



# Modelos neurais para classificação de falhas em poços de petróleo usando o 3W dataset

CPE 727 - Aprendizado Profundo

**Luíza Helena de Andrade** ([luiza.andrade@lps.ufrj.br](mailto:luiza.andrade@lps.ufrj.br))

9 de dezembro de 2025





# Conteúdo da Apresentação

## 1 Introdução

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas



# Classificação de Falhas em Poços de Petróleo

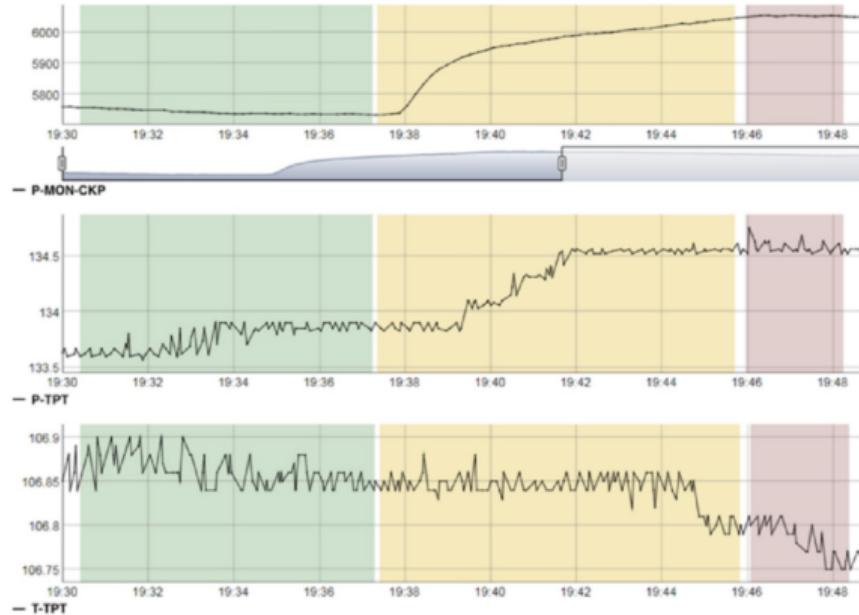
## 1 Introdução

- A classificação correta de falhas é essencial; diagnósticos tardios ou imprecisos podem gerar **perda de produção e danos a equipamentos**.
- O trabalho propõe a aplicação de modelos de **Aprendizado Profundo** para análise e classificação de falhas no **3W Dataset**.
- O foco está em cinco arquiteturas: **LSTM**, **GRU**, **CNN 1D**, **LSTM-AE** e **GRU-AE**, utilizando uma **MLP** como baseline.

# Conjunto de Dados 3W

Base de Dados Pública

- Contém **séries temporais multivariadas** de leituras de sensores de poços.
- **Dados reais:** Registros de **21 poços** ao longo de 4 anos de operação.
- **Simulações:** Complementado por simulações do simulador de fluxo multifásico dinâmico OLGA [1].
- **Rótulos:** 10 classes, incluindo 1 classe normal (Classe 0) e 9 classes de falhas específicas.



**Figura:** Instância real rotulada como anormal (restrição rápida no PCK, classe 6). Fonte: [1]



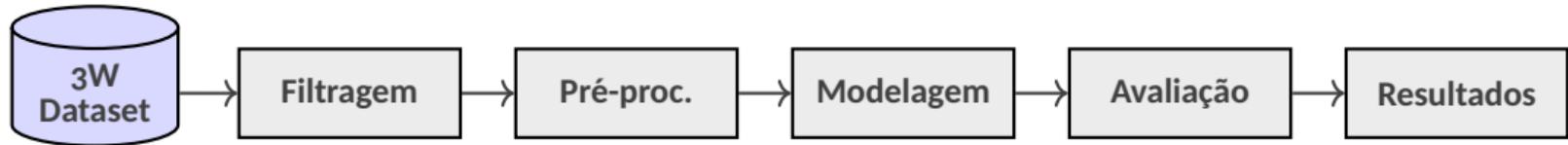
# Conteúdo da Apresentação

## 2 Metodologia e Modelos

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas

# Pipeline de Classificação de Falhas

## 2 Metodologia e Modelos



- **Filtragem:** Seleção apenas de eventos reais do 3W Dataset. Exclusão de dados simulados para análise de casos reais de operação.
- **Pré-processamento:** Janelamento temporal, normalização e padronização de 5 variáveis sensoriais.
- **Modelos avaliados:**
  - **MLP:** baseline feedforward (sem recorrência);
  - **LSTM e LSTM-AE:** Long Short-Term Memory (com autoencoder);
  - **GRU e GRU-AE:** Gated Recurrent Unit (com autoencoder);
  - **CNN 1D:** Convolutional Neural Network 1D.
- **Avaliação:** Validação cruzada K-Fold (5 folds). Métricas: Acurácia e F1-Score com desvio padrão. Matrizes de confusão normalizadas e curvas de loss.

