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 sem perda de informações usados no aprendizado não supervisionado de máquina aplicado ao teste de software

---

Orientado
Isaac Santos Soares

Orientador
Prof. Dr. Paulo Sergio Lopes de Souza

São Carlos - São Paulo
Setembro de 2025

Sumário

1 Introdução

Este é o relatório do projeto de Iniciação Científica, financiado pelo Programa Unificado de Bolsas de Estudo para Apoio e Formação de Estudante de Graduação (PUB), intitulado "**Estudo de compressores sem perda de informações usados no aprendizado não supervisionado de máquinas aplicado ao teste de software**". Esse projeto de pesquisa foi desenvolvido durante o período de setembro/2024 a agosto/2025

As principais atividades previstas envolveram a pesquisa e estudo dos compressores sem perda, revisão da literatura, projeto e implementação de otimizações dos compressores, proposição de novos compressores e métodos, planejamento e execução de experimentos, análise crítica dos resultados, disponibilização do corpo de conhecimento (dados, gráficos e demais informações) e submissão de artigos e relatórios.

A detecção de falhas em sistemas de software é um desafio significativo para a garantia da qualidade, especialmente devido aos altos custos e à complexidade dos processos de teste. Neste contexto, a metodologia TRICORDER apresenta uma abordagem para identificar anomalias através do monitoramento do consumo de recursos computacionais (como CPU e memória) [1]. A análise dos perfis de consumo coletados é realizada pela DAMICORE [2], uma abordagem de aprendizado de máquina não supervisionado. A DAMICORE agrupa perfis com comportamentos similares e isola os anômalos, utilizando para isso a métrica de Distância de Compressão Normalizada (NCD) [3], [4]. Sendo a NCD uma aproximação prática da teoria da complexidade de Kolmogorov que depende de algoritmos de compressão de dados, a eficácia de todo o processo de detecção está diretamente condicionada à performance e adequação do compressor utilizado.

Os perfis de consumo de recursos provenientes da execução de aplicações e utilizados nos nossos experimentos foram obtidos pela monitoração da aplicação *Decision Tree* [5], uma biblioteca que executa um algoritmo de aprendizado de máquina para diferentes conjuntos de dados. A *Decision Tree* foi escolhida por permitir a variação do seu funcionamento dependendo do *Dataset* utilizado, além do tipo de anomalia que foi injetada em sua execução.

Os diferentes *Datasets* representam diferentes cargas de trabalho classificadas conforme a Tabela 1.

Dataset	Carga de Trabalho
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.

Esses *Datasets* são conjuntos de dados públicos utilizados em experimentos de aprendizado de máquina, provenientes do *UCI Machine Learning Repository* [5].

As anomalias injetadas são classificadas conforme a Tabela 2. Essa classificação foi apresentada na tese de mestrado de Braga. As anomalias injetadas são defeitos inseridos no código-fonte da aplicação de forma proposital para simular um comportamento anômalo. A aplicação original, sem nenhuma anomalia injetada, denomina-se perfil de controle (*CONTROL*) neste projeto.

Tipo de anomalia	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 anomalias injetadas na *Decision Tree*.

Este trabalho foi desenvolvido paralelamente ao trabalho de Iniciação Científica *Estudo de compressores sem perda de informações usados no aprendizado não supervisionado de máquina aplicado ao teste de software* do aluno Hugo Hiroyuki Nakamura. O projeto também contou com as co-orientações de Caio Guimarães Herrera, aluno de mestrado, e da Profa. Dra. Simone do Rocio Senger de Souza do ICMC/USP.

2 Objetivos do Projeto Proposto

O presente investiga e avalia algoritmos de compressão de dados sem perdas aplicados à geração da Matriz de Distância na ferramenta DAMICORE, no contexto da

Tricorder. A análise esteve focada na relação entre a eficiência da compressão e o custo computacional do processo.

A pesquisa abrange a otimização dos compressores estudados, com ênfase em sua performance, tempo de resposta e no consumo de recursos de hardware, como CPU, memória RAM e acessos a disco.

O projeto também teve como meta secundária a imersão do estudante de graduação na Computação de Alto Desempenho aplicada ao aprendizado de máquina não supervisionado e à resolução de desafios em teste de software. A participação do aluno neste projeto proporcionou a expansão dos conhecimentos ao mesmo em Engenharia de Software, Inteligência Artificial e Computação de Alto Desempenho, aliando o estudo teórico com a implementação de soluções em uma pesquisa aplicada.

3 Metodologia Prevista

A metodologia deste projeto foi estruturada em uma sequência lógica de etapas, iniciando-se com um aprofundamento teórico sobre compressores sem perda de informação, com foco particular em sua aplicação nos contextos das metodologias DAMICORE e TRICORDER. Em paralelo, foi conduzida uma ampla revisão da literatura para solidificar os conceitos chave da pesquisa, como a otimização de compressores de arquivos aplicada ao aprendizado não supervisionado, a métrica NCD e as inter-relações entre os resultados da DAMICORE e da TRICORDER. Com base nesse embasamento teórico, a fase seguinte consistiu no projeto de possíveis otimizações para os algoritmos de compressão, que foram posteriormente implementadas computacionalmente. Para avaliar a eficácia das modificações propostas, foi elaborado um plano de experimentos comparativos entre os compressores otimizados e suas versões tradicionais. A execução desses experimentos buscou identificar as características positivas e negativas das otimizações, especialmente no cálculo da métrica NCD na TRICORDER. Os dados coletados foram então tabulados e submetidos a uma análise para extrair conhecimentos que guiassem pesquisas futuras e ser aplicados na própria DAMICORE. Por fim, o projeto encerrou com a disseminação científica, que incluiu a organização e disponibilização pública de todos os dados e artefatos gerados, bem como a submissão de artigos e relatórios técnicos.

