

Construção de Modelos Markovianos por Redução de Entropia sobre Árvores

Higor Í. Santos, Daniel P. B. Chaves e Cecilio Pimentel

Resumo—Sistemas dinâmicos discretos são amplamente utilizados em aplicações científicas. Este trabalho apresenta um novo algoritmo para a modelagem desses sistemas utilizando autômatos probabilísticos de estados finitos. Inicialmente, aplica-se o conceito de entropia para se gerar uma árvore. Diferente de outras propostas, garante-se que todas as folhas da árvore se tornem estados do modelo markoviano. Empregando séries temporais de falhas em motores elétricos, demonstra-se que a proposta supera outros métodos tanto na sua capacidade em reproduzir as estatísticas da sequência modelada quanto no seu emprego em algoritmos de detecção de falha.

Palavras-Chave—Filtragem por dinâmica simbólica, PFSA, entropia, detecção de falha.

Abstract—Discrete dynamic systems are widely used in scientific applications. This paper presents a new algorithm for modeling these systems using probabilistic finite state automata. Initially, the concept of entropy is applied to generate a tree. Unlike other proposals, it is guaranteed that all leaves of the tree become states of the Markov model. By employing time series data of faults in electric motors, it is demonstrated that the proposal outperforms other methods both in its ability to reproduce the statistics of the modeled sequence and in its application in fault detection algorithms.

Keywords—Symbolic dynamics filtering, PFSA, entropy, fault detection.

I. INTRODUÇÃO

Dois métodos se destacam na modelagem de sistemas dinâmicos: baseado em modelo e orientado a dados [1]. O primeiro busca descrever analiticamente o sistema dinâmico, enquanto o segundo estabelece uma relação entre as variáveis do espaço de estado, mesmo sem o conhecimento prévio de seu comportamento dinâmico. O método orientado a dados é adequado para sistemas complexos, nos quais a determinação de modelos analíticos pode ser impraticável [1], [2]. Esse é especialmente valioso na análise de séries temporais, principalmente em sistemas não lineares, capturando o mecanismo subjacente que gera os dados observados [3]. A filtragem por dinâmica simbólica (SDF, *Symbolic Dynamic Filtering*) é uma técnica de processamento de dados que captura o comportamento dinâmico de uma série temporal [4], [5]. Nesse caso, uma série temporal contínua passa por um processo de simbolização por meio de processos adequados de quantização [6].

Higor Santos, Daniel Chaves e Cecilio Pimentel, Departamento de Eletrônica e Sistemas, Universidade Federal de Pernambuco, Recife-PE, e-mails: {higor.italo, daniel.chaves, cecilio.pimentel}@ufpe.br; Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Ciência e Tecnologia do Estado de Pernambuco (FACEPE), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

A forma usual de aplicar SDF na análise de séries temporais é determinar um PFSA (*Probabilistic Finite State Automata*) que capture a estrutura causal de uma sequência discreta observada, S . O D -Markov é um algoritmo clássico para determinar um PFSA a partir de uma sequência S . Constrói-se um processo de Markov de ordem finita D a partir da rotulação dos estados do PFSA por todas as sequências de comprimento D . Isso gera um crescimento exponencial do número de estados com D , o que explicita uma limitação do método, já que se pressupõe que todas as sequências de comprimento D são necessárias para refletir a estrutura causal da dinâmica. A técnica de construção de PFSA proposta em [7], chamada DCGraM (*D-Markov with Clusterization and Graph Minimization*), utiliza algoritmos de agrupamento e minimização de estados para obter um PFSA reduzido a partir de um D -Markov. Contudo, a memória máxima capturada pelo PFSA é limitada ao parâmetro D do modelo inicial D -Markov.

