tipo da fonte de alimento encontrada (Karaboga, 2005). Dentro da organização da colmeia, as abelhas responsáveis por encontrar fontes de alimentos e repassar estas informações às outras abelhas são chamadas de abelhas forrageadoras. Elas são divididas em dois grupos: forrageadoras livres e forrageadoras empregadas. Ainda, no grupo das abelhas forrageadoras livres existem dois tipos de abelhas. O primeiro tipo engloba as abelhas exploradoras, que procuram randomicamente por novas fontes de alimentos próximas a colmeia. O segundo tipo é composto por abelhas espectadoras que aguardam a dança das abelhas empregadas a fim de selecionar a melhor fonte de alimento. Já as abelhas forrageadoras empregadas são as abelhas que possuem informações sobre fontes de alimentos. Elas são responsáveis também por passar essas informações às outras abelhas da colmeia. As abelhas empregadas se tornam abelhas exploradoras assim que a fonte de alimento se esgotar.

Para implementação do algoritmo ABC, três componentes devem ser considerados: as fontes de alimentos, as abelhas forrageadoras empregadas e as abelhas forrageadoras livres. As fontes de alimentos representam a posição da solução do problema. A qualidade da fonte é representada por um valor de *fitness*. Além disso, há dois comportamentos básicos a serem seguidos: recrutamento para uma fonte de alimento e o abandono de uma fonte de alimento. Na equação 1, P_i é calculada como sendo a probabilidade associada a uma fonte de alimentos i. Uma abelha espectadora irá escolher uma fonte de alimento com base no valor de P_i. Na mesma equação, fit_i representa a qualidade da fonte de alimento, medida pelas abelhas empregadas.

$$P_i = \frac{fit_i}{\sum fit_i} \quad \text{(eq. 1)}$$

O algoritmo ABC para a tarefa de classificação foi proposto por Shukran em 2011 (Shukran et al., 2011), sendo brevemente apresentado aqui. Na fase de treinamento, as abelhas visitam as fontes de alimento, coletando padrões e organizando-os na forma de regras. O conjunto de regras produzido representa o modelo construído coletivamente. O algoritmo utiliza as principais definições apresentadas a seguir: (a) descoberta da regra; (b) função de *fitness das regras*; (c) poda; (d) estratégia de previsão.

2.1 Descoberta das Regras

A fase da descoberta da regra é a parte crucial do algoritmo de classificação. Uma regra é construída a partir dos atributos do conjunto de dados de treinamento (*dataset*). Para cada atributo são guardados dois valores: o limite inferior (menor valor atribuído a ele) e o limite superior (maior valor). Existem outros três valores associados a uma regra: classe de predição, o valor do *fitness* e porcentagem de cobertura da regra. O algoritmo da regra de classificação deve descobrir automaticamente a regra para cada classe. Para cada classe selecionada, o algoritmo vai gerar regras iterativamente até que o conjunto de regras possa cobrir todas as instâncias pertencentes à classe.

2.2 Função de Fitness das Regras

O valor do *fitness* de uma regra é calculado conforme a equação 2.

$$Fitness = \frac{VP}{VP + FN} x \frac{VN}{VN + FP} \quad \text{(eq.2)}$$

As variáveis VP, FN, FP e VN representam respectivamente o número de verdadeiros positivos, falsos negativos, falsos positivos e verdadeiros negativos dado o dataset analisado. Diz-se que uma regra "cobre" uma instância do conjunto de dados quando todos os seus valores de atributo estejam entre os limites inferior e superior de cada atributo da regra. Se todos os atributos de uma instância respeitos os limites de uma regra, então esta instância é coberta pela regra. Se a classe da instância avaliada for igual à classe prevista pela regra, ela constitui um verdadeiro positivo. Caso a instância não seja coberta pela regra, mas seja da mesma classe da regra, ela é um falso negativo. Nos falsos positivos conta-se as instâncias cobertas pela regra, mas que não possuem a classe prevista pela regra. Por fim, nos verdadeiros negativos conta-se o número de instâncias não cobertas pela regra e que não possuem a classe prevista pela regra. A partir destes valores pode-se calcular o *fitness* das regras.

