UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Departamento de Sistemas de Computação

RELATÓRIO DE PROJETO PUB (vertente pesquisa)

Estudo de compressores com perda de informações usados no aprendizado não supervisionado de máquina aplicado ao teste de software

---

Orientado
Hugo Hiroyuki Nakamura

Orientadora
Profa. Dra. Simone do Rocio Senger de Souza

São Carlos - São Paulo
Setembro de 2025

Sumário

1 Introdução

Este relatório apresenta a redação final sobre o desenvolvimento do projeto de iniciação científica, pelo Programa Unificado de Bolsas (PUB). O projeto teve duração de 12 meses, com início em 01 de setembro de 2024 e fim em 31 de agosto de 2025. Ele foi desenvolvido paralelamente ao trabalho de *Estudo de compressores sem perda de informações usados no aprendizado não supervisionado de máquina aplicado ao teste de software*, do aluno de graduação Isaac Santos Soares, e, por isso, algumas etapas do desenvolvimento foram realizadas em conjunto. Além da orientação da Prof. Dra. Simone do Rocio Senger de Souza, o projeto também contou com as co-orientações de Caio Guimarães Herrera, aluno de mestrado, e do Prof. Dr. Paulo Sérgio Lopes de Souza, docente do ICMC.

A proposta do uso de compressores com perdas de informação se baseia no aprendizado não supervisionado da DAMICORE [1]. Ela é uma metodologia de agrupamento, independente do tipo de dados de entrada, que aplica o aprendizado não supervisionado a partir da geração de uma matriz de distâncias, formada pelo cálculo da *Normalized Compression Distance* (NCD).

A NCD é uma métrica de similaridade universal baseada em compressão de dados de dois arquivos. O seu funcionamento consiste em obter separadamente o tamanho da compressão de ambos os arquivos e também da concatenação deles, de modo que se possa calcular:

$$NCD_z(x, y) = \frac{Z(xy) - \min\{Z(x), Z(y)\}}{\max\{Z(x), Z(y)\}} \tag{1}$$

onde x e y representam os arquivos de entrada, xy a concatenação de ambos e $Z(\alpha)$ a função que retorna o tamanho da compressão de um arquivo α. O valor resultante do cálculo da NCD está entre 0 e 1, tal que aquele mais próximo de 0 indica maior semelhança entre os arquivos x e y. Dado um conjunto de arquivos de entrada $\vec{x}$ e $\vec{y}$, o resultado da NCD entre cada par (x,y) forma uma matriz de distâncias.

A TRICORDER é uma metodologia de detecção de anomalias de software pela análise de perfis de consumo de recursos computacionais [2]. Os perfis, obtidos a partir da monitoração de um software em execução, são analisados pela DAMICORE, que os separa em diferentes grupos. A detecção de uma anomalia ocorre quando um grupo formado exclusivamente por perfis de execução anômalos é gerado. Portanto, no contexto da TRICORDER e da DAMICORE, a qualidade da detecção de defeitos depende principalmente do compressor utilizado.

Os perfis de consumo de recursos, ou perfis de execução, são arquivos que descrevem o uso de memória em *Bytes*, percentual de CPU e bytes usados em operações de E/S. Neste trabalho, a aplicação monitorada foi a *Decision Tree* [3], uma biblioteca que executa um algoritmo de aprendizado de máquina para diferentes conjuntos de dados. Dois fatores são importantes para a geração dos perfis de execução: as cargas de trabalho utilizadas na

aplicação e o defeito de software injetado nesta.

As cargas de trabalho se referem aos arquivos de diferentes tamanhos utilizados como entrada no software. Um volume maior de dados pode acentuar as anomalias do perfil de execução. Para a *Decision Tree*, as cargas de trabalho utilizadas são provenientes de conjuntos de dados públicos amplamente utilizados em experimentos de aprendizado de máquina, principalmente do *UCI Machine Learning Repository* [3]. A Tabela 1 apresenta as cargas de trabalho e seus respectivos tamanhos.

Carga de trabalho	**Tamanho**
IRIS-HARBER	Muito pequeno
WINE-HEART	Pequeno
BANK-PIMA	Médio-pequeno
IONO-SOLAR	Médio-grande
CANCER-PHON	Grande
OIL-MAMMO	Muito grande

Tabela 1: cargas de trabalho utilizadas na *Decision Tree* e seu respectivo tamanho.

O defeito injetado se refere a uma variação da *Decision Tree*, em que um defeito específico é propositalmente inserido, a fim de forçar uma execução anômala para os experimentos. Os defeitos impactam a aplicação em diferentes níveis, podendo gerar anomalias mais ou menos evidentes. A Tabela 2 apresenta os defeitos inseridos e o seu impacto na aplicação [4].

Tipo de defeito	**Descrição**	**Impacto**
LOGIC	Erros de lógica	Baixo
MODEL	Problemas de configuração de aprendizagem de máquina	
MEMORY	Problemas de uso de memória	
API	Falta ou chamada de API no local errado	Médio
TRAIN	Falta de dados de treinamento	
CONC	Problemas de sincronização	Alto
PROCESS	Problema na inicialização de objetos ou uso de métodos	

Tabela 2: Tipos de defeito inseridos na aplicação monitorada *Decision Tree*.