Obtém-se maior eficácia na expressão da relação causal da dinâmica ao se empregar uma estrutura de árvore. Nesse caso, a grande diferença em relação aos métodos citados anteriormente é que os estados do PFSA são rotulados por sequências de comprimento variado, associadas às folhas de uma árvore. A forma como essas folhas são determinadas caracteriza o método, sendo dois desses: por identificação de contexto e por redução de entropia. Em [8] um nó da árvore é identificado como folha quando a probabilidade de geração de símbolos a partir desse é semelhante a dos nós que o seguem. Esse algoritmo é denominado VL-DCGraM (*Variable Length DCGraM*), pois, assim como o DCGraM, minimiza um PFSA inicial ao aplicar algoritmos de agrupamento e minimização de estados. Em [4], a árvore é gerada pela minimização de uma função de entropia definida sobre as folhas da árvore, sendo denominado S2ER (*State-Splitting by Entropy Reduction*).

Esses algoritmos de construção de PFSA são avaliados pelas suas capacidades de reproduzir as estatísticas da sequência modelada e com o menor número possível de estados. Nesse contexto, há duas contribuições neste artigo: (1) Um algoritmo por redução de entropia (PS2ER, *Preserved S2ER*), o qual reproduz as estatísticas da sequência modelada de forma análoga ou superior aos métodos citados e com menor número de estados; (2) Um método de identificação de anomalias sobre PFSA empregando redes neurais que se mostra superior a outras técnicas [7], [9].

Este artigo apresenta mais seis seções. Na Seção II apresentamos os fundamentos sobre PFSA e os algoritmos: D -Markov, DCGraM e VL-DCGraM. Na Seção III, descrevemos os algoritmos por redução de entropia, em particular o PS2ER proposto. Os quantificadores de desempenho para PFSA são

discutidos na Seção IV. A comparação dos algoritmos de detecção de falhas é realizada na Seção V e, por fim, a Seção VII traz as conclusões.

II. PRELIMINARES

Nessa seção abordaremos conceitos relevantes para o entendimento do algoritmo proposto [4], [10]. O comprimento de uma palavra ω é denotado por $|\omega|$.

Definição 1 (PFSA): Um PFSA \mathcal{G} é uma quádrupla (Q, Σ, δ, π) na qual:

- Q é o conjunto finito de estados com cardinalidade $|Q|$;
- Σ é o alfabeto finito com cardinalidade $|\Sigma|$ que rotula as transições direcionadas entre estados;
- δ é a função de transição de estados, tal que, $Q \times \Sigma \rightarrow Q$;
- π é uma atribuição de probabilidade $\pi : Q \times \Sigma \rightarrow [0, 1]$ que satisfaz $\sum_{\sigma \in \Sigma} \pi(q, \sigma) = 1, \forall q \in Q$.

Dado um estado $q \in Q$, o vetor distribuição de probabilidade $\mathcal{V}(q) = (\pi(q, \sigma_1), \pi(q, \sigma_2), \dots, \pi(q, \sigma_{|\Sigma|}))$ é chamado de *morph* desse estado, em que $\pi(q, \sigma) = \Pr(\sigma | q), \forall \sigma \in \Sigma$.

Definição 2 (Contexto à direita): Uma palavra $\omega\sigma$ pertence ao contexto à direita do estado $q \in Q$ de um PFSA se, e somente se, $\delta^*(q, \omega\sigma) = \delta^*(\delta^*(q, \omega), \sigma) \in Q$, em que $\delta^*(q, \omega)$ é a extensão de δ sobre palavras, ou seja, o contexto à direita de um estado $q \in Q$ é definido como o conjunto de todas as palavras geradas por caminhos que começam no estado q e terminam em algum estado de Q .

Há dois algoritmos para particionar o conjunto Q de um PFSA em estados com mesmo contexto à direita: Moore e Hopcroft [11]; conhecidos como algoritmos de minimização de estados. A seguir apresentaremos brevemente alguns algoritmos desenvolvidos na literatura para a construção de PFSA a partir de uma sequência discreta sobre um alfabeto Σ .

A. *D*-Markov

A máquina *D*-Markov [4] é um PFSA que gera símbolos que dependem apenas de D símbolos passados. Essa gera um processo Markoviano de ordem D . Para construir uma máquina *D*-Markov, $|\Sigma|^D$ estados são criados gerando o conjunto Q em que cada estado é rotulado com uma palavra de comprimento D sobre um alfabeto Σ .