2.3 Poda

Depois de todas as classes serem processadas e de todos os conjuntos de regras serem gerados, cada regra é colocada no processo de poda. O objetivo principal da poda é remover limitações de regras redundantes que possam ser desnecessariamente incluídas no conjunto de regras. Uma vez que atributos sem relação irão influenciar de forma negativa o resultado da classificação, a poda de regras pode aumentar a precisão dos resultados. O processo é repetido até que todas as regras tenham sido avaliadas.

2.4 Estratégia de Previsão

O conjunto de regras resultante é utilizado para prever a classe de novos dados. Algumas vezes, o teste dos dados poderá ser coberto por mais de uma regra para diferentes classes. Quando isto acontecer, a estratégia de previsão irá determinar qual a classe da instância. Existem três passos que devem ser seguidos para realizar a previsão:

1. Calcular o valor da previsão para todas as regras que foram cobertas pelo teste dos dados.
2. Acumular estes valores de previsão de acordo com as possíveis classes.
3. Selecionar a classe com o maior valor de previsão como a classe final.

Após o processo de estratégia de previsão, o núcleo é a função de previsão que irá ser utilizada para calcular o valor de previsão para cada regra. Este valor é definido pela equação 3.

$$previsão = (\alpha * fitness) + (\beta * porcentagem\,de\,cobertura) \quad \text{(eq. 3)}$$

Veja que α e β são dois parâmetros onde $\alpha \in [0,1]$ e $\beta=(1-\alpha)$. A porcentagem de cobertura da regra define a proporção de dados cobertos pela regra que tem a classe prevista pela regra (verdadeiros positivos). Este cálculo é feito pela equação 4.

$$porcentagem\,de\,cobertura = \frac{TP}{N} \quad \text{(eq.4)}$$

Note que N é o total de instâncias que pertencem as classes previstas pelas regras.

3. TRABALHOS RELACIONADOS

O algoritmo ABC tem sido utilizado para resolver diversos problemas tais como: otimização global numérica (Kang, 2013), questões de projeto em engenharia como otimização da modelagem do transistor e capacidade ótima das instalações de produção de gás (Sharma, 2012), controle de emissão dos efeitos da energia eólica (Jadhav, 2012), busca de construções genéticas potenciais explorando interações genéticas possíveis e dados cronológicos (Rakshitl ,2012), algoritmos que auxiliem no desenvolvimento de sistemas que realizem a previsão de falhas nos negócios para impedir a perda significativa de custos sociais causados pela falência inesperada de organizações (Lee, 2012) sistemas de biometria para autenticação de usuários (Tsai, 2012) e programação de tarefas em indústrias siderúrgicas (Pan, 2013).

Méndeza (2012) desenvolveu um conjunto de ferramentas para aumentar o desempenho do software SpamAssassin utilizando técnicas de *cross-validation*. Sua aplicabilidade foi verificada comparando seus resultados com os resultados obtidos pelo software de SpamAssassin e outras técnicas anti-Spam (redes Baysianas e Máquinas Vetoriais) em dois estudos de caso diferentes. Salcedo-Campos (2012) apresentou uma nova técnica baseada unicamente na informação do cabeçalho do e-mail. Os caracteres são tratados como sinais e parametrizados de acordo com técnicas de processamento do sinal padrão extraindo parâmetros relevantes do cabeçalho. Esta técnica se baseia no modelo de Markov. Para validar a parametrização proposta, foram utilizadas as máquinas vetoriais e o vizinho mais próximo (KNN). Para realizar a classificação, Razmara (2012) propôs um método de variação da frequência do termo (TFV - Term Frequency Variance) para reduzir a complexidade computacional, removendo os termos menos informativos. Para realizar a filtragem dos dados foi utilizada a técnica Random Forest.