# LSTM: Arquitetura

## 2 Metodologia e Modelos

- Utiliza três portas: **entrada, esquecimento e saída**.
- A memória interna é controlada para decidir o que manter, escrever ou expor.
- O estado de célula funciona como **trilha de gradiente preservada**.
- Atualização da célula:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

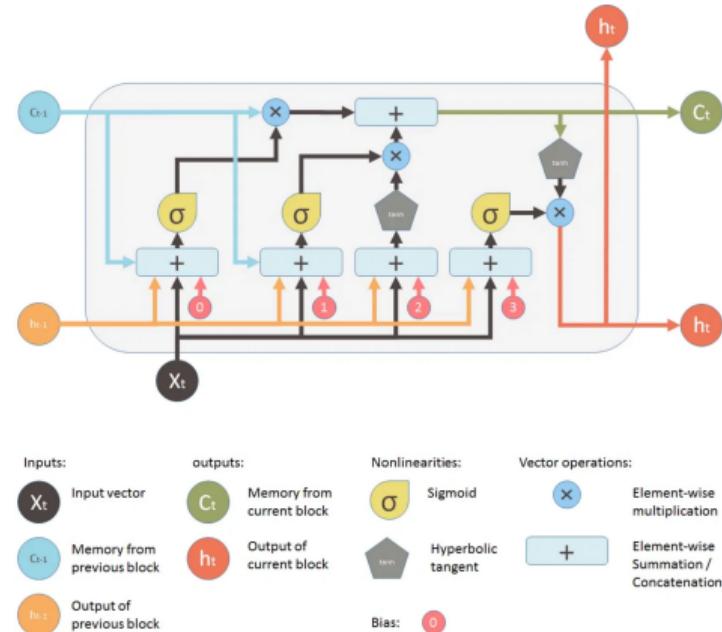


Figura: Estrutura da LSTM.[2]

# GRU: Arquitetura

## 2 Metodologia e Modelos

- Usa apenas duas portas: **atualização** e **reset**.
- Não possui estado de célula separado — toda a memória está no estado oculto  $h_t$ .
- A porta de atualização controla quanto do passado é mantido.
- Atualização do estado oculto:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

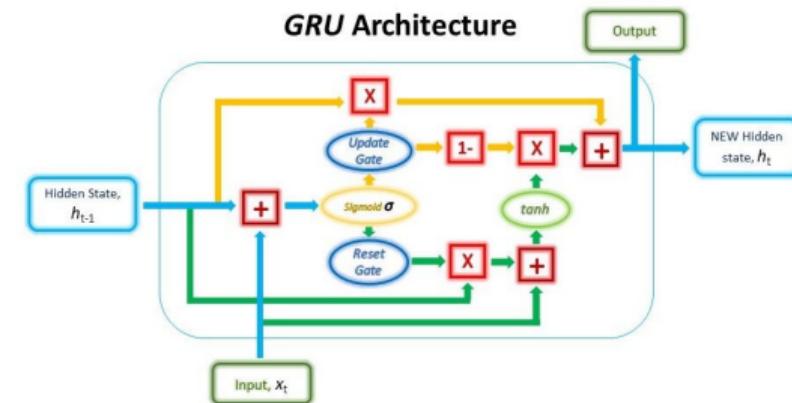


Figura: Estrutura da GRU.[3]

# CNN1D: Arquitetura

## 2 Metodologia e Modelos

- Extrai **padrões locais** da série temporal por meio de convoluções 1D.
- Cada filtro analisa pequenas janelas do sinal.
- Pooling reduz dimensionalidade e destaca características relevantes.
- A saída convolucional é achatada e enviada a camadas densas para classificação.
- Pipeline típico:

$$y = \text{Dense}(\text{Flatten}(\text{Conv1D}(x)))$$

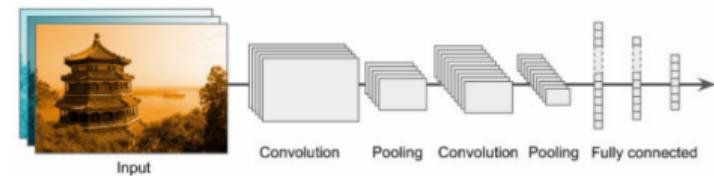


Figura: Arquitetura típica de CNN.[4]

# Autoencoder: Arquitetura

## 2 Metodologia e Modelos

- Composto por duas partes: **encoder** e **decoder**.
- O encoder comprime a entrada em uma **representação latente** reduzida.
- O decoder reconstrói a entrada original a partir dessa representação.
- O objetivo é aprender:

$$x' = g_\phi(f_\theta(x)) \approx x$$

- Estrutura típica:

Entrada → Encoder →  $h$  → Decoder → Reconstrução

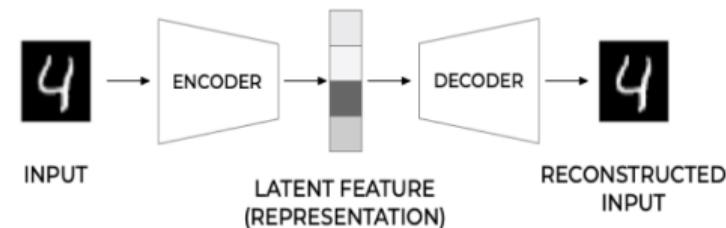


Figura: Estrutura de um autoencoder com representação latente. [5]



