

Comparação de Arquiteturas de Deep Learning na detecção de anomalias em sensores de poços de petróleo

CPE 727 - Aprendizado Profundo

Bruno Coelho Martins (bruno.martins@smt.ufrj.br)

December 10, 2025

Table of Contents

1 Introdução

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Introdução

1 Introdução

- Classificação de séries temporais é um problema fundamental em Aprendizado de Máquina, com aplicações em sensores industriais, saúde, sinais fisiológicos e previsão de eventos. Nesse trabalho, será estudado modelos de classificação na base 3W.
- Este trabalho estuda a criação de classificadores usando diferentes arquiteturas:
 - MLP (baseline)
 - CNN
 - LSTM
 - GRU
- Utilização do **3W Toolkit**, que facilita experimentos com modelos unificados utilizando a base 3W.

Table of Contents

2 Problema a ser abordado

- ▶ Introdução
- ▶ **Problema a ser abordado**
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Problema a ser abordado

2 Problema a ser abordado

- Como construir um classificador de séries temporais eficiente usando arquiteturas de Deep Learning.
- Dificuldades clássicas:
 - Capturar padrões locais e globais.
 - Lidar com dados ruidosos.
 - Ajustar hiperparâmetros e evitar overfitting.
- Objetivo do trabalho:

Comparar arquiteturas profundas e verificar qual apresenta melhor desempenho no conjunto de dados.

Table of Contents

3 Revisão Bibliográfica

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ **Revisão Bibliográfica**
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Revisão Bibliográfica

3 Revisão Bibliográfica

- A literatura apresenta modelos bem estabelecidos para séries temporais:
 - **CNNs**: capturam padrões locais e variações rápidas [1, 2].
 - **LSTM/GRU**: modelam dependências de longo prazo [3, 4].
 - **MLPs**: úteis como baseline, mas menos robustos.
- Trabalhos recentes mostram que combinações CNN+LSTM superam modelos isolados em diversos benchmarks [5].
- Este trabalho adota o pipeline experimental do **3W Toolkit** baseado no dataset descrito em [6].

Table of Contents

4 Base de Dados

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ **Base de Dados**
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Base de Dados — 3W Dataset

4 Base de Dados

- O **3W Dataset** é um conjunto de dados público voltado para tarefas de **classificação de séries temporais multivariadas**.
- Contém sinais provenientes de sensores industriais.
- Possui dados reais, simulados e criados manualmente.
- Inclui múltiplas classes correspondentes a diferentes eventos e comportamentos operacionais (0 a 9).
- É acompanhado pelo **3W Toolkit**.

Table of Contents

5 Método Proposto

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ **Método Proposto**
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Método Proposto

5 Método Proposto

- Pipeline utilizado:
 1. Pré-processamento das séries temporais.
 2. Seleção de arquiteturas (MLP, CNN, LSTM, GRU).
 3. Treinamento com divisão treinamento/validação/teste.
 4. Métricas avaliadas:
 - Balanced Accuracy (principal que será mostrada/comparada)
 - Precision
 - Recall
 - F1-score
 - CrossEntropyLoss para treinamento
- Uso do **3W Toolkit** para padronização dos experimentos.
- Notebook do workshop usado como referência
(7_model_training_and_evaluation.ipynb)

Método Proposto — Pré-processamento da base

5 Método Proposto

- `clean_data = True`
- Apenas sinais reais
- Divisão dos eventos em treino (75%), validação (15%) e teste (15%)
- Windowing de cada divisão (`window_size = 100`, `overlap = 0.5`)
 - Sinais: hann
 - Labels: boxcar

Método Proposto — Escolha das Classes

5 Método Proposto

- Retirada de classes que geravam poucos dados após windowing:
 - Antes: 234542, 1288(X), 1725(X), 9806, 55383, 6245(X), 269(X), 92521, 41362, 2344(X)
 - Depois: 196703, 9805, 52385, 66321, 64387
- Classes usadas: 0, 3, 4, 7, 8
- Total de eventos: de 1119 para 1019 (train=713, val=153, test=153)
- Total de janelas: train=389601, val=86549, test=108457