Os testes foram realizados em um ambiente virtual controlado, utilizando uma máquina virtual com o sistema operacional Ubuntu 22.04.5 LTS, executando via WSL2 (Windows Subsystem for Linux 2). A máquina virtual foi hospedada em um notebook Lenovo Ideapad 3i, com processador de oito núcleos Intel Core i7-10510U @ 1.80GHz, 8 GB de RAM e sistema Windows 11 (24H2).

Nesta primeira seção foi apresentada a contextualização do projeto. Na seção 2 são apresentados os objetivos que este projeto de iniciação científica busca atingir. Na seção 3 é apresentada a metodologia prevista para a execução dos experimentos. Na seção 4

são apresentadas a execução dos experimentos e os resultados obtidos. Na seção 5 são apresentadas as publicações de artigos baseados nos experimentos realizados. Por fim, a seção 6 traz as conclusões finais da iniciação científica.

2 Objetivos do projeto proposto

Tendo como base a contextualização e os resultados já obtidos, este projeto tem como objetivo central analisar o uso de algoritmos de compressão com perda de informação na geração da matriz de distância do DAMICORE, especialmente quando aplicados no contexto da ferramenta TRICORDER. A investigação busca avaliar o equilíbrio entre a eficácia da compressão e o custo computacional necessário para sua obtenção.

Nesse sentido, é considerada a otimização dos compressores em diferentes aspectos, como eficiência da compressão, tempo de execução e consumo de recursos computacionais (processador, memória e armazenamento).

Além do objetivo principal, o projeto também envolve o(a) aluno(a) de graduação em atividades ligadas ao estudo de computação de alto desempenho aplicada ao aprendizado de máquina não supervisionado, bem como na análise de desafios associados ao teste de software. Essa participação contribui para o aprofundamento dos conhecimentos do estudante em áreas-chave da engenharia de software, inteligência artificial e computação de alto desempenho, tanto por meio de estudos teóricos quanto pelo desenvolvimento prático de soluções em um projeto real de geração de conhecimento.

3 Metodologia prevista

A execução deste projeto contemplou um conjunto de etapas previamente planejadas no projeto inicial, descritas a seguir:

1. **Estudo dos compressores com perda de informações:** nesta fase foi previsto um exame detalhado dos compressores quando aplicados no contexto do *Damicore* e da *Tricorder*.

2. **Revisão da literatura:** esta etapa previa um levantamento bibliográfico de conceitos diretamente relacionados ao projeto, com ênfase na otimização de compressores de arquivos voltados ao aprendizado não supervisionado. Ainda foram explorados, de modo introdutório, a métrica NCD e as demais etapas vinculadas à *DAMICORE*, bem como sua aplicação na *TRICORDER*.

3. **Proposição de possíveis otimizações nos compressores:** a partir dos resultados obtidos nas etapas anteriores, foram previstas sugestões de melhorias potenciais para os compressores utilizados na *DAMICORE*.

4. **Implementação das otimizações propostas:** nesta etapa foi prevista a aplicação das melhorias sugeridas na etapa anterior, diretamente nos algoritmos de compressão de dados.

5. **Planejamento dos experimentos:** nesta fase, os experimentos com compressores foram planejados para identificar seus pontos positivos e suas limitações, considerando o cálculo da métrica NCD no contexto da *TRICORDER*.

6. **Execução dos experimentos:** nesta etapa os experimentos foram realizados de acordo com o planejamento, aplicando os algoritmos de compressão tanto em sua forma original quanto nas versões otimizadas.

7. **Análise crítica dos resultados:** os dados obtidos nos experimentos foram tabulados e analisados, permitindo verificar o desempenho dos compressores utilizados e avaliar os impactos das otimizações propostas, com foco na aplicação na *DAMICORE*.

8. **Disponibilização dos resultados:** finalmente, os resultados alcançados foram organizados em relatórios e apresentados de forma estruturada, viabilizando sua utilização em pesquisas futuras.

9. **Submissão de artigos e relatórios com os resultados obtidos**.

4 Atividades realizadas e resultados

4.1 Revisão da literatura e otimização dos compressores

Em *Avaliação do uso de agrupamento de dados de desempenho para apoiar o teste de software no domínio de aprendizagem de máquina* [4], Braga realiza breves experimentos com alguns compressores na DAMICORE, o ZLIB, PPMD, BZ2 e GZIP.

O ZLIB faz uso do algoritmo DEFLATE [5], que combina o método LZ77 [6] — responsável por substituir sequências repetitivas por referências a ocorrências anteriores — reduzindo a redundância, com a codificação de Huffman [7], que atribui códigos curtos aos símbolos mais frequentes. O resultado da compressão é dividido em blocos, que podem ser dinâmicos, fixos ou armazenados sem compressão, acompanhados de metadados que permitem sua correta descompressão.

O GZIP é também um compressor sem perdas baseado no DEFLATE [8], mas adiciona um cabeçalho e um rodapé próprios. Esses elementos extras preservam informações adicionais, como o nome do arquivo original, a data de criação e um CRC32 para verificação de integridade.

O PPMd é uma implementação do algoritmo PPM (*Prediction by Partial Matching*) [9], que realiza compressão estatística a partir da previsão do próximo símbolo, baseada nos últimos n caracteres já observados. Para organizar esses contextos, o PPMd utiliza uma

estrutura em árvore TRIE, onde cada caminho da raiz até um nó representa uma sequência do histórico. Cada nó armazena estatísticas de frequência para os símbolos que podem suceder aquele contexto. As distribuições de probabilidade obtidas nos nós são então compactadas por codificação aritmética. Essa abordagem baseada em modelos probabilísticos é especialmente eficiente em textos, onde há dependências de longo alcance, embora seu uso demande mais memória e poder de processamento, quando comparado a métodos como o LZ77.