4 Atividades Realizadas com Resultados

As etapas iniciais desta pesquisa foram realizadas em colaboração com aluno Hugo Nakamura. Tais etapas consistiram no estudo de compressores sem perda de informação utilizados em trabalhos anteriores, em testes de otimização dos algoritmos **ZLIB** e **PPMD** e na avaliação de desempenho desses compressores na DAMICORE.

4.1 Estudo dos Compressores sem Perdas de Informações

A compressão de dados refere-se ao processo de codificação da informação digital para que ela ocupe menos espaço. Utilizando diferentes métodos computacionais para reduzir redundâncias, essa técnica diminui a necessidade de armazenamento e otimiza a velocidade de transferência de dados pela rede. A forma como esses métodos lidam com a integridade dos dados durante a redução define as duas abordagens principais de compressão.

O objetivo na abordagem sem perdas (*lossless*), é garantir a reversibilidade total do processo. Isso significa que, ao reverter a compressão, o arquivo resultante é perfeitamente idêntico ao original, sem que informações sejam alteradas. Este método é crucial em aplicações onde a exatidão é fundamental, como em bancos de dados, documentos textuais e programas de computador.

Em oposição, a abordagem com perdas (*lossy*) alcança taxas de compressão maiores ao eliminar permanentemente informações do arquivo original. Essas informações descartadas são escolhidas por serem menos perceptíveis aos sentidos humanos. Embora impeça a recuperação exata dos dados originais, essa técnica é usual em arquivos de multimídia, como áudio (**MP3**) e imagens (**JPEG**), onde a diminuição na qualidade é aceitável em troca de economia de espaço.

Inicialmente os compressores sem perdas estudados foram o **ZLIB**, **GZIP**, **PPMD** e o **BZ2** (ou **BZIP2**), pois eles já haviam sido implementados na **Damicorepy**, uma implementação da metodologia DAMICORE [7].

4.1.1 ZLIB

O **ZLIB** [8] é uma biblioteca de compressão de dados baseada no algoritmo **DEFLATE** [9]. O **DEFLATE** é a junção de outros dois algoritmos: o **LZ77** [10] e o **Huffman Encoding** (Codificação de Huffman) [11]. A estratégia de compressão do **DEFLATE** ocorre em duas etapas principais.

Primeiramente, o algoritmo **LZ77** é aplicado para eliminar redundâncias nos dados de entrada. Este método utiliza uma técnica de janela deslizante para identificar sequências de dados repetidas. Ao encontrar uma duplicata, a sequência é substituída por um ponteiro que referencia sua ocorrência anterior, indicando a distância e o comprimento da sequência correspondente. Este processo reduz o volume de dados ao substituir padrões recorrentes por referências compactas.

Subsequentemente, o algoritmo de Codificação de **Huffman** é aplicado sobre o resultado processado pelo **LZ77** (incluindo os literais e os ponteiros). **Huffman** é um método de codificação estatística que atribui códigos binários de comprimento variável aos símbolos de entrada com base em sua frequência de ocorrência. Símbolos mais frequentes recebem representações binárias mais curtas, enquanto símbolos menos frequentes recebem representações mais longas. A combinação desses dois métodos permite que o **DEFLATE** alcance altas taxas de compressão, explorando tanto a redundância estrutural (via **LZ77**) quanto a probabilística (via **Huffman**).

4.1.2 GZIP

O **GZIP** [12], assim como o **ZLIB**, se baseia no algoritmo **DEFLATE**. Porém, por ser uma ferramenta de compressão e não uma biblioteca, ele possui um padrão de arquivo que também possui o mesmo nome **GZIP**, cuja extensão é **.gz** [13]. Este padrão estabelece que os arquivos comprimidos no padrão **gzip** devem ter um cabeçalho e um rodapé, que armazenam metadados como nome do arquivo, data e soma de verificação **CRC32**.

4.1.3 PPMD

O **PPMD PPMD** implementa o algoritmo **PPM** (*Prediction by Partial Matching*) [14], que opera através da modelagem estatística dos dados. Sua função principal é estimar a probabilidade do próximo símbolo com base em um contexto formado por um número variável de caracteres precedentes. Diferentemente de abordagens baseadas em dicionário, como o **LZ77** que busca repetições literais de sequências, o **PPM** calcula a probabilidade de ocorrência de um símbolo dentro de múltiplos níveis de contexto hierárquicos.

A compressão é efetivada pela codificação aritmética, que converte as probabilidades estimadas em uma representação binária compacta, alocando menos bits para os símbolos considerados mais prováveis em um determinado contexto. Para realizar a

gestão desses múltiplos contextos, a implementação **PPMD** utiliza uma estrutura de dados em árvore digital (**TRIE**).

O mecanismo de predição hierárquica funciona da seguinte maneira: o algoritmo tenta prever o símbolo no contexto de ordem mais alta disponível. Se o símbolo não for encontrado nesse nível (ou seja, se for uma novidade para aquele contexto específico), um código de escape é gerado, forçando a transição para um contexto de ordem inferior (mais curto). Esse processo de regressão continua até que o símbolo seja previsto ou que se alcance um modelo base de ordem zero. Embora essa metodologia baseada em probabilidades seja altamente eficaz para dados com dependências complexas e de longo alcance, ela geralmente demanda maior consumo de memória e capacidade de processamento em comparação com técnicas mais simples.

4.1.4 BZ2

O **BZ2** comprime blocos de dados por meio de um processo de três estágios: a Transformada de Burrows-Wheeler (**BWT**) [15], a transformação *Move-to-Front* (**MTF**) e a codificação de **Huffman** [11].

