DOI: 10.48011/asba.v2i1.1539

Detecção de Anomalias em Logs para Manutenção Preditiva baseada em Sistema Fuzzy Evolutivo Fracamente Supervisionado

Leticia Decker * **and Daniel Leite** **

* *University of Bologna (UNIBO), Itália*
(e-mail: leticia.deckerde@unibo.it)
** *Universidade Federal de Lavras (UFLA), Brasil*
(e-mail: daniel.leite@ufla.br)

Abstract: Anomaly detection of system behaviors is a main issue for predictive maintenance and data security in computer centers. A computing center is any computer network that allows users to share data and resources. Broadly speaking, logs are unstructured data (files), produced by a stochastic and non-stationary process. We propose a real-time computational-intelligence approach to monitor and classify the behavior of log-based systems using a sliding window scheme in conjunction with statistical control chart to find structures and classify logs in relation to anomaly degrees. Classification results are improved through the eGFC (evolving Gaussian Fuzzy Classifier) method, which generates and updates a fuzzy granular model from weak labelled data streams. The log-based monitoring evolving fuzzy system has provided encouraging results regarding accuracy and real-time processing efficiency to online machine learning applications.

Resumo: A detecção de anomalia de comportamento de sistemas é crucial para a manutenção preditiva e a segurança dos dados em centros de computação. Centro de computação é qualquer rede de computadores que permita aos usuários compartilhar dados e recursos computacionais. Em geral, logs são dados não-estruturados (arquivos) produzidos por processos estocásticos não-estacionários. Propomos uma abordagem de inteligência computacional em tempo real para monitorar e classificar o comportamento de sistemas baseado em logs usando um esquema de janela deslizante em conjunto com um gráfico de controle estatístico para encontrar estruturas e classificar logs em relação a graus de anomalia. Os resultados de classificação são melhorados a partir do eGFC (*evolving Gaussian Fuzzy Classifier*), que gera e atualiza um modelo granular fuzzy a partir de fluxo de dados contínuo fracamente rotulados. O sistema fuzzy evolutivo de monitoramento de centrais de computação tem produzido resultados encorajadores em termos de acurácia e eficiência em processamento em tempo real para aplicações de aprendizado de máquina online.

Keywords: Fuzzy System, Evolving System, Machine Learning, Anomaly Detection, Predictive Maintenance.

Palavras-chaves: Sistema Fuzzy, Sistema Evolutivo, Aprendizado de Máquina, Detecção de Anomalia, Manutenção Preditiva.

1. INTRODUÇÃO

1.1 Preliminares

Uma central de computação oferece suporte a infraestruturas de recursos computacionais sob demanda, de forma dinâmica, escalável e nas nuvens. Os recursos são disponíveis diretamente ou através de serviços (Furht and Escalante, 2010). Para manter essa complexa infraestrutura confiável e eficiente, é necessário o desenvolvimento de ferramentas de manutenção preditiva.

A manutenção preditiva é classificada em relação à complexidade das operações e ao momento em que as operações são ativadas. Geralmente, a manutenção de centrais de computação pode ser: (i) reativa, (ii) preventiva, (iii) preditiva, e (iv) avançada. A manutenção reativa acontece quando um conjunto de procedimentos são executados após a ocorrência de uma falha, visando restaurar o comportamento anterior. A preventiva consiste de procedimentos executados para diminuir a probabilidade de uma falha do sistema. A preditiva é baseada na determinação do *status* do sistema para prever eventos futuros. Por fim, a manutenção avançada é uma combinação dos paradigmas supracitados, envolvendo previsão e diagnóstico de falhas (Trojan and Marçal, 2017).

Geralmente, a manutenção é reativa e baseada em análise estatística dos arquivos de log após a ocorrencia da falha; ou preventiva, i.e., em intervalos regulares de tempo. No entanto, recentemente, sistemas online baseados em

inteligência computacional têm sido usados em problemas como detecção de anomalias sob demanda, previsão, aprendizado autônomo e manutenção preditiva em uma grande variedade de aplicações (Lopez et al., 2019; Precup et al., 2014; Lobo et al., 2018; Angelov and Gu, 2018; Pratama et al., 2016; Decker et al., 2020b; Sousa et al., 2012; Sousa, 2011). Uma das abordagens mais promissoras é a evolução de estruturas de modelos fuzzy (Škrjanc et al., 2019; Donato and Serra, 2016; Silva et al., 2013; Casalino et al., 2019; Leite et al., 2020; Silva et al., 2018; Sadaei et al., 2019; Blažič and Škrjanc, 2020) usando algoritmos incrementais de aprendizado de máquina.

A maior parte dos serviços geram registros de log não-estruturados e orientados a eventos que podem ser usados em aplicações de manutenção preditiva. Com isso, o tratamento dos dados caso a caso seria o caminho mais simples, já que soluções de uso geral baseadas no conteúdo dos arquivos de log são um desafio a longo prazo. Em sistemas que geram esse tipo de arquivo, os dados são muito detalhados e textuais, sendo caro extrair informações úteis dos dados brutos. A quantidade de dados é enorme, e grande parte, redundante. Em centros de computação, serviços executados pelos usuários geram dados de log possivelmente em mais de um arquivo, complicando ainda mais as análises e a tomada de decisão.

Já que todas as atividades do centro de computação são registradas em arquivos de log, algoritmos podem usá-los para monitorar eventos e prever o status do sistema. Além disso, no caso preditivo, é possível identificar comportamentos anômalos como uma etapa intermediária do processamento de log (Sousa et al., 2019). Para reduzir o processamento de dados, uma característica comum a esses arquivos pode ser utilizada: o momento do registro de cada linha. Assim sendo, uma suposição razoável é que o nível de atividade do sistema seja proporcional à taxa em que linhas são gravadas no arquivo de log relacionado. Falhas ocorreriam com mais frequência durante períodos de atividade anômala do sistema.