O BZ2, por sua vez, aplica um processo de três passos: a Transformada de Burrows–Wheeler (BWT) [10], seguida do Move-to-Front (MTF) e da codificação de Huffman [7]. A BWT reorganiza os dados agrupando símbolos iguais, o MTF converte-os em referências posicionais que favorecem valores pequenos e repetidos (complementados pelo uso de RLE), e finalmente o Huffman reduz o custo de representação dos símbolos frequentes. O resultado é que o BZ2 geralmente alcança compressões melhores que o ZLIB, embora com maior custo em memória e tempo de processamento.

Os primeiros experimentos realizados consistiram em construir a matriz de distâncias com os perfis de monitoramento de um tipo de defeito e controle, pois, dessa forma, era possível entender o comportamento de cada compressor por meio da NCD. Os compressores com maior taxa de compressão apresentaram uma melhor diferenciação de perfis controle de perfis defeito segundo a métrica NCD. Os resultados revelaram que o PPMD tem diferença média de NCD maior que os demais, o que pode ser um índicio de uma capacidade maior de detecção de anomalias dentro da DAMICORE. Entretanto, Braga apontou o ZLIB como o compressor com maior taxa de detecção para seus experimentos, enquanto neste o ZLIB apresentou uma diferença média de NCD menor que o do PPMD.

Visando melhorar a diferenciação de arquivos com a métrica NCD, os parâmetros dos compressores foram modificados. Os tempos de processamento também foram medidos para que se pudesse descobrir uma possível vantagem em tempo. Neste experimento, apenas o ZLIB e o PPMD foram escolhidos, pois os estudos de Braga os indicam como os dois melhores compressores, além de diminuir a quantidade de testes. O ZLIB apresenta os seguinte parâmetros modificados:

- **Nível de compressão**: indica o esforço que deve ser feito na compressão;
- **Tamanho de janela deslizante**: define o volume de consultas ao histórico de padrões identificados;
- **Quantidade de memória alocada**: refere-se à alocação de recursos para o estado interno da compressão.

Os parâmetros do PPMD são:

- **Ordem**: está diretamente associado à taxa de compressão;
- **Quantidade de memória alocada**: limita o espaço total que sua árvore digital pode ocupar.

Os resultados indicaram que o PPMd, especialmente com ordens de compressão mais elevadas, alcança uma distinção significativamente maior na métrica de NCD. Contudo, essa

maior sensibilidade é obtida ao custo de um tempo de processamento consideravelmente superior, chegando a ser uma ordem de magnitude maior que a do ZLIB nas configurações testadas. Observa-se, também, que o ZLIB se beneficia de uma configuração com nível de compressão, memória e janela deslizante em seus valores mais altos para maximizar a diferença de NCD. Para o PPMd, a configuração ótima é relativa: uma ordem 6 é suficiente para gerar uma diferença de NCD comparável ao valor de máxima compressão, se acompanhada de um valor de memória suficiente. Isto acontece porque o cálculo NCD, apresentado na Equação 1, é influenciada pelo poder de compressão empregado nos arquivos, em que uma compressão maior, gera resultados melhores.

Ao fim desta etapa, entendeu-se que o ZLIB chega a custar 10 vezes menos tempo que o PPMd, mas sua diferenciação de NCD está na ordem de magnitude de centésimos. Entretanto, foi preciso observar se a diferença de NCD é suficiente para que os agrupamentos ocorram da devida forma. A partir dos testes realizados, não foi possível observar uma vantagem definitiva de um compressor em relação ao outro.

Ao utilizar ambos os compressores em uma ferramenta que implementa a DAMICORE, a *DamicorePy* [11], foi possível observar o impacto das mudanças de parâmetro na detecção de anomalias. Com os resultados dos testes realizados dentro da DAMICORE, observou-se, que a proximidade da diferenciação da NCD por parte do ZLIB é suficiente para o agrupamento, apresentando resultados melhores que o próprio PPMd.

4.2 Levantamento e teste de compressores com perdas

Os compressores de imagem pertencem a uma classe de compressores amplamente distribuída no meio digital, principalmente porque o armazenamento e envio de imagens de alta qualidade em tamanho original é muito custoso em tempo e processamento. Como estes compressores conseguem comprimir e descomprimir grandes imagens em arquivos pequenos sem perda significativa da qualidade, espera-se que apresentem uma alta taxa de compressão e, consequentemente, uma acurácia de detecção comparável aos compressores já estudados. Os compressores utilizados são, especificamente, os compressores de imagem com perdas, visto que alguns tem variações sem perdas. Eles utilizam a incapacidade humana de pecepção de certas variações de cores para reduzí-las a uma única cor.

Para utilizar compressores de imagem, é necessário inicialmente converter os dados de entrada, já que os perfis de execução são binários e, portanto, incompatíveis com esse tipo de compressor. A solução adotada consiste em transformar esses arquivos binários em imagens não comprimidas. Para isso, escolheu-se o formato BITMAP, que é um tipo de imagem sem compressão no qual cada três bytes representa um componentes de cor vermelho, verde e azul (RGB).

Os compressores avaliados estão listados na Tabela 3, mantendo-se os compressores de texto a fim de obter uma base de comparação.