Almeida (2012) utilizou um método de inferência indutiva denominando princípio mínimo do comprimento da descrição (MDL - Minimum Description Length) para realizar a filtragem dos SPAMs. Laorden (2012) explorou o uso da semântica na filtragem do Spam aplicando diversos modelos de aprendizagem usando cross-validation: redes Baysianas, árvores de decisão, vizinhos mais próximo (KNN - k-Nearest Neighbour) e máquinas de suporte vetorial. Além de utilizar diversos tipos de redes Bayesianas e máquinas vetoriais, Pérez-Díaz (2012) também utilizou Adaboost para realizar a identificação de e-mails do tipo SPAM. Já Li (2012) propõe utilizar máquinas vetoriais lineares para classificar os e-mails.

Moniza (2012) comparou a utilização das técnicas aprendizagem supervisionadas da máquina tais como algoritmos Bayesianos, *lazy*, árvores de decisão, redes neurais e máquinas vetoriais, para realizar o reconhecimento de SPAMs. O modelo de rede neural é mais robusto se comparado com as outras técnicas devido a sua capacidade de prever melhor na presença de ruídos, além de prover uma escalabilidade melhor. Por outro lado, a interpretabilidade das redes neurais é mais baixa se comparada às árvores de decisão, às regras de decisão e aos algoritmos de Naïve Bayes. Du Toit (2012) analisou o desempenho de uma rede neural aditiva generalizada, comparando com redes Bayesianas e a uma técnica baseada em memória. Através de seus testes, chegou a conclusão que a rede neural aditiva generalizada é mais eficaz que as outras técnicas testadas. Observou-se entretanto nos artigos mencionados que o algoritmo ABC não foi utilizado. Nas seções seguintes pretende-se analisar a adequação deste algoritmo para realizar a classificação dos e-mails.

4. IMPLEMENTAÇÃO

Esta seção descreve o Sistema de Classificação de Mensagens baseado em Inteligência de Enxames (SCMIE). A sua implementação foi desenvolvida em linguagem orientada a objetos C# utilizando a versão 4 do framework .NET. O sistema realiza a classificação de mensagens de *e-mail* em uma seguintes classes:

(a) ***Phishing:*** Classe que representa os e-mails que possuem alta probabilidade de serem e-mails fraudulentos com o objetivo de obter dados pessoais do usuário.

(b) ***Spam:*** Classe que representa os e-mails que são recebidos de forma não solicitada normalmente contendo material publicitário.

(c) **Seguro:** Classe que representa os e-mails que não se enquadram em nenhuma das categorias anteriores e que podem ser considerados seguros por não conterem conteúdo agressivo ao usuário.

Para a criação das regras e validação das mensagens são utilizados atributos previamente avaliados em estudos realizados por FETTE et. Al., (2007) :

(a) **Contém links:** um link em um e-mail pode indicar que este não é seguro.

(b) **URLs formadas por endereço IP:** empresas reais não utilizam links com um endereço IP.

(c) **Contém javascript:** e-mails com código javascript podem ser utilizados para esconder informações.

(d) **Concentração de termos de *spam*:** calculada a partir de uma lista de termos.

(e) **Concentração de termos de *phishing*:** calculada a partir de uma lista de termos.

(f) **Concentração de termos de mensagens seguras:** calculada a partir de uma lista de termos.

4.1 Treinamento e Testes

Nas etapas de treinamento e teste foram elaborados *datasets* de e-mails contendo mensagens de *phishing*, *spam* e seguras previamente classificadas. Para a criação dos *datasets* foram utilizadas mensagens das bases de Nazário (Nazário, 2007), Ling-spam (Androutsopoulos,2000) e SpamAssasin (Apache Software, 2003). As mensagens foram selecionadas aleatoriamente e divididas da seguinte forma: 500 mensagens para cada uma das três classes de mensagens de treinamento e 200 mensagens para cada uma das três classes de teste.

