Avaliação Comparativa de Estratégias de Deep Learning para Detecção Multiclasse de Malware em Memória

CPE 727 - Aprendizado Profundo

Brenno Rodrigues de Carvalho da Silva (brennorcs@poli.ufrj.br, brenno@lps.ufrj.br)

10 de dezembro de 2025

Sumário

1 Introdução

Motivação e Problema

Migração da Classificação Trivial (Binária) para a Multiclasse

O trabalho aborda a detecção de *malware* em memória (Dump de Processo), focando na superação do limite da classificação binária.

Problema Principal: O *dataset* CIC-MalMem-2022 para classificação binária é **trivial** (AUC ≈ 1.0).
Foco: do trabalho Migrar para a tarefa **Multiclasse**, exigindo modelos robustos e estratégias de inicialização eficazes.

Dataset

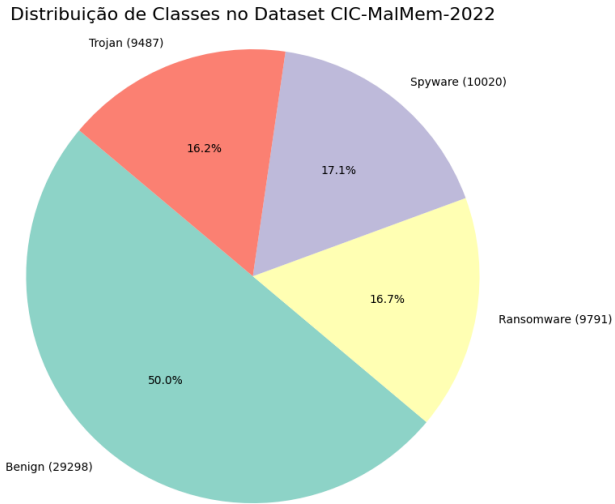
1 Introdução

O *dataset* utilizado é o *CIC-MalMem-2022* (Dump de Memória de Processo) ”O dataset foi criado para representar o mais próximo possível de uma situação do mundo real usando malwares predominantes no mundo real”[1]. O dataset é composto por 58,6k amostras e quatro classes:

1. Trojan;
2. Spyware;
3. Ransomware;
4. Benigno;

Dataset

1 Introdução



Sumário

2 Metodologia

Pré-processamento

2 Metodologia

- **Dados Tabulares**: 55 *features* extraídas de processos.
- **Feature Selection**: Aplicação do **ANOVA F-test** para reduzir as *features* para **16**, aumentando a eficiência dos modelos tabulares [2]
- **Escalonamento**: Uso de **MinMaxScaler** aplicado de forma segura (*anti-leakage*) dentro do *K-Fold*.
- Para utilizar a **CNN** (Rede Neural Convolucional) nos dados tabulares, as 55 *features* originais foram transformadas em uma matriz (imagem) [3].
- **Técnica de Mapeamento:** As 55 *features* são mapeadas em uma matriz $\mathbf{8 \times 8}$ (totalizando 64 ”pixels”), onde as 9 células não utilizadas foram preenchidas com zeros (padding).

Tabela: Estratégias para DNN (Baseline)

Características	DNN
Tipo de Dados	Tabular (16 Features)
Estrutura do Encoder	Linear (Fully Connected)
Camadas Ocultas (Neurônios/Filtros)	256 → 128 → 64
Funções de Ativação	ReLU + Softmax (Saída)
Estratégia de Inicialização	Supervisionado do Zero (Random Initialization)
Peculiaridade Metodológica	Requer ANOVA para Feature Selection (16 features).

Arquiteturas e Estratégias em Teste

2 Metodologia

Tabela: Estratégias para CNN

Características	CNN
Tipo de Dados	Imagem (Matriz 8×8)
Estrutura do Encoder	Convolucional (Conv2D)
Camadas Ocultas (Neurônios/Filtros)	2x Conv Layers
Funções de Ativação	ReLU + Softmax (Saída)
Estratégia de Inicialização	Supervisionado do Zero (Random Initialization)
Peculiaridade Metodológica	Lida com todas as 55 features (Visão).

