Evolving Fuzzy Set-based and Cloud-based Unsupervised Classifiers for Spam Detection

Eduardo Soares, Cristiano Garcia, Ricardo Poucas, Heloisa Camargo, Daniel Leite

***Resumo*—Technological advancements has made individuals and organizations more dependent on e-mails to communicate and share information. The increasing use of e-mails has led to an increased production of unsolicited commercial messages, known as spam. Spam classification systems able to self-adapt over time, with no human intervention, are rare. Adaptation is interesting as spams vary over time due to the use of different message-masking techniques. Moreover, classification models that handle large volumes of data are essential. Evolving intelligent systems are able to adapt their parameters and structure according to the data stream. This study applies the evolving methods TEDA (Typicality and Eccentricity based Data Analytics) and FBeM (Fuzzy Set-Based Evolving Modeling) for online unsupervised classification of spams. TEDA and FBeM are compared in terms of accuracy, model compactness, and processing time. For dimensionality reduction, a non-parametric Spearman-correlation-based feature selection method is employed. A dataset containing 25,745 samples, being 7,830 spams and 17,915 legitimate e-mails, is considered. 711 features extracted from an e-mail server describe each sample.**

***Index Terms*—Unsupervised Classification, Spam Detection, Evolving Intelligent Systems, Clustering, Data Streams.**

I. Introdução

COM os avanços recentes das tecnologias de informação e de redes, organizações e indivíduos dependem cada vez mais de e-mails para se comunicarem e para compartilhar informações. No entanto, a crescente utilização de e-mails acarretou problemas causados por mensagens não solicitadas, conhecidas como spams. O envio de spam através de e-mails sobrecarrega servidores SMTP (*Simple Mail Transfer Protocol*), desperdiça banda, aborrece usuários, e gera prejuízos morais e até financeiros [1]. Devido à grande quantidade de técnicas de envio e mascaramento de mensagens por parte de spammers e às diferenças do conteúdo da mensagem que variam com a cultura, época, entre outros, o problema de detecção de spams está longe de ser solucionado.

De acordo com [2], em uma escala global, a porcentagem de mensagens spam chega a 95% do total de mensagens trafegadas. Já um estudo divulgado pela Symantec revelou que no ano de 2010 o percentual de spams foi de 89,1% do número total de e-mails [3]. Em 2010, a quantidade registrada de mensagens spam foi de cerca de 62 bilhões [3].

* Corresponding author

E. Soares* e H. Camargo, Departamento de Computação, Universidade Federal de São Carlos, São Carlos, SP, Brasil, edu.soares999@gmail.com, heloisa@dc.ufscar.br.

C. Garcia, R. Poucas e D. Leite, Departamento de Engenharia, Universidade Federal de Lavras, Lavras, MG, Brasil, cristianooo@gmail.com, ricardo.poucas@gmail.com, daniel.leite@deg.ufla.br.

O problema de classificação de e-mails de texto se refere à discriminação entre spams e e-mails legítimos. Um sistema anti-spam verifica as palavras contidas no cabeçalho e no corpo do e-mail e compara se as mesmas ocorrem com certa frequência em mensagens spam típicas. Em geral, faz-se uso de uma base de dados contendo palavras recorrentes em spams [1]. Tal técnica é, por vezes, ineficaz visto que os spammers trocam caracteres corretos por outros, mantendo o sentido original. Isso confunde sistemas classificadores já que a combinação de caracteres é diferente.

De um modo geral, métodos de reconhecimento de padrões [4], [5] têm sido investigados para classificação de spam. Estes métodos têm obtido taxas relativamente altas de classificações corretas em contexto estacionário. Ambientes não-estacionários requerem a adaptação de parâmetros de modelos classificadores a partir do uso de algoritmos incrementais de aprendizado de máquina [6]. A idéia é que modelos classificadores acompanhem as mudanças dos padrões de spam ao longo do tempo [7], [8], [9]. Atualmente, a detecção de spams é feita marjoritariamente por algoritmos de reconhecimento de padrões não iterativos como SVM e Naive Bayes, ou seja, algoritmos que requerem retreinamento constante para se adaptarem às mudanças nos padrões de envio de spam [10].

No presente trabalho são considerados modelos inteligentes evolutivos, que aprendem em tempo real e não exigem retreinamento, baseados em nuvem de dados – *Typicality and Eccentricity Data Analytics* (TEDA) [7] – e em grânulos e regras fuzzy – *Fuzzy-Set-Based evolving Modeling* (FBeM) [8] – para classificação de e-mails de texto, legítimos e spams. TEDA conta com um método de aprendizado que se baseia nos conceitos de nuvem de dados, excentricidade e tipicidade. A idéia é que nuvens TEDA não têm um formato geométrico específico, como clusters convencionais. FBeM usa objetos fuzzy granulares para sumarizar a informação extraída de um fluxo de dados. FBeM é baseado no conceito de cobertura (granulação fuzzy) do espaço dos dados. Suas regras são interpretáveis linguisticamente e, portanto, úteis para auxílio à tomada de decisão [8].

Os métodos TEDA e FBeM são comparados em termos de acurácia de classificação, tempo de processamento e tamanho da base de regras gerada. Ademais, um método de redução da dimensionalidade dos dados baseado na correlação de Spearman, conforme [7], foi considerado para ordenação e seleção dos atributos mais discriminativos de classes. Portanto, as contribuições deste trabalho são as seguintes:

- aplicação de métodos de inteligência artificial, inter-

pretáveis e adaptáveis, para a classificação de e-mails legítimos e spams de forma *online*;
- utilização de um seletor de atributos baseado na correlação de Spearman para a redução da dimensionalidade da base de dados;
- proposta de uma nova base de dados para classificação de e-mail legítimos utilizando dados de um servidor real e o filtro *SpamAssassin*.

Após esta introdução, a próxima seção apresenta a literatura relacionada. A Seção 3 descreve os métodos TEDA e FBeM para agrupamento de fluxos de dados e classificação de padrões, e o método de Spearman para redução do número de atributos. A Seção 4 investiga o desempenho dos classificadores evolutivos baseados em grânulos e nuvens de dados quando submetidos a uma grande base de dados real, coletada ao longo de 2 anos e 10 meses, contendo e-mails legítimos e spams. A conclusão e sugestões para trabalhos futuros se encontram na Seção 5.