Compressor	Com perdas	Sem perdas
ZLIB	-	X
PPMd	-	X
BZ2	-	X
GZIP	-	X
WEBP	X	-
JP2	X	-
HEVC	X	-

Tabela 3: compressores utilizados na *DamicorePy*.

O experimento carregou todos os perfis de um tipo de anomalia na *DamicorePy* por vez. Assim, para cada carga de trabalho, um valor de acurácia pode ser obtido a partir da quantidade de detecções corretas. A média de acertos por cada carga gera a acurácia total do compressor. A Tabela 4 apresenta os resultados do experimento.

COMPRESSOR	BANK-PIMA	CANCER-PHON	IONO-SOLAR	IRIS-HABER	OIL-MAMMO	WINE-HEART	Total	(%)
GZIP	6	7	7	7	6	7	40	95,2
ZLIB	7	6	6	7	7	7	40	95,2
BZ2	7	5	7	6	6	6	37	88,1
HEVC	6	6	6	6	5	7	36	85,7
PPMD	7	5	6	6	6	6	36	85,7
JP2	6	5	7	5	7	4	34	81,0
WEBP	6	2	4	3	4	6	25	59,5

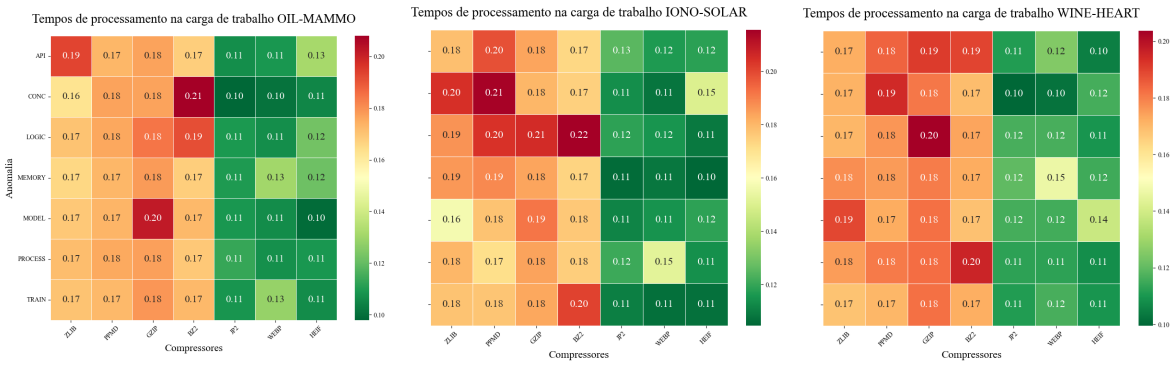
Tabela 4: quantidade de detecções corretas e acurácia para cada compressor em cada carga de trabalho.

Também foi realizado um teste de geração de falsos positivos, no qual a *DamicorePy* foi carregada com perfis do tipo ANOTHER-CONTROL, representando uma versão da aplicação sem defeitos inseridos. Espera-se que os compressores não o detectem como uma anomalia. A Tabela 5 apresenta as detecções incorretas de cada compressor em cada carga.

COMPRESSOR	BANK-PIMA	CANCER-PHON	IONO-SOLAR	IRIS-HABER	OIL-MAMMO	WINE-HEART	Total	(%)
WEBP	1	0	1	1	1	1	5	83,3
HEIF	1	1	1	1	0	0	4	66,7
JP2	1	1	0	0	0	1	3	50.0
ZLIB	0	0	0	0	0	0	0	00,0
GZIP	0	0	0	0	0	0	0	00,0
PPMD	0	0	0	0	0	0	0	00,0
BZ2	0	0	0	0	0	0	0	00,0

Tabela 5: Acertos por compressor sob o único tipo ANOTHER-CONTROL.

Os tempos de processamento da execução da DAMICORE para cada tipo de defeito e compressor, por carga de trabalho, foram medidos. A Figura 1 apresenta esses tempos para as cargas OIL-MAMMO, IONO-SOLAR e WINE-HEART, escolhidas por representar cenários de carga de tamanho muito grande, médio e pequeno, respectivamente.



(a) OIL-MAMMO (b) IONO-SOLAR (c) WINE-HEART

Figura 1: tempos de processamento dos compressores na detecção das anomalias para algumas cargas.

Da Tabela 4, observa-se que o compressor HEVC apresentou a quarta melhor taxa de acerto, com acurácia de 85,7 %, igual a do PPMD. O JP2 e o WEBP foram os dois piores resultados, com 81.0% e 59.9 %, respectivamente, sendo este último um resultado muito baixo em relação aos outros. Da Tabela 5, os três compressores com perdas geraram falsos positivos ao detectar o tipo ANOTHER-CONTROL, que pode indicar que, na tarefa de detecção de anomalias, esses compressores não são confiáveis. Da Figura 1, entretanto, foi possível observar que os compressores com perdas tem tempos de processamento menores que os compressores sem perdas, o que pode indicar uma vantagem de eficiência.

4.3 Testes de eficácia e eficiência

O experimento anterior realizou um teste simples de detecção de defeitos, considerando o carregamento de todos os perfis de um tipo de anomalia para gerar os agrupamentos. Entretanto, o método da TRICORDER utiliza a DAMICORE incrementalmente e deve seguir as seguintes etapas:

1. carregamento dos perfis não anômalos, para gerar os grupos de referência;
2. para um mesmo tipo de anomalia, ocorre a inserção de um perfil;
3. se, após a inserção do perfil, a DAMICORE detectar um grupo exclusivo de perfis anômalos, o processo finaliza e a anomalia foi detectada. Caso contrário, o item 2 ocorre novamente. Se os perfis acaberem sem detecção, entende-se como execuções normais.