O cenário deste estudo são os centros de computação do WLCG (*Worldwide LHC Computing Grid*), que dão suporte aos experimentos de física do CERN, o maior acelerador de partículas do mundo. Os dados de log utilizados foram gerados pelo maior Tier-1 italiano da rede, localizado em Bologna, mantido pelo centro de computação do Instituto Italiano de Física Nuclear (INFN-CNAF). O WLCG envolve cerca de 170 centros de computação em 42 países. O Tier-1 italiano possui cerca de 40 mil núcleos de CPUs e 130 PB de dados armazenados. A largura de banda de transmissão é superior a 200 Gbps. Atualmente, o Tier-1 coleta dados de cerca de 1200 computadores.

O INFN-CNAF fornece os recursos computacionais para os serviços prestados ao Tier-1. Ele atua como um serviço subjacente ao sistema de gerenciamento de carga de trabalho, permitindo que o agendamento de tarefas acesse diretamente os dados dos experimentos na infraestrutura. Em média, cerca de 100 mil programas são executados por dia nesse centro de computação. Ele também fornece acesso contínuo a uma infraestrutura de rede distribuída (Minarini, 2019).

Como o Tier-1 é uma infraestrutura dedicada aos experimentos de física (Di Girolamo et al., 2020), é necessário otimizar os recursos usados para manter a operacionalidade do sistema. Para tal, uma possível abordagem é identificar quais trechos de log têm prioridade de processamento, baseado em uma maior probabilidade de encontrar informações úteis para a manutenção do sistema. Este trabalho aborda a detecção de anomalias do comportamento do sistema para auxílio à manutenção preditiva.

1.2 Contribuições à Manutenção Preditiva

Atualmente, se encontra em execução o projeto LHC de alta luminosidade (HL-LHC) no CERN. Ele é a principal atualização programada do instituto de pesquisa, visando aumentar a luminosidade usada em um fator de 10 em relação ao design original, sendo que luminosidade é a taxa de colisões potenciais por área. Dessa forma, a quantidade de dados experimentais gerados e suas respectivas análises aumentarão ao menos pelo mesmo fator, intensificando, assim, a complexidade da manutenção do sistema.

Vários projetos de manutenção preditiva têm sido criados nos centros de computação do WLCG para criar ferramentas que extraiam informações dos logs. Um primeiro trabalho baseado em *Elastic Stack Suite* cataloga os registros de log e as anomalias usando uma ferramenta de aprendizado de máquina não-supervisionado (Diotalevi et al., 2019). Outra iniciativa usa abordagens supervisionadas de inteligência computacional para prever anomalias no comportamento do sistema em uma solução ad-hoc (Giommi et al., 2019). Em (Minarini, 2019) foi criado um protótipo para diferenciar comportamentos usuais e anômalos do sistema via máquinas de vetor de suporte.

1.3 Logs do Serviço Storm

O StoRM é o serviço de gerenciamento de armazenamento em disco do Tier-1 em Bologna, útil para otimizar o desempenho de sistemas de arquivos paralelos. Ele possui arquitetura modular baseada em dois componentes principais: Front-end (FE) e Back-end (BE), que são conectados a um sistema de banco de dados. O módulo FE gerencia a autenticação do usuário e armazena/recupera informações no banco de dados através do módulo BE. Por outro lado, o módulo BE é o núcleo do serviço StoRM, executando todas as suas funcionalidades síncronas e assíncronas, além de gerenciar as atividades através da rede. Em geral, cada linha do log BE possui um identificador do operador que solicitou a ação (DN), os endereços dos arquivos envolvidos, e o resultado da respectiva operação. Neste trabalho, os arquivos de log de BE do experimento ATLAS são usados sem nenhum motivo especial.

A Seção 2 descreve o classificador evolutivo baseado em regras fuzzy. Ele aprende padrões espaço-temporais em registros de log. A Seção 3 descreve como os experimentos computacionais foram conduzidos. Os resultados são fornecidos na Seção 4 e as conclusões na Seção 5.

2. EGFC: CLASSIFICADOR EVOLUTIVO FUZZY

Essa seção descreve um método de construção e adaptação incremental da estrutura e parâmetros de um modelo fuzzy a partir de fluxos de dados. O método, denominado eGFC, foi proposto recentemente em (Leite et al., 2020; Decker et al., 2020a). Ele é desenvolvido em um ambiente de computação granular evolutiva – um paradigma amplamente discutido nos últimos anos (Leite et al., 2012, 2013). eGFC usa funções de pertinência Gaussianas para descrever o espaço de dados com grânulos fuzzy (modelos locais) e associar novos rótulos de classe a novos dados. Grânulos são reorganizados (abordagem *scattering*) sempre que necessário para englobar novas informações locais. Um algoritmo de aprendizado incremental constrói e atualiza sua base de regras.

Modelos locais são criados se os novos dados são suficientemente diferentes do conhecimento atual expresso em regras fuzzy. O algoritmo de aprendizado é responsável pela dinâmica de atualização da base de regras, podendo expandir, comprimir, deletar e mesclar os seus grânulos. As regras podem ser revistas a partir de relações inter-grânulos. eGFC fornece fronteiras fuzzy entre classes. As fronteiras são não-lineares, não-abruptas e não-estacionárias.

2.1 Base de Regras

O aprendizado do modelo eGFC não depende de regra inicial nem fase de treinamento. As regras são criadas e atualizadas de acordo com o comportamento do sistema ao longo do tempo. Quando uma amostra de dados é disponível, uma regra pode ser adicionada ou os parâmetros de uma regra muito ativa podem ser adaptados.