II. Trabalhos Relacionados

São consideradas técnicas anti-spam: (i) listas de bloqueio (bloqueio de mensagens provindas de endereços de IPs contidos em listas negras); (ii) palavras-chave (bloqueio de mensagens contendo em seu corpo determinadas sequências de caracteres típicas em e-mails spam); (iii) recusa intencional (recusa em primeira instância, e adição do e-mail a uma *Gray List*, para posterior aceite em caso de re-envio); (iv) políticas de remetente (prevenção da substituição do cabeçalho do e-mail remetente); (v) técnicas baseadas em assinatura (o destinatário deve reconhecer a assinatura digital do remetente); e (vi) reconhecimento inteligente de padrões de spam. A última consiste essencialmente em modelos classificadores cujos parâmetros e estrutura são identificados a partir de um conjunto de amostras numéricas extraídas de e-mails e da combinação de algumas ou várias das técnicas mencionadas acima. Os modelos devem prover superfícies de separação entre as classes 'e-mail legítimo' e 'e-mail spam' em um espaço multidimensional.

O modelo classificador Naive Bayes (NB) é um dos mais usados em e-mails de texto [11]. Por ser um classificador simples e que apresenta boas taxas de detecções corretas de spam, NB é muito utilizado em sistemas comerciais de filtragem [12]. Em [13], comparou-se três métodos baseados em NB, viz. *Locally Weighted Naive Bayes* (LWNB), *Discretized Locally Weighted Naive Bayes* (LWNBD) e *Lazy Bayesian Rules* (LBR), além de outros métodos como *Averaged One-Dependence Estimator* (AODE) e *K-nearest Neighbours With Distance Weighting* (KNNDW). Propuseram-se melhorias e customizações no método NB original para tratar o problema de spam. Várias bases de dados de e-mails foram avaliadas, apontando os métodos NB modificados como aqueles com as maiores taxas de classificações corretas de spam.

Em [14] foi apresentada uma abordagem que leva em consideração a característica dinâmica de spams, i.e., consideraram-se várias técnicas utilizadas por spammers para burlar sistemas anti-spam. Em pré-processamento, mensagens são transformadas e ordenadas em bases de dados numéricas utilizando o método Huffman Adaptativo baseado em árvore. Um dos benefícios da utilização de árvores adaptativas é a possibilidade de acrescentar novas folhas sem a necessidade de se criar uma nova árvore. Na classificação foi utilizado o método *Support Vector Machine* (SVM) [15], e a adaptação contou com uma técnica de envelhecimento exponencial. Em [14], somente as mensagens mais recentes são consideradas. A abordagem considera que spams mudam ao longo do tempo. A ênfase é dada, então, às mensagens mais recentes.

Em [16] enfatiza-se que modelos que se baseiam em algoritmos de aprendizado de máquina *offline* usualmente falham quando um novo tipo de mensagem spam é recebido, sendo que o sistema anti-spam já se encontra em operação. É necessário rever e re-projetar um novo modelo capaz de detectar o novo padrão de mensagem. Porém, até que isso ocorra, o e-mail spam pode alcançar vários usuários. De uma maneira geral, os trabalhos procuram comparar modelos classificadores para verificar quais são os métodos de aprendizado e as estruturas de modelo mais eficientes [17]. Contudo, tais métodos e modelos muitas vezes estão ao alcance dos spammers, que podem explorar suas vulnerabilidades.

Em [17] destaca-se que a aplicação de redes neurais artificiais (RNA) para detecção e filtragem de e-mails spam tem obtido melhores resultados que máquinas de vetor suporte, SVM. Adicionalmente, o método *Random Forest* (RF), que é constituído de um comitê de árvores de decisão, é discutido para a classificação supervisionada e semi-supervisionada de e-mails spam. Modelos RF têm apresentado melhor acurácia do que modelos NB, SVM e RNA [17].

Atualmente, estudos relacionados à classificação *online* de spam em e-mails de texto utilizando modelos adaptativos ao longo do tempo são raros. Os sistemas inteligentes evolutivos TEDA e FBeM, tratados na próxima seção, são abordagens apropriadas para ambiente não-estacionário, isto é, sujeitos à mudanças. Em outras palavras, modelos TEDA e FBeM são equipados com algoritmos incrementais para adaptação de seus parâmetros e estrutura de regras. Conforme novos padrões de spam surgem em um fluxo de dados *online*, a essência da informação é armazenada em novos modelos locais (nuvens ou grânulos). Ao mesmo tempo, os padrões identificados anteriormente são mantidos na memória. Os métodos e modelos mencionados nesta seção não possuem capacidade de evolução recursiva de parâmetros e estrutura de regras ao longo do tempo, conforme as propostas de TEDA e FBeM.

III. Classificadores Evolutivos

Modelos classificadores evolutivos têm como principal característica a capacidade de adaptação a novidades [18], [19], [20], [21]. A adaptação no contexto de fluxos de dados *online* é obtida a partir de algoritmos de aprendizado de máquina recursivos que visam melhorar gradativamente ou manter a acurácia do modelo em ambiente variante no tempo, i.e., em ambiente em que a distribuição que gera e governa os dados muda. Além disso, o processamento de dados de teste por parte de classificadores evolutivos pode se iniciar a partir de uma base de conhecimento vazia. Em outras palavras, os

parâmetros de modelos locais e as propriedades dos dados não precisam ser conhecidos *a priori*. As abordagens TEDA e FBeM, tratadas no presente trabalho, são orientadas e foram customizadas para problemas de classificação em ambiente não-estacionário. Modelos de detecção de spam são obtidos sem a necessidade da introdução de conhecimento humano prévio sobre o domínio do problema de spam.

A. TEDA Class

TEDA é um método de modelagem baseado em regras capaz de lidar incrementalmente com grandes volumes de dados [7], [22]. O aprendizado é baseado nos conceitos de excentricidade e tipicidade, relacionados às características de proximidade e densidade dos dados. O método se baseia em estimativas recursivas de densidade [23].