Inicialmente o **BWT** reorganiza os dados para agrupar símbolos iguais, facilitando a compressão. Em seguida, o **MTF** converte a sequência reorganizada em valores que representam a posição de cada símbolo em uma lista dinâmica; símbolos que aparecem repetidamente tendem a ter posições próximas ao início dessa lista, resultando em números pequenos.

Isso favorece longas repetições de valores baixos, que são ideais para serem reduzidas pelo **RLE**. Por fim, a codificação de **Huffman** atribui códigos menores a símbolos mais frequentes, reduzindo ainda mais o tamanho. O **BZ2** tende a gerar taxas de compressão maiores que as do **ZLIB**, mas com maior uso de memória e processamento [15].

4.2 Revisão de Literatura

A metodologia TRICORDER [1] é projetada para identificar anomalias em sistemas de software através do monitoramento do consumo de recursos computacionais, como CPU e memória. A abordagem dispensa a análise das saídas funcionais do software, focando-se em detectar desvios no comportamento esperado dos perfis de execução. O processo da TRICORDER divide-se em duas fases principais: monitoramento, para coleta de dados e geração dos perfis de consumo; e análise, onde os perfis são processados para identificar padrões.

A fase de análise é executada pela DAMICORE [2], uma metodologia de aprendizado de máquina não supervisionado para agrupamento e classificação de dados. A DAMICORE opera de forma genérica, sendo capaz de processar diferentes tipos de dados. Sua principal função no contexto da TRICORDER é agrupar os perfis de execução com comportamentos similares e isolar os anômalos. Para isso, a DAMICORE emprega um processo de agrupamento hierárquico baseado em uma matriz de distâncias.

A métrica central utilizada para calcular essas distâncias é a Distância de Compressão Normalizada (NCD) [3], [4]. Fundamentada na teoria da complexidade de Kolmogorov, a NCD oferece uma abordagem poderosa e agnóstica ao formato dos dados, pois mede a similaridade com base em quão bem eles podem ser comprimidos juntos. A lógica subjacente é que, se dois arquivos (**x** e **y**) são muito similares, a compressão da sua concatenação (**xy**) será mais eficiente do que a compressão de cada um separadamente.

O valor da NCD é normalizado no intervalo $[0, 1]$, facilitando a interpretação: valores próximos a 0 significam alta similaridade, enquanto valores próximos a 1 indicam alta dissimilaridade. A distância é calculada pela Equação 1, onde x e y são os arquivos, xy é a concatenação dos arquivos e $Z(\cdot)$ é a função que retorna o tamanho (em bytes) do arquivo comprimido por um compressor Z.

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

Uma análise comparativa realizada por Braga ([6]) destacou o desempenho superior do **ZLIB** em termos de tempo de resposta e taxa de acertos na identificação de anomalias, posicionando-o como a escolha preferencial para experimentos subsequentes. O **PPMD**, compressor padrão da ferramenta, também demonstrou ser uma alternativa promissora. Em contrapartida, os compressores **BZ2** e **GZIP** apresentaram um desempenho significativamente inferior, com tempos de identificação de duas a três vezes maiores que as opções mais eficientes.

4.2.1 Teste Experimental da NCD

Para este experimento, foram selecionados perfis de três tipos de anomalias apresentadas na Tabela 2: *PROCESS*, *TRAIN* e *MEMORY*. Essa escolha foi deliberada para representar, respectivamente, os níveis de impacto alto, médio e baixo, permitindo uma análise da sensibilidade da métrica. A fim de isolar o efeito de cada anomalia, todos os perfis, incluindo os de referência (*CONTROL*), foram gerados sob uma carga

de trabalho única e consistente, correspondente à classificação "Grande"(**Dataset: CANCER-PHON**) da Tabela 1.

O procedimento de análise foi realizado de forma independente para cada tipo de anomalia (*PROCESS*, *TRAIN* e *MEMORY*). Para cada caso, o script trabalhou com um conjunto de dados combinado de 20 perfis: os 10 de controle e os 10 da anomalia em questão. Sobre este conjunto de 20 perfis, foi realizada uma comparação "todos para todos", calculando-se a distância entre todas as combinações possíveis de pares de arquivos através da NCD (Equação 1). O resultado para cada anomalia foi uma matriz de distâncias de 20x20. Consequentemente, a diagonal principal de cada matriz (onde o índice da linha é igual ao da coluna) representa a comparação de um arquivo consigo mesmo, seja ele de controle ou de anomalia. Os demais valores quantificam a dissimilaridade: quanto menor o valor, maior a semelhança entre os perfis.

A análise dos resultados partiu de duas considerações básicas. Para pares heterogêneos (controle-anomalia), esperavam-se valores de NCD próximos de 1, indicando perfis "distintos". Para pares homogêneos, ou seja, dentro da mesma categoria (controle-controle ou anomalia-anomalia), a expectativa era de valores próximos de 0, indicando perfis "semelhantes". Comparações entre anomalias de tipos diferentes (e.g., *PROCESS vs. MEMORY*) foram excluídas deliberadamente, pois a distinção entre diferentes falhas foge ao escopo da ferramenta TRICORDER, cujo objetivo é a detecção de anomalias em relação a um comportamento de referência.

Os resultados indicaram desempenhos variados: o **PPMD** foi o mais eficaz, apresentando a maior diferença entre a NCD de arquivos similares e distintos. **ZLIB** e **GZIP** tiveram um desempenho parecido e mais modesto, enquanto o **BZ2** se mostrou ineficaz para esta tarefa, com resultados que não refletiam a dissimilaridade esperada.