B. Algoritmo DCGraM

O algoritmo DCGraM [7] identifica e colapsa estados de um *D*-Markov com o mesmo contexto à direita e respectivas distribuições de probabilidade semelhantes. Esse é dividido essencialmente em três passos: 1) Geração de um *D*-Markov; 2) Aplicação do algoritmo de clusterização *K-Means* ao conjunto de *morphs* de cada estado; 3) Aplicação de algum algoritmo de minimização de estados (e.g., Moore ou Hopcroft), o que permite identificar estados que geram sequências com probabilidades distintas.

C. Algoritmo VL-DCGraM

O emprego de um *D*-Markov como PFSA de entrada para o DCGraM pode ser proibitivo, uma vez que o número de

estados cresce exponencialmente com D . A diferença fundamental entre o DCGraM e VL-DCGraM [8] é a substituição do *D*-Markov na primeira etapa do algoritmo por uma cadeia de Markov de comprimento variável (VLMC - *Variable length Markov Chain*) cuja memória varia a depender dos símbolos passados. Na construção do VLMC, uma árvore é construída até que todas as extensões a partir de um dado nó apresentem o mesmo *morph* ou, na prática, que tenham *morphs* similares (distância euclidiana menor que um certo limiar T). A árvore é estendida até uma profundidade máxima, denominada L . Alguns estados que atingem esta profundidade com alta probabilidade de visita são estendidos até atingirem a profundidade máxima M . Um parâmetro α controla esta divisão de estados.

III. ALGORITMO PROPOSTO: PS2ER

Nesta seção tratamos dos algoritmos para geração de PFSA por redução de entropia. Para simplificar a notação adotada, nos referimos aos nós de uma árvore como palavras sobre o alfabeto Σ , tal que, o nó raiz é a palavra nula ε . De cada nó da árvore partem $|\Sigma|$ ramos cada um rotulado com um elemento de Σ . O nó alcançado a partir de ε por um caminho de rótulo ω recebe a mesma rotulação. Para cada nó ω da árvore atribui-se a probabilidade P_ω e a função $\pi : \omega \times \Sigma \rightarrow [0, 1]$, a qual satisfaz $\sum_{\sigma \in \Sigma} \pi(\omega, \sigma) = 1$. O conjunto de nós alcançados por ramos que partem de um nó ω são denominados nós filhos desse, sendo ω o correspondente nó pai. As folhas de uma árvore não possuem nós filhos. O conjunto de folhas de uma árvore é definido por $\Gamma = \{w_1, w_2, \dots, w_m\}$. Tanto P_ω como π são estimados a partir da série temporal modelada.

Isso posto, o algoritmo S2ER [4] atua iterativamente estendendo a folha que mais reduz a expressão

$$H(\Gamma) = - \sum_{\omega \in \Gamma} P_\omega \sum_{\sigma \in \Sigma} \pi(\omega, \sigma) \log \pi(\omega, \sigma), \quad (1)$$

a qual definimos como a entropia do conjunto de folhas da árvore. Considere Γ^i o conjunto obtido de Γ pela substituição de ω_i pelos correspondentes nós filhos, ou seja, $\Gamma^i = \Gamma \setminus \{\omega_i\} \cup \{\sigma\omega_i : \sigma \in \Sigma\}$ para $\omega_i \in \Gamma$. No algoritmo S2ER, ω_i será substituído por seus filhos na árvore se $H(\Gamma^i) < H(\Gamma^j)$ para qualquer outro $\omega_j \in \Gamma$. Portanto, concluída essa iteração do algoritmo, teremos uma árvore expandida cujo conjunto de folhas é Γ^i . Como abordado nas Seções IV e V, o S2ER apresenta valores iniciais melhores nos quantificadores de desempenho quando comparado com os métodos descritos na Seção II para o banco de dados considerado. Contudo, com o aumento no número de estados do PFSA, há uma inflexão na evolução de quantificadores, acarretando em sua saturação. Esse efeito provém da construção da função de transição δ , que não utiliza todos os elementos em Γ na determinação do conjunto de estados Q do PFSA final.