Seja $X = [x_{(1)}, ..., x_{(K)}]'$, $X \in \Re^{K \times n}$, um conjunto de dados; e $k = 1, ..., K$, o índice de tempo. A soma das distâncias, $\pi(.)$, da k-ésima amostra, $x_{(k)}$, para todas as amostras de X é dada por

$$\pi(x_{(k)}) = \sum_{j=1}^{K} d(x_{(k)}, x_{(j)}), \tag{1}$$

onde $d(.)$ pode ser qualquer métrica de distância. Recursivamente, $\pi(.)$ no índice de tempo k, i.e., a soma das distâncias entre as k-ésimas amostras de dados e a amostra anterior de um fluxo de dados, é dada por

$$\pi(x_{(k)}) = \pi(x_{(k-1)}) + d(x_{(k)}, x_{(k-1)}). \tag{2}$$

A excentricidade $\varepsilon(.)$ de $x_{(k)}$ em relação a todas as amostras contidas na base de dados é obtida de

$$\varepsilon(x_{(k)}) = \frac{2\pi(x_{(k)})}{\sum_{j=1}^{K} \pi(x_{(j)})}. \tag{3}$$

A tipicalidade $\gamma(.)$ é o complemento da excentricidade, i.e.,

$$\gamma(x_{(k)}) = 1 - \varepsilon(x_{(k)}). \tag{4}$$

Uma regra TEDA é descrita por

$$R_i(x) : \text{Se } (x \sim x_i^*), \text{ então Classe } z_k \tag{5}$$

onde $i = 1, ..., N$; e $\sim$ indica o relacionamento de uma amostra x à regra R_i; e x_i^* é um ponto representativo na nuvem de dados (ponto focal). Em particular, a regra R_i mapeia $x \in \Re^n$ na classe z_k, que rotula a i-ésima nuvem de dados.

A excentricidade local, ε_i, e a tipicalidade local, γ_i, são definidas conforme as amostras são assimiladas à i-ésima nuvem de dados. Uma nuvem e uma regra correspondente são criadas quando o valor da tipicalidade local $\gamma_i \; \forall i$ é menor que um limiar α, i.e.,

$$\max_{i=1,...,N} \gamma_i(x_{(k)}) \leq \alpha. \tag{6}$$

Caso contrário, a amostra $x_{(k)}$ é atribuída a regra mais ativa. O nível de ativação da i-ésima regra R_i para uma amostra $x_{(k)}$ é

$$w_i(x_{(k)}) = \frac{\gamma_i(x_{(k)})}{\sum_{j=1}^{N} \gamma_j(x_{(k)})}. \tag{7}$$

Assim que uma amostra é atribuida a uma regra R_{i^*}, a excentricidade local, ε_{i^*}, e a tipicalidade local, γ_{i^*}, são atualizadas.

Detalhes da formulação TEDA, como um algoritmo geral, podem ser encontradas em [7], [22], [23]. O algoritmo é sumarizado como segue.

Algoritmo TEDA Class

DETERMINAR α, $N = 0$;
LER a primeira amostra $x_{(1)}$;
CRIAR a regra R_N, $N = N + 1$;
PARA k = 2, ...
  LER amostra $x_{(k)}$;
  CALCULAR $\gamma_i(x_{(k)})$ e $\varepsilon_i(x_{(k)})$, $i = 1, ..., N$;
  CALCULAR a ativação $w_i(x_{(k)})$, $i = 1, ..., N$;
  FORNECER estimação de classe, $\bar{y}_{(k)}$;
  SE Eq. (6) é verdade
    CRIAR regra R_{N+1}, $N = N + 1$;
  SENÃO
    ATUALIZAR γ_{i^*} e ε_{i^*}, $i^* = argmax \; w_i(x_{(k)})$;
  FIM
FIM

B. FBeM Class

O modelo FBeM consiste de um conjunto de regras fuzzy. FBeM é capaz de lidar com problemas *online* em que as amostras são ilimitadas e algoritmos *offline* convencionais apresentam problema de escalabilidade. FBeM é baseado em aprendizado incremental e fornece fronteiras não-lineares de separação de classes [24]. O método não requer dados *a priori* para aprender. Regras e grânulos de informação fuzzy são criados dinamicamente e adaptados ao longo do tempo. Para cada grânulo de informação existe uma regra correspondente. A parte antecedente das regras FBeM consiste de hiper-retângulos fuzzy, e a parte consequente é uma classe basicamente. Neste trabalho, as regras FBeM têm a forma

$$R_i(x) : \text{Se } (x_1 \text{ é } A_1^i) \text{ E ... E } (x_n \text{ é } A_n^i)$$
$$\text{Então } (\bar{y} \text{ é } Classe \; z_k)$$

sendo x_j, $j = 1, ..., n$, a variável de entrada; $\bar{y}$ a classe estimada; e R^i, $i = 1, ..., c$, a coleção de regras. $(x, y)^{[h]}$, $h = 1, ...$ é um fluxo de dados; y é a classe verdadeira. A classe verdadeira é conhecida somente após a estimação. Ademais, $A_j^i = (l_j^i, \lambda_j^i, \Lambda_j^i, L_j^i)$, são funções de pertinência trapezoidais criadas e desenvolvidas conforme os dados. A

classe relativa à regra mais ativa para uma amostra $x^{[h]}$ é a saída global do modelo. A regra mais ativa é dada por

$$\alpha^* = S(\alpha^1, ..., \alpha^i, ..., \alpha^c), \tag{8}$$

onde considera-se o operador máximo, $max_{i\ \forall i}$, como conorma triangular S. Além disso,

$$\alpha^i = T(A_1^i, ..., A_j^i, ..., A_n^i), \tag{9}$$

onde adotou-se o operador de agregação mínimo, $min_{i\ \forall i}$, como norma triangular T.

A largura máxima que funções $A_j^i\ \forall i$ podem assumir ao longo do j-ésimo universo de discurso é limitada pela granularidade ρ_j. A largura de A_j^i é dada por

$$wdt(A_j^i) = L_j^i - l_j^i. \tag{10}$$

Logo,

$$wdt(A_j^i) \leq \rho_j,\ i = 1, ..., c. \tag{11}$$

Diferentes valores de ρ produzem diferentes representações do fluxo de dados em diferentes níveis de granularidade. A região de expansão de A_j^i é denotada por

$$E_j^i = [mp(A_j^i) - \frac{\rho_j}{2}, mp(A_j^i) + \frac{\rho_j}{2}], \tag{12}$$

onde

$$mp(A_j^i) = \frac{\lambda_j^i + \Lambda_j^i}{2} \tag{13}$$

é o ponto central de A_j^i. A região E_j^i auxilia na determinação de quando uma amostra $x^{[h]}$ deve ser considerada ou não como pertencente a um grânulo já existente. Para dados normalizados, o valor de ρ varia em $[0, 1]$. Se $\rho = 1$, então o grânulo não é capaz de se expandir. Portanto, o processo de aprendizado cria uma regra para cada amostra de dados, o que pode levar ao overfitting e a um alto número de regras que não generalizam o comportamento dos dados. Se $\rho = 0$, então cria-se um único grânulo para representar todos os dados, o que não é interessante em ambiente não não-estacionário. FBeM utiliza um procedimento rápido para evoluir ρ ao longo do tempo.