Uma regra R^i tem a forma:

SE (x_1 é A_1^i) E ... E (x_n é A_n^i)

ENTÃO (y é C^i)

em que x_j, $j = 1, ..., n$, são atributos, e y é uma classe. O fluxo de dados é dado por $(\mathbf{x}, y)^{[h]}, h = 1, ...$ Além disso, A_j^i, $\forall j$; $i = 1, ..., c$, são funções de pertinência Gaussianas construídas a partir dos dados; e C^i é o rótulo de classe da i-ésima regra. As regras R^i, $\forall i$, formam a base de regras. O número de regras c é variável, o que é uma característica notável da abordagem, visto que é dispensável a suposição de quantas partições existem (Škrjanc et al., 2019).

Uma função Gaussiana normal, $A_j^i = G(\mu_j^i, \sigma_j^i)$ tem valor modal μ_j^i, dispersão σ_j^i e altura 1 (Pedrycz and Gomide, 1998). Essa função é interessante visto que fornece: (i) facilidade de adaptação recursiva a partir de fluxo de dados; (ii) suporte infinito, i.e., uma vez que os dados são previamente desconhecidos, seu suporte se estende por todo o domínio; e (iii) provisão de uma superfície suave para os grânulos fuzzy, $\gamma^i = A_1^i \times ... \times A_j^i \times ... \times A_n^i$, no espaço Cartesiano n-dimensional, obtido pela extensão cilíndrica de Gaussianas unidimensionais, e agregação via T-norma *min* (Pedrycz and Gomide, 1998).

2.2 Adicionando Regras

As regras eGFC não precisam existir *a priori*. Elas são criadas e atualizadas quando dados são disponibilizados. Um novo grânulo γ^{c+1} e sua respectiva regra R^{c+1} são criados se nenhuma das regras existentes $\{R^1, ..., R^c\}$ são suficientemente ativadas por $\mathbf{x}^{[h]}$. Nesse caso, $\mathbf{x}^{[h]}$ traz novas informações. A granularidade $\rho^{[h]} \in [0, 1]$ é um limiar adaptativo que define se é necessária uma nova regra. Se

$$T\left(A_1^i(x_1^{[h]}), ..., A_n^i(x_n^{[h]})\right) \leq \rho^{[h]}, \ \forall i, \ i = 1, ..., c, \quad (1)$$

sendo T é uma norma triangular, então a estrutura do modelo é expandida. A T-norma *min* de Gödel é usada neste artigo. Se $\rho^{[h]} = 0$, então o modelo tem estrutura estável e é incapaz de acompanhar mudanças. Em contraste, se $\rho^{[h]} = 1$, uma regra é criada para cada nova instância, o que não é prático. A adaptabilidade estrutural e paramétrica é balanceada para valores intermediários (Leite et al., 2013).

O limiar $\rho^{[h]}$ limita o quão grande os grânulos podem ser. Diferentes valores impactam na acurácia e compacidade do modelo, resultando em diferentes perspectivas granulares do mesmo problema. A Seção 2.3 detalha o procedimento de atualização de $\rho^{[h]}$.

Um novo grânulo γ^{c+1} é inicialmente representado pelas funções de pertinência, A_j^{c+1}, $j = 1, ..., n$, com

$$\mu_j^{c+1} = x_j^{[h]}, \quad (2)$$

e

$$\sigma_j^{c+1} = 1/2\pi. \quad (3)$$

A Eq. (3) é a abordagem de Stigler para Gaussianas padrões (Stigler, 1982; Leite et al., 2020). A ideia é começar com um valor grande e convergir gradualmente para um valor menor de dispersão conforme necessário. Essa estratégia gera uma estrutura compacta.

Em geral, a classe C^{c+1} da regra R^{c+1} é indefinida até que um rótulo seja fornecido. Quando a saída correspondente, $y^{[h]}$, associada a $\mathbf{x}^{[h]}$, se torna disponível, então

$$C^{c+1} = y^{[h]}. \quad (4)$$

Caso contrário, o rótulo da primeira amostra rotulada que chegar após o instante h e ativar R^{c+1} define C^{c+1}.

Caso uma amostra rotulada ativar uma regra com outro rótulo, um novo grânulo (parcialmente sobreposto) é criado para representar a nova informação. Grânulos parcialmente sobrepostos com rótulos diferentes tendem a ter sua dispersão e área de sobreposição reduzidas ao longo do tempo pelo procedimento de adaptação de parâmetros (Sec. 2.3). Além disso, seus valores modais tendem a se afastar naturalmente.

2.3 Atualização de Parâmetros

Atualizar o modelo consiste em: reduzir ou expandir o grânulo mais ativo γ^{i^*} representado por $A_j^{i^*} \forall j$; mover esse grânulo em direção a regiões mais povoadas; e rotular regras quando dados rotulados estão disponíveis. A adaptação desenvolve modelos locais mais específicos (Yager, 2008) e fornece cobertura para novos dados.

A regra R^i é candidata a ser atualizada se ela é suficientemente ativada por uma amostra $\mathbf{x}^{[h]}$,

$$min\left(A_1^i(x_1^{[h]}), ..., A_n^i(x_n^{[h]})\right) > \rho^{[h]}. \tag{5}$$

Geometricamente, $\mathbf{x}^{[h]}$ pertence a uma região altamente influenciada por γ^i. Para $\mathbf{x}^{[h]}$ sem rótulo, se mais de uma regra é ativada, apenas a regra mais ativa R^{i^*} é adaptada. Para amostras com rótulo $(\mathbf{x}, y)^{[h]}$, a classe da regra R^{i^*}, se já definida, deve coincidir com $y^{[h]}$. Caso contrário, a segunda regra mais ativa dentre aquelas que alcançaram o nível $\rho^{[h]}$ é escolhida para ser adaptada, e assim por diante. Se nenhuma das regras for apta, uma nova regra é criada conforme explanação da Sec. 2.2.