A partir dos dados de treinamento, o algoritmo das abelhas ABC recebe cada instância com o objetivo de encontrar uma relação entre os atributos da instância e a classe a que pertence e assim definir um conjunto de regras. Com um conjunto de regras já definidos, estas regras vão ser aplicadas ao *dataset* de teste e assim produzir resultados que posteriormente são utilizados para poder avaliar e melhorar a performance do algoritmo. A figura 1 apresenta as etapas da aprendizagem de regras (treinamento).

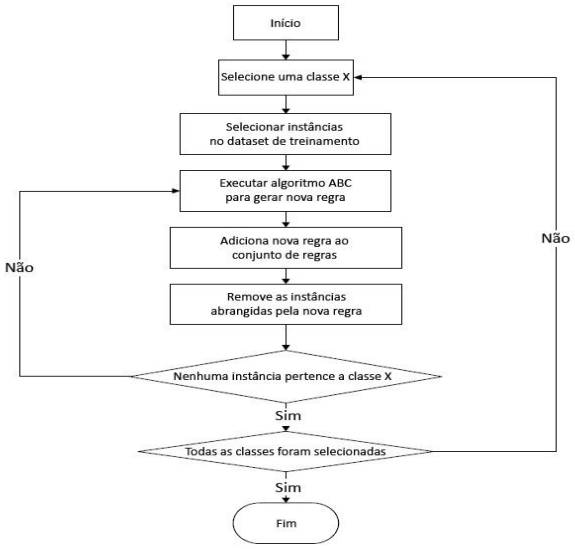

Figura 1. Fluxograma da descoberta de regra do algoritmo ABC (Shukran et al., 2011)

4.2 Classificação de uma Nova Mensagem

A classificação das mensagens se dá através da avaliação das regras que foram criadas nas etapas do treinamento. Em termos do algoritmo ABC, considera-se que a análise da mensagem é realizada pela abelha trabalhadora. Dentro da colmeia existem as regras que foram trazidas pelas abelhas exploradoras. Cada abelha trabalhadora memoriza uma regra e analisa a fonte de alimento, que no caso é a mensagem que deve ser classificada. A abelha que conseguir o maior índice de aproveitamento da regra que possui na memória, classificará a mensagem de acordo com a classe a que sua regra pertence.

Uma abelha trabalhadora executa o um procedimento para definir o aproveitamento de uma regra:

(a) Define uma variável para pontuação com valor inicial em zero.

(b) Verifica se o valor de algum dos três atributos especiais definidos (*javascript*, contém *link* e *link* com endereço IP) da regra possuem o mesmo valor que a mensagem. Em caso positivo a variável da pontuação recebe 2 pontos e em caso negativo não recebe pontuação nenhuma.

(c) Verifica palavras contidas na mensagem. Para cada palavra que estiver presente na mensagem, a variável de pontuação recebe 1 ponto, em caso negativo não recebe pontuação nenhuma.

(d) Após analisar todos os atributos é calculada a pontuação obtida.

(e) Uma vez que todas as regras foram avaliadas, então são comparados os aproveitamentos obtidos e assim definida qual regra melhor se encaixa para a classificação da mensagem.

O processo do cálculo do aproveitamento de uma regra para uma mensagem é representando na figura 2. Pode-se verificar como a quantidade de atributos é obtida e como os seus pesos são distribuídos. A mensagem do exemplo alcançou nove pontos (sobre quatorze possíveis da regra). Sendo assim, a taxa de compatibilidade foi de 64.28% para a regra aplicada.