Seja r a diferença entre o número de grânulos atual e o número de grânulos a h_r iterações passadas, i.e., $r = c^{[h]} - c^{[h-h_r]}$. Se a quantidade de grânulos aumenta mais rápido do que uma taxa de crescimento η, ou seja, $r > \eta$, então ρ é aumentado conforme

$$\rho^{[h]} = (1 + \frac{r}{h_r})\rho^{[h-h_r]}. \tag{14}$$

Isso controla o valor de ρ e evita o aumento da complexidade do modelo. Se a granularidade ρ cresce a uma taxa menor do que η, ou seja, $r < \eta$, então ρ é reduzido,

$$\rho^{[h]} = (1 - \frac{\eta - r}{h_r})\rho^{[h-h_r]}. \tag{15}$$

A granularidade de modelos FBeM varia no tempo de acordo com o fluxo de dados. A criação de regras ocorre quando uma entrada $x = (x_1, ..., x_n)$ não pertence à região de expansão $(E_1^i, ..., E_n^i)$, $i = 1, ..., c$. Nesse caso,

$$A_j^{c+1} = (l_j^{c+1}, \lambda_j^{c+1}, \Lambda_j^{c+1}, L_j^{c+1}) = (x_j, x_j, x_j, x_j), \tag{16}$$

$j = 1, ..., n$. O número de regras c é acrescido de uma unidade.

A adaptação de parâmetros ocorre quando $x \in E^i$ para algum i. Existem seis possibilidades:

Se $x^{[h]} \in [mp(A_j^i) - \frac{\rho}{2}, l_j^i]$
Então $l_j^i(new) = x^{[h]}$ (expansão do suporte)
Se $x^{[h]} \in]l_j^i, \lambda_j^i]$
Então $\lambda_j^i(new) = x^{[h]}$ (expansão do núcleo)
Se $x^{[h]} \in]\lambda_j^i, mp(A_j^i)]$
Então $\lambda_j^i(new) = x^{[h]}$ (contração do núcleo)
Se $x^{[h]} \in]mp(A_j^i), \Lambda_j^i]$
Então $\Lambda_j^i(new) = x^{[h]}$ (contração do núcleo)
Se $x^{[h]} \in]\Lambda_j^i, L_j^i]$
Então $\Lambda_j^i(new) = x^{[h]}$ (expansão do núcleo)
Se $x^{[h]} \in]L_j^i, mp(A_j^i) + \frac{\rho}{2}]$
Então $L_j^i(new) = x^{[h]}$ (expansão do suporte)

Procedimentos para remoção e mescla de grânulos e regras FBeM podem ser encontrados em [8]. Procedimentos alternativos de mescla são reportados em [25], [26]. O aprendizado FBeM é sumarizado a seguir.

Algoritmo FBeM Class

DETERMINAR ρ, h_r, η;
PARA $h = 1, ...$
  RECEBER amostra $x^{[h]}$;
  PROVER classe estimada $C^{[h]}$;
  // A classe real se torna disponível
  CALCULAR erro de estimação;
  // Adaptação do modelo
  SE $x^{[h]} \notin E^i \forall i$
    CRIAR novo grânulo $\gamma^{[c+1]}$;
  SENÃO
    ADAPTAR grânulo mais ativo γ^{i*};
  FIM
  REMOVER amostra $x^{[h]}$;
  SE $h = \beta h_r,\ \beta = 1, ...$
    MESCLAR grânulos semelhantes;
    ATUALIZAR a granularidade ρ;
    REMOVER grânulos inativos;
  FIM
FIM

C. Correlação de Spearman para Seleção de Atributos

A ordenação e seleção de atributos possibilitam a redução da complexidade de um problema. Um subconjunto de atributos mais relevantes ou discriminativos facilita a interpretação de um modelo, reduz a chance de *overfitting*, e pode produzir melhores resultados ao eliminarem variáveis

que podem confundir o processo descoberta de padrões, tendências e relacionamentos. Seja $(x_1, ..., x_i, ..., x_n)$ um conjunto de atributos, e y a variável dependente. Os atributos podem ser pontuados por

$$S_i = \frac{1}{n}\sum_{j=1}^{n}|\rho(x_i, x_j) - \rho(x_i, y)|, \tag{17}$$

sendo $i = 1, ..., n$; ρ é o coeficiente de Spearman nesta sub-seção [7],

$$\rho(x_i, x_j) = 1 - \frac{6\sum_{k=1}^{K} d_{ij(k)}^2}{K(K^2-1)}, \tag{18}$$

sendo K o número de amostras. Além disso,

$$d_{ij(k)}^2 = rank(x_{i(k)}) - rank(x_{j(k)}) \tag{19}$$

é a diferença entre as classificações das amostras após a ordenação destas de forma ascendente. Se valores consecutivos de uma variável, $x_{i(k)}$ e $x_{i(k+1)}$, são idênticos, então valores fracionários, iguais à média de suas posições na ordem ascendente de valores, são atribuídos. Isto é equivalente a uma média sobre todas as permutações possíveis.

O ρ de Spearman varia de -1 a 1. Um valor de 1 (ou -1) implica que uma função monotonamente crescente (decrescente) descreve a relação entre as variáveis perfeitamente. Uma correlação de 0 indica que não há tendência do valor de uma variável aumentar ou diminuir quando outra variável aumenta. O coeficiente de Spearman aumenta conforme as variáveis tornam-se mais próximas de serem funções monótonas perfeitas umas das outras. Diferentemente do coeficiente de Pearson [4], o ρ de Spearman não assume linearidade ou normalidade.

Quanto mais próximo de 0 é o valor de $S_i \in [-1, 1]$, mais importante é o i-ésimo atributo. A idéia é que um atributo interessante possui baixa correlação com outros atributos, o que implica que este transmite informações únicas e diferentes. Uma alta correlação com a variável dependente indica que um atributo auxilia na previsão. Atributos são deixados de fora do modelo conforme suas pontuações S_i. Quando um atributo gera uma redução significativa de desempenho do modelo, então o processo de eliminação de novos atributos da lista S_i é encerrado.

Como a velocidade de processamento é essencial em ambiente *online*, o método de seleção de atributos descrito nesta seção é útil para reduzir o problema de dimensionalidade.