Para incluir $\mathbf{x}^{[h]}$ em R^{i^*}, o algoritmo de aprendizado eGFC atualiza os valores modais e dispersões de $A_j^{i^*} \forall j$:

$$\mu_j^{i^*}(\text{new}) = \frac{(\varpi^{i^*} - 1)\mu_j^{i^*}(\text{old}) + x_j^{[h]}}{\varpi^{i^*}}, \tag{6}$$

e

$$\sigma_j^{i^*}(\text{new}) = \left(\frac{(\varpi^{i^*} - 1)}{\varpi^{i^*}} \left(\sigma_j^{i^*}(\text{old})\right)^2 + \right. \\ \left. + \frac{1}{\varpi^{i^*}} \left(x_j^{[h]} - \mu_j^{i^*}(\text{old})\right)^2\right)^{1/2}, \tag{7}$$

em que ϖ^{i^*} é o número de vezes que a i^*-ésima regra venceu e foi atualizada. Note que (6)-(7) são recursivas, e, portanto, não requerem armazenamento de dados. Como σ^{i^*} define uma região de influência convexa ao redor de μ^{i^*}, valores muito grandes ou muito pequenos podem ditar a existência de um único ou excesso de grânulos por classe. Uma possível estratégia é confinar $\sigma_j^{i^*}$ entre limites inferior, $1/4\pi$, e de Stigler, $1/2\pi$.

2.4 Adaptação do Nível ρ

O limiar de ativação $\rho^{[h]} \in [0, 1]$ varia no tempo baseado na dispersão média global,

$$\sigma_{avg}^{[h]} = \frac{1}{cn} \sum_{i=1}^{c} \sum_{j=1}^{n} \sigma_j^{i[h]}; \tag{8}$$

em que c e n são o número de regras e atributos tal que

$$\rho(\text{new}) = \frac{\sigma_{avg}^{[h]}}{\sigma_{avg}^{[h-1]}} \rho(\text{old}). \tag{9}$$

O nível de ativação de regras para $\mathbf{x}^{[h]}$ é comparado com $\rho^{[h]}$ para decidir entre mudança paramétrica ou estrutural do modelo. Em geral, o modelo eGFC começa a aprender com uma base de regras vazia. A prática sugere $\rho^{[0]} = 0.1$ como valor inicial. Este valor é adaptado conforme o fluxo e leva a estrutura do classificador à estabilidade. Classes novas e não-estacionárias guiam $\rho^{[h]}$ a valores que melhor refletem as necessidades do ambiente atual.

2.5 Mescla de Grânulos

A similaridade entre dois grânulos que apontam para a mesma classe pode ser alta tal que formar um grânulo único seja melhor. Uma medida de distância entre Gaussianas n-dimensionais, digamos γ^{i_1} e γ^{i_2}, é

$$d(\gamma^{i_1}, \gamma^{i_2}) = \frac{1}{n}\left(\sum_{j=1}^{n} |\mu_j^{i_1} - \mu_j^{i_2}| + \sigma_j^{i_1} + \right. \\ \left. \sigma_j^{i_2} - 2\sqrt{\sigma_j^{i_1}\sigma_j^{i_2}} \right). \tag{10}$$

A medida considera a especificidade da informação que, por vez, é inversamente relacionada à dispersão (Leite et al., 2020), e.g., se $\sigma_j^{i_1}$ e $\sigma_j^{i_2}$ são 'mais diferentes', então a distância entre as Gaussianas é maior.

O par de grânulos que apresentar o menor $d(.)$ – desde que o par não tenha classe definida, ou ambos os grânulos apontem para a mesma classe – é mesclado. A decisão de mescla é baseada no limiar Δ. Para dados dentro do hipercubo unitário, sugere-se $\Delta = 0.1$ como valor padrão.

O novo grânulo, γ^i, resultado de mescla de γ^{i_1} e γ^{i_2}, é representado por Gaussianas com valor modal

$$\mu_j^i = \frac{\frac{\sigma_j^{i_1}}{\sigma_j^{i_2}}\mu_j^{i_1} + \frac{\sigma_j^{i_2}}{\sigma_j^{i_1}}\mu_j^{i_2}}{\frac{\sigma_j^{i_1}}{\sigma_j^{i_2}} + \frac{\sigma_j^{i_2}}{\sigma_j^{i_1}}},\ j = 1, ..., n, \tag{11}$$

e dispersão:

$$\sigma_j^i = \sigma_j^{i_1} + \sigma_j^{i_2},\ j = 1, ..., n. \tag{12}$$

Essas relações levam em conta a incerteza granular para definir a posição e o tamanho do novo grânulo (Škrjanc et al., 2019).

2.6 Remoção de Grânulos

Um grânulo é removido do modelo se ele é inconsistente com o ambiente atual. Em outras palavras, se uma regra não for ativada durante um certo número de iterações h_r, ela é apagada. Contudo, se ela apontar para uma classe rara, então podemos fazer $h_r = \infty$ e manter a regra.

