Predição probabilística do tempo até mudança de estado em compressores industriais usando RNNs e VAEs

Fernanda Mickosz Villa Verde

CPE727 – Deep Learning

9 de dezembro de 2025

Sumário

Motivação e Objetivos do trabalho

Motivação:

- Compressores industriais são ativos críticos (segurança, disponibilidade, custo).
- Mudanças de regime operacional (estados) afetam: Desempenho energético; Risco de falha; Planejamento de manutenção.
- Predizer **quando** e com que incerteza ocorrerá a próxima mudança de estado é útil para:
  - Detecção precoce de desvios;
  - Replanejamento operacional;
  - Suporte à decisão para operadores.

Objetivos do trabalho:

- Aprender uma **modelagem probabilística do tempo até a próxima mudança de estado**, estimando distribuições preditivas condicionadas ao histórico observado, utilizando RNNs e VAEs;

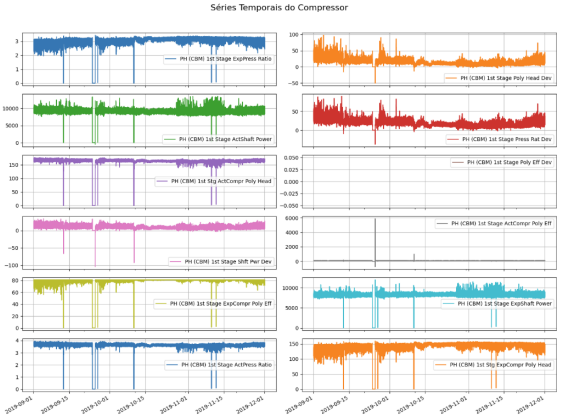
Comparar arquiteturas:

- Modelos DEEP, LSTM, GRU, BiLSTM e BiGRU com GRU fuser, todos com otimização Adam;
- Variações de treinamento (warmup, scheduler, RAdam, AdamW, variational dropout, difusão, log-likelihood e layernorm).

Base de dados e pré-processamento

Base de dados

Cognite — sinais reais de sensores de um compressor industrial offshore.



Pré-processamento

- Estados operacionais:
  - Estados originais: 6 (0 a 5);
  - Estados finais: 3 (normal (0), falha (1), anômalo (2)).
- Normalização robusta (RobustScaler) por sensor;
- Série temporal com amostras a cada 5 minutos.
- Segmentação em janelas:
  - `window_size` = 40, `window_step` = 40
  - janelas não sobrepostas.

Definição do problema

- Série temporal multivariada: $x_t \in \mathbb{R}^d$, $t = 1, \ldots, T$, com $d = 11$ sensores.
- Estados discretos: $s_t \in \{0, 1, 2\}$ (após fusão dos 6 estados iniciais).
- Definimos o **tempo até a próxima mudança de estado**:

$$\tau_t = \min\{\Delta > 0 \ : \ s_{t+\Delta} \neq s_t\}.$$

- Para cada janela de histórico $X = (x_{t-L+1}, \ldots, x_t)$, condicionada ao estado atual s_t, queremos modelar a distribuição preditiva:

$$p(\tau \mid X, s_t).$$

- Na prática, o modelo produz, para cada estado $s \in \{0, 1, 2\}$, parâmetros $\mu_s(X)$ e $\sigma_s^2(X)$ que definem as **PDFs do tempo até mudança de estado**.

Assim, tratamos o problema como um **forecast probabilístico de tempo até evento**.

Revisão Bibliográfica

O que já existe na área:

- Deep learning para RUL (Remaining Useful Life) de máquinas rotativas e motores [1];
- RNNs para séries multivariadas com dados ausentes [2];
- Estimativa de RUL (Remaining Useful Life) com incerteza via redes Bayesianas e métodos probabilísticos [3] e *neural temporal point processes* [4].

Lacuna atacada neste trabalho:

- Poucos trabalhos modelam explicitamente a **distribuição do tempo até a próxima mudança de estado** em equipamentos industriais, sendo a maioria focada em estimativas pontuais de tempo até falha (RUL) [5, 6];
- Não encontramos, na literatura, aplicações com **RNN + VAE** que estimem **PDFs condicionais por estado** e avaliem **calibração probabilística** (cov90, width90) em compressores industriais.