Para sanar as limitações do S2ER, propomos o algoritmo PS2ER apresentado em **Algoritmo 1**, o qual associa a cada estado em Q uma única folha em Γ e vice-versa. O resultado é um ganho consistente nos quantificadores de desempenho do PFSA em relação a todas as demais propostas. Pode-se dividir o PS2ER em três etapas:

- Etapa 1 (Extensão de nós): Calcula-se Γ^i para todo $w_i \in \Gamma$, tal que, $|w_i| < l_{max}$. Lembrando que Γ^i é obtido de Γ pela substituição de w_i pelos correspondentes nós filhos. Por fim, mantém-se a expansão da árvore que gerou Γ^i , desde que, $H(\Gamma^i) < H(\Gamma^j)$ para todo $w_j \in \Gamma$.
- Etapa 2 (Determinação dos sufixos dos nós folhas): Para cada $w_i \in \Gamma$ e $\sigma \in \Sigma$ verifica-se a existência de algum sufixo de $w_i\sigma$ em Γ . Se não existir, estende-se a folha w_i , substituindo-a por seus filhos em Γ . Deve-se repetir esse processo até não ser mais possível estender a árvore.
- Etapa 3 (Construção do PFSA): Uma vez atingido o critério de parada, $|\Gamma| \geq N_{max}$, adota-se $Q = \Gamma$ como o conjunto de estados do PFSA. Por fim, para cada $q \in Q$ e $\sigma \in \Sigma$ calculam-se as funções $\delta(q, \sigma)$ e $\pi(q, \sigma)$. Supondo que ω é o rótulo de q , herdado da correspondência de q com as folhas em Γ , então $\delta(q, \sigma)$ é o estado cujo rótulo é sufixo de $\omega\sigma$. Já $\pi(q, \sigma)$ é estimado pela frequência relativa com que σ segue ω na sequência S modelada.

É importante frisar que a Etapa 2 garante que na construção do PFSA na Etapa 3 todo q em Q sempre é alcançado por outro estado em Q .

Algoritmo 1 PS2ER

```

1: Entradas:  $\Sigma, N_{max}, l_{max}$ 
2: Saídas: PFSA  $(Q, \Sigma, \delta, \pi)$ 
3: Inicialização:  $\Gamma \leftarrow \{\varepsilon\}$ 
4: repita
5:   # Etapa 1: Extensão de nós
6:    $\Gamma' \leftarrow \{\varepsilon\}$ 
7:    $H(\Gamma') = 1$ 
8:   para  $w_i \in \Gamma$  faça
9:     se  $|w_i| < l_{max}$  então
10:       $\Gamma^i \leftarrow (\Gamma \setminus \{w_i\}) \cup (\{\sigma w_i : \sigma \in \Sigma\})$ 
11:      se  $H(\Gamma^i) < H(\Gamma')$  então
12:         $\Gamma' \leftarrow \Gamma^i$ 
13:         $H(\Gamma') \leftarrow H(\Gamma^i)$ 
14:    $\Gamma \leftarrow \Gamma'$ 
15:   # Etapa 2: Identificação de máximos sufixos
16:    $\Gamma'' \leftarrow \{\varepsilon\}$ 
17:   enquanto  $\Gamma'' \neq \Gamma$  faça
18:      $\Gamma'' \leftarrow \Gamma$ 
19:     para  $w_i \in \Gamma''$  faça
20:       se  $\text{suffix}(w_i\sigma) \notin \Gamma''$  para algum  $\sigma \in \Sigma$  então
21:          $\Gamma \leftarrow (\Gamma \setminus \{w_i\}) \cup (\{\sigma w_i : \sigma \in \Sigma\})$ 
22: até que  $|\Gamma| \geq N_{max}$ 
23: # Etapa 3: Construção do PFSA
24:  $Q \leftarrow \Gamma$ 
25: para  $q \in Q$  faça
26:   para  $\sigma \in \Sigma$  faça
27:      $\delta(q, \sigma) \leftarrow \text{suffix}(q\sigma)$ 
28:      $\pi(q, \sigma) = \text{Pr}(\sigma | q)$ 
29: retorne  $\{Q, \Sigma, \delta, \pi\}$ 
    
```