2.7 Aprendizado Semi-Supervisionado Online

O algoritmo de aprendizado para construir e atualizar um modelo eGFC é dado abaixo.

eGFC: Aprendizado Online Semi-Supervisionado

Número inicial de regras, $c = 0$;
Meta-parâmetros iniciais, $\rho^{[0]} = \Delta = 0.1$, $h_r = 200$;
Ler instância $\mathbf{x}^{[h]}, h = 1$;
Criar grânulo γ^{c+1} (Eqs. (2)-(3)), classe C^{c+1} indef.;
FOR $h = 2$, ... **DO**
  Ler $\mathbf{x}^{[h]}$, calcular ativação de regras (Eq. (1));
  Determinar regra mais ativa R^{i^*};
  Fornecer classe estimada C^{i^*};
  // Adaptar modelo
  IF $T(A_1^i(x_1^{[h]}), ..., A_n^i(x_n^{[h]})) \leq \rho^{[h]}\ \forall i,\ i = 1, ..., c$
    IF classe $y^{[h]}$ está disponível
      Criar grânulo γ^{c+1}, classe def. (Eqs. (2)-(4));
    ELSE
      Criar grânulo γ^{c+1}, classe indef. (Eqs. (2)-(3));
    END
  ELSE
    IF classe $y^{[h]}$ está disponível
      Atualizar grânulo mais ativo γ^{i^*} cuja classe C^{i^*} é igual a $y^{[h]}$ (Eqs. (6)-(7));
      Rotular grânulos ativos sem rótulos;
    ELSE
      Atualizar grânulo γ^{i^*} mais ativo (Eqs. (6)-(7));
    END
  END
  Atualizar nível ρ (Eqs. (8)-(9));
  Deletar regras inativas baseado em h_r;
  Mesclar grânulos baseado em Δ (Eqs. (10)-(12));
END

3. METODOLOGIA

Descrevemos o gráfico dinâmico de controle para extração de atributos e rotulação fraca de logs. Dissertamos também sobre os dados utilizados e as medidas de desempenho.

3.1 Gráfico de Controle: Rótulos Fracos para Logs

Um gráfico de controle é uma representação de série temporal usada para monitorar a evolução de um sistema, fenômeno ou variável com base no teorema do Limite Central (Qiu, 2014). A ideia é que a média $\mu(u)$ de uma variável aleatória independente u com distribuição desconhecida, segue uma distribuição normal.

Seja

$$\mathbf{u}_j = [u_1\ \ldots\ u_i\ \ldots\ u_n] \tag{13}$$

uma sequência de valores que representam o número de linhas escritas em um arquivo de *log* durante uma janela de tempo $w_j = [\underline{w}_j\ \overline{w}_j]$; $\mathbf{u}_j \in \mathbb{N}^n$. O intervalo de tempo de u_1 a u_n coincide com os limites da janela, $\underline{w}_j$ e $\overline{w}_j$. Além disso, seja μ_j a média de $\mathbf{u}_j$, logo

$$\mu_j = \frac{1}{n}\sum_{i=1}^{n} u_i,\ u_i \in [\underline{w}_j\ \overline{w}_j]. \tag{14}$$

Uma série temporal de médias, com cardinalidade m, é

$$\mu = [\mu_1\ \ldots\ \mu_j\ \ldots\ \mu_m]. \tag{15}$$

Se μ tem distribuição normal, uma amostra μ_j pode ser rotulada usando gráfico de controle, como mostra a Fig. 1. A média $\bar{\mu}$ da série temporal das médias μ é

$$\overline{\mu} = \frac{1}{m}\sum_{j=1}^{m} \mu_j. \tag{16}$$

As k-ésimas linhas horizontais superiores e inferiores em relação a $\overline{\mu}$ referem-se a $\overline{\mu} \pm k$ vezes o desvio padrão

$$\sigma_k(\mu) = k\sqrt{\frac{1}{m}\sum_{j=1}^{m}(\overline{\mu} - \mu_j)^2}. \tag{17}$$

Se

$$\mu_j\ \subset\ [\overline{\mu} - \sigma_k(\mu_j \forall j),\ \overline{\mu} + \sigma_k(\mu_j \forall j)], \tag{18}$$

para $k = 1$, então μ_j é rotulada como 'Classe 1', indicando condição normal do sistema. Em (18), para $k = 2, 3$ e 4, então μ_j é rotulada como 'Classe 2 ', 'Classe 3' e 'Classe 4', conforme a ocorrência de anomalia de baixa, média e alta gravidade. Portanto, quanto maior o valor de k, maior a severidade da anomalia.

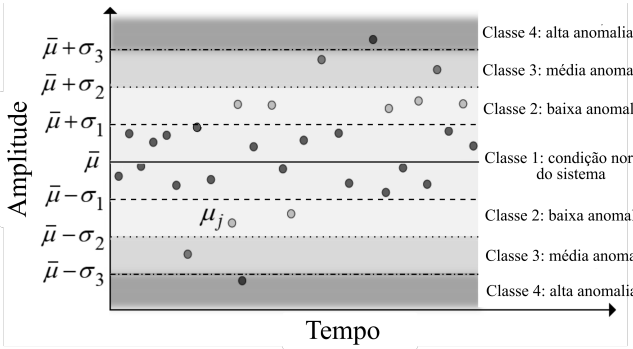

Figura 1. Exemplo de gráfico de controle para gerar rótulos fracos de trechos de *logs*

Gráficos de controle são amplamente utilizados em monitoramento de qualidade e detecção de anomalias. A probabilidade de uma amostra μ_j pertencer as faixas (classes) de 1 a 4 é 68,3%, 27,1%, 4,3% e 0,3%, respectivamente. Trata-se, portanto, de um problema de classificação desbalanceado.

3.2 Sobre a Base de Dados

O serviço StoRM produz dados de *log* continuamente. Uma linha de um arquivo de log contém o momento da escrita seguido da própria mensagem. A análise do conteúdo verbal da mensagem está fora do escopo deste artigo.