Função de custo

- O modelo admite a função de custo:

$$\mathcal{L} = \mathcal{L}_{\text{vae}} = \mathbb{E}_{q(z|\tau)}\big[-\log p(\tau \mid z)\big] + \beta\,\mathrm{KL}\big(q(z \mid \tau)\,\|\,\mathcal{N}(0, I)\big)$$

- O primeiro termo corresponde à verossimilhança negativa do tempo até a mudança de estado, modelado como Gaussiana, enquanto o segundo termo regulariza o espaço latente via divergência KL ponderada por β.

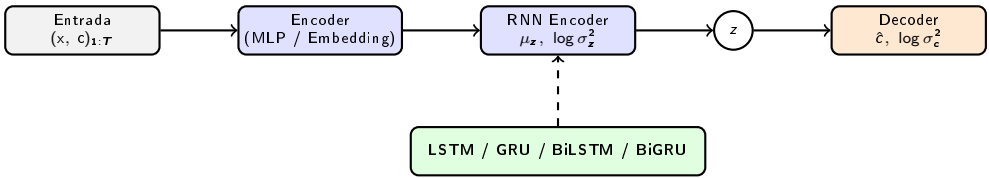
- Para o modelo com difusão de missingness, a função de custo é:

$$\mathcal{L}_{\text{difusão}} = \lambda_m \mathcal{L}_{\text{miss}} + \lambda_v \mathcal{L}_{\text{vae}}$$

$$\mathcal{L}_{\text{miss}} = \mathrm{BCE}(m, \hat{m}), \quad \hat{m} = \sigma(f_{\text{miss}}(h))$$

- O modelo permite estimar **incerteza** e a **cobertura probabilística** do tempo até mudança de estado.

Arquitetura base e arquiteturas avaliadas



- **Arquitetura base**: Encoder temporal recorrente acoplado a um VAE latente.
- A entrada $X = (x, c)_{1:T}$ é codificada e processada pelo **RNN encoder**, que parametriza a distribuição latente $(\mu_z, \log \sigma_z^2)$.
- A partir da amostragem $z \sim q(z \mid X)$, o **decoder** estima a distribuição sobre a mudança futura, produzindo $\hat{c}$ e sua incerteza associada $\log \sigma_c^2$.
- **Arquiteturas avaliadas**:
  - **Deep MLP**: baseline feedforward (sem recorrência);
  - **LSTM**: LSTM unidirecional e **BiLSTM**;
  - **GRU**: GRU unidirecional e **BiGRU**.
  - **VAE**: Variational Autoencoder.

Variações de treinamento em relação à arquitetura base

Arquitetura base fixa: Encoder temporal **BiLSTM** + VAE latente.

1. **Variações de otimização**
   - **BiLSTM + Warmup & Scheduler**: Aumenta gradualmente a taxa de aprendizado no início e depois aplica um *scheduler* de decaimento, reduzindo instabilidades nas primeiras épocas e melhorando a convergência inicial.
   - **LSTM/BiLSTM + RAdam**: Substitui o Adam por **RAdam**, que corrige a variância adaptativa nos primeiros passos, tornando o treinamento mais estável.

2. **Variações de regularização**
   - **BiLSTM + Variational Dropout**: Aplica *variational dropout* ($p = 0{,}2$) compartilhado no tempo, reduzindo overfitting sem quebrar dependências temporais.
   - **BiLSTM + Difusão (missingness)**: Acrescenta um objetivo probabilístico extra $\mathcal{L} = \lambda_m \mathcal{L}_{miss} + \lambda_v \mathcal{L}_{vae}$, forçando o modelo a ser robusto a *missing data* e a calibrar melhor incerteza.
   - **BiLSTM + LayerNorm**: Aplica **LayerNorm** às ativações da BiLSTM, estabilizando as distribuições internas com efeito regularizante indireto.
   - **BiLSTM + AdamW (L2/Weight Decay)**: Usa **AdamW** com *weight decay* desacoplado ($\lambda = 10^{-4}$), acrescentando regularização L2 explícita nos pesos e melhorando generalização.

Configuração Experimental

- Divisão treino/teste: 80% / 20%.
- Dimensões do espaço latente foram validadas com a divisão 60% / 20% / 20%.
- Treinamento por 500 épocas, batch size 256.
- Paciênica de 50 épocas para early stopping.
- Função de perda multi-tarefa conforme descrito.
- Otimização e Regularização tratadas como diferentes modelos.
- Otimizador: Adam (exceto variações).
- Taxa de aprendizado inicial: 3×10^{-4}.
- Early stopping baseado no NELBO do teste.
- Avaliação final no conjunto de teste.