A fim de ilustrar o processo de extensões da árvore na Fig. 1, realizado pelo Algoritmo 1, seja $\Gamma = \{00, 010, 110, 01, 11\}$ o conjunto de folhas da árvore em determinada iteração do algoritmo. Supondo os valores de $H(\Gamma^i)$ para cada $w_i \in \Gamma$:

$H(\Gamma^1) = 0,505, H(\Gamma^2) = 0,500, H(\Gamma^3) = 0,506, H(\Gamma^4) = 0,508, H(\Gamma^5) = 0,509$; sendo $\arg \min_i H(\Gamma^i) = 2$, conclui-se que $w_2 = 010$ deve ser estendida, o que leva ao novo conjunto de folhas $\Gamma = \{00, 0010, 1010, 110, 01, 01, 11\}$. Em seguida, de acordo com a Etapa 2, verifica-se para cada elemento w_i de Γ a existência de uma folha no próprio Γ cujo rótulo é sufixo de $w_i\sigma, \sigma \in \Sigma$. Observa-se que a extensão com símbolo $\sigma = 0$ da folha 01 (isto é, 010) não consta entre os elementos de Γ . Isso indica a necessidade de realizar uma nova extensão na árvore (folhas 001 e 101 partindo da folha pontilhada 01 na figura), resultando em $\Gamma' = \{00, 0010, 1010, 110, 001, 101, 11\}$. Esse conjunto satisfaz a verificação de sufixos na Etapa 2, o que acarreta na conclusão desta etapa.

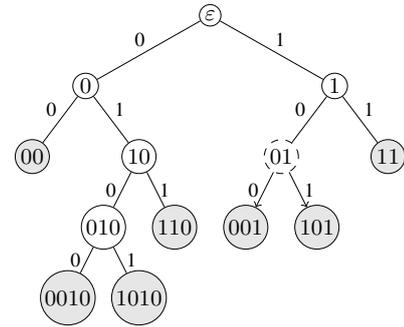


Fig. 1. Representação em árvore da geração de estados do PFSA construído pelo Algoritmo 1.

IV. QUANTIFICADORES DE DESEMPENHO

Dois quantificadores são adotados para comparar o desempenho do PFSA gerado pelo algoritmo proposto com outros presentes na literatura. Os quantificadores são: a entropia condicional de ordem ℓ e o divergência de Kullback-Leiber de ordem ℓ . O primeiro é definido por $h_\ell \triangleq H(X_\ell | X_1 X_2 \dots X_{\ell-1})$, em que $H(X|Y)$ é a entropia condicional de X dado Y [12] e permite estimar a memória da série temporal. O segundo quantificador, definido como

$$D_\ell = \sum_{\omega \in \Sigma^\ell} P_{ST}(\omega) \log \left(\frac{P_{ST}(\omega)}{P_M(\omega)} \right), \quad (2)$$

em que, Σ^ℓ é o conjunto de todas as palavras de comprimento ℓ sobre o alfabeto Σ , permite avaliar a semelhança entre as distribuições de probabilidade $P_{ST}(\omega)$ e $P_M(\omega)$, estimadas a partir da sequência S modelada e da gerada pelo PFSA, respectivamente.

V. RESULTADOS

O sistema utilizado para análise de resultados neste trabalho corresponde ao banco de dados MaFaulDa [13], composto de séries temporais obtidas a partir de leituras realizadas por sensores em um simulador de falhas de alinhamento-equilíbrio-vibração em motores. O aparato experimental emula a dinâmica destas máquinas e permite o monitoramento e análise de seus modos de operação. Dois desses modos foram considerados neste trabalho: operação normal (correspondente à atuação do sistema sem presença de falhas) e Desalinhamento Horizontal de 2mm (falha induzida no sistema pelo

deslocamento de 2mm do eixo do motor horizontalmente em relação a sua posição normal.)