Neste experimento não foi levado em consideração o tempo de resposta dos compressores.

4.3 Projeto e Implementação de Otimizações dos Compressores

Os experimentos iniciais, alinhados aos resultados de Braga, destacaram os compressores **PPMD**, **ZLIB** e **GZIP**. Dada a similaridade entre **ZLIB** e **GZIP**, que compartilham o algoritmo **DEFLATE**, a análise subsequente focou na otimização do **PPMD** e do **ZLIB**.

O compressor **PPMD** possui dois parâmetros configuráveis principais: a ordem e

o limite de memória. A ordem determina a profundidade do contexto estatístico usado para prever símbolos, onde ordens maiores tendem a aumentar a compressão. O limite de memória restringe o espaço alocado para a estrutura de dados do modelo preditivo. Os limites de ordem e de memória do **PPMD** são relativos à máquina que o executa.

Já o **ZLIB** oferece três parâmetros ajustáveis: nível de compressão, tamanho da janela deslizante e quantidade de memória. O nível de compressão regula o esforço computacional aplicado nas etapas de busca por repetições (**LZ77**) e otimização (Huffman Encoding), podendo ser configurado de 0 (sem compressão) a 9 (máxima compressão). O tamanho da janela deslizante define o volume do histórico de dados analisado para encontrar padrões, com valores que podem ir de 9 a 15. Por fim, o parâmetro de quantidade de memória, que varia de 1 a 9, gerencia a alocação de recursos para *buffers* internos.

4.3.1 Teste das Otimizações na NCD

Este experimento replicou o teste anterior, com foco em analisar o impacto dos parâmetros configuráveis do **PPMD** e **ZLIB** sobre a métrica NCD. A metodologia consistiu em variar um único parâmetro de cada vez, percorrendo toda a sua faixa de valores (do mínimo ao máximo). Para isolar a influência do parâmetro sob análise, os demais parâmetros permaneceram fixos em seus valores máximos durante cada execução do teste.

A análise dos parâmetros do compressor **PPMD** demonstra uma interdependência entre a ordem de compressão e a memória alocada. Para obter uma maior diferenciação de perfis (**NCD**), é necessário aumentar a ordem de compressão. No entanto, o aumento da ordem só se traduz em melhoria da **NCD** se a memória for suficiente para suportá-la. Caso a memória seja inadequada para uma ordem elevada, a capacidade de diferenciação do algoritmo diminui. Em termos de eficiência, o tempo de processamento aumenta com ordens mais complexas, mas pode ser otimizado, pois a diferenciação (**NCD**) tende a se estabilizar a partir de um certo limite de memória.

Para o compressor **ZLIB**, a relação entre os parâmetros e os resultados mostrou-se menos linear. O aumento do nível de compressão resultou em maior tempo de processamento, como esperado, mas não demonstrou uma correlação consistente com a melhoria da diferenciação NCD; os resultados variaram de forma não monotônica. Em relação à memória e ao tamanho da janela deslizante, os testes indicaram que, uma vez ultrapassados os valores mínimos, ajustes adicionais nesses parâmetros tiveram

impacto negligenciável na métrica NCD, sugerindo um platô de desempenho para essas configurações.

A partir dessas observações, a análise comparativa entre os compressores **PPMD** e **ZLIB** revela um claro trade-off: o PPMD oferece uma capacidade de diferenciação de perfis (NCD) substancialmente superior, mas com um custo computacional muito elevado. O **ZLIB** opera de forma mais eficiente (até dez vezes mais rápido), porém sua capacidade de distinção é consideravelmente inferior, mesmo em configurações otimizadas.

4.3.2 Teste das Otimizações na DAMICORE

Para expandir a análise para além da métrica NCD, o experimento subsequente focou na performance de agrupamento da metodologia DAMICORE, utilizando a implementação Damicorepy. Nesta fase, a variação experimental foi restrita aos parâmetros que influenciam diretamente a taxa de compressão: o nível para o **ZLIB** e a ordem para o **PPMD**.

A metodologia consistiu em executar o algoritmo de agrupamento para cada um dos 6 **datasets**. Para cada execução, o conjunto de dados de entrada era formado por 60 perfis: os 30 perfis de referência (*CONTROL*) e os 30 perfis de um dos 7 tipos de anomalia. O critério de sucesso para cada execução era a capacidade do algoritmo de formar ao menos um cluster composto exclusivamente por perfis anômalos. Considerando as 7 anomalias e os 6 **datasets**, o número total de cenários de teste foi 42.

Os resultados revelam uma forte correlação positiva entre a intensidade da compressão (nível ou ordem) e a precisão do agrupamento. O **ZLIB** demonstrou uma melhoria consistente, enquanto o **PPMD** apresentou uma exceção a essa tendência. O número de acertos em 42 testes foi:

- **ZLIB**: 37 (nível 3), 39 (nível 6) e 40 (nível 9).
- **PPMD**: 35 (ordem 4), 36 (ordem 8), 38 (ordem 12) e 36 (ordem 16).

Fica evidente que o **ZLIB** alcança seu desempenho máximo no nível de compressão mais alto testado (nível 9). Para o **PPMD**, a maior precisão ocorre na ordem 12; subsequentemente, a performance declina na ordem 16, confirmando-a como uma exceção à tendência geral de melhoria.

4.4 Proposição de Novos Compressores e Métodos

Em busca de alternativas aos compressores de dados convencionais, a pesquisa, conduzida em colaboração com Hugo Nakamura, direcionou-se para os algoritmos de compressão de imagem. Embora majoritariamente conhecidos por sua compressão com perdas (*lossy*), identificou-se que muitos desses formatos também oferecem modos de operação totalmente sem perdas (*lossless*).