IV. Metodologia

A base de dados considerada neste trabalho foi obtida a partir de 25745 e-mails enviados para o endereço <cemes@cemes.edu.br> entre julho de 2014 e abril de 2017. O filtro anti-spam *SpamAssassin*, disponível em <http://spamassassin.apache.org/>, foi utilizado para verificar a veracidade dos e-mails. O *SpamAssassin* é um filtro de spam de código aberto amplamente utilizado. Ele é composto de 711 testes que verificam a presença de palavras-chave contidas no corpo do e-mail e realizam testes de cabeçalho e URL (*Uniform Resource Locator*). Por exemplo, o teste 584 verifica se o nome do remetente contém nomes de drogas. Já o teste 362 verifica se o endereço do remetente consta em uma 'lista negra'.

Cada teste realizado é associado à uma pontuação. Quando padrões de spam são identificados, um valor numérico maior do que 0 é atribuído ao teste. Quanto maior é o valor de um teste, maior é a chance de que o e-mail em questão seja spam. A pontuação dos 711 testes é somada. Caso a soma seja maior ou igual a um limiar, cujo valor padrão é 5, então o e-mail é definido como spam. Caso contrário, o e-mail é considerado legítimo.

A partir da API disponível em <http://spam check.postmarkapp.com/> carregou-se os 25745 e-mails coletados e obteve-se uma pontuação nos 711 testes *SpamAssassin* para cada um. O resultado indicou a existência de 7830 e-mails spam e 17915 e-mails legítimos. Os dados obtidos nos 711 testes individuais foram normalizados na escala $[0, 1]$ para que as variáveis fossem consideradas na mesma proporção. A normalização do i-ésimo elemento do k-ésimo vetor de dados é dada por

$$x_{i(k)}^r = \frac{x_{i(k)} - \min\limits_{\forall j}(x_{i(j)})}{\max\limits_{\forall j}(x_{i(j)}) - \min\limits_{\forall j}(x_{i(j)})}, \tag{20}$$

sendo $j = 1, ..., k-1$; e $x_{i(k)}^r \in [0, 1]$.

Classificação de fluxo de dados envolve pares $(x, C)^{[h]}$, $h = 1, ...$ Assume-se C desconhecido em princípio e, portanto, algoritmos de aprendizado não-supervisionados. Não-estacionariedades ao longo do tempo podem ser capturadas devido a adaptação incremental dos classificadores evolutivos TEDA e FBeM. Logo, a fronteira discriminativa de classes de ambos é variável no tempo.

A acurácia dos classificadores é calculada conforme

$$Acc(\%) = \frac{VP + VN}{VP + FP + VN + FN}, \tag{21}$$

onde VP, FP, VN, e FN referem-se a verdadeiros e falsos, positivos e negativos. A classe negativa (classe 0) indica e-mail legítimo. A classe positiva (classe 1) refere-se à e-mails spam. Ademais, utilizou-se também a área sob a curva ROC (*Receiver Operating Characteristic*) para análise da acurácia de classificação [5]. A análise ROC fornece um método conveniente para avaliar a qualidade dos classificadores evolutivos, pois é insensível às mudanças nas distribuições das classes e à proporção de amostras por classe. A área ROC é definida a partir das taxas de VP e FP, conforme

$$VP_{taxa} = \frac{VP}{VP + FN}, \tag{22}$$

$$FP_{taxa} = \frac{FP}{FP + VN}. \tag{23}$$

Logo,

$$A = \int_0^1 VP_{taxa}(t) FP_{taxa}(t) dt, \quad (24)$$

onde A é a área sob a curva ROC; t é o tempo; VP_{taxa} e FP_{taxa} são as taxas de verdadeiro-positivo e falso-positivo, respectivamente.

Para cada classe, a análise ROC aplica um valor limiar no intervalo [0,1] para a saída. Quanto mais a curva ROC se aproxima do limite superior esquerdo do gráfico $VP_{taxa} \times FP_{taxa}$, melhor é o desempenho do classificador.

Os parâmetros iniciais do classificador FBeM são $\rho = 0.06$; $h_r = 10000$; $\eta = 1$; e TEDA usa $m = 2$. Estes valores foram obtidos por tentativa e têm se mostrado apropriados em diferentes simulações. O classificador Naive Bayes também foi avaliado. NB utilizou os parâmetros estabelecidos como padrão. O método para seleção de atributos baseado na correlação de Spearman foi empregado. Os modelos classificadores foram avaliados a partir dos 711 atributos originais, e também a partir de 75%, 50% e 25% dos atributos dentre aqueles indicados como os mais discriminativos de classes. O número de regras (modelos locais – grânulos ou nuvens) desenvolvido pelos diferentes classificadores, assim como o tempo de processamento, também são considerados nas comparações. Os experimentos foram realizados no ambiente Matlab 2018a, em computador com sistema operacional macOS High Sierra, Intel Core i5 com 1,8 GHz e 8GB de memória RAM DDR3.

V. RESULTADOS

Simulações computacionais foram realizadas para avaliar a acurácia dos modelos evolutivos FBeM e TEDA, e também do modelo mais comum na área, Naive Bayes, combinados com o método de seleção de atributos utilizando o seletor de atributos baseado na correlação de Spearman. A Tabela 1 sumariza os resultados obtidos para diferentes conjuntos de atributos. O número de regras ou modelos locais desenvolvidos por FBeM e TEDA foram controlados por meio dos parâmetros ρ e m, respectivamente, para que não mais de 10 modelos locais fossem desenvolvidos. Esta abordagem é útil para uma comparação justa. Maior acurácia pode ser obtida conforme o número de grânulos e nuvens aumenta, ou seja, a partir do uso de classificadores com maior estrutura e parâmetros. O modelo NB não retorna nenhum tipo de regra.

Nota-se na Tabela 1 que os classificadores são mais precisos quando os 711 atributos são considerados. Informações são perdidas quando certos atributos não são utilizados. Entretanto, excluem-se os atributos menos relevantes a partir do método de correlação de Spearman. O classificador granular FBeM obteve maior acurácia do que o classificador TEDA baseado em nuvens em todos os casos. Um número relativamente grande de atributos pôde ser removido sem perda significativa de acurácia. Logo, o método baseado na correlação de Spearman tem um papel fundamental na redução da dimensão do espaço. Como conseqüência, um menor tempo total de processamento é percebido com a redução do número de atributos. O tempo de processamento é um fator importante em modelagem *online*. Por exemplo, um servidor de e-mail deve processar milhares de e-mails por segundo. Ressalta-se que neste contexto, algoritmos de agrupamento *offline*, como o Naive Bayes, K-Means, Fuzzy C-Means, Gustafson-Kessel e Gath-Geva, são infactíveis, pois precisam realizar múltiplas análises (épocas de treinamento) sobre um conjunto de e-mails.