Método Proposto — Configurações gerais

5 Método Proposto

- epochs = 100 (com early stopping de 10 épocas)
- Otimizador Adam
- learning_rate = $1e-3$
- weight_decay = $1e-5$

Método Proposto — Configurações dos modelos

5 Método Proposto

- **MLP (Implementado no toolkit):**

- `hidden_sizes = (32, 16)`
- `activation = relu`
- `regularization = 0.01`

- **CNN:**

- `conv_channels = [16, 32]`
- `kernel_sizes = [3, 3]`
- `activation = relu`

- **LSTM e GRU:**

- `hidden_size = 64`
- `num_layers = 2`

Table of Contents

6 Resultados Obtidos

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ **Resultados Obtidos**
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Resultados Obtidos

6 Resultados Obtidos

- Resultados de validação calculados com a classe `ModelAssessment` do toolkit.
- Treinados na CPU AMD Ryzen 5 2600.
- Com o early stopping tradicional:

Modelo	Balanced Acc.	Época Final	Tempo (s)
MLP	0.8752	44	525.39
CNN	0.7724	25	552.47
LSTM	0.8132	26	3987.82
GRU	0.7738	15	7791.70

Resultados Obtidos

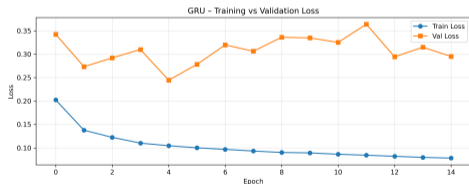
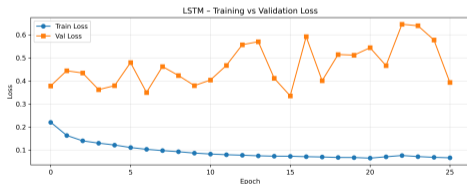
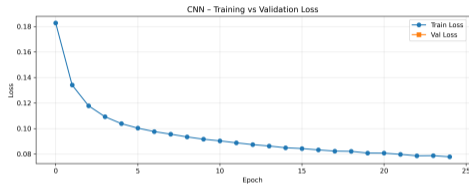
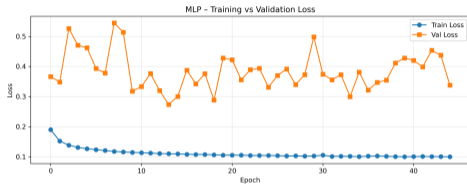
6 Resultados Obtidos

- Utilizando early stop só quando a validation loss ficar abaixo de 0.1 (não ocorreu):

Modelo	Balanced Acc.	Época Final	Tempo (s)
MLP	0.7886	100	1156.52
CNN	0.7520	100	2006.39
LSTM	0.8319	100	14949.02
GRU	0.8508	100	53357.29

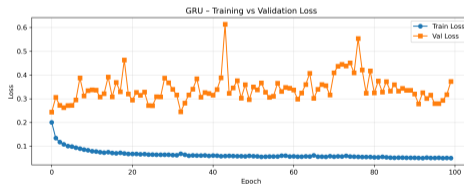
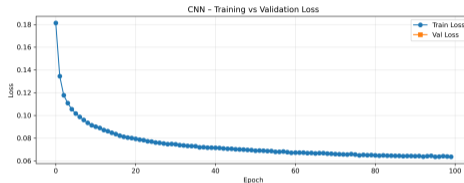
Resultados — Curvas de Loss (early stopping tradicional)

6 Resultados Obtidos



Resultados — Curvas de Loss (early stopping modificado)