Com base nessa distinção, foi definida uma estratégia de aplicação e a análise dos compressores foi dividida, de modo que cada pesquisador se aprofundou nos formatos mais alinhados aos seus respectivos temas. Os algoritmos selecionados para este estudo foram o **PNG**, um formato intrinsecamente sem perdas, e as versões *lossless* do JPEG 2000 (**JP2**) e do **WEBP**, que são primariamente projetados para compressão com perdas.

Adicionalmente aos próprios compressores, foi explorado um método analítico alternativo ao processo de compressão efetiva. Esta abordagem consiste em estimar o tamanho mínimo teórico de um arquivo após a compressão sem perdas, utilizando o conceito da **Entropia de Shannon**. O objetivo é prever o resultado da compressão sem incorrer no custo computacional de executá-la.

4.4.1 PNG

O **PNG** (*Portable Network Graphics*) é um compressor de imagens sem perdas baseado no **DEFLATE** [16]. Embora o **DEFLATE** não seja especializado para dados de imagem, o **PNG** otimiza sua eficiência através de uma etapa de pré-processamento. Nessa etapa, são aplicados filtros preditivos que analisam os *pixels* adjacentes para transformar os dados da imagem, realçando padrões e redundâncias. Esse processo prepara os dados para que o **DEFLATE** alcance uma taxa de compressão superior à que obteria com os dados brutos.

4.4.2 JP2

O **JPEG 2000** (JP2) [17] foi desenvolvido como uma evolução do padrão **JPEG**, introduzindo maior flexibilidade e eficiência. Ele suporta tanto compressão com perdas quanto sem perdas, sendo capaz de lidar com imagens de grandes dimensões e oferecendo recursos como decodificação progressiva e maior tolerância a erros. A arquitetura do **JPEG 2000** é baseada na **Transformada Wavelet Discreta** (DWT), utilizada para a compressão com perdas, e na sua variante reversível, a **Transformada**

Wavelet de Inteiros, para a compressão sem perdas. Essa dualidade permite que o mesmo codificador alterne entre máxima fidelidade e altas taxas de compressão. Após a aplicação da transformada, os coeficientes resultantes são quantizados (no modo com perdas) e, em seguida, codificados pelo algoritmo **EBCOT** (*Embedded Block Coding with Optimal Truncation*). O **EBCOT** organiza os dados em blocos independentes, gerando um fluxo de bits escalável que permite a extração de diferentes resoluções e qualidades a partir de um único arquivo.

4.4.3 WEBP

O **WEBP** é um compressor que oferece compressão com e sem perdas, combinando diferentes técnicas para otimizar a compactação de imagens. No modo com perdas, o **WEBP** utiliza um método de predição *intra-frame* derivado do **codec** de vídeo VP8 [18], no qual blocos de *pixels* são previstos com base em blocos vizinhos já decodificados. Já no modo sem perdas, o formato emprega um conjunto de estratégias mais sofisticadas antes da compressão final com o algoritmo **DEFLATE**. Essas estratégias incluem a predição espacial, que estima o valor de um pixel com base nos seus vizinhos para explorar a correlação local; o uso de um cache de cores, que armazena cores utilizadas recentemente para referenciá-las de forma eficiente; e a predição por blocos, que analisa padrões locais para diminuir a entropia dos dados. Ao combinar essas técnicas, o **WEBP** explora as redundâncias específicas de imagens de forma mais eficaz que algoritmos genéricos como o **DEFLATE** puro, resultando em arquivos menores com alta fidelidade visual.

4.4.4 ENTROPY

O **ENTROPY** é uma implementação do cálculo da **Entropia de Shannon**. A entropia não realiza uma compressão efetiva, mas estima o limite inferior teórico para o tamanho que um arquivo pode atingir ao ser comprimido por um algoritmo de compressão sem perdas ideal. Esse conceito é introduzido por Claude Shannon em seu artigo *"A Mathematical Theory of Communication"* ([19]). A definição formal da entropia de uma fonte discreta é dada pela Equação 2.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{2}$$

onde $H(X)$ é a entropia da fonte X, e $p(x_i)$ é a probabilidade de ocorrência de

cada caractere x_i do arquivo. Quanto mais imprevisível ou variado é o conteúdo, maior é a sua entropia e, portanto, menor o seu potencial de compressão.

4.4.5 Ressalva aos Compressores de Imagem

A aplicação de algoritmos de compressão de imagem aos perfis de monitoramento requer uma etapa de pré-processamento, dado que os dados de origem são texto e não imagens. A metodologia reinterpreta o arquivo de entrada como um fluxo de dados binários, onde cada byte é sistematicamente mapeado para os canais de cor, vermelho, verde e azul (RGB) de um *pixel*. O resultado desse processo é uma imagem em formato **BITMAP**, uma representação visual não comprimida dos dados originais. A escolha por essa transformação se justifica por produzir imagens com resolução suficiente para que os compressores atuem de forma eficaz e por estabelecer um método de conversão agnóstico ao conteúdo do arquivo de entrada.

4.5 Planejamento de Experimentos

O planejamento dos experimentos subsequentes foi elaborado em colaboração com Hugo Nakamura, a fim de garantir a comparabilidade dos resultados finais entre os dois projetos. Foram definidos três testes principais, cujos resultados serão detalhados nas subseções seguintes. A distinção fundamental entre os experimentos reside no tipo de compressor utilizado: a pesquisa de Nakamura foca em compressores com perdas, enquanto este trabalho aborda os compressores sem perdas.