O experimento seguinte utilizou a DAMICORE incrementalmente, para se aproximar mais de uma execução padrão da detecção. Dois testes foram realizados nessa etapa: o teste de eficácia e o teste de eficiência.

Para esta etapa, considerou-se a eficácia como a capacidade qualitativa do compressor de identificar corretamente um defeito. A eficácia, portanto, é estimada a partir de medidas

quantitativas, que descrevem os resultados dos agrupamentos. Estas medidas são acurácia, precisão, *recall* e *f1-score*. Para calculá-las, é preciso, primeiramente, entender os quatro componentes da matriz de confusão, apresentada na Tabela 6.

- **Verdadeiro Positivo (VP)**: quando o agrupamento de arquivos de um tipo defeituoso junto a arquivos controle **gera pelo menos um grupo** somente com arquivos do tipo defeituoso;
- **Falso Negativo (FN)**: quando o agrupamento de arquivos de um tipo defeituoso junto a arquivos controle **não gera nenhum grupo exclusivo**;
- **Falso Positivo (FP)**: quando o agrupamento de arquivos de um tipo não defeituoso junto a arquivos controle **gera pelo menos um grupo** somente com arquivos do tipo não defeituoso;
- **Verdadeiro Negativo (VN)**: quando o agrupamento de arquivos de um tipo não defeituoso junto a arquivos controle **não gera nenhum grupo exclusivo**

		Defeito detectado	
		Sim	**Não**
Arquivos defeituosos	**Sim**	Verdadeiro Positivo (VP)	Falso Negativo (FN)
	Não	Falso Positivo (FP)	Verdadeiro Negativo (VN)

Tabela 6: matriz de confusão da detecção da TRICORDER.

Dessa forma, pode-se definir as medidas quantitativas de eficácia, cujas equações estão apresentadas na Tabela 7.

- **Acurácia**: mede a proporção de agrupamentos corretamente identificados em relação ao total de testes.
- **Precisão**: mede a proporção de tipos defeituosos detectados que são, de fato, defeituosos. É útil quando falsos positivos são um problema.
- **Recall**: mede a quantidade de acertos na identificação do total de defeitos. Um recall alto indica que a maioria dos arquivos do tipo defeituoso é detectada. É útil quando falsos negativos são um problema.
- **F1-Score**: é a média harmônica entre precisão e recall. Esta métrica equilibra as duas anteriores e é útil quando há desbalanceamento entre VP, FP e FN. Um resultado alto significa que o compressor está detectando bem os agrupamentos. Um valor baixo indica a ocorrência de muitos falsos positivos ou falsos negativos.

Métrica	**Fórmula**
Acurácia	$\frac{VP+VN}{VP+VN+FP+FN}$
Precisão	$\frac{VP}{VP+FP}$
Recall	$\frac{VP}{VP+FN}$
F1-score	$2 \cdot \frac{\text{Precisão} \cdot \text{Recall}}{\text{Precisão}+\text{Recall}}$

Tabela 7: Fórmulas das métricas de avaliação

Em relação a eficiência, ela é considerada como a capacidade quantitativa do compressor de gerar resultados no menor tempo de execução possível. A eficiência é medida pelo tempo de resposta de toda a execução da DAMICORE, desconsiderando neste momento a validade ou a qualidade dos resultados gerados. A partir de repetidas medições de tempo, é possível obter dados estatísticos analisados sobre algumas métricas. A que foi utilizada para os tempos foi a mediana, o valor central da distribuição de tempos. Também é possível avaliar a eficiência pela quantidade de iterações necessárias em que um compressor, quando aplicado na DAMICORE, detecta uma anomalia.

A Figura 2 apresenta o resultado das execuções da DAMICORE com os compressores com perdas e sem perdas. A matriz de iteração apresenta a iteração em que anomalia foi detectada, com os espaços cinzas representando a não detecção. A matriz de mediana apresenta a mediana do tempo de resposta em que o sistema realizou o processo de identificação. Observa-se que os compressores com perda continuaram detectando o tipo ANOTHER-CONTROL e gerando falsos positivos. Em relação ao tempo de resposta, é possível ver que os compressores WEBP, JP2 e HEVC são mais lentos que os demais compressores. Mas em compensação, eles detectam as anomalias em menos iterações, na maioria dos casos, em especial o HEVC.