Figuras de Mérito utilizadas

- **NELBO** (Negative Evidence Lower Bound):
  - Loss probabilística minimizada no treinamento;
  - Maximiza implicitamente a verossimilhança (ELBO);
  - Balanceia reconstrução e regularização latente (KL).

  $$NELBO = \mathbb{E}_{q(z|x)}[-\log p(x \mid z)] + \mathrm{KL}(q(z \mid x) \,\|\, p(z))$$

- **NLL** (Negative Log-Likelihood):
  - Generaliza o MSE ao modelar explicitamente a variância da distribuição predita;
  - Penaliza erros grandes *e* variâncias mal calibradas (super ou subestimação de incerteza).

  $$\mathrm{NLL} = -\log p(x \mid \mu, \sigma^2) \quad = \frac{(x-\mu)^2}{2\sigma^2} + \frac{1}{2}\log \sigma^2 + \mathrm{cte}$$

Figuras de Mérito utilizadas

- **MSE** (Mean Squared Error):
  - Erro médio de reconstrução das séries;
  - Mede fidelidade gerativa.
- **Cobertura 90%** (cov_90):
  - Mede a fração de amostras reais que caem dentro do intervalo preditivo teórico [5%, 95%];
  - Avalia se o desvio padrão estimado produz uma **calibração probabilística consistente** com a cobertura nominal de 90%.
- **Largura do intervalo 90%** (width_90):
  - Corresponde à largura teórica do intervalo [5%, 95%] derivado da distribuição predita;
  - Quantifica a **sharpness** do modelo: intervalos menores indicam maior confiança, desde que a cobertura permaneça bem calibrada.

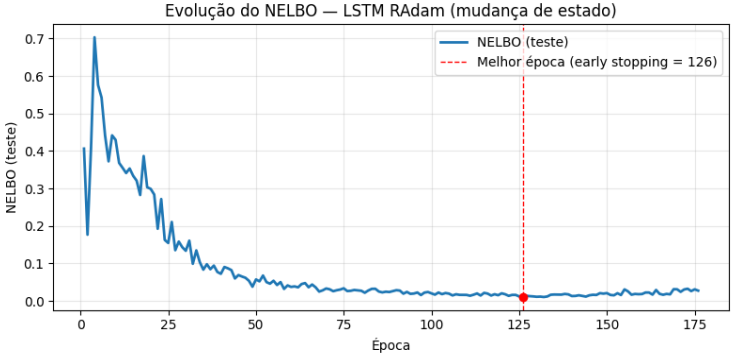
Resultados quantitativos

Modelo	NELBO ↓	NLL ↓	MSE (micro) ↓	Cov90 → 0.90	Width90 ↓
LSTM RAdam	**0.010**	**0.000**	0.001 ± 0.000	**1.000**	**0.270**
BiLSTM RAdam	0.023	0.023	0.002 ± 0.000	0.946	0.270
BiLSTM Warmup/Sched.	0.033	0.033	0.002 ± 0.000	0.946	0.270
LSTM	0.064	0.063	0.002 ± 0.000	0.937	0.270
BiLSTM	0.150	0.150	0.002 ± 0.000	0.920	0.270
BiGRU	0.203	0.202	0.001 ± 0.000	0.875	0.270
BiLSTM AdamW	0.230	0.229	0.001 ± 0.001	0.866	0.270
BiLSTM Difusão	0.329	0.328	0.002 ± 0.001	0.839	0.270
BiLSTM LayerNorm	0.383	0.332	0.054 ± 0.001	0.839	0.270
GRU	0.464	0.463	0.003 ± 0.001	0.812	0.270
BiLSTM VarDrop 0.2	0.992	0.991	0.002 ± 0.001	0.705	0.270
DEEP	8016.760	33.711	0.819 ± 0.011	0.250	0.270

Tabela: Comparação probabilística entre modelos. O critério principal de seleção é a minimização do NELBO. A métrica Cov90 avalia calibração probabilística, cujo valor ideal é próximo de 0.90. A métrica Width90 mede a precisão dos intervalos de previsão e é comparada apenas entre modelos com cobertura adequadamente calibrada.

