

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

CPE727 – Deep Learning

**Fernanda Mickosz Villa Verde** ([fernandamvv@lps.ufrj.br](mailto:fernandamvv@lps.ufrj.br))

10 de dezembro de 2025

# Motivação e Objetivos do trabalho

## 1 Introdução

### Motivação

- Mudanças de estado em compressores impactam segurança, desempenho e manutenção.
- Prever o **tempo até a próxima mudança**, com incerteza, apoia decisões operacionais.

### Objetivo

- Modelar probabilisticamente o tempo até a próxima mudança de estado usando RNNs e VAEs.

### Avaliação

- Comparar RNNs (LSTM, GRU, BiRNN) e estratégias de treinamento com regularização e otimização

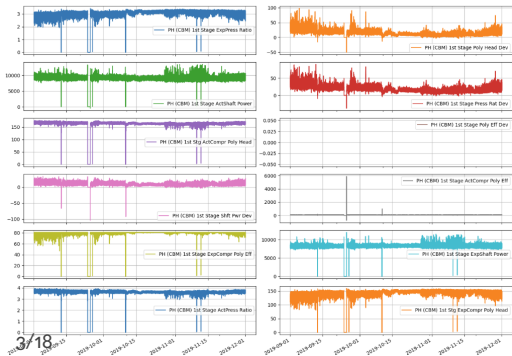
# Base de dados e pré-processamento

## 2 Base de dados e pré-processamento

### Base de dados

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

Séries Temporais do Compressor



### 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

## 3 Formulação do Problema

- Série temporal multivariada:  $\mathbf{x}_t \in \mathbb{R}^d$ ,  $t = 1, \dots, T$ , 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  $\mathbf{X} = (x_{t-L+1}, \dots, x_t)$ , condicionada ao estado atual  $s_t$ , queremos modelar a distribuição preditiva:

$$p(\tau \mid \mathbf{X}, s_t).$$

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

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

# Revisão Bibliográfica

## 4 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

## 4 Revisão Bibliográfica

- $\mathcal{L}_{\text{vae}} = \mathbb{E}_{q(z|\tau)}[-\log p(\tau | z)] + \beta \text{KL}(q(z | \tau) \| \mathcal{N}(0, I))$
- 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  $\beta$ .
- Para o modelo com difusão de missingness, a função de custo é:

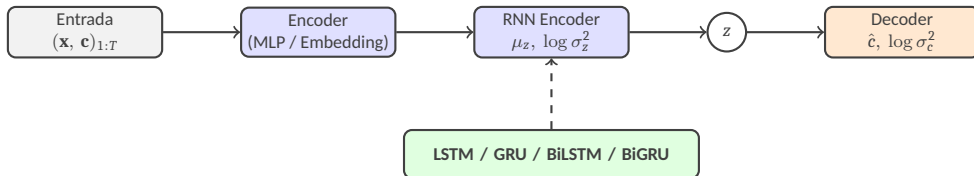
$$\mathcal{L}_{\text{difusão}} = \lambda_m \mathcal{L}_{\text{miss}} + \lambda_v \mathcal{L}_{\text{vae}}$$

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

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

# Arquitetura base e arquiteturas avaliadas

## 4 Revisão Bibliográfica



- **Arquitetura base:** Encoder recorrente acoplado a um VAE latente para previsão probabilística de mudanças de estado.
- **Arquiteturas avaliadas:**
  - **Deep MLP:** baseline feedforward (sem recorrência);
  - **LSTM:** LSTM unidirecional e **BiLSTM**;
  - **GRU:** GRU unidirecional e **BiGRU**.
  - **VAE:** Variational Autoencoder.

# Variações de treinamento

## 5 Arquiteturas e Variações

**Base fixa:** BiLSTM + VAE latente.

### Otimização

- **Warmup & Scheduler:** estabiliza as primeiras épocas.
- **RAdam:** reduz instabilidade do Adam no início do treino.

### Regularização

- **Variational Dropout:** reduz overfitting temporal.
- **Difusão (missingness):** melhora robustez e calibração.
- **LayerNorm:** estabiliza ativações internas.
- **AdamW:** L2 desacoplado para melhor generalização.



# Configuração Experimental

## 6 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.
- Paciência 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:  $3 \times 10^{-4}$ .
- Early stopping baseado no NELBO do teste.
- Avaliação final no conjunto de teste.