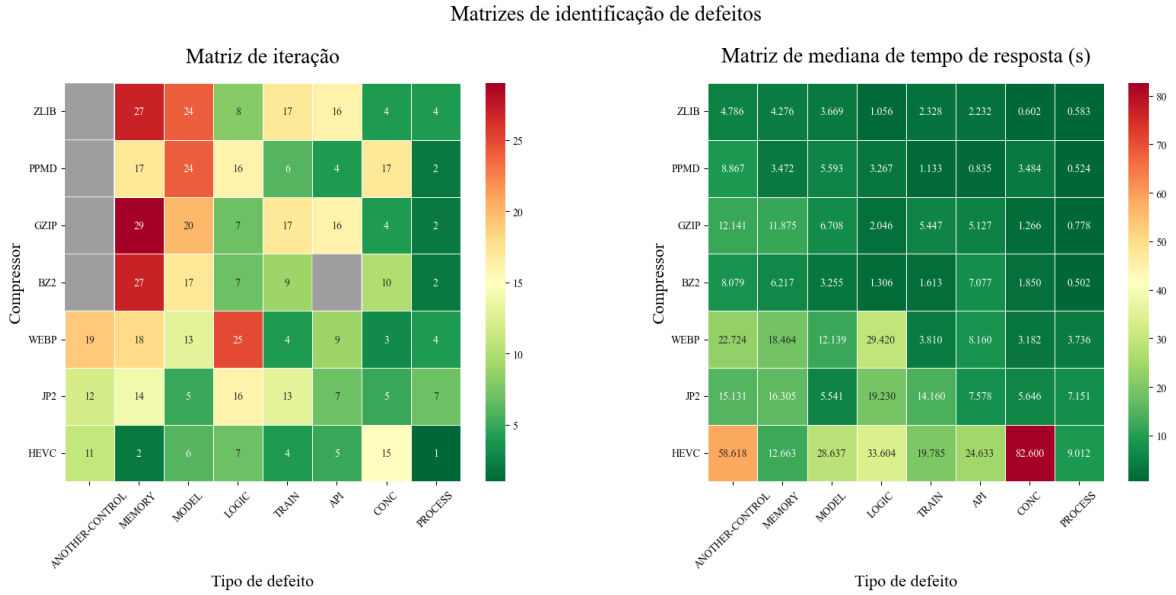


Figura 2: matrizes de iteração e mediana do tempo de resposta.

A análise da eficácia foi feita a partir dos resultados obtidos. A Figura 3 apresenta os gráficos de acurácia, precisão, recall e f1-score da carga grande CANCER-PHON. Nesse caso, como tínhamos apenas um grupo controle alternativo, foi necessário balanceá-lo com os tipos de defeito, ou seja, devido à existência de 7 tipos de defeito, a detecção do tipo ANOTHER-CONTROL foi considerada como a detecção de 7 tipos não anômalos.

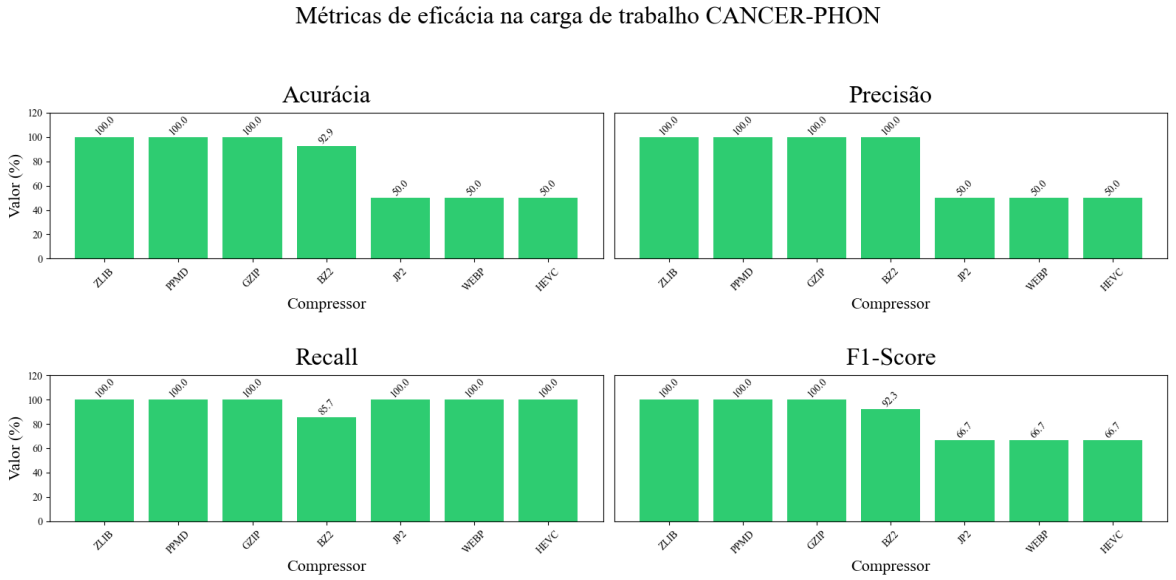


Figura 3: gráfico de barras das métricas de eficácia com a carga de trabalho OIL-MAMMO.

Ao fim do teste de eficácia da Figura 3, observou-se que os compressores de imagem apresentaram precisão de 50%, que é baixa em relação aos 100% dos outros compressores. O mesmo comportamento de geração de falsos positivos é observado na Figura 2.

Paralelamente, eles apresentaram o máximo recall. Estes dados mostram que eles são compressores competentes na tarefa de detecção de defeitos, mas não aparentaram ser confiáveis ao identificar perfis normais como perfis anômalos.

Da análise de eficiência, a Figura 4 apresenta as distribuições de tempo de resposta, por meio de boxplots, com a carga de trabalho CANCER-PHON e os tipos de defeito PROCESS, API e CONTROL, escolhidos por terem impacto alto, médio e baixo nos perfis.