Inicialmente, obtemos o atributo *atividade de log* de um arquivo, i.e., a quantidade de linhas geradas por unidade de tempo. Deste novo fluxo de dados, extraímos um conjunto de atributos estatísticos em intervalo de tempo fixo a partir de janelas deslizantes w_j. Assim, produzimos um fluxo de vetores com 5 atributos

$$\mathbf{x}^{[h]} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5], \quad h = 1, ... \tag{19}$$

que significam, respectivamente, $\overline{\mu}$, $\sigma(\mu_j \; \forall j)$, $min(\mu_j \; \forall j)$, $max(\mu_j \; \forall j)$ e $max(\Delta\mu_j \; \forall j)$. O último atributo é a diferença máxima de amplitude entre dois μ_j consecutivos dentro da janela w_j.

Um vetor $\mathbf{x}^{[h]}$ é associado a uma das classe $C = \{1, 2, 3, 4\}$. O rótulo C gerado pelo gráfico de controle é disponibilizado após a estimativa $\hat{C}$ ser fornecida pelo modelo eGFC. O par $(\mathbf{x}, C)^{[h]}$ é, então, usado pelo algoritmo de aprendizado online de eGFC para atualização.

3.3 Análise de Desempenho

A acurácia da classificação, $Acc \in [0, 1]$, é obtida de forma recursiva conforme

$$Acc(\text{new}) = \frac{h-1}{h} \; Acc(\text{old}) + \frac{1}{h} \; \tau, \tag{20}$$

em que $\tau := 1$ se a estimativa é correta, ou seja, $\hat{C}^{[h]} = C^{[h]}$ e $\tau := 0$, caso contrário.

O número médio de regras fuzzy ao longo do tempo, c_{avg}, é uma medida da compacidade do modelo. Assim, temos recursivamente,

$$c_{avg}(\text{new}) = \frac{h-1}{h} \; c_{avg}(\text{old}) + \frac{1}{h} \; c^{[h]}. \tag{21}$$

4. EXEMPLO DE APLICAÇÃO

Avaliamos eGFC no problema de detecção de anomalia da central de computadores Tier-1 Bologna. Nenhum conhecimento prévio sobre os dados é assumido. Os classificadores são desenvolvidos online – sem fase de treinamento – a partir do fluxo de logs.

Os meta-parâmetros de eGFC são iniciados com valores padrão (ver *Algoritmo*). A Tabela 4 mostra os resultados médios em 5 rodadas do algoritmo para amostras sorteadas dos registros de log. Foram produzidos 4 destes conjuntos variando o tamanho da janela de tempo, viz., 5, 15, 30 e 60 minutos. Janelas maiores produzem efeito de filtro passa-baixas de ordem maior e tendem a isolar a componente de tendência das componentes aleatórias, vide Fig. 2. Cada conjunto de dados contém 1436 instâncias e 5 atributos; 4 classes são possíveis: operação normal; e baixa, média e alta gravidade de anomalia.

Tabela 1. Desempenho de eGFC no monitoramento de anomalias com confiança de 99%

Janela (min)	$Acc(\%)$	# Regras	Tempo (s)
60	$92,48 \pm 1,21$	$13,42 \pm 4,32$	$0,36 \pm 0,10$
30	$88,01 \pm 4,96$	$17,22 \pm 2,59$	$0,45 \pm 0,04$
15	$82,57 \pm 5,64$	$18,13 \pm 4,79$	$0,49 \pm 0,10$
5	$81,97 \pm 5,02$	$16,09 \pm 2,51$	$0,41 \pm 0,06$

A Tabela 4 mostra que janelas de sumarização maiores facilitam o algoritmo de aprendizado do eGFC a detectar e classificar melhor os padrões espaço-temporais em classes de anomalia. Observe que, usando uma estrutura compacta com aproximadamente 14 regras fuzzy, eGFC obteve acurácia média de $92,48\%$. O tempo de CPU em um notebook quad-core i7-8550U com 1,80 GHz e 8 GB de RAM é semelhante em todos os cenários.

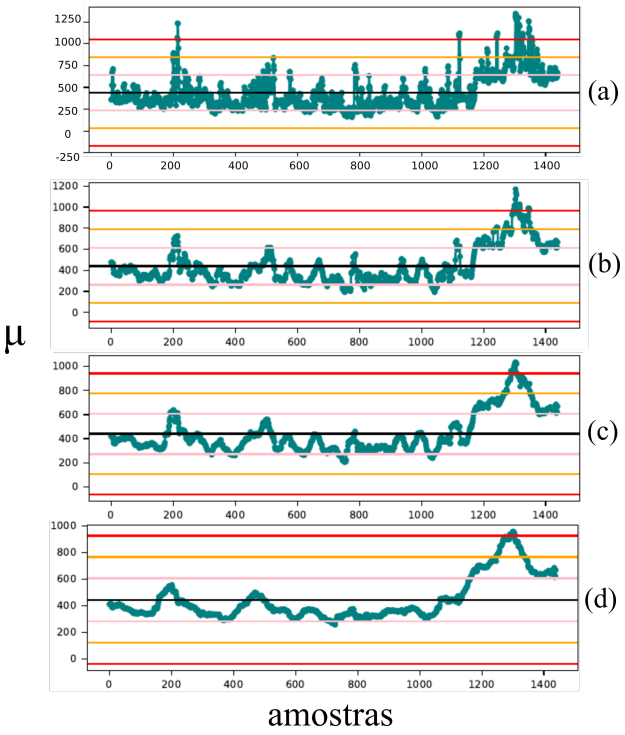

Figura 2. Efeito de filtro passa-baixas nos dados para janelas de: (a) 5, (b) 15, (c) 30, e (d) 60 minutos

A Fig. 3 mostra um exemplo da evolução da acurácia e número de regras ao longo do tempo. Já a Fig. 4 mostra 4 dimensões dos grânulos finais, em $h = 1436$. Observe que os grânulos das classes 1 e 2 não são linearmente separáveis. Há grande sobreposição destas classes. Elas precisam ser representadas por mais de um grânulo. As demais classes são concentradas em uma região comum. Os dados da classe 4 (anomalia de alta gravidade) estão em uma região mais compacta e isolada e, portanto, são representados por um único grânulo. Quanto maior o número de grânulos usados para representar uma classe, maior tende a ser a não-linearidade da fronteira de classe. Note que a geometria dos grânulos é elipsoidal.