# Processo de Treinamento e Avaliação

## 2 Metodologia e Modelos

- Pré-processamento Padronizado:
  - Janelamento
  - Features: 5 variáveis sensoriais
  - Dataset: eventos reais
- Avaliação:
  - Métricas: Accuracy e F1 (weighted)  
± desvio padrão
  - Matrizes de Confusão:  
normalizadas

- Balanceamento de Classes:
  - Pesos Adaptativos por fold
  - CrossEntropyLoss ponderada
- Configurações de Treinamento:
  - K-Fold: 5 folds (shuffle,  
random\_state=42)
  - AdamW (LR=1e-4,  
weight\_decay=1e-5)
  - 100 épocas, batch size 32
- Análise do Espaço Latente:
  - t-SNE: bottleneck 128D/64D → 2D
  - Scatter plots por classe
  - Modelo *best\_fold* retreinado

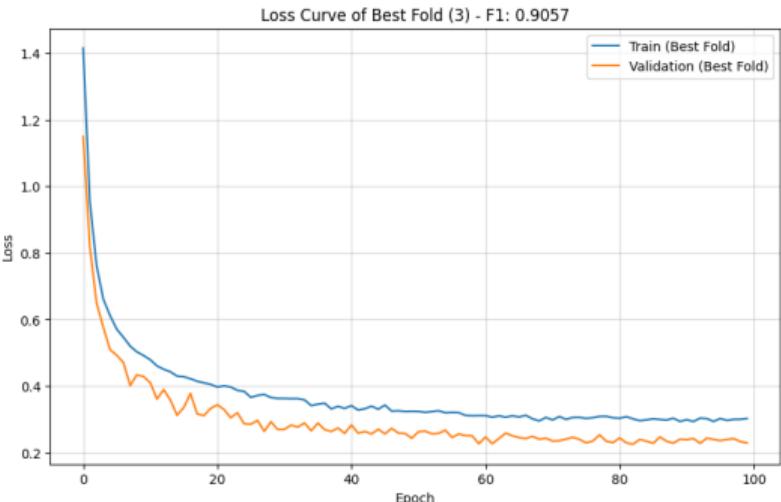
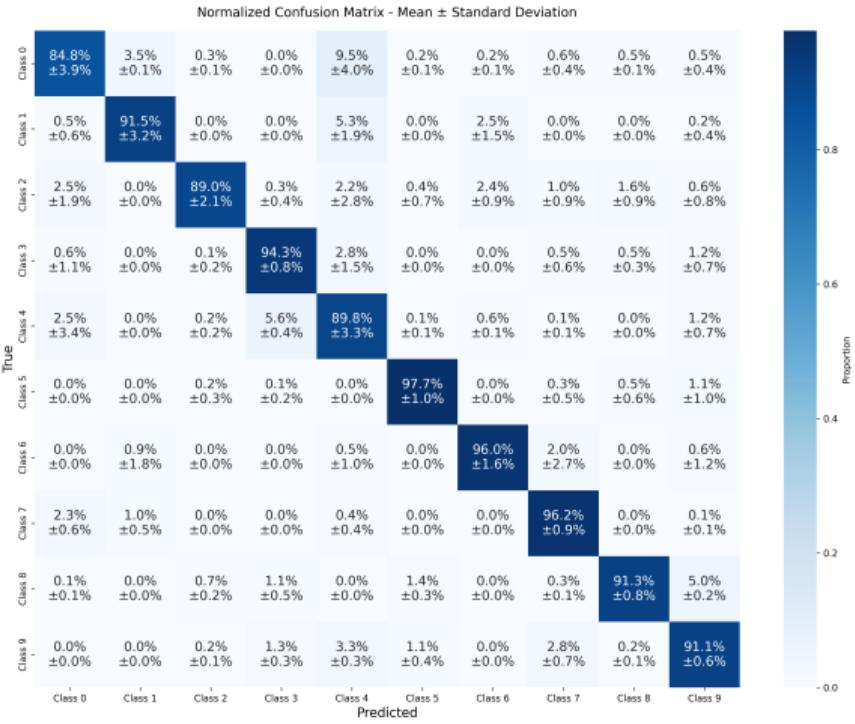
# Conteúdo da Apresentação