O primeiro experimento, denominado Detecção de Anomalias por Agrupamento, consistiu em um teste de classificação utilizando a biblioteca **Damicorepy**. A metodologia empregada foi a mesma dos testes anteriores com os compressores **ZLIB** e **PPMD**, consistindo em: para cada um dos 6 conjuntos de dados (**datasets**), executar o algoritmo de agrupamento em uma entrada de 60 perfis, sendo 30 de referência (*CONTROL*) e 30 de um dos 7 tipos de anomalia. O critério de sucesso para cada execução foi a capacidade do algoritmo de formar ao menos um *cluster* composto exclusivamente por perfis anômalos. A combinação dos 7 tipos de anomalia com os 6 **datasets** resultou em um total de 42 cenários de teste. Adicionalmente, o tempo de processamento de cada compressor foi monitorado durante as execuções para calcular uma média, fornecendo uma base inicial para a análise de eficiência.

O segundo experimento foi projetado para a Análise de Falsos Positivos. Para isso, realizou-se um teste de classificação comparando um conjunto de perfis de con-

trole (*CONTROL*) com uma cópia idêntica deste conjunto (denominada *ANOTHER-CONTROL*). Nesse cenário, o resultado esperado era a não formação de um *cluster* exclusivo de perfis *ANOTHER-CONTROL*, pois a sua detecção como um grupo distinto indicaria a identificação incorreta de uma anomalia, ou seja, um falso positivo.

Por fim, o terceiro experimento realizou uma Análise de Detecção Incremental, modificando a abordagem de comparação para emular a ferramenta TRICORDER. Diferentemente dos testes anteriores, que processavam 30 perfis anômalos e 30 de controle simultaneamente, esta nova metodologia partiu de um conjunto com todos os perfis de controle. A cada iteração, um único perfil anômalo era adicionado ao conjunto, e a classificação era executada. O processo repetia-se até que um *cluster* exclusivo de anomalias fosse formado ou todos os 30 perfis anômalos fossem utilizados.

Devido ao alto custo computacional da análise incremental, este experimento concentrou-se exclusivamente no **dataset** *CANCER-PHON*, selecionado por representar uma carga de trabalho de grande impacto. Para garantir a robustez estatística dos resultados de tempo, o teste para cada um dos sete tipos de anomalia foi executado 100 vezes, utilizando-se a mediana dos tempos de resposta para a análise de eficiência, visto que a mediana é menos sensível a valores extremos (*outliers*) do que a média.

A avaliação dos compressores no contexto do terceiro experimento foi realizada a partir de duas dimensões: eficiência e eficácia. A eficiência foi tratada como uma medida de custo computacional, correspondendo à duração total do processo incremental, ou seja, calculou-se o somatório do tempo de cada iteração até que o sistema obtivesse sucesso na detecção (formando um *cluster* exclusivo) ou esgotasse o conjunto de dados anômalos. Por outro lado, a eficácia representou a capacidade de detecção correta do modelo, sendo quantificada por meio de métricas consolidadas na literatura de aprendizado de máquina: acurácia, precisão, *recall* e *F1-score*.

As métricas de eficácia baseiam-se nos seguintes eventos, que definem a matriz de confusão no contexto deste trabalho:

- Verdadeiro Positivo (VP): Um acerto na detecção de anomalia, caracterizado pela correta formação de um *cluster* isolado contendo apenas perfis defeituosos.
- Falso Negativo (FN): Uma falha de detecção, registrada quando os perfis defeituosos não são distinguidos dos perfis de controle, impossibilitando a criação de um *cluster* exclusivo para eles.
- Falso Positivo (FP): Um alarme falso. Este evento acontece durante o teste de

controle (*CONTROL* vs. *ANOTHER-CONTROL*), caso o algoritmo identifique diferenças inexistentes e crie um *cluster* exclusivo de perfis normais.

- Verdadeiro Negativo (VN): Um acerto na rejeição de anomalia. Ocorre quando, no teste de controle, o sistema corretamente não encontra anomalias e, portanto, não gera nenhum *cluster* exclusivo.

Como havia apenas um cenário de teste para o FP (resultando em no máximo 1 Falso Positivo) contra sete cenários para as outras métricas da matriz de confusão, foi aplicado um balanceamento para evitar que os resultados fossem enviesados. Para isso, os eventos do FP receberam um peso 7, equiparando sua importância à soma dos sete cenários das outras métricas da matriz de confusão.

Com base nas métricas de eficiência (tempo de resposta) e eficácia (métricas da matriz de confusão) coletadas, a análise dos resultados procedeu com uma validação estatística rigorosa. Inicialmente, o teste de Shapiro-Wilk foi empregado para verificar a normalidade da distribuição dos dados de tempo. A partir dessa verificação, o teste não paramétrico de Wilcoxon foi utilizado para comparar o desempenho entre os compressores e identificar diferenças estatisticamente significativas. Finalmente, para consolidar os achados e proporcionar uma visão integrada, foi apresentado um mapa de desempenho que relacionou a eficácia e a eficiência, permitindo uma avaliação visual do equilíbrio entre a precisão e o custo computacional de cada compressor.

4.6 Execução de Experimentos

Os resultados obtidos na execução dos experimentos são apresentados a seguir. Esta análise foca especificamente nos testes com compressores sem perdas (*lossless*) e **ENTROPY**.