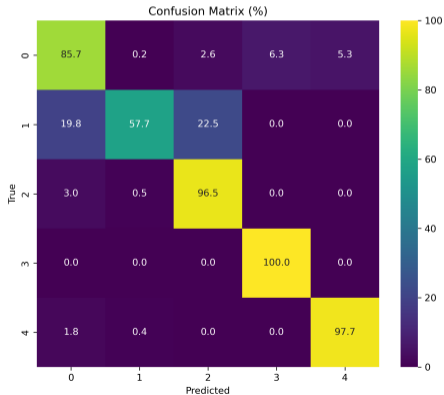
6 Resultados Obtidos



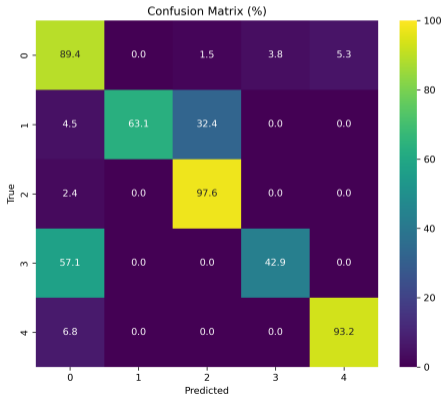
Resultados — Matrizes de Confusão (early stopping tradicional)

6 Resultados Obtidos

MLP



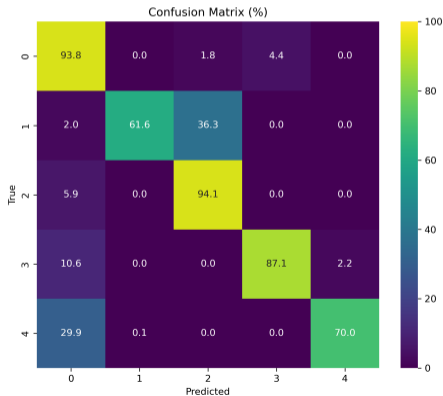
CNN



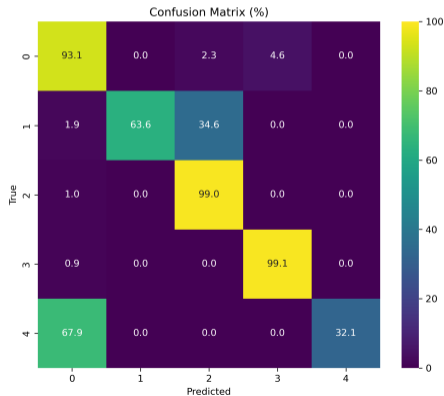
Resultados — Matrizes de Confusão (early stopping tradicional)

6 Resultados Obtidos

LSTM



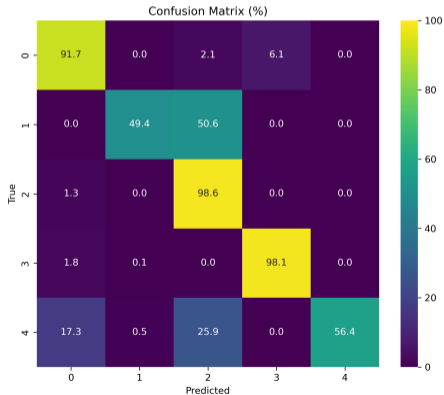
GRU



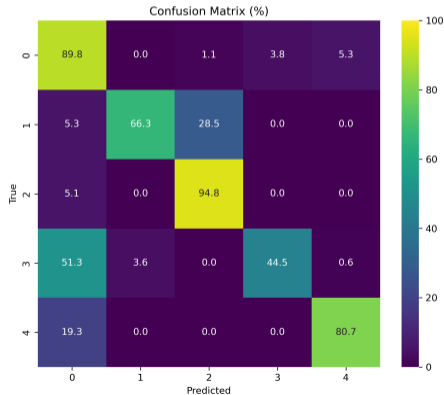
Resultados — Matrizes de Confusão (early stopping modificado)