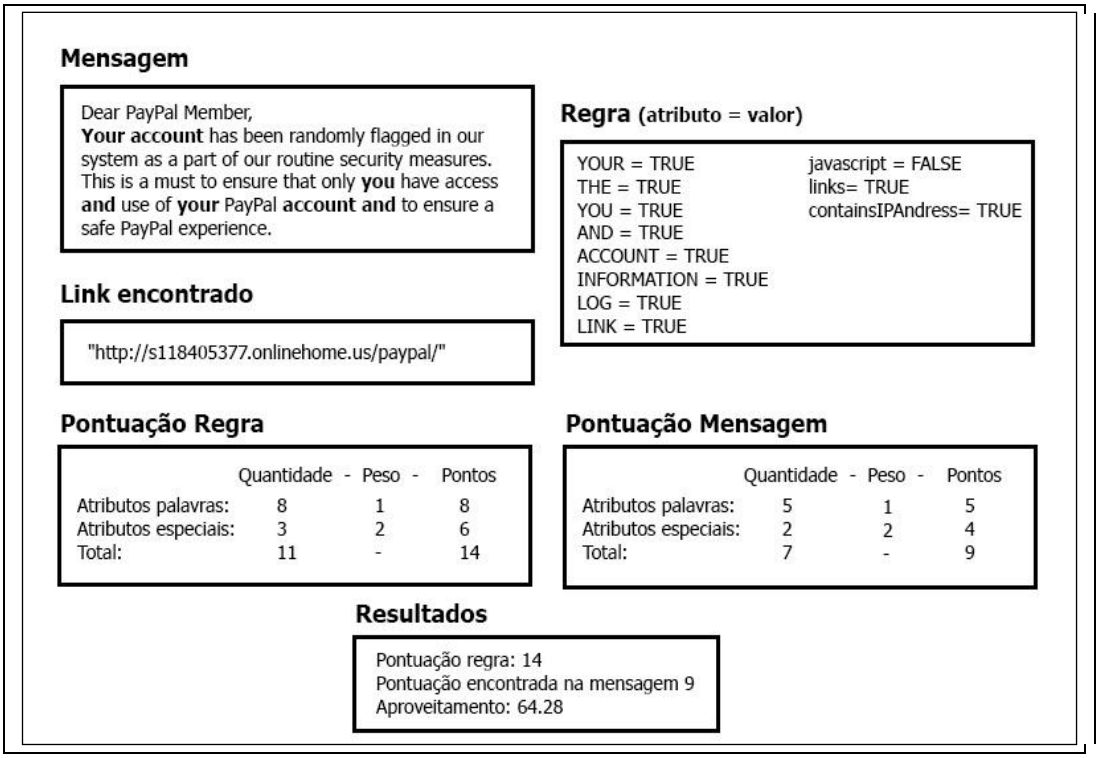

Figura 2. Representação da avaliação de uma regra.

4.3 Teste e Análise dos Resultados

Foram testados três cenários: treinamento do sistema, testes e classificação de mensagens. No primeiro cenário foi realizado o treinamento. Como resultado desta etapa foram criadas as regras que serão utilizadas para a classificação das mensagens. Foi produzido um relatório com os resultados do treinamento que são exibidos na interface do sistema e um relatório completo com a classificação para cada uma das mensagens (na forma de *log*). No cenário 2 foram utilizados os datasets de teste para verificação do desempenho da aprendizagem das regras. Após a conclusão dos testes são gerados relatórios com os resultados da classificação cada uma das mensagens. No cenário 3 são feitas solicitações de classificação de mensagem distintas das utilizadas previamente. O resultado desta etapa é classificação da mensagem selecionada como *spam*, *phishing* ou mensagem segura.

5. RESULTADOS E CONCLUSÕES

Esta seção apresenta os resultados da avaliação do *dataset* de teste com base nas regras produzidas na etapa de treinamento. Para isso foram gerados casos com a quantidade de termos diferentes para comparação e utilizando a técnica de validação *cross-validation*. A configuração que obteve o melhor resultado foi posteriormente utilizada para comparação com outros métodos de classificação.