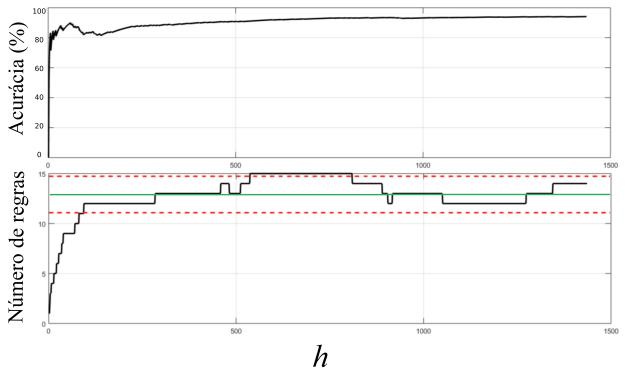

Figura 3. Evolução no tempo da acurácia e número de regras fuzzy

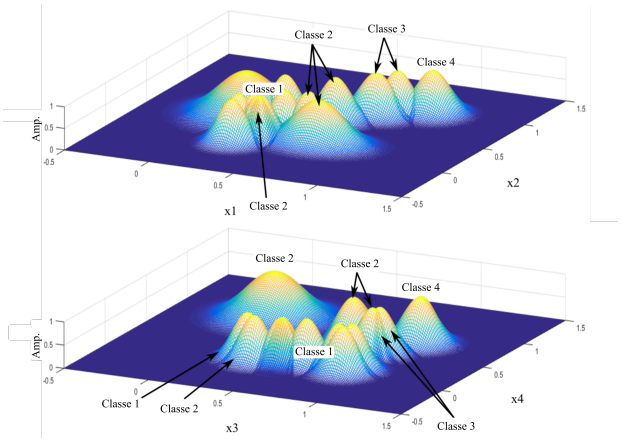

Figura 4. Grânulos Gaussianos eGFC finais, em $h = 1436$

A Tabela 4 mostra um exemplo de matriz de confusão de um cenário com acurácia de $94,2\%$. A confusão predomina na vizinhança da classe alvo, o que é positivo e é o principal aspecto a se observar. Se um número maior de amostras for disponibilizado, o modelo eGFC pode melhorar sua acurácia adaptando sua fronteira de separação. Porém, algumas vezes isso não é aconselhável, pois induz ao problema de *overfitting*, i.e., ao mascaramento do resultado prático de fato.

Tabela 2. Exemplo: matriz de confusão eGFC

–	Classe Alvo				–
–	Usual	Baixa	Média	Alta	Total
Usual	**1084**	10	0	0	99,1%
Baixa	61	**186**	3	0	74,4%
Média	1	1	**66**	2	94,3%
Alta	0	0	8	**14**	63,6%
Total	94,6%	94,4%	85,7%	87,5%	**94,0%**

* Diagonal principal: acertos de classe

Em resumo, o classificador evolutivo, com suporte da abordagem de gráfico de controle e janela deslizante, é usado para monitorar uma central de computação. Ele identifica trechos de logs que exigem análise de conteúdo. O status geral do sistema é modelado por grânulos fuzzy de atributos estatísticos. Ademais, o status corrente do sistema pode ser usado para prever os próximos status. Como eGFC se auto-atualiza em ambiente não-estacionário, ele tem se mostrado uma hipótese de solução interessante para o problema de manutenção preditiva.

5. CONCLUSÃO

Descrevemos uma solução de inteligência granular evolutiva, eGFC, para o problema de detecção de anomalia baseado em logs considerando dados da central de computação Tier-1 em Bologna. Modelos eGFC alcançaram uma acurácia média de $92.5\% \pm 1.2$ com um intervalo de confiança de 99% usando uma janela deslizante de 60 minutos. O problema de detecção de anomalia é sensível ao contexto. Logo, eGFC provê uma estratégia para atualizar e evoluir grânulos de informação e os parâmetros de classificadores fuzzy em tempo real.

A abordagem eGFC, orientada a fluxo de dados variantes no tempo, tem mostrado ser eficiente em uma diversidade de formulações de problemas de classificação de logs em centrais de computação. Adicionalmente, a estratégia de rotulação fraca baseada em gráfico de controle foi aplicada com sucesso e permitiu o aprendizado supervisionado do modelo em ambiente dinâmico. No futuro, mineração de texto em arquivos de log também será considerada para auxílio à tomada de decisão quanto a manutenção preditiva em tempo real.

AGRADECIMENTO

Este trabalho recebeu apoio do Instituto Serrapilheira (processo Serra – 1812-26777).

REFERÊNCIAS

Angelov, P. and Gu, X. (2018). Deep rule-based classifier with human-level performance and characteristics. *Inf. Sci.*, 463-464, 196–213.

Blažič, S. and Škrjanc, I. (2020). Hybrid system identification by incremental fuzzy c-regression clustering. In *2020 IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, 1–7. doi:10.1109/FUZZ48607.2020.9177678.