4.6.1 Detecção de Anomalias por Agrupamento

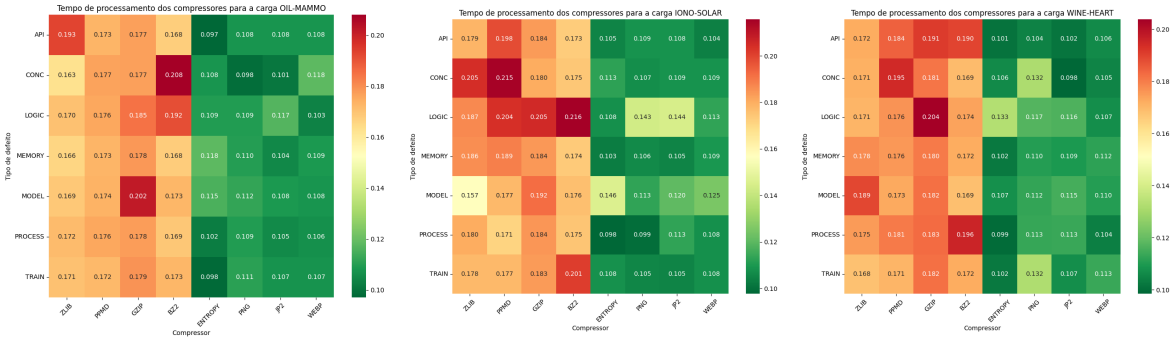
A primeira avaliação experimental teve como objetivo verificar a capacidade fundamental do método em distinguir perfis anômalos de perfis de controle nos 42 cenários propostos. Os resultados a seguir focam na eficácia (taxa de sucesso) e na eficiência (tempo de processamento) de cada compressor.

A Tabela 3 consolida a eficácia de cada compressor, apresentando o número de detecções bem-sucedidas e a taxa de sucesso correspondente.

COMPRESSOR	BANK-PIMA	CANCER-PHON	IONO-SOLAR	IRIS-HABER	OIL-MAMMO	WINE-HEART	Sucessos	Taxa de Sucesso (%)
GZIP	6	7	7	7	6	7	40	95,2
ZLIB	7	6	6	7	7	7	40	95,2
PPMD	7	5	6	6	6	6	36	85,7
BZ2	5	7	4	4	7	7	34	81,0
JP2	6	5	7	5	7	4	34	81,0
ENTROPY	6	5	6	3	6	5	31	73,8
WEBP	4	4	5	5	6	6	30	71,4
PNG	6	6	4	4	5	3	28	66,7

Tabela 3: Acertos por compressor em diferentes cargas.

Os mapas de calor apresentados na Figura 1, consolidam os tempos de processamento para as cargas de trabalho (ou datasets): *OIL-MAMMO* (Figura 1a), *IONO-SOLAR* (Figura 1b) e *WINE-HEART* (Figura 1c). Foram escolhidos esses datasets por representarem, respectivamente, cargas de trabalho Muito Grande, Médio-Grande, Pequeno.



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

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

A análise da Tabela 3 revela uma clara distinção no desempenho entre os grupos de compressores. Os algoritmos de uso geral, como **GZIP** e **ZLIB**, destacaram-se como os mais eficazes, alcançando uma taxa de sucesso idêntica de 95,2%. O **PPMD** e o **BZ2** também apresentaram boa performance, com taxas acima de 80%. Notavelmente, as falhas do **BZ2** se concentraram nos **datasets** *BANK-PIMA*, *IONO-SOLAR* e *IRIS-HABER*, sugerindo uma sensibilidade particular a certos tipos de dados.

Em contrapartida, o desempenho dos compressores de imagem foi mais variado. O **JP2** foi o destaque do grupo, com uma taxa de sucesso respeitável de 81%. Em um patamar intermediário, o **WEBP** alcançou 71,4% de eficácia, enquanto o **PNG** ficou significativamente atrás com 66,7%, sendo o menos eficaz entre os compressores de imagem. A abordagem teórica baseada em entropia (**ENTROPY**) apresentou um resultado moderado de 73,8%, servindo como um interessante ponto de referência entre o **JP2** e os demais.

Ao avaliar a eficiência nos mapas de calor (Figura 1), a tendência se inverte. Os compressores de imagem (**PNG**, **JP2** e **WEBP**) e o **ENTROPY** foram consistentemente mais rápidos que os compressores de uso geral. Essa vantagem de velocidade mostra-se consistente entre os diferentes volumes de dados, com os compressores genéricos demonstrando um custo computacional visivelmente maior em todos os cenários apresentados (Figura 1).

Em síntese, os resultados deste primeiro experimento apontam para um claro *trade-off* entre eficácia e eficiência, medida aqui pelo tempo de processamento. Por um lado, **GZIP** e **ZLIB** ofereceram a maior confiabilidade na detecção, mas com um custo computacional mais elevado. Por outro lado, compressores como **PNG**, **JP2**, **WEBP** e **ENTROPY** garantiram um menor tempo de processamento, porém sacrificando a precisão da detecção. A escolha, neste primeiro momento, parece depender da prioridade da aplicação: máxima acurácia ou menor custo de processamento.

4.6.2 Análise de Falsos Positivos

A segunda avaliação experimental teve como objetivo verificar a robustez do método contra falsos positivos. Para isso, o algoritmo de agrupamento foi executado em cenários contendo apenas perfis de controle (*CONTROL* vs. *ANOTHER-CONTROL*) para cada *dataset*. O resultado esperado é a ausência de formação de *clusters* exclusivos, indicando que o sistema não identifica anomalias onde elas não existem.

Os resultados deste teste de robustez são sumarizados na Tabela 4, que reporta a incidência de falsos positivos para cada compressor nos seis *datasets* de controle. O sucesso, neste caso, é a ausência total de detecções.