O melhor resultado apresentado na Tabela 1 depende de um compromisso entre a capacidade de processamento disponível no servidor e a acurácia de classificação. Nota-se na Tabela 1, que a capacidade adaptativa de FBeM e TEDA permite que estes captem as mudanças provenientes da fonte geradora de dados. Isto faz com que estes modelos apresentem melhor desempenho em termos de acurácia quando comparados à modelos não-evolutivos, como o modelo NB. Nota-se também vantagem dos modelos evolutivos em tempo de processamento comparados ao modelo NB, pois estes não precisam ser re-treinados. Valores em negrito e sublinhados na Tabela 1 representam o melhor desempenho no critério em questão.

Tabela I
DESEMPENHO COMPARATIVO DOS MODELOS CLASSIFICADORES FBEM, TEDA E NAIVE BAYES

–	**FBeM**			
# Atributos	*Acc* (%)	Tempo (s)	# Regras	A-ROC
711	**94,78**	137,2	10	**0,951**
533	91,32	100,1	7	0,924
355	90,91	89,7	7	0,912
178	81,47	73,7	**7**	0,833
–	**TEDA**			
# Atributos	*Acc* (%)	Tempo (s)	# Regras	A-ROC
711	93,89	92,3	10	0,944
533	89,95	89,6	10	0,905
355	88,97	73,0	10	0,899
178	83,47	**57,3**	10	0,854
–	**Naive Bayes**			
# Atributos	*Acc* (%)	Tempo (s)	# Regras	A-ROC
711	86,57	188,1	-	0,872
533	85,94	180,5	-	0,864
355	85,42	124,4	-	0,859
178	84,49	82,2	-	0,856

As matrizes de confusão dos melhores casos de acurácia de FBeM e TEDA, i.e., usando os 711 atributos originais, são mostrados na Fig. 1. É importante observar alguns pontos. Primeiro, a classificação FBeM é superior a classificação TEDA em todos os quadrantes. Isto se deve essencialmente à particularidades dos algoritmos de aprendizado e da estrutura de suas regras. As áreas sob as curvas ROC (conforme Tabela 1) também demonstram uma maior eficiência de classificação do modelo FBeM com relação ao modelo TEDA. No melhor cenário, FBeM tem área ROC igual a 0,951, enquanto TEDA apresenta área 0,944, a Fig. 2 ilustra a curva ROC para ambos os classificadores.

Outro ponto é que ambas as matrizes de confusão indicam um melhor desempenho de classificação de FBeM e de TEDA

para a classe 0, i.e., 'e-mails legítimos'. Isto se deve ao fato de que e-mails spam mudam de padrão ao longo do tempo. Note que as amostras foram coletadas em um período de 2 anos e 10 meses em um ambiente real. As fronteiras de separação de classes de ambos os modelos classificadores, FBeM e TEDA, são adaptadas após algumas iterações e erros, porém a variação temporal de padrões spam justifica o erro ligeiramente maior para e-mails spam. Um terceiro ponto a se notar é que os quadrantes da diagonal secundária da Figura 1 (em vermelho) são indicadores de erro. Em geral, considera-se ser um erro mais grave indicar que um e-mail é spam quando se trata de um e-mail legítimo já que os usuários não costumam verificar a *caixa de spam*. Do contrário, se um e-mail spam não for filtrado, o usuário ainda pode reconhecer que se trata de um spam. Há um risco associado, porém a exclusão de um e-mail importante pode ser uma falha mais significativa. Nesse aspecto, em 2,3% dos casos, FBeM indicou erroneamente que um e-mail legítimo era spam, enquanto TEDA fez o mesmo em 3,0% dos casos.

Várias mensagens legítimas possuem atributos que se confundem com mensagens spam. Por exemplo, se o usuário assinou uma lista de e-mails em um site de comércio eletrônico, este, por sua vez, pode enviar mensagens contendo promoções, imagens, HTML, o que confunde mecanismos de classificação e até mesmo a nossa noção intuitiva do que é um e-mail spam, dado o conteúdo generalista de um e-mail que contém promoções e links para sites externos. Desta forma, conclui-se que a acurácia de ambas as abordagens, FBeM e TEDA, são 'muito boas' e, acima disso, ambas são factíveis para lidar com o problema de variação temporal de padrões de mensagens spam.

VI. CONCLUSÃO

Classificadores inteligentes evolutivos baseados em grânulos fuzzy, FBeM, e nuvens de dados, TEDA, foram propostos para classificação não-supervisionada de e-mails spam e de e-mails legítimos. Ademais, um método não-paramétrico de seleção de atributos baseado na correlação de Spearman foi empregado para redução da dimensionalidade das amostras.

Experimentos mostraram que os métodos evolutivos foram eficientes na classificação não-supervisionada, apresentando acurácias entre 93% e 95% para os melhores casos, superando o modelo Naive Bayes, comumente utilizado na área. Isso é possível visto que os primeiros são capazes de acompanhar as mudanças da fonte geradora de dados. A análise de matrizes de confusão mostrou maior dificuldade na classificação de instâncias spam. Isto ocorre devido às mudanças na fonte geradora. Em outras palavras, as técnicas utilizadas por spammers variam no tempo. O método FBeM foi ligeiramente superior ao método TEDA em acurácia, conforme comprovado pela análise ROC. No entanto, o método TEDA demonstrou maior velocidade para processar os dados de e-mails, o que pode ser determinante em um ambiente de fluxo de dados em alta freqüência. Em 2,3% dos casos, FBeM indicou erroneamente que um e-mail legítimo era spam, contra 3,0% de indicações equivocadas de TEDA – sendo esta a situação mais crítica.