A partir de seqüências binárias ($\Sigma = 2$) formadas via quantização por máxima entropia [6] das series temporais obtidas dos sensores do sistema MaFaulDa para o modo de operação normal, foram gerados modelos PFSA com os algoritmos descritos neste trabalho. Máquinas D -Markov são geradas para D entre 4 e 10 e utilizadas no algoritmo DCGraM com parâmetro $K = 5$ do K -means; modelos VL-DCGraM são gerados com L de 4 a 9 e utilizando o algoritmo K -means ponderado com $K = 5$ e parâmetros $T = 0,001$, $M = L + 1$ e $\alpha = 2$; S2ER e PS2ER foram gerados para diferentes valores de N_{max} (entre 16 a 343 para o S2ER e entre 18 e 262 para o PS2ER, com esse último adotando um comprimento máximo de palavra $l_{max} = 9$). Uma vez construídos, os desempenhos dos PFSA foram comparados usando os quantificadores adotados.

A Fig. 2 mostra h_ℓ , para $\ell = 10$, versus o número de estados para os PFSA obtidos pelos algoritmos: D -Markov [4], DCGraM [7], VL-DCGraM [8], S2ER [4] e PS2ER. A linha horizontal mostrada nesta figura corresponde ao valor de h_{10} obtido com a série temporal e serve como um valor de referência de desempenho. Observa-se que o PFSA gerado com o algoritmo PS2ER atinge o valor de entropia h_{10} do sistema original com menor número de estados. A Fig. 3 traz D_ℓ versus o número de estados para $\ell = 10$. Observa-se que o PFSA gerado pelo PS2ER se destaca entre os métodos, apresentando o menor D_ℓ e menor número de estados entre todos os PFSA que atingem o referencial h_{10} da série temporal modelada.

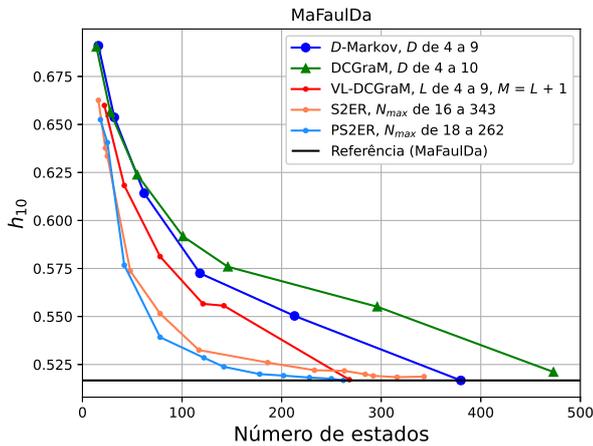


Fig. 2. Entropia condicional h_{10} versus número de estados para os PFSA construídos com D -Markov, DCGraM, VL-DCGraM, S2ER e PS2ER para valores distintos de D , L e N_{max} . Sistema MaFaulDa com operação normal.

VI. DETECÇÃO DE FALHAS

Vetores de probabilidades estacionárias de PFSA obtidos nas condições de operação normal e anômala têm sido usados para realizar a detecção de falhas em sistemas dinâmicos [9], [7]. Nesta seção, propomos o emprego de uma rede neural que tem como entrada esses vetores e comparamos a sua eficiência com outros métodos da literatura.

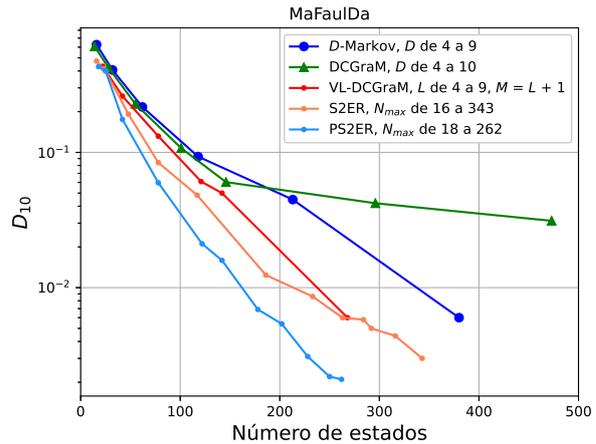


Fig. 3. Divergência de Kullback-Leibler D_{10} versus número de estados para os PFSA construídos com D -Markov, DCGraM, VL-DCGraM, S2ER e PS2ER para valores distintos de D , L e N_{max} . Sistema MaFaulDa com operação normal.