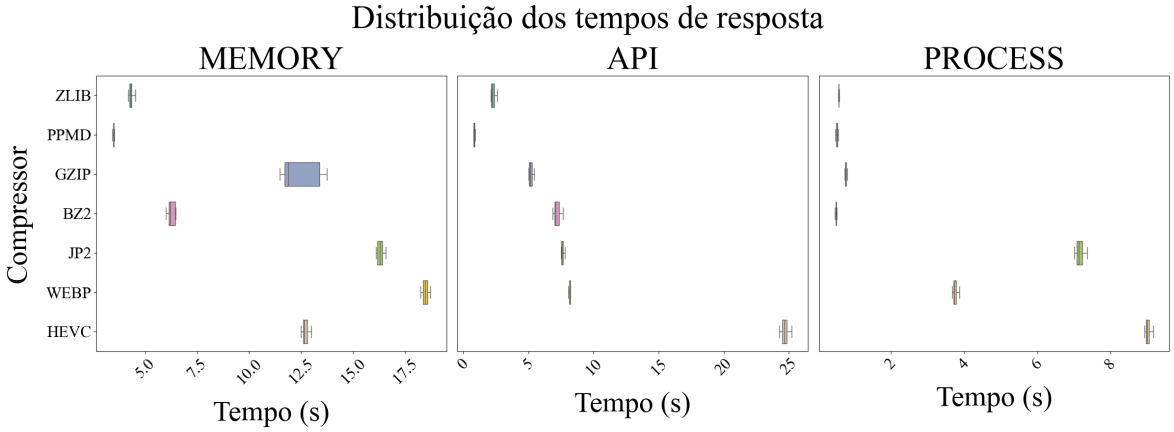


Figura 4: distribuições de tempo de execução para diferentes compressores em 3 tipos de anomalia.

A representação da Figura 4 permite a comparação visual entre os tempos, mas é necessário realizar uma análise estatística de diferenças significativas entre os resultados. Para verificar diferenças significativas, é preciso primeiramente avaliar se os dados estão em uma distribuição normal, com o teste de Shapiro Wilk. A Tabela 8 apresenta os compressores que geraram dados de tempo com distribuição normal, na tarefa de detecção de um defeito. Neste caso, como não há comparação dos dados de tempo destes compressores no mesmo defeito, é possível utilizar o teste não-paramétrico de Wilcoxon para diferenças significativas.

Compressor	Defeito	Valor-P	Dist. Normal
HEIF	MEMORY	0,246	SIM
HEIF	TRAIN	0,630	SIM
HEIF	CONC	0,145	SIM
BZ2	PROCESS	0,057	SIM

Tabela 8: compressores com tempos de execução com distribuição normal pelo teste de Shapiro-Wilk.

A partir da escolha do teste de Wilcoxon, a Tabela 9 apresenta os resultados do teste que retornaram um Valor-P maior que $0,05$, ou seja, que não possuem diferenças significativas.

Comparação	Defeito	Valor-P	Significante
BZ2 x JP2	API	0,177	Não

Tabela 9: Diferenças de tempo entre compressores consideradas não significativas pelo teste de Wilcoxon (p > 0,05).

Observou-se, a partir da Figura 4, que há diferenças visíveis nos tempos de execução da detecção de anomalias pelos compressores. Na Figura 2 foi observado que os compressores com perdas apresentam medianas de tempo de execução maiores que os compressores sem perdas. Como a Tabela 9 põe apenas a comparação BZ2 x JP2 na identificação do defeito API como não significativa, é possível confirmar estatisticamente essa diferença de tempos. Há, ainda, a percepção de que os compressores com perdas detectaram as anomalias em menos iterações que os demais, na maioria dos casos, em especial o HEVC.

Por fim, a Figura 5 apresenta uma relação entre eficácia e eficiência para cada compressor em um mapa, a partir dos resultados da execução com a carga de trabalho CANCER-PHON. Nesse caso, a eficácia foi tomada como a acurácia normalizada, e a eficiência foi tomada como a mediana do tempo de resposta invertida e normalizada.

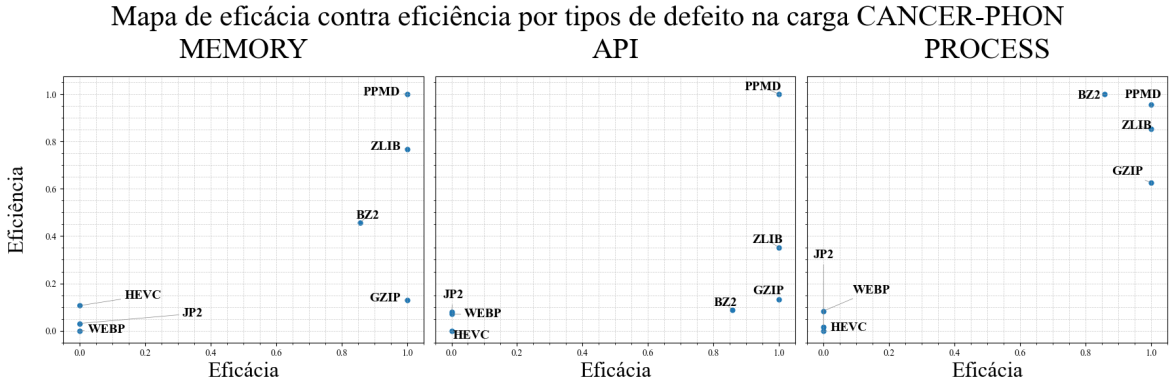


Figura 5: mapa de eficiência por eficácia nos dados da carga CANCER-PHON.