Arquiteturas e Estratégias em Teste

2 Metodologia

Tabela: Estratégias para SAE (Pré-treino Reconstrução)

Características	SAE (Pré-treino Reconstrução)
Tipo de Dados	Tabular (16 Features)
Estrutura do Encoder	Linear (Fully Connected)
Camadas Ocultas (Neurônios/Filtros)	256 → 128 → 64
Funções de Ativação	ReLU + Sigmoid/ReLU
Estratégia de Inicialização	Não Supervisionada (Reconstrução MSE)
Peculiaridade Metodológica	Requer MinMaxScaler e treinamento em duas fases
Loss Usada no Pré-treino	MSE Loss (Erro de Reconstrução)

Arquiteturas e Estratégias em Teste

2 Metodologia

Tabela: Estratégias para DBN (Pré-treino Probabilístico)

Características	DBN (Pré-treino Probabilístico)
Tipo de Dados	Tabular (16 Features)
Estrutura do Encoder	RBM (Máquina de Boltzmann Restrita)
Camadas Ocultas (Neurônios/Filtros)	256 → 128 → 64
Funções de Ativação	Sigmoid
Estratégia de Inicialização	Não Supervisionada (Contrastive Divergence - CD)
Peculiaridade Metodológica	O pré-treino falhou, estagnando em BCE Loss 0.69
Loss Usada no Pré-treino	MSE/BCE Loss (Erro de Reconstrução)

Metodologia de Treinamento

Protocolo de Validação Stratified K-Fold

- **Hold-Out Estratificado**: Conjunto de teste final (20%) isolado.
- **Validação Cruzada (K=10)**: Uso de *Stratified K-Fold* nos dados de desenvolvimento para obter métricas de validação robustas.
- **Early Stopping (Patience=25)**: Aplicado em cada *fold* para evitar *overfitting* e capturar o melhor ponto de generalização do modelo.
- 500 épocas de treinamento.

Seleção do Melhor Modelo (”Modelo Campeão”)

- Após o *Stratified K-Fold* o modelo salvo foi avaliado no hold-out;
- O melhor dos 10 modelos treinados foi selecionado como o ”campeão”;
- O critério de seleção foi a maior acurácia.

Fase de Pré-Treinamento

Modelos SAE e DBN

- **50 épocas**
- **SAE**: Treinado **end-to-end** para reconstruir a entrada ($X \rightarrow \hat{X}$) com **MSE Loss**. Os pesos do Encoder são transferidos para a DNN.
- **DBN**: Treinado em fases (*greedy layer-wise*) usando o algoritmo **Contrastive Divergence (CD)** e **BCE Loss**.
- **Falha do DBN**: O treinamento falhou, com a *Loss* estagnada em ≈ 0.693 ($\ln(2)$).

Fase de Pré-Treinamento - SAE e DBN

2 Metodologia

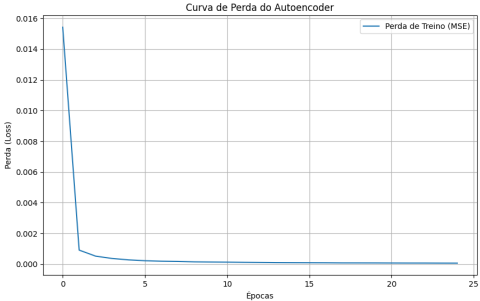


Figura: SAE MSE Loss

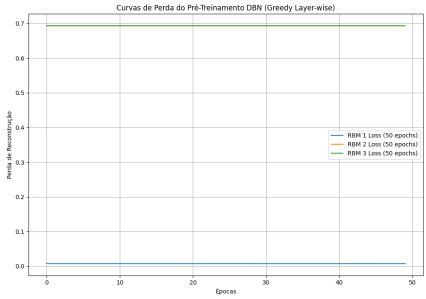


Figura: DBN BCE Loss

Curvas de Loss para Múltiplas Arquiteturas - I

Análise Comparativa do Treinamento Supervisionado (K-Fold)

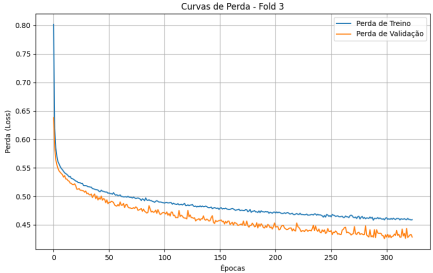


Figura: A: DNN (Baseline)

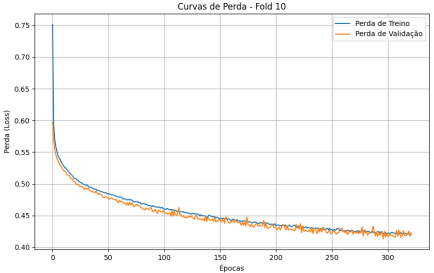


Figura: CNN

Curvas de Loss para Múltiplas Arquiteturas - II

Análise Comparativa do Treinamento Fine-Tuning (K-Fold)