6 Resultados Obtidos

MLP



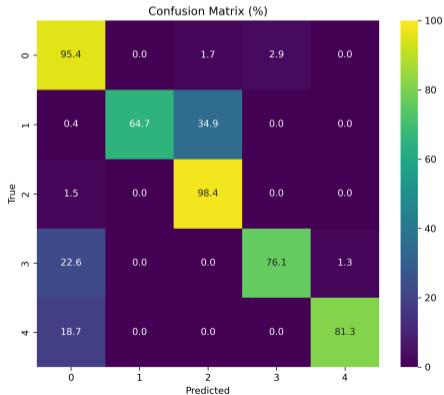
CNN



Resultados — Matrizes de Confusão (early stopping modificado)

6 Resultados Obtidos

LSTM



GRU

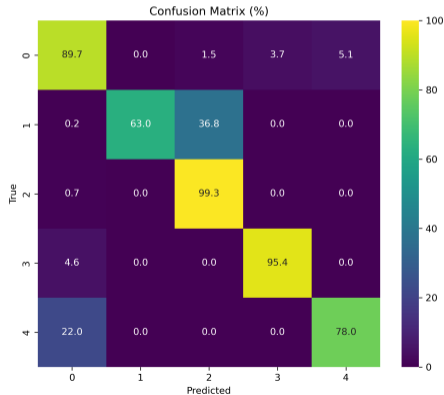


Table of Contents

7 Conclusões

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ **Conclusões**
- ▶ Trabalhos Futuros
- ▶ Referências Bibliográficas

Conclusões

7 Conclusões

- Problema de alta dimensionalidade, difícil convergência.
- Early stopping tradicional fazia os modelos de RNN pararem de treinar muito cedo.
- Mesmo modificando o early stopping, a validação não converge. Os modelos não estão aprendendo padrões generalizáveis.
- Neste caso em específico, a escolha com melhores resultados é a MLP com early stopping tradicional. Porém, talvez outras configurações para a CNN e as RNNs (como com o uso de dropout) pudessem mudar esse quadro.

Table of Contents

8 Trabalhos Futuros

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ **Trabalhos Futuros**
- ▶ Referências Bibliográficas

Trabalhos Futuros

8 Trabalhos Futuros

- Testar: uso de dropout, diminuir learning rate, outros tamanhos de redes.
- Fazer validação cruzada
- Escolher menos sensores
- Múltiplos modelos para diferentes classes
- Testar arquiteturas híbridas (CNN+LSTM)
- Implementar CNN e as RNNs na versão mais expansível do toolkit
- Uso de Foundation Models

Table of Contents

9 Referências Bibliográficas

- ▶ Introdução
- ▶ Problema a ser abordado
- ▶ Revisão Bibliográfica
- ▶ Base de Dados
- ▶ Método Proposto
- ▶ Resultados Obtidos
- ▶ Conclusões
- ▶ Trabalhos Futuros
- ▶ **Referências Bibliográficas**

Referências Bibliográficas

9 Referências Bibliográficas

- [1] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. Zhao, “Time series classification using multi-channels deep convolutional neural networks,” in *International Conference on Web-Age Information Management*, pp. 298–310, Springer, 2014.
- [2] Z. Wang, W. Yan, and T. Oates, “Time series classification from scratch with deep neural networks: A strong baseline,” *2017 IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585, 2017.
- [3] Z. Lipton, “A critical review of recurrent neural networks for sequence learning,” *arXiv preprint arXiv:1506.00019*, 2015.

Referências Bibliográficas

9 Referências Bibliográficas

- [4] K. Cho, B. v. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [5] F. J. Ordóñez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” in *Proceedings of the 2016 ACM International Symposium on Wearable Computers*, pp. 213–220, ACM, 2016.
- [6] N. Autor and N. Autor2, “3w dataset v2: A multimodal benchmark dataset for wearable, web and world signals,” *Technical Report / arXiv*, 2025.
Disponível no repositório oficial do 3W Toolkit.

Comparação de Arquiteturas de Deep Learning na detecção de anomalias em sensores de poços de petróleo

Obrigado pela Atenção!

Alguma Pergunta?

Bruno Coelho Martins

bruno.martins@smt.ufrj.br