Casalino, G., Castellano, G., and Mencar, C. (2019). Data Stream Classification by Dynamic Incremental Semi-Supervised Fuzzy Clustering. *Int J Artif Intell T*, 28(8), 26p.

Decker, L., Leite, D., Giommi, L., and Bonacorsi, D. (2020a). Real-time anomaly detection in data centers for log-based predictive maintenance using an evolving fuzzy-rule-based approach. In *IEEE World Congress on Comput. Intell. (WCCI, FUZZ-IEEE), Glasgow*, 8p.

Decker, L., Leite, D., Viola, F., and Bonacorsi, D. (2020b). Comparison of evolving granular classifiers applied to anomaly detection for predictive maintenance in computing centers. In *IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), Bari*, 8p.

Di Girolamo, A. et al. (2020). Operational intelligence for distributed computing systems for exascale science. In *Int Conf on Computing in High Energy and Nuclear Physics (CHEP), AU*, 8p.

Diotalevi, T. et al. (2019). Collection and harmonization of system logs and prototypal analytics services with the elastic (ELK) suite at the INFN-CNAF computing centre. In *Int Symposium on Grids & Clouds (ISGC), Taiwan: Proceedings of Science*, 15p.

Donato, O. and Serra, G. (2016). Evolving fuzzy clustering algorithm based on maximum likelihood with participatory learning. In *IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 8p.

Furht, B. and Escalante, A. (2010). *Handbook of Cloud Computing*. Springer Publishing Company, Incorporated, 1st edition.

Giommi, L. et al. (2019). Towards predictive maintenance with machine learning at the INFN-CNAF computing centre. In *Int. Symposium on Grids & Clouds (ISGC). Taipei, TW: Proceedings of Science*, 17p.

Leite, D., Andonovski, G., Skrjanc, I., and Gomide, F. (2020). Optimal rule-based granular systems from data streams. *IEEE Tran. on Fuzzy Systems*, 28(3), 583–596.

Leite, D., Ballini, R., Costa Jr, P., and Gomide, F. (2012). Evolving fuzzy granular modeling from nonstationary fuzzy data streams. *Evolving Systems*, 3, 65–79.

Leite, D., Costa Jr, P., and Gomide, F. (2013). Evolving granular neural networks from fuzzy data streams. *Neural Networks*, 38, 1–16.

Leite, D., Decker, L., Santana, M., and Souza, P. (2020). EGFC: Evolving Gaussian fuzzy classifier from never-ending semi-supervised data streams - with application to power quality disturbance detection and classification. In *IEEE World Congress on Computational Intelligence (WCCI – FUZZ-IEEE), Glasgow*, 8p.

Lobo, J.L., Lana, I., Ser, J.D., Bilbao, M.N., and Kasabov, N. (2018). Evolving spiking neural networks for online learning over drifting data streams. *Neural Networks*, 108, 1–19.

Lopez, M., Mattos, D., Duarte, O., and Pujolle, G. (2019). Toward a monitoring and threat detection system based on stream processing as a virtual network function for big data. *Concurrency Comput Pract Exper*, e5344, 17p.

Minarini, F. (2019). *Anomaly Detection Prototype for log-based Predictive Maintenance at INFN-CNAF*. Master's thesis, U. of Bologna.

Pedrycz, W. and Gomide, F. (1998). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press.

Pratama, M., Lughofer, E., Lim, C., Rahayu, W., Dillon, T., and Budiyono, A. (2016). pclass+: A novel evolving semi-supervised classifier. *INT J FUZZY SYST*, 19, 863–880.

Precup, R.E., Filip, H.I., Radac, M.B., Petriu, E., Preitl, S., and Dragos, C.A. (2014). Online identification of evolving takagi–sugeno–kang fuzzy models for crane systems. *Applied Soft Computing*, 24, 1155–1163.

Qiu, P. (2014). *Introduction to Statistical Process Control*. Wiley: India.

Sadaei, H.J., de Lima e Silva, P.C., Guimarães, F.G., and Lee, M.H. (2019). Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 175, 365–377.

Silva, A., Caminhas, W., Lemos, A., and Gomide, F. (2013). Evolving neo-fuzzy neural network with adaptive feature selection. In *11th Brazilian Congress on Computational Intelligence (BRICS)*, 9p.

Silva, T., Schick, L., Lopes, P., and Camargo, H. (2018). A fuzzy multiclass novelty detector for data streams. In *IEEE Int Conf on Fuzzy Syst (FUZZ-IEEE)*, 8p.

Sousa, L.D. (2011). *Detecção de Eventos em Redes de Sensores Sem Fio*. Master's thesis, Universidade Federal de Minas Gerais.

Sousa, L.D. et al. (2012). Event detection framework for wireless sensor networks considering data anomaly. In *IEEE Symp. Comput. and Commun. (ISCC)*, 500–507.

Sousa, L.D. et al. (2019). Big data analysis for predictive maintenance at the INFN-CNAF data center using machine learning approaches. In *Conf. of Open Innovations Association (FRUCT), Helsinki*, 448–451.

Stigler, S.M. (1982). A modest proposal: A new standard for the normal. *The American Statist.*, 36(2), 137–138.

Trojan, F. and Marçal, R. (2017). Proposal of maintenance-types classification to clarify maintenance concepts in production and operations management. *Journal of Business Economics*, 8, 562–574.

Yager, R.R. (2008). Measures of specificity over continuous spaces under similarity relations. *Fuzzy Sets and Systems*, 159(17), 2193 – 2210.

Škrjanc, I., Iglesias, J., Sanchis, A., Leite, D., Lughofer, E., and Gomide, F. (2019). Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. *Inf. Sci.*, 490, 344–368.