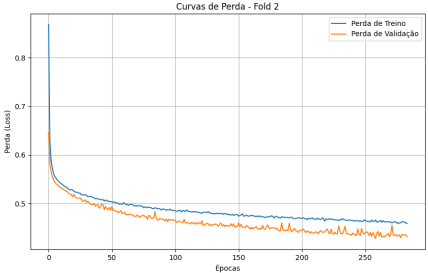


Figura: SAE (Fine-Tuning)

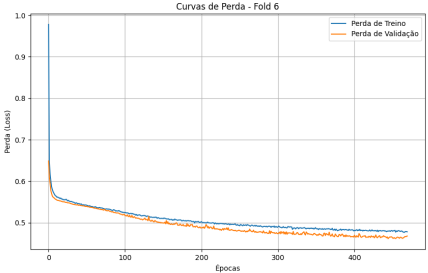


Figura: DBN (Fine-Tuning)

Otimização do Limiar de Decisão

Regra de Decisão Multiclasse

- A classificação primária utiliza a regra **Argmax** (seleciona a classe com a maior probabilidade, regra padrão do Softmax).
- Para cálculo de métricas de classificação (*Accuracy*, *F1-Score*) em cada *fold*:
  - **Binário (Sens/Espec)**: O limiar ótimo é calculado usando a métrica $\mathbf{sp} = \sqrt{\mathbf{G} \cdot \mathbf{A}}$ (*Youden's J Modified*).
  - **Multiclasse (F1/Acc)**: O limiar é implicitamente 0.5, pois a decisão é puramente por argmax.

Figuras de mérito

2 Metodologia

A comparação final foi baseada no desempenho do modelo no conjunto **Hold-Out** (Teste Final), com foco nas classes minoritárias (Malware).

Figuras de Mérito:

- **AUC-OVR** (*One-vs-Rest*): Métrica de ranking universal.
- **F1-Score Macro**: **Métrica de Alerta Crítica.** Essencial, pois dá peso igual a cada classe.
- **Acurácia** e **F1-Score Weighted**.
- **Custo Computacional** (Tempo e Pico de Memória).

Sumário

3 Resultados

Comparação de performance

3 Resultados

Tabela: Comparação de performance dos modelos no conjunto de Hold-out

Modelo	Figuras de mérito		
	AUC-OVR $\pm$ desvio padrão	Acurácia $\pm$ desvio padrão	$F1_{weighted}$
DeepNN	0.957 ± 0.004	0.795 ± 0.009	0.795 ± 0.009
CNN	0.959 ± 0.002	0.803 ± 0.007	0.802 ± 0.007
SAE - DeepNN	0.966 ± 0.003	0.819 ± 0.008	0.819 ± 0.008
DBN - DeepNN	0.950 ± 0.003	0.778 ± 0.006	0.777 ± 0.006

Comparação de performance

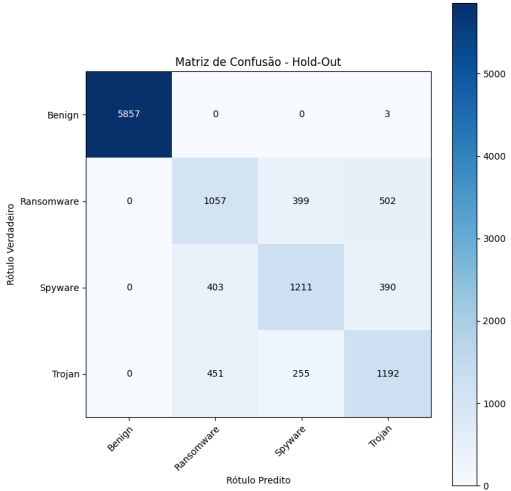
3 Resultados

Tabela: Comparação de performance dos modelos no conjunto de Hold-out

Modelo	Figuras de mérito F1 Score Macro
DeepNN	0.69 ± 0.01
CNN	0.70 ± 0.01
SAE - DeepNN	0.73 ± 0.01
DBN - DeepNN	0.666 ± 0.009