Para comparar metodologias de detecção de falhas baseadas em um classificador binário associado a um limiar de decisão, adotamos como medida de desempenho a área sob a curva (AUC, *Area Under Curve*) da curva característica de operação do receptor (ROC, *Receiver Operating Characteristic Curve*) [14]. Quanto maior a AUC de um modelo, maior sua capacidade de distinguir entre um comportamento normal ou anômalo.

A seguir descrevemos brevemente duas metodologias para avaliação da capacidade de detecção de falha usando vetores de probabilidades estacionárias de um PFSA, e na subseção seguinte é apresentada uma metodologia proposta para detecção de falhas.

- Metodologia 1 [9]: Busca identificar alterações no comportamento do processo estacionário pelo cálculo de múltiplas máquinas D -Markov construídas a partir de palavras de comprimento W para avaliação da variação do valor médio dos correspondentes vetores de probabilidade estacionária de seus respectivos estados.
- Metodologia 2 [7]: Indicação de comportamento anômalo pela comparação da divergência de Kullback-Leibler entre o vetor de probabilidade estacionária de um PFSA gerado a partir de uma seqüência associada à operação normal e a probabilidade estimada de seus estados através de um passeio aleatório de comprimento W sobre o mesmo empregando uma seqüência de teste.

A. Metodologia proposta para detecção de falhas

Propomos como metodologia alternativa para detecção de seqüências com característica anômala a construção de uma rede neural que adota como dados de treinamento vetores de probabilidades de ocupação de estados obtidos por um passeio aleatório de comprimento W sobre um PFSA que modela operação normal de seqüências discretas que representam operações normal e anômala. A rede é constituída por uma camada interna composta por r neurônios, função de ativação sigmoide e saída única (classificação binária) que fornece

valores no intervalo $[0, 1]$ indicando a associação do vetor de entrada à operação normal (saídas com valores próximos a 0) ou anômala (valores de saída próximos a 1).

A Tabela I apresenta uma comparação em termos de valores AUCs normalizados (pelo maior valor em cada coluna) calculados através das três metodologias ao aplicarmos D -Markov (16 estados) e DCGraM (14 estados) com $D = 4$, VL-DCGraM com $L = 4$ e $M = L$ (11 estados), e por fim S2ER e PS2ER (11 estados) com $N_{max} = 11$. Para todos os casos, o número de neurônios na camada interna das redes neurais é $r = 11$, o qual coincide com o menor número de entradas das redes. Curvas ROC são geradas utilizando $J = 5000$ janelas de símbolos de comprimento $W = 100$ geradas a partir do banco de dados MaFaulDa, sendo 2500 referentes à operação Normal e 2500 à falha Desalinhamento Horizontal de 2mm. A Metodologia 1 [9] pressupõe o emprego de máquinas D -Markov, por isso o seu cálculo está restrito a esse caso. Verificamos pela tabela que a metodologia proposta fornece maiores valores de AUC para qualquer PFSA considerado, demonstrando que a geração de curvas ROC com uso da rede neural adotando vetores de probabilidades estacionárias de um PFSA como entrada constitui uma forma eficaz para detecção de falhas no sistema MaFaulDa.

A Fig. 4 apresenta curvas ROC geradas de acordo com a metodologia proposta para os PFSA comparados neste trabalho a fim de verificar a capacidade de cada um quanto à classificação da falha Desalinhamento Horizontal de 2 mm no sistema MaFaulDa. Observa-se que o modelo PS2ER alcança valor de AUC igual aos maiores valores calculados a partir de PFSA com maior número de estados.

VII. CONCLUSÕES

Dois metodologias abordadas neste trabalho para construção de PFSA empregam princípios complementares, sendo estes: a fusão de estados e a quebra de estados. A primeira é caracterizada pela criação de uma apresentação sobredimensionada, como um D -Markov, seguida da fusão de estados com *mophs* similares. Esse é o método adotado no DCGraM e no VL-DCGraM. Por sua vez, o segundo método estabelece sucessivas quebras de estados com base em uma função custo, como em (1), sendo o caso da S2ER e PS2ER. Em decorrência das especificidades do processo de cálculo da função de transição do PFSA final, os resultados indicam que a estratégia S2ER é limitada, ainda sendo necessário garantir a permanência de todos os estados gerados pelo processo inicial de quebra. A eficácia do método proposto é ratificada pela entropia condicional e o divergência de Kullback-Leibler, os quais refletem a capacidade do modelo em capturar a memória e a estatística da sequência modelada. Por fim, o destaque da proposta em relação a outros métodos é reiterado ao aplicarmos os PFSA gerados ao processo proposto de identificação de falta baseado em redes neurais.