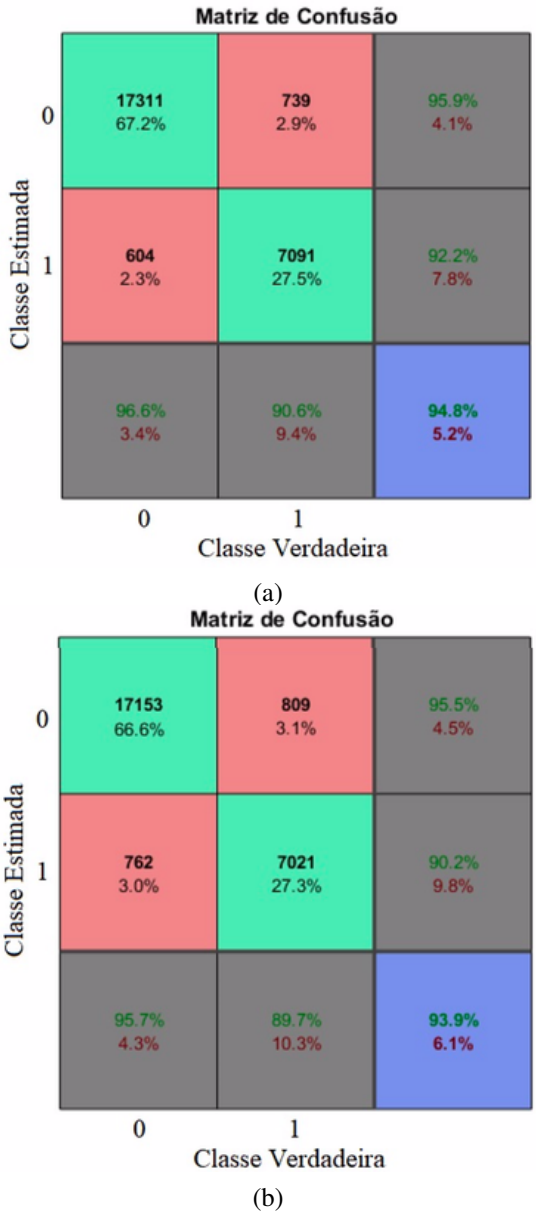


Figura 1. Matriz de Confusão: (a) FBeM, (b) TEDA

Os índices de erro, tanto de FBeM quanto de TEDA, são relativamente muito baixos nesta aplicação. Ambos os métodos evolutivos são adequados para serem utilizados na classificação de spam, visto que os atributos de e-mails spam são modificados ao longo do tempo para lubridiar usuários e sistemas. Essas mudanças nos atributos podem ser assimiladas por abordagens *online*. Abordagens *offline* requerem re-treinamento do modelo anti-spam, o que pode ser custoso e válido por curto prazo.

Trabalhos futuros considerarão seleção incremental de atributos e diferentes variáveis estatísticas associadas às regras fuzzy FBeM e nuvens TEDA. Por exemplo, será avaliada a utilidade de capturar e manter informações sobre especificidade, entropia, correntropia e cardinalidade associadas às amostras de e-mails. Estas variáveis podem ser úteis

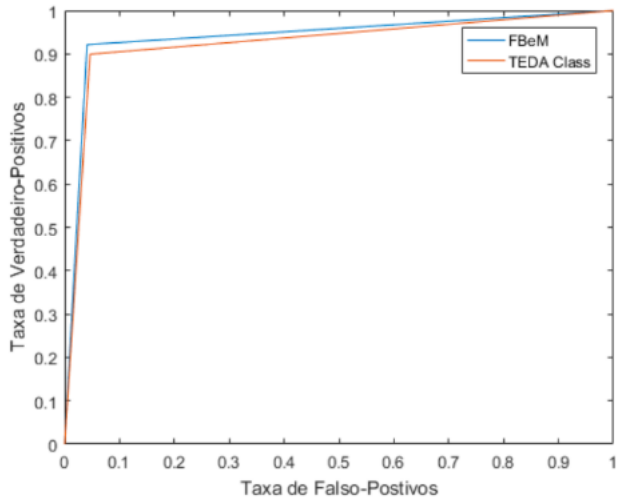


Figura 2. Curva ROC para os classificadores TEDA e FBeM

na otimização estrutural dos modelos evolutivos durante o aprendizado *online*. Processamento de dados incertos, ou seja, de misturas de dados numéricos e fuzzy, também é um tópico. A análise spam em imagens anexas e arquivos GIF também será considerada.

AGRADECIMENTOS

Eduardo Soares é grato à Ford Motor Company e à Lancaster University por uma bolsa de pesquisa. Daniel Leite agradece ao Instituto Serrapilheira pelo suporte e gratificação para pesquisas.

REFERÊNCIAS

[1] C. K. Olivo, A. O. Santin, and L. E. S. Oliveira, "Abordagens para detecção de spam de e-mail," *XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2015.

[2] B. Whitworth and E. Whitworth, "Spam and the social-technical gap," *Computer*, vol. 37, no. 10, pp. 38–45, 2004.

[3] M. Fossi, G. Egan, K. Haley, E. Johnson, T. Mack, T. Adams, J. Blackbird, M. K. Low, D. Mazurek, D. McKinney *et al.*, "Symantec internet security threat report trends for 2010," *Volume XVI*, 2011.

[4] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[6] V. C. Mota, F. A. Damasceno, E. A. Soares, and D. F. Leite, "Fuzzy clustering methods applied to the evaluation of compost bedded pack barns," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[7] E. Soares, P. Costa Jr, B. Costa, and D. Leite, "Ensemble of evolving data clouds and fuzzy models for weather time series prediction," *Applied Soft Computing*, vol. 64, pp. 445–453, 2018.

[8] D. Leite, R. Ballini, P. Costa, and F. Gomide, "Evolving fuzzy granular modeling from nonstationary fuzzy data streams," *Evolving Systems*, vol. 3, no. 2, pp. 65–79, 2012.

[9] E. Soares, V. Mota, R. Poucas, and D. Leite, "Cloud-based evolving intelligent method for weather time series prediction," in *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2017, pp. 1–6.

[10] A. Heydari, M. ali Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: A survey," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3634–3642, 2015.

[11] K.-M. Schneider, "A comparison of event models for naive bayes anti-spam e-mail filtering," in *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 307–314.

[12] C. Chen, Y. Tian, and C. Zhang, "Spam filtering with several novel bayesian classifiers," in *2008 19th International Conference on Pattern Recognition*. IEEE, 2008, pp. 1–4.

[13] E. Frank, M. Hall, and B. Pfahringer, "Locally weighted naive bayes," in *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 249–256.

[14] Í. A. Braga and M. Ladeira, "Um modelo adaptativo para a filtragem de spam," in *VI Encontro Nacional de Inteligência Artificial, Rio de Janeiro–RJ, Anais do XXVII Congresso da Sociedade Brasileira de Computação*, 2007, pp. 1381–1390.

[15] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[16] M. Soranamageswari and C. Meena, "Statistical feature extraction for classification of image spam using artificial neural networks," in *2010 Second International Conference on Machine Learning and Computing*. IEEE, 2010, pp. 101–105.