Dessa forma, com a Figura 5, foi possível observar visualmente como os compressores com perdas não apresentam vantagens de eficácia e eficiência, principalmente por estarem mais próximos do canto inferior esquerdo. Os melhores resultados foram do ZLIB e do PPMD, que estiveram muito próximos do canto superior direito, o de máxima eficácia e eficiência.

5 Publicações

Três artigos científicos foram escritos e estão em análise no momento da escrita deste relatório.

Os artigos *Avaliação de Compressores com Perdas de Informações na Detecção de Anomalias de Software com a TRICORDER* [12] e *Analisando Compressores sem Perdas*

de Informações na Detecção de Anomalias de Software com a TRICORDER [13] foram submetidos para o 33° Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP).

O artigo *Efeitos da Compressão no Aprendizado Não Supervisionado para Detecção de Anomalias em Software* [14] foi escrito e submetido para a trilha principal do Simpósio de Sistemas Computacionais de Alto Desempenho (SSCAD), edição de 2025.

Há ainda o planejamento da submissão de um artigo sobre o cálculo da entropia como uma estimativa de compressão sem perdas.

6 Conclusões

Ao fim dos experimentos, considera-se que todos os objetivos foram atingidos, principalmente sobre a análise de eficácia e eficiência dos compressores na DAMICORE, a otimização de compressores e o desenvolvimento do aluno de iniciação científica.

Segundos os experimentos realizados, os compressores com perdas de informação possuem eficácia e eficiência baixos quando aplicados à TRICORDER. Os compressores genéricos, sem perdas, em especial o ZLIB e o PPMD, se apresentaram como os mais eficazes e eficientes, nesse contexto. Entretanto, os compressores com perdas realizaram a detecção das anomalias em iterações menores que os demais, o que pode ser uma vantagem em cenários em que a iteração é custosa. Em relação à otimização dos compressores, descobriu-se que a melhor configuração é a de máxima compressão, visto que a NCD, que é a base da matriz de distâncias, é afetada conforme a maior compressão dos arquivos utilizados.

Os resultados também possibilitaram a escrita e submissão de três artigos até o momento. Isto não somente desenvolveu maturidade na escrita científica, como colaborou com habilidades de pesquisa, análises estatísticas, conhecimento de engenharia de confiabilidade de software, aprendizado não supervisionado etc.

Todos os códigos, imagens e resultados podem ser encontrados no repositório do GitHub, LossyCompressors_IC [15].

Referências

[1] A. Sanches, J. M. P. Cardoso, and A. C. B. Delbem, "Identifying merge-beneficial software kernels for hardware implementation," in *2011 International Conference on Reconfigurable Computing and FPGAs*. San José, Costa Rica: IEEE, 2011.

[2] V. S. Montes, "Detecção de defeitos de software utilizando agrupamento de perfis de desempenho," Dissertação de Mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP, 2021, programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC).

[3] S. Santos, B. Silveira, V. Durelli, R. Durelli, S. Souza, and M. Delamaro, "On using decision tree coverage criteria for testing machine learning models." ACM, 2021.

[4] D. Braga, "Avaliação do uso de agrupamento de dados de desempenho para apoiar o teste de software no domínio de aprendizagem de máquina," Dissertação de Mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, SP, 2021, programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC).

[5] P. Katz, "DEFLATE Compressed Data Format Specification version 1.3," Internet Engineering Task Force (IETF), Request for Comments RFC 1951, May 1996. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1951

[6] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[7] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, September 1952.

[8] P. Deutsch, "GZIP file format specification version 4.3," Internet Engineering Task Force (IETF), Request for Comments RFC 1952, May 1996. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1952

[9] A. Moffat, "Implementing the ppm data compression scheme," *IEEE Transactions on Communications*, vol. 38, no. 11, pp. 1917–1921, 1990.

[10] M. Burrows, D. J. W. D. I. G. I. T. A. L, R. W. Taylor, D. J. Wheeler, and D. Wheeler, "A block-sorting lossless data compression algorithm," 1994. [Online]. Available: https://api.semanticscholar.org/CorpusID:2167441

[11] K. Polo, "Damicore-python3 (fork)," 2024, fork do repositório de Alexandre C. B. Delbem. [Online]. Available: https://gitlab.uspdigital.usp.br/kevin.polo/damicore-python3

[12] H. H. Nakamura, I. S. Soares, C. G. Herrera, P. S. L. de Souza, and S. do Rocio Senger de Souza, “Avaliação de compressores com perdas de informações na detecção de anomalias de software com a tricorder,” in *33ª Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP)*. São Carlos, SP, Brasil: Universidade de São Paulo, 2025, [SUBMETIDO].

[13] I. S. Soares, H. H. Nakamura, C. G. Herrera, S. do Rocio Senger de Souza, and P. S. L. de Souza, “Analisando compressores sem perdas de informações na detecção de anomalias de software com a tricorder,” in *33ª Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP)*. São Carlos, SP: Universidade de São Paulo, 2025, [SUBMETIDO].

[14] H. H. Nakamura, I. S. Soares, C. G. Herrera, S. do Rocio Senger de Souza, and P. S. L. de Souza, “Efeitos da compressão no aprendizado não supervisionado para detecção de anomalias em software,” in *Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD)*. Brasil: Sociedade Brasileira de Computação, 2025, [SUBMETIDO].

[15] H. H. Nakamura, “Lossycompressors_ic,” https://github.com/ikuyorih9/LossyCompressors_IC, 2025, acessado em: 05/03/2024.