A tabela 1 apresenta os resultados obtidos na etapa de treinamento. Os resultado exibidos representam a convergência média do sistema após 5 processamentos. Em cada uma das execuções o sistema foi configurado para utilizar diferentes quantidades de atributos mais significativos: 20, 50, 80, 100 e 150.

Nota-se que o melhor resultado obtido foi utilizando 80 atributos, com a taxa de acerto das mensagens de *phishing* em 93,6%, *spam* 89,5% e seguras 87,5%. Por este motivo a configuração com 80 atributos foi escolhida para realizar as comparações com outros algoritmos.

Tabela 1. Taxas de acerto obtidas após diversas execuções com atributos distintos

	20 atributos	50 atributos	80 atributos	100 atributos	150 atributos
Phishing	58,5	90,5	93,5	92	93
Spam	26	73,5	89,5	92	84
Seguras	93	80,5	87,5	75	87

A fim de se realizar a comparação com principais algoritmos de classificação de dados utilizou-se os resultados do estudo de Hepp (2010) que analisou os mesmos datasets para mensagens de *phishing*. Portanto, para esta comparação utilizou-se apenas as mensagens de *phishing* e seguras (tabela 2). Do total de 198 mensagens de *phishing*, 187 foram classificadas corretamente o que gerou uma taxa de acerto 96,79%. As mensagens de *phishing* classificadas como seguras foram 11 o que gerou uma taxa de erro de 3,2%. Hepp analisou os algoritmos *Multilayer Perceptron* (MP), J48, *BayesNet* e E-M para o mesmo problema. Em um teste utilizou-se exatamente o mesmo *dataset* de Hepp (514 mensagens obtidas de bases de dados públicas e selecionadas de forma aleatória). Das 514 mensagens, 122 foram mensagens de *phishing* e 392 mensagens seguras. Foram ainda aplicadas configurações específicas aos algoritmos que estão definidas em seu trabalho. A taxa de acerto do SCMIE foi satisfatória, sendo levemente melhor que o algoritmo MP que havia obtido o melhor resultado das mensagens classificadas de forma correta.

Tabela 2. Comparação de acertos entre algoritmos

Algoritmo	Acerto	Erro
SCMIE	96,79	3,2
MP	96,5	3,5
J48	94,94	5,06
BayesNet	93,19	6,81
E-M	89,69	10,31

Considerando-se que esta é uma primeira versão do sistema e a técnica do algoritmo das abelhas não é tão popularmente utilizada para classificação de dados, os resultados podem ser considerados altamente satisfatórios. O índice das mensagens classificadas corretamente foi próximo a de outros algoritmos de classificação, sendo inclusive superior em alguns casos. Para finalizar, percebe-se que através da modelagem e do desenvolvimento do sistema foi possível perceber que o algoritmo das abelhas pode ser adaptado a outras áreas de classificação.

REFERÊNCIAS

Almeida, T.; Yamakami, A., 2012. Occam's razor-based spam filter. *Journal of Internet Services and Applications*, Vol.3(3), pp.245-253.

Du Toit, T. ; Kruger, H., 2012. Filtering spam e-mail with Generalized Additive Neural Networks. . ISSA, page 1-8. IEEE.

Han, J.; Kamber, M.; Pei, J., 2012. Data Mining Concepts and Techniques, Morgan Kaufmann publications, USA.

Hepp, F. S., 2010. Aprendizagem Automática de Phishing por Mineração de Dados. Universidade de Caxias do Sul, Monografia.

Jadhav, H.T. ; Sharma, U. ; Patel, J. ; Roy, R.. 2012. Gbest guided artificial bee colony algorithm for emission minimization incorporating wind power. 11th International Conference on Environment and Electrical Engineering (EEEIC), pp. 1064-1069.

Kang, F.; Li, J.; Li, H., 2013. Artificial bee colony algorithm and pattern search hybridized for global optimization. Applied Soft Computing, Volume 13, Issue 4, pp 1781–1791.