COMPRESSOR	BANK-PIMA	CANCER-PHON	IONO-SOLAR	IRIS-HABER	OIL-MAMMO	WINE-HEART	Falhas	Taxa de Falha (%)
ZLIB	0	0	0	0	0	0	0	0.0
PPMD	0	0	0	0	0	0	0	0.0
GZIP	0	0	0	0	0	0	0	0.0
BZ2	0	0	0	0	0	0	0	0.0
JP2	1	1	0	0	0	1	3	50.0
ENTROPY	0	0	0	0	0	0	0	0.0
PNG	0	0	0	0	0	0	0	0.0
WEBP	0	0	0	0	0	0	0	0.0

Tabela 4: Incidência de Falsos Positivos por Compressor e *Dataset*

A análise de robustez revela um resultado extremamente positivo para a grande maioria dos compressores. Como detalhado na Tabela 4, praticamente todos os algoritmos, incluindo os de uso geral, os de imagem (**PNG** e **WEBP**) e a abordagem teórica (**ENTROPY**), demonstraram confiabilidade ideal, não gerando nenhum falso

positivo. Este desempenho de 100% de sucesso é um forte indicativo de que suas representações de dados são estáveis e consistentes.

A única e notável exceção foi o compressor **JP2**. Ele identificou anomalias inexistentes em três dos seis **datasets**: *BANK-PIMA*, *CANCER-PHON* e *WINE-HEART*, resultando em uma taxa de falha de 50% neste experimento.

Este achado introduz uma dimensão crítica para a avaliação do **JP2**. A sua eficácia na detecção de anomalias reais, observada no experimento anterior, contrasta diretamente com sua propensão a gerar falsos positivos em cenários de controle. A implicação deste comportamento é que, embora seja capaz de identificar problemas existentes, o **JP2** também pode sinalizar anomalias inexistentes. Portanto, a sua adequação para uma aplicação prática dependerá da tolerância do sistema a alarmes falsos versus a necessidade de garantir a detecção.

4.6.3 Análise de Detecção Incremental

O experimento avaliou os compressores em duas dimensões: eficácia e eficiência. A eficácia, que mede a capacidade de detecção correta, foi quantificada pelas métricas de acurácia, precisão, *recall* e *F1-score*. A eficiência foi medida pelo custo computacional, correspondente à duração total do processo.

Os resultados de eficácia, detalhados na Figura 2, mostram que os compressores **PNG**, **ENTROPY** e a maioria dos genéricos atingiram desempenho máximo em todas as métricas (100%). O compressor **BZ2** também apresentou alta performance, com 92,9% de acurácia, mas com uma leve redução no recall (85,7%), resultando em um F1-score de 92,3%.

Em contraste, os compressores **JP2** e **WEBP** tiveram o desempenho mais baixo. O **JP2** apresentou 50% de acurácia e precisão, resultando em um *F1-score* de 66,7%. Já o **WEBP** registrou os piores índices: seu *recall* de 71,4% foi o mais baixo, acompanhado por 35,7% de acurácia e 41,7% de precisão, o que culminou no menor *F1-score* do teste (52,6%).

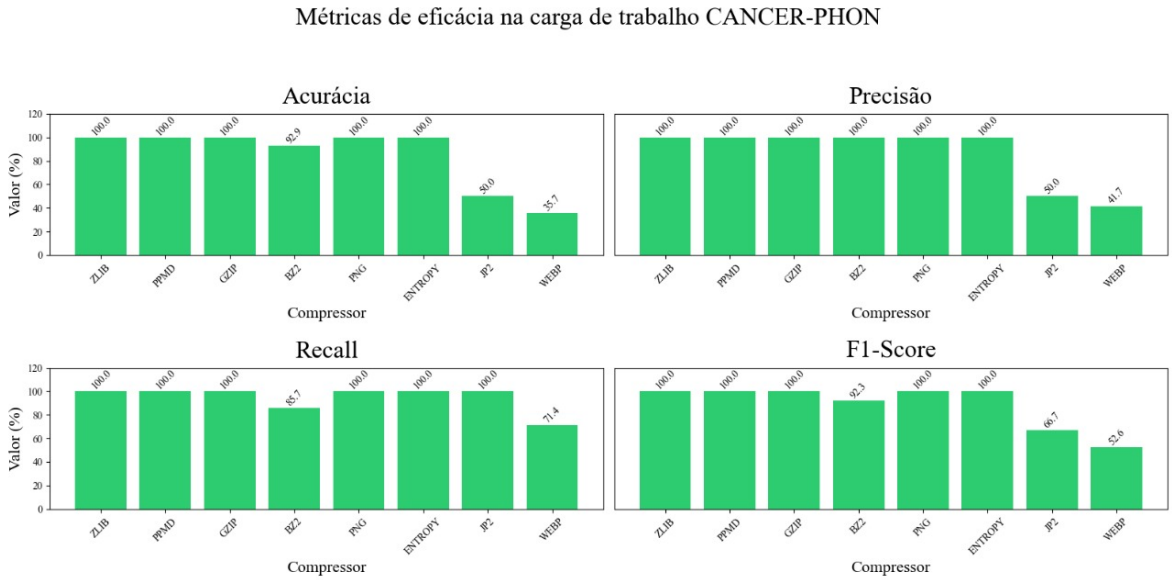


Figura 2: eficácia dos compressores na carga *CANCER-PHON*.

A Figura 3 apresenta os resultados de eficiência da detecção em dois mapas de calor, baseados na mediana de 100 execuções. O primeiro mapa ("Matriz de Iteração") exibe o número de iterações, e o segundo ("Matriz de Mediana de Tempo de Resposta") mostra o tempo em segundos. Falhas na detecção são indicadas em cinza. Confirmando a análise de eficácia, os mapas de calor evidenciam que **JP2** e **WEBP** foram os únicos compressores a gerar falsos positivos, identificando erroneamente anomalias no grupo *ANOTHER-CONTROL*.