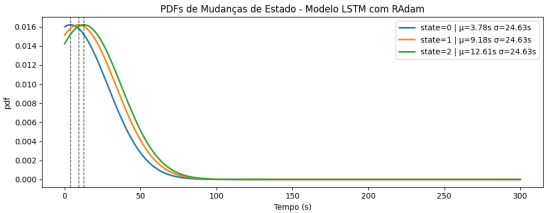
Curva de treinamento — Modelo **LSTM RAdam**

- A curva de NELBO apresenta oscilações significativas nas primeiras épocas, refletindo o desbalanceamento entre estados e a raridade de eventos de mudança.
- Após as primeiras dezenas de épocas, o modelo entra em um regime estacionário, operando em patamar baixo de NELBO, com oscilações locais associadas à natureza estocástica da estimação probabilística.
- A época 126 corresponde a um mínimo consistente do NELBO no conjunto de teste, sendo adotada como ponto de early stopping por representar o melhor desempenho probabilístico observado.

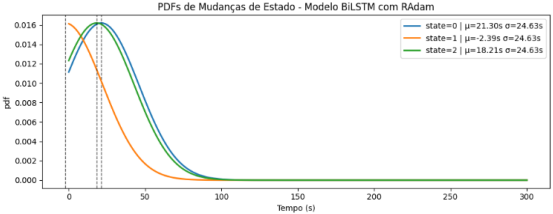


PDFs do tempo até mudança de estado — Comparação LSTM vs BiLSTM (com RAdam)

LSTM (RAdam)



BiLSTM (RAdam)



Distribuições preditivas do tempo até mudança de estado condicionadas ao histórico observado, evidenciando diferenças de incerteza e separabilidade entre arquiteturas.

- PDFs do tempo até a mudança, condicionadas ao histórico $X_{1:t}$.
- Curvas por estado futuro, com médias distintas e incerteza associada.
- A arquitetura bidirecional apresenta maior separação entre distribuições, indicando melhor desambiguação entre estados futuros.

Discussão dos resultados — Mudança de estado

- **Qualidade probabilística**:
  - O **LSTM RAdam** atinge o **menor NELBO** entre os modelos, indicando melhor ajuste probabilístico para a tarefa de mudança de estado;
  - Baixos valores de NLL e MSE micro confirmam alta precisão na modelagem de eventos de transição raros.
- **Calibração das incertezas**:
  - O **LSTM RAdam** apresenta **cobertura próxima de 1.0**, evidenciando excelente calibração probabilística;
  - Width90 constante sugere intervalos preditivos estáveis e comparáveis entre arquiteturas.
- **Impacto arquitetural**:
  - Arquiteturas unidirecionais mostraram-se mais adequadas ao caráter **causal** do problema de mudança de estado;
  - Em cenários fortemente desbalanceados, o principal desafio é a **otimização sob gradientes raros**, e não o overfitting;
  - Otimizadores adaptativos (RAdam, warmup) estabilizam o treinamento e facilitam a captura de eventos raros, enquanto arquiteturas excessivamente complexas tendem a diluir esses sinais.

Conclusões — Mudança de estado

- A predição de mudanças de estado é um problema fortemente desbalanceado e altamente sensível à calibração probabilística.
- Arquiteturas recorrentes probabilísticas são essenciais:
  - Capturam dependências temporais relevantes ao longo da janela;
  - Modelam explicitamente incertezas associadas à transição de estado.
- O **LSTM RAdam** é o modelo mais adequado neste estudo:
  - Melhor NELBO global entre as arquiteturas avaliadas;
  - Excelente calibração (Cov90 ≈ 1.0) com largura de intervalo controlada;
  - Melhor alinhamento com a natureza causal da tarefa.
- Esses resultados indicam potencial para:
  - Detecção antecipada de mudanças de regime;
  - Suporte a sistemas de monitoramento e alerta industrial.
- Melhorias futuras:
  - Utilizar modelos do tipo *Mixture of Gaussians* com *VAMP Prior* para capturar o próximo estado operacional;

Referências I

[1] L. Wang *et al.*, "Deep learning approaches for remaining useful life prediction: A comprehensive review," in *IEEE International Conference on Prognostics and Health Management (ICPHM)*, 2025.

[2] Z. Che *et al.*, "Recurrent neural networks for multivariate time series with missing values," *Scientific Reports*, vol. 8, no. 1, p. 6085, 2018.

[3] J. Rivas *et al.*, "Remaining useful life estimation with uncertainty quantification using bayesian neural networks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3255–3264, 2022.

[4] O. Shchur, M. Biloš, and S. Günnemann, "Neural temporal point processes: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[5] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long short-term memory network for remaining useful life estimation," in *Annual Conference of the PHM Society*, 2017.

[6] A. Rivas, P. Santos, D. Peres, and S. Vinga, "Predictions of component remaining useful lifetime and its uncertainty using bayesian neural networks," *Reliability Engineering & System Safety*, vol. 222, p. 108469, 2022.