Karaboga, D., 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization. (TECHNICAL REPORT-TR06).

Karaboga, D.; Ozturk, C., 2009. A Novel Clustering approach: Artificial Bee Colony (ABC) Algorithm – Elsevier B.V.

Karaboga, D.; Ozturk, C., 2010. Fuzzy Clustering With Artificial Bee Colony Algorithm – Scientific Research and Eassays Vol. 5(14), pp. 1899-1902.

Kaspersky Lab. Disponível em: <http://www.kaspersky.com>. Acesso em 26 mar. 2012.

Kumbhar, P.Y.; Krishnan, S., 2011. Use of Artificial Bee Colony (ABC) Algorithm in Artificial Neural Network Synthesis – Int Journal of Advanced Engineering Sciences and Technologies, Vol No. 11, Issue No. 1, pp.162-171.

Laorden, C. ; Santos, I. ; Sanz, B. ; Alvarez, G. ; Bringas, P. , 2012. Word sense disambiguation for spam filtering Electronic Commerce Research and Applications, Vol.11(3), pp.290-298.

Lee, T. ; Cheng, J. ; Jiang, L., 2012. A New Artificial Bee Colony Based Clustering Method and Its Application to the Business Failure Prediction. International Symposium on Computer, Consumer and Control (IS3C), pp. 72-75.

Li, M. ; Park, Y. ; Ma, R. ; Huang, H., 2012. Business email classification using incremental subspace learning. 21st International Conference on Pattern Recognition (ICPR), pp. 625- 628.

Méndez, J.R.; Reboiro-Jato, M.; Díaz, F. ; Díaz, E.; Fdez-Riverola, F., 2012. Grindstone4Spam: an optimization toolkit for boosting e-mail classification. Journal of Systems and Software. Volume 85/12, pp.2909-2920.

Moniza, P. ; Asha, P., 2012 An assortment of spam detection system. International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp. 860-867.

Pan, Q. ; Wang, L. ; Mao, K. ; Zhao, J. ; Zhang, M., 2013. An Effective Artificial Bee Colony Algorithm for a Real-World Hybrid Flowshop Problem in Steelmaking Process. IEEE Transactions on Automation Science and Engineering, Volume: 10 , Issue: 2, pp. 307-322.

Pérez-Díaz, N. ; Ruano-Ordás, D. ; Méndez, J. ; Gálvez, J. ; Fdez-Riverola, F., 2012. Rough sets for spam filtering: Selecting appropriate decision rules for boundary e-mail classification. Applied Soft Computing Journal, 2012, Vol.12(11), pp. 3671-3682.

Rakshit, P. ; Das, P. ; Konar, A. ; Nasipuri, M. ; Janarthanan, R., 2012. A recurrent fuzzy neural model of a gene regulatory network for knowledge extraction using invasive weed and artificial bee colony optimization algorithm. 1st International Conference on Recent Advances in Information Technology (RAIT), pp. 385-391.

Razmara, M. ; Asadi, B. ; Narouei, M. ; Ahmadi, M., 2012 A novel approach toward spam detection based on iterative patterns. 2nd International eConference on Computer and Knowledge Engineering (ICCKE), pp. 318-323.

Salcedo-Campos, F.; Diaz-Verdejo, J. ; Garcia-Teodoro, P., 2012. Segmental parameterisation and statistical modelling of e-mail headers for spam detection. Information Sciences, Vol.195, p.45(17).

Sharma, T.K. ; Pant, M., 2012. Golden Search Based Artificial Bee Colony Algorithm and Its Application to Solve Engineering Design Problems. Second International Conference on Advanced Computing & Communication Technologies (ACCT), pp 156-160.

Shukran, M.A.M. Chung, Y.Y; Yeh, W.-C.; Wahdi, N.; Zaidi, A.M.A., 2011. Artificial Bee Colony based Data Mining Algorithms for Classification Tasks, Canadian Center of Science and Education.