## 3 Resultados

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas

# Matriz e Loss Baseline MLP

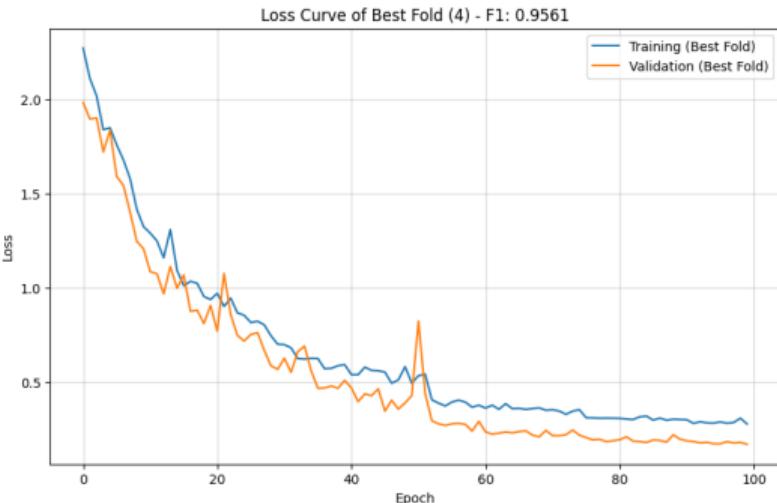
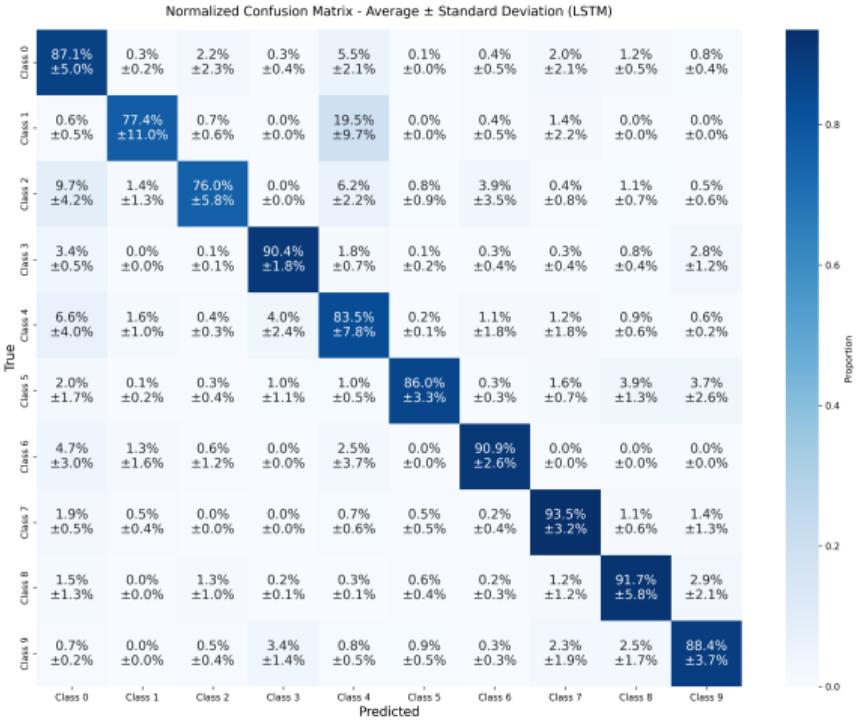
## 3 Resultados



- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

# Matriz e Loss LSTM

## 3 Resultados

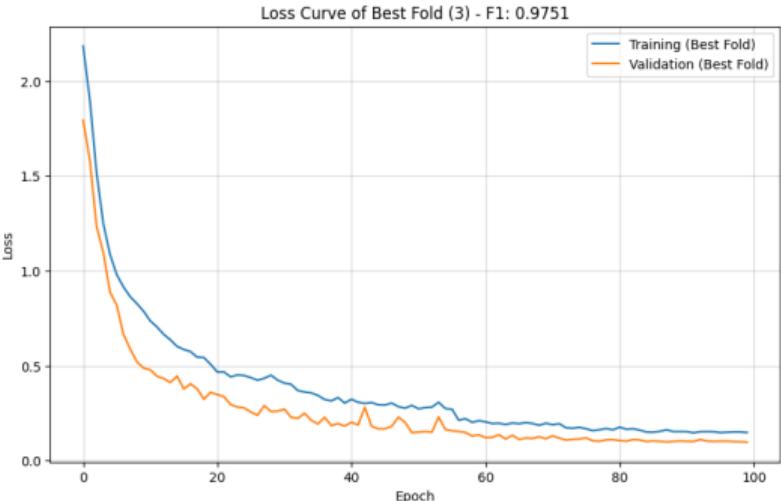


- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

# Matriz e Loss GRU

## 3 Resultados

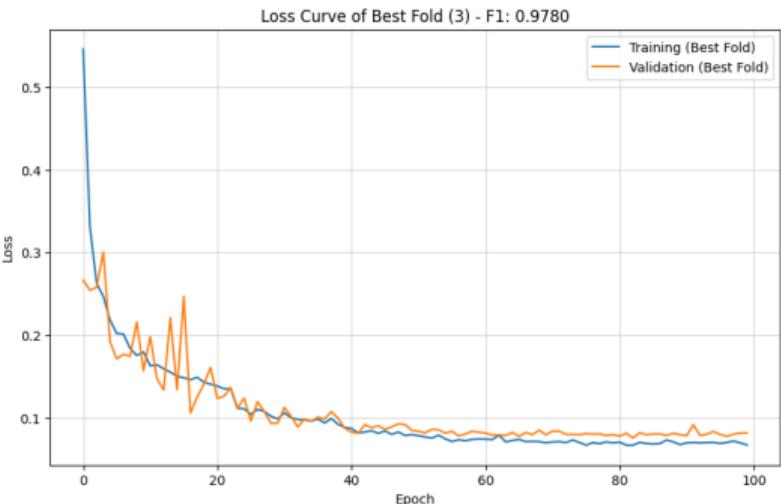
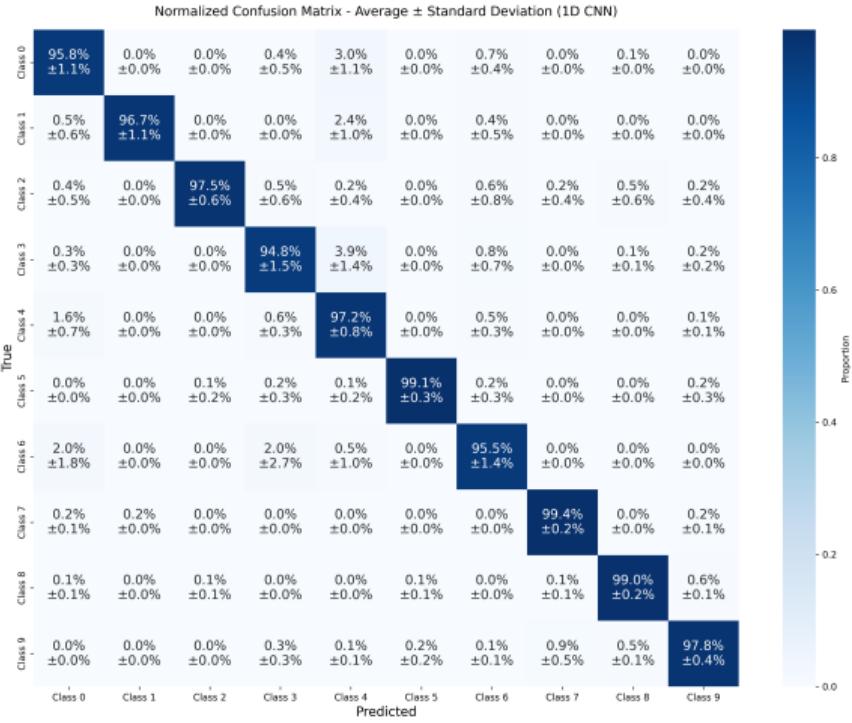
		Normalized Confusion Matrix - Average $\pm$ Standard Deviation (Pure GRU)									
		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
True	Class 0	96.5% $\pm$ 0.9%	0.0% $\pm$ 0.0%	0.2% $\pm$ 0.1%	0.0% $\pm$ 0.0%	2.5% $\pm$ 0.4%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%	0.2% $\pm$ 0.1%	0.5% $\pm$ 0.3%	0.0% $\pm$ 0.0%
	Class 1	0.7% $\pm$ 0.9%	92.2% $\pm$ 4.9%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%	6.8% $\pm$ 5.0%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%	0.2% $\pm$ 0.4%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%
	Class 2	8.8% $\pm$ 2.6%	1.1% $\pm$ 2.2%	84.4% $\pm$ 5.3%	0.0% $\pm$ 0.0%	0.9% $\pm$ 1.5%	0.3% $\pm$ 0.4%	1.1% $\pm$ 1.0%	0.5% $\pm$ 0.6%	1.8% $\pm$ 1.6%	1.1% $\pm$ 0.7%
	Class 3	2.4% $\pm$ 0.9%	0.0% $\pm$ 0.0%	0.0% $\pm$ 0.0%	90.8% $\pm$ 2.4%	3.7% $\pm$ 1.3%	0.3% $\pm$ 0.4%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%	0.4% $\pm$ 0.2%	2.3% $\pm$ 0.8%
	Class 4	2.9% $\pm$ 0.7%	0.1% $\pm$ 0.0%	0.1% $\pm$ 0.1%	0.3% $\pm$ 0.1%	95.9% $\pm$ 0.8%	0.1% $\pm$ 0.1%	0.0% $\pm$ 0.0%	0.2% $\pm$ 0.1%	0.3% $\pm$ 0.3%	0.2% $\pm$ 0.1%
	Class 5	0.8% $\pm$ 0.6%	0.1% $\pm$ 0.2%	0.2% $\pm$ 0.3%	0.1% $\pm$ 0.2%	0.1% $\pm$ 0.2%	95.0% $\pm$ 2.3%	0.1% $\pm$ 0.2%	0.0% $\pm$ 0.0%	1.9% $\pm$ 1.5%	1.6% $\pm$ 0.9%
	Class 6	2.6% $\pm$ 2.5%	0.0% $\pm$ 0.0%	2.1% $\pm$ 1.8%	0.0% $\pm$ 0.0%	2.7% $\pm$ 2.6%	0.0% $\pm$ 0.0%	92.0% $\pm$ 2.0%	0.0% $\pm$ 0.0%	0.6% $\pm$ 1.2%	0.0% $\pm$ 0.0%
	Class 7	0.6% $\pm$ 0.3%	0.4% $\pm$ 0.3%	0.1% $\pm$ 0.1%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%	0.0% $\pm$ 0.1%	0.0% $\pm$ 0.0%	98.7% $\pm$ 0.2%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%
	Class 8	0.5% $\pm$ 0.4%	0.0% $\pm$ 0.0%	0.3% $\pm$ 0.2%	0.1% $\pm$ 0.1%	0.1% $\pm$ 0.1%	0.3% $\pm$ 0.1%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%	97.3% $\pm$ 0.9%	1.2% $\pm$ 1.1%
	Class 9	0.5% $\pm$ 0.3%	0.0% $\pm$ 0.0%	0.1% $\pm$ 0.1%	0.6% $\pm$ 0.2%	0.3% $\pm$ 0.2%	0.4% $\pm$ 0.3%	0.0% $\pm$ 0.0%	0.9% $\pm$ 0.4%	1.5% $\pm$ 0.8%	95.7% $\pm$ 1.1%
Predicted		Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9



- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

# Matriz e Loss CNN 1D

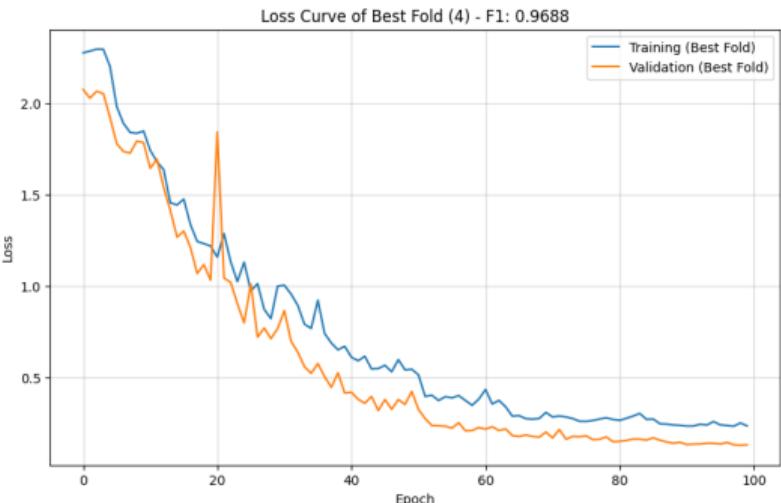
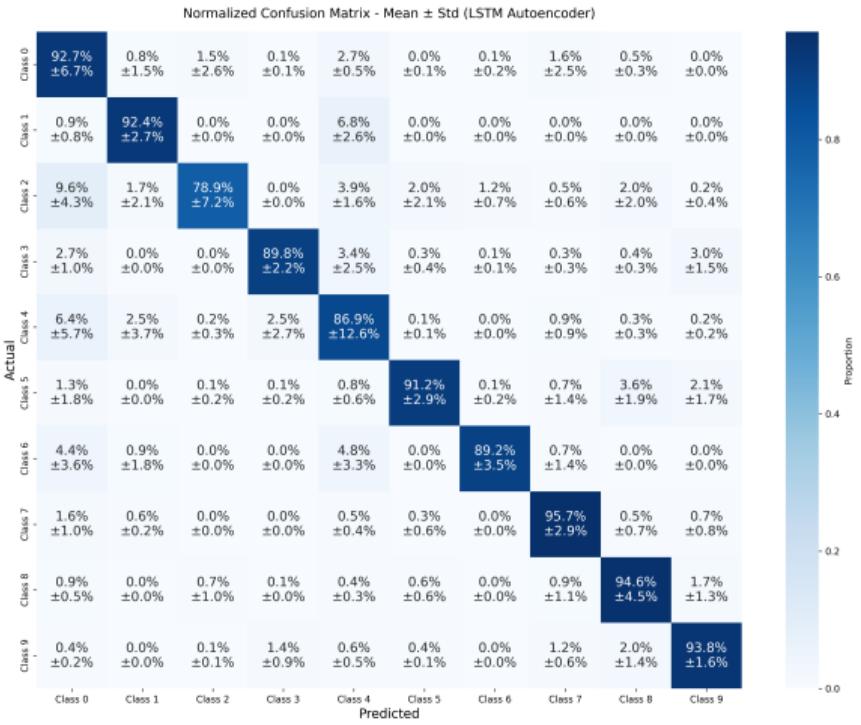
## 3 Resultados



- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

# Matriz e Loss LSTM-AE

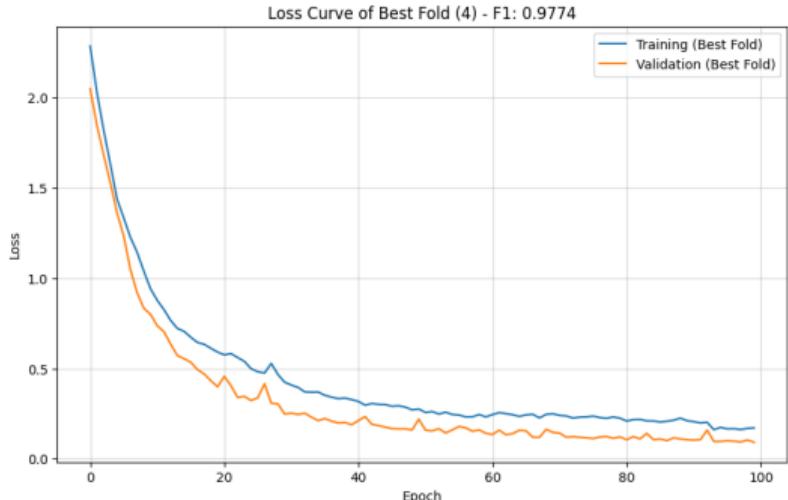
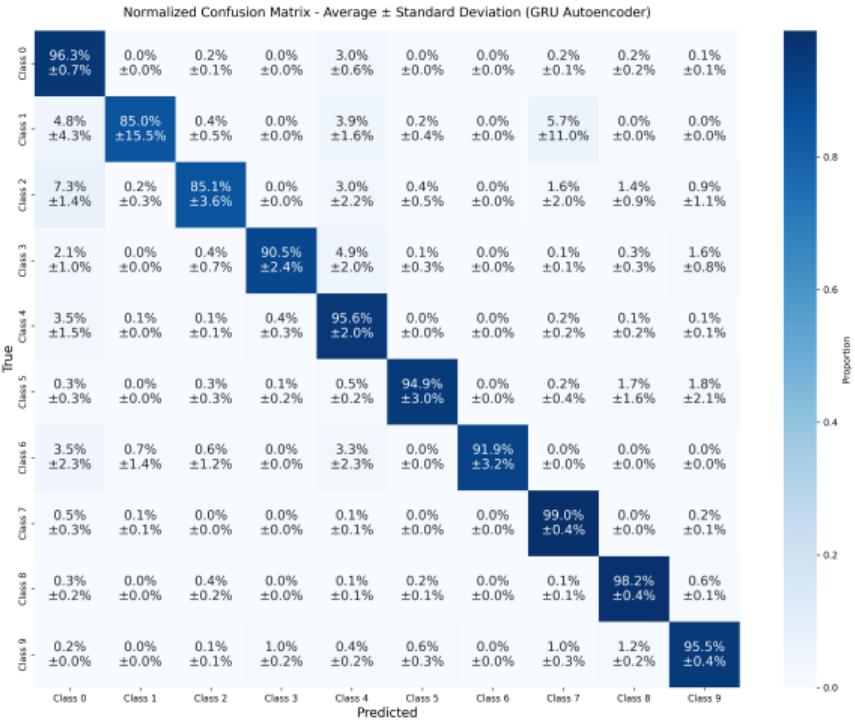
## 3 Resultados



- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

# Matriz e Loss GRU-AE

## 3 Resultados

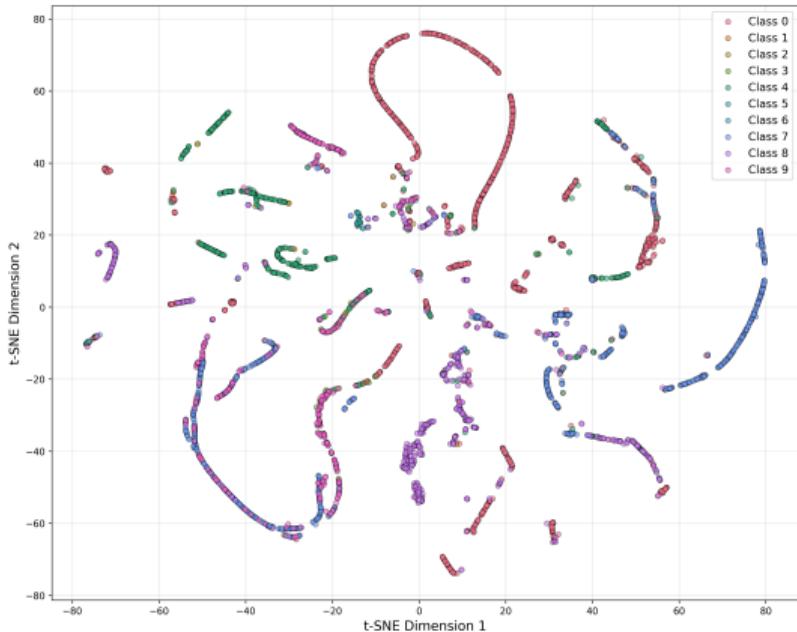


- **Obs:** Dois comportamentos diferentes:
  - **Conjunto de treino:** Atualizações de pesos por Gradiente Estocástico
  - **Conjunto de validação:** Cálculo da loss sem backpropagation para monitorar overfitting e controlar o scheduler.