# Figuras de Mérito utilizadas

## 7 Resultados

- **NELBO** (Negative Evidence Lower Bound):

- Loss probabilística minimizada no treinamento;
- Maximiza implicitamente a verossimilhança (ELBO);
- Balanceia reconstrução e regularização latente (KL).
- $NELBO = \mathbb{E}_{q(z|x)}[-\log p(x | z)] + \text{KL}(q(z | x) || p(z))$

- **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).

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

# Figuras de Mérito utilizadas

## 7 Resultados

- **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.

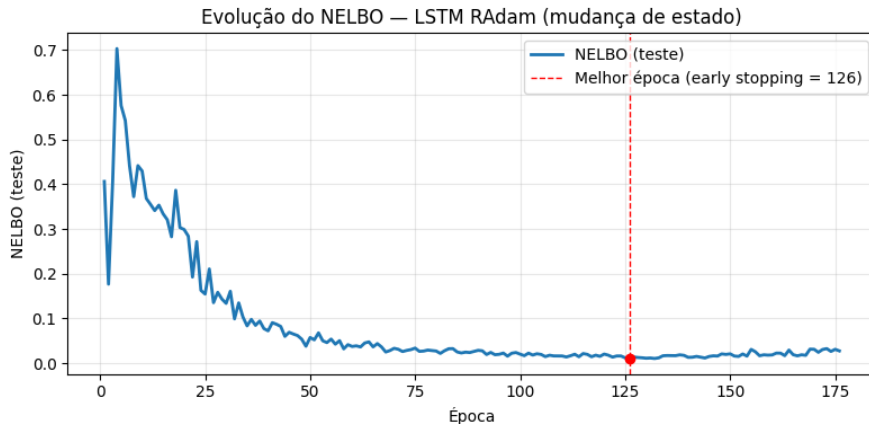
# Resultados quantitativos

## 7 Resultados

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

# Curva de treinamento — Modelo LSTM RAdam

7 Resultados

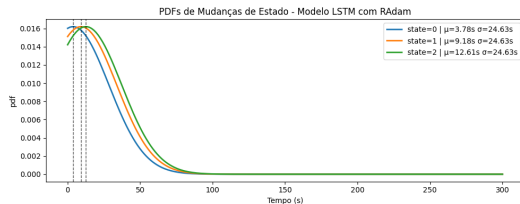


**Figura:** Evolução do NELBO no conjunto de teste do modelo LSTM RAdam.

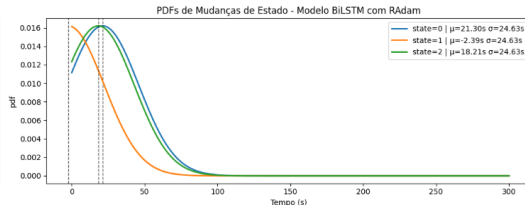
# Distribuições preditivas do tempo até mudança de estado

## 7 Resultados

### LSTM (RAdam)



### BiLSTM (RAdam)



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

# Discussão dos resultados

## 8 Discussão e Conclusões

- **Qualidade probabilística:** **LSTM RAdam** apresenta o menor NELBO, indicando melhor ajuste probabilístico ao tempo até mudança de estado.
- **Calibração das incertezas:** Cobertura próxima de 1.0 (Cov90), com intervalos estáveis entre arquiteturas.
- **Impacto arquitetural:**
  - O BiLSTM aprende representações latentes mais suaves, resultando em PDFs visualmente melhor separadas, enquanto o LSTM unidirecional apresenta melhor calibração e desempenho probabilístico global.
  - Em cenários fortemente desbalanceados, o principal desafio é a **otimização sob gradientes raros**, e não o overfitting;

# Conclusões

## 8 Discussão e Conclusões

- Predição de mudança de estado é um problema desbalanceado e sensível à calibração probabilística.
- Modelos recorrentes probabilísticos são essenciais para capturar dependências temporais e incerteza.
- **LSTM RAdam** é o modelo mais adequado neste estudo: melhor NELBO, boa calibração e alinhamento causal.
- Aplicações potenciais: monitoramento antecipado e suporte à decisão industrial.
- Trabalhos futuros: modelos multimodais (Mixture of Gaussians, VAMP Prior).



## Referências Bibliográficas

### 9 Referências Bibliográficas

- [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.

## Referências Bibliográficas

### 9 Referências Bibliográficas

- [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.