REFERÊNCIAS

[1] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques — Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.

TABELA I
COMPARAÇÃO ENTRE DIFERENTES TÉCNICAS DE DETECÇÃO DE ANOMALIAS SOB PFSA. FALHA DESALINHAMENTO HORIZONTAL DE 2 MM NO SISTEMA MAFAULDA.

Modelo Método	D -Markov	DCGraM	VL-DCGraM	S2ER	PS2ER
[9]	0,2453	-	-	-	-
[7]	0,8444	0,8786	0,9270	0,7738	0,7863
Proposta	1	1	1	1	1

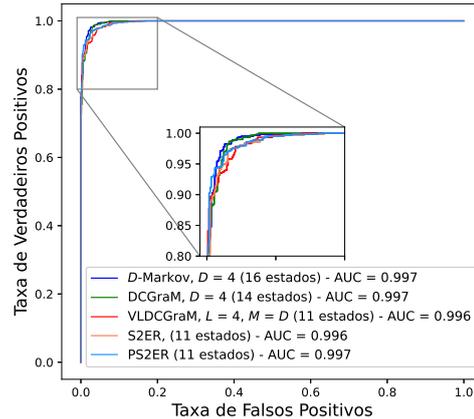


Fig. 4. Curvas ROC geradas a partir de modelos D -Markov, DCGraM, VL-DCGraM, S2ER, PS2ER com relação à falha Desalinhamento Horizontal de 2 mm no sistema MaFaulDa. Rede neural com $r = 11$.

[2] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, Apr. 2014.

[3] E. Bradley and H. Kantz, "Nonlinear time-series analysis revisited," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 9, p. 097610, Apr. 2015.

[4] K. Mukherjee and A. Ray, "State splitting and merging in probabilistic finite state automata for signal representation and analysis," *Signal Processing*, vol. 104, pp. 105–119, Apr. 2014.

[5] L. Wang, Y. Zhong, and W. Ma, "GPS-data-driven dynamic destination prediction for on-demand one-way carsharing system," *IET Intelligent Transport Systems*, vol. 12, no. 10, pp. 1291–1299, 2018.

[6] Y. Li and A. Ray, "Unsupervised symbolization of signal time series for extraction of the embedded information," *Entropy*, vol. 19, no. 4, p. 148, Mar. 2017.

[7] D. K. Franch, D. P. Chaves, C. Pimentel, and D. M. Hamilton, "Markov modeling of dynamical systems via clustering and graph minimization," *Digital Signal Processing*, vol. 104, p. 102769, Sept. 2020.

[8] H. Í. dos Santos, D. P. Chaves, and C. Pimentel, "Determinação de modelo markoviano para canal sem fio empregando aprendizado de máquina," in *XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*, 2020.

[9] N. F. Ghalyan and A. Ray, "Measure invariance of ergodic symbolic systems for low-delay detection of anomalous events," *Mechanical Systems and Signal Processing*, vol. 159, p. 107746, Oct. 2021.

[10] D. Lind and B. Marcus, *An Introduction To Symbolic Dynamics and Codings*, 2nd ed. Cambridge, United Kingdom: Cambridge University Press, 2021.

[11] J. Berstel, L. Boasson, O. Carton, and I. Fagnot, "Minimization of automata," *arXiv preprint arXiv:1010.5318*, 2010. [Online]. Available: <https://arxiv.org/pdf/1010.5318>

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

[13] COPPE/Poli/UFRJ. Mafaulda – machinery fault database. [Online]. Available: https://www02.smt.ufrj.br/offshore/mfs/page_01.html#SEC2

[14] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3rd ed. Sebastopol, California: O'Reilly Media, Inc., 2022.