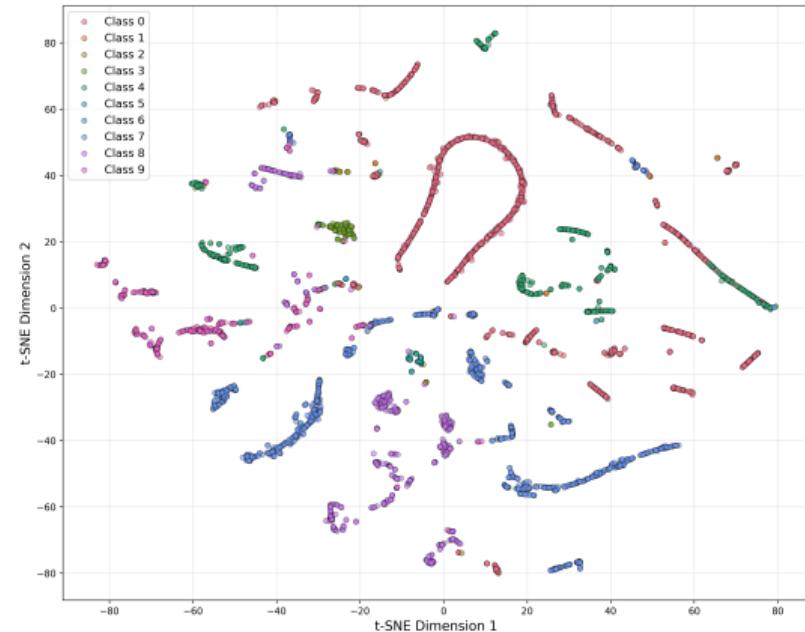
# T-SNE LSTM-AE e GRU-AE

## 3 Resultados

t-SNE of Bottleneck - LSTM Autoencoder



Bottleneck t-SNE - GRU Autoencoder





# Conteúdo da Apresentação

## 4 Comparação

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas

# Comparação dos Resultados

## 4 Comparação

Modelo	Acurácia ( $\mu \pm \sigma$ )	F1-Score ( $\mu \pm \sigma$ )
MLP (Baseline)	89,99 $\pm$ 0,97%	90,60 $\pm$ 0,90%
LSTM	88,90 $\pm$ 4,06%	89,40 $\pm$ 3,65%
GRU	96,73 $\pm$ 0,55%	96,75 $\pm$ 0,54%
CNN 1D	97,49 $\pm$ 0,34%	97,61 $\pm$ 0,28%
LSTM-AE	92,88 $\pm$ 5,35%	93,21 $\pm$ 4,82%
GRU-AE	96,74 $\pm$ 0,56%	96,75 $\pm$ 0,57%

- CNN 1D apresentou o melhor desempenho geral.
- GRU e GRU-AE mantiveram desempenho alto e consistente.
- Modelos com autoencoder tiveram desempenho próximo às versões originais.
- LSTM apresentou a maior variabilidade entre execuções.
- A MLP superou a LSTM, mas ficou abaixo dos modelos temporais avançados e da CNN 1D.

# Comparação Estatística — Acurácia

## 4 Comparação

Comparação	Estatística	Valor-p	Significância
LSTM vs GRU	0.0000	0.0625	Não significativa
LSTM vs CNN1D	0.0000	0.0625	Não significativa
LSTM vs LSTM-AE	2.0000	0.1875	Não significativa
LSTM vs GRU-AE	0.0000	0.0625	Não significativa
GRU vs CNN1D	0.0000	0.0625	Não significativa
GRU vs LSTM-AE	2.0000	0.1875	Não significativa
GRU vs GRU-AE	7.0000	1.0000	Não significativa
CNN1D vs LSTM-AE	0.0000	0.0625	Não significativa
CNN1D vs GRU-AE	1.0000	0.1250	Não significativa
LSTM-AE vs GRU-AE	0.0000	0.0625	Não significativa

**Tabela:** Comparação entre modelos usando o teste de Wilcoxon — Acurácia.

# Comparação Estatística — F1-score

## 4 Comparação

Comparação	Estatística	Valor-p	Significância
LSTM vs GRU	0.0000	0.0625	Não significativa
LSTM vs CNN1D	0.0000	0.0625	Não significativa
LSTM vs LSTM-AE	2.0000	0.1875	Não significativa
LSTM vs GRU-AE	0.0000	0.0625	Não significativa
GRU vs CNN1D	0.0000	0.0625	Não significativa
GRU vs LSTM-AE	2.0000	0.1875	Não significativa
GRU vs GRU-AE	5.0000	1.0000	Não significativa
CNN1D vs LSTM-AE	0.0000	0.0625	Não significativa
CNN1D vs GRU-AE	0.0000	0.0625	Não significativa
LSTM-AE vs GRU-AE	0.0000	0.0625	Não significativa

**Tabela:** Comparação entre modelos usando o teste de Wilcoxon — F1-score.



# Conteúdo da Apresentação

5 Conclusões

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas

# Conclusões

## 5 Conclusões

- A **CNN 1D** apresentou o melhor desempenho geral no 3W Dataset.
- **GRU** e **GRU-AE** obtiveram resultados estáveis.
- Os modelos **LSTM-AE** e **GRU-AE** mantiveram desempenho próximo às versões originais.
- A **LSTM** apresentou a maior variabilidade entre execuções.
- A **MLP (baseline)** superou a LSTM, mas ficou abaixo dos modelos avançados.
- Em todos os testes de Wilcoxon, as hipóteses nulas foram rejeitadas; assim, **não é possível afirmar diferença estatisticamente significativa entre as distribuições dos modelos.**



# Conteúdo da Apresentação

## 6 Referências Bibliográficas

- ▶ Introdução
- ▶ Metodologia e Modelos
- ▶ Resultados
- ▶ Comparação
- ▶ Conclusões
- ▶ Referências Bibliográficas



# Referências Bibliográficas

## 6 Referências Bibliográficas

- [1] R. E. V. Vargas, C. J. Munaro, P. M. Ciarelli, A. G. Medeiros, B. G. d. Amaral, D. C. Barrionuevo, J. C. D. d. Araújo, J. L. Ribeiro, and L. P. Magalhães, "A realistic and public dataset with rare undesirable real events in oil wells," *Journal of Petroleum Science and Engineering*, vol. 181, p. 106223, 2019.
- [2] "Understanding lstm and its diagrams."
- [3] "Capítulo 52 – arquitetura de redes neurais gated recurrent unit (gru)."
- [4] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.
- [5] U. Michelucci, "An introduction to autoencoders," *arXiv preprint arXiv:2201.03898*, 2022.