[17] T. S. Guzella and W. M. Caminhas, "A review of machine learning approaches to spam filtering," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 206–10 222, 2009.

[18] F. J. Ordóñez, J. A. Iglesias, P. De Toledo, A. Ledezma, and A. Sanchis, "Online activity recognition using evolving classifiers," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1248–1255, 2013.

[19] A. Bouchachia, B. Gabrys, and Z. Sahel, "Overview of some incremental learning algorithms," in *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007, pp. 1–6.

[20] J. Iglesias, A. Ledezma, A. Sanchis, and P. Angelov, "Real-time recognition of calling pattern and behaviour of mobile phone users through anomaly detection and dynamically-evolving clustering," *Applied Sciences*, vol. 7, no. 8, p. 798, 2017.

[21] I. Škrjanc, A. S. de Miguel, J. A. Iglesias, A. Ledezma, and D. Dovžan, "Evolving cauchy possibilistic clustering based on cosine similarity for monitoring cyber systems," in *2017 Evolving and Adaptive Intelligent Systems (EAIS)*. IEEE, 2017, pp. 1–5.

[22] P. Angelov and R. Yager, "A new type of simplified fuzzy rule-based system," *International Journal of General Systems*, vol. 41, no. 2, pp. 163–185, 2012.

[23] C. G. Bezerra, B. S. J. Costa, L. A. Guedes, and P. P. Angelov, "An evolving approach to unsupervised and real-time fault detection in industrial processes," *Expert systems with applications*, vol. 63, pp. 134–144, 2016.

[24] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural networks from fuzzy data streams," *Neural Networks*, vol. 38, pp. 1–16, 2013.

[25] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, and F. Gomide, "Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey," *Information Sciences*, 2019.

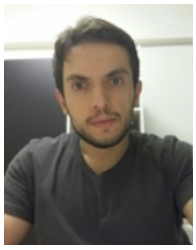
[26] E. A. Soares, H. A. Camargo, S. J. Camargo, and D. F. Leite, "Incremental gaussian granular fuzzy modeling applied to hurricane track forecasting," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2018, pp. 1–8.

Eduardo Soares É doutorando em Ciência da Computação pela Lancaster University (UK), financiado pela Ford Motor Company. É mestre em Engenharia de Sistemas e Automação pela Universidade Federal de Lavras (UFLA), e bacharel em Sistemas de Informação também pela UFLA, com período sanduíche na Kwantlen Polytechnic University (Canadá). Atualmente atua como Associate Lecturer pela Lancaster University, tendo atuado anteriormente como professor substituto pelo Instituto Federal do Sul de Minas (IfSuldeMinas). Ganhou o prêmio *Best Student Paper Award* no *World Congress of Computational Intelligence*, 2018. Sua área de interesse é Inteligência Computacional, com ênfase em sistemas inteligentes evolutivos, sistemas fuzzy e neuro-fuzzy, redes neurais, reconhecimentos de padrões e previsões de séries temporais.

Cristiano Garcia Atualmente é servidor público docente no Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, desde maio de 2019. Trabalhou por 5 anos na Universidade Federal de Lavras, trabalhando na integração de sistemas de informação e outros projetos de TI da Instituição. Possui mestrado em Engenharia de Sistemas e Automação na mesma Instituição. Foi bolsista do Ciência sem Fronteiras, na Hungria, de responsabilidade da Capes, por 11 meses. Foi bolsista PIBIC/FAPEMIG de Iniciação Científica entre 03/2011 e 05/2013. Foi monitor (Algoritmos e Estruturas de Dados I) durante 3 semestres, e é bacharel em Sistemas de Informação pela Universidade Federal de Lavras/MG. É também especialista em Tecnologias para Aplicações Web. Fez curso técnico em Eletrônica com ênfase em Telecomunicações na Escola Técnica de Eletrônica quot;Francisco Moreira da Costaquot;, em Santa Rita do Sapucaí/MG, com duração de 3 anos.

Ricardo Pouças Formado em Sistemas de Informação, Pós Graduado em Engenharia de Sistemas e mestrando em Engenharia de Sistemas e Automação, atua no desenvolvimento de websites, portais educacionais, aplicações WEB, projetos em Redes de Computadores, Tecnologia da Informação e infraestrutura de TI. Professor na Faculdade CEMES de campo belo, foi também professor substituto no Centro Federal de Educação Tecnológica de Nepomuceno . Atualmente trabalha com pesquisas sobre modelagem matemática, sistemas inteligentes evolutivos, sistemas fuzzy e inteligência computacional no laboratório de inteligência computacional da UFLA.

Heloisa Camargo Possui graduação em Ciência da Computação pelo Instituto de Ciências Matemáticas e de Computação Usp São Carlos (1978), mestrado em Ciências da Computação pelo Instituto de Ciências Matemáticas e de Computação Usp São Carlos (1984) e doutorado em Engenharia Elétrica pela Universidade Estadual de Campinas (1993). Realizou pós-doutorado em Edmonton, AB, Canada, na University of Alberta (2001-2002). Atualmente é professora associada da Universidade Federal de São Carlos, no Departamento de Computação. É credenciada no Programa de Pós-Graduação em Ciências da Computação, na UFSCar. Tem experiência na área de Ciência da Computação, com ênfase em Inteligência Artificial, atuando principalmente nos seguintes temas: Lógica Fuzzy, Sistemas Fuzzy Genéticos, Aprendizado de Máquina, Agrupamento de dados, Ontologias Fuzzy e Fuzzy Petri Nets.



Daniel Leite É doutor pela Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas (UNICAMP). Realizou pós-doutorado na Universidade Federal de Minas Gerais (UFMG) e na University of Ljubljana (UL), Eslovênia. Atualmente é professor do Departamento de Engenharia da Universidade Federal de Lavras (UFLA). Recebeu os prêmios *NAFIPS Early Career Award 2017*; *Best PhD Thesis Award da IEEE Computational Intelligence Society* (2017), da NAFIPS (2015), e da Sociedade Brasileira de Computação (2014); e *Outstanding Student Paper na IEEE Joint Conference on Neural Networks* 2009, e *World Congress on Computational Intelligence* 2012. Recebeu o prêmio *Best Student Paper Award*, como orientador, no *World Congress of Computational Intelligence*, 2018. Tem interesse em aprendizado de máquina, inteligência computacional, computação granular, sistemas fuzzy, modelagem de sistemas dinâmicos e reconhecimento de padrões.