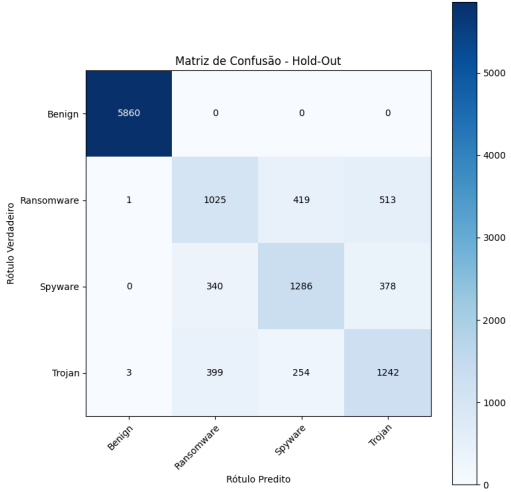
Confusion Matrix - DeepNN

3 Resultados



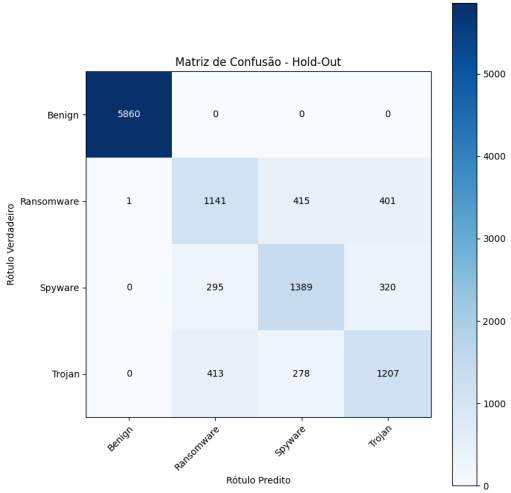
Confusion Matrix - CNN

3 Resultados



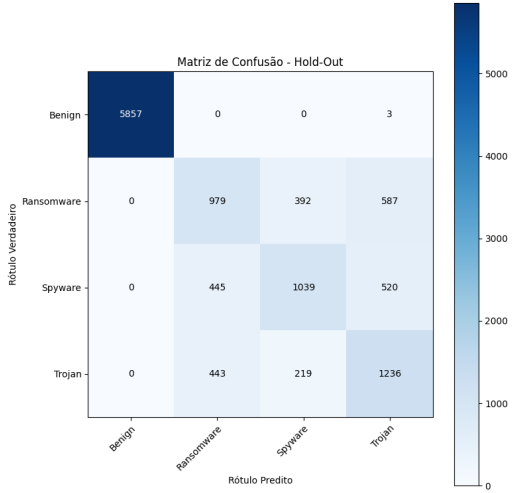
Confusion Matrix - SAE

3 Resultados



Confusion Matrix - DBN

3 Resultados



Sumário

4 Conclusão

Conclusão e Principais Achados

4 Conclusão

- **Validação da Metodologia:** O protocolo *Stratified K-Fold* com mitigação de *leakage* e filtro ANOVA se mostrou eficaz.
- **Falha Crítica do DBN:** Falha total durante o Pré-treinamento.
- **Sucesso do Pré-Treinamento:** O SAE superou a DNN *baseline*, comprovando a eficácia da inicialização por reconstrução.
- **Vantagem da CNN:** A demonstrou bom desempenho sem a necessidade de seleção manual de *features* (ANOVA).

Sumário

5 Referências Bibliográficas

Referências Bibliográficas

[1] Tristan Carrier,, “Detecting Obfuscated Malware using Memory Feature Engineering.” `https://www.unb.ca/cic/datasets/malmem-2022.html`, 2022. Acessado em: 09 de dezembro de 2025.

[2] H. Mourad, M. Mohammed, W. Ferhi, M. Djillali, B. Al Baraa, and H. M. Hicham, “Obfuscated malware detection using deep neural network with anova feature selection on cic-malmem-2022 dataset,” *Journal Scientific and Technical Of Information Technologies, Mechanics and Optics*, vol. 157, no. 5, p. 849, 2024.

[3] Y. Zhu, T. Brettin, F. Xia, A. Partin, M. Shukla, H. Yoo, Y. A. Evrard, J. H. Doroshow, and R. L. Stevens, “Converting tabular data into images for deep learning with convolutional neural networks,” *Scientific reports*, vol. 11, no. 1, p. 11325, 2021.

Avaliação Comparativa de Estratégias de Deep Learning para Detecção Multiclasse de Malware em Memória

Obrigado pela Atenção!
Alguma Pergunta?

Brenno Rodrigues

brennorcs@poli.ufrj.br, brenno@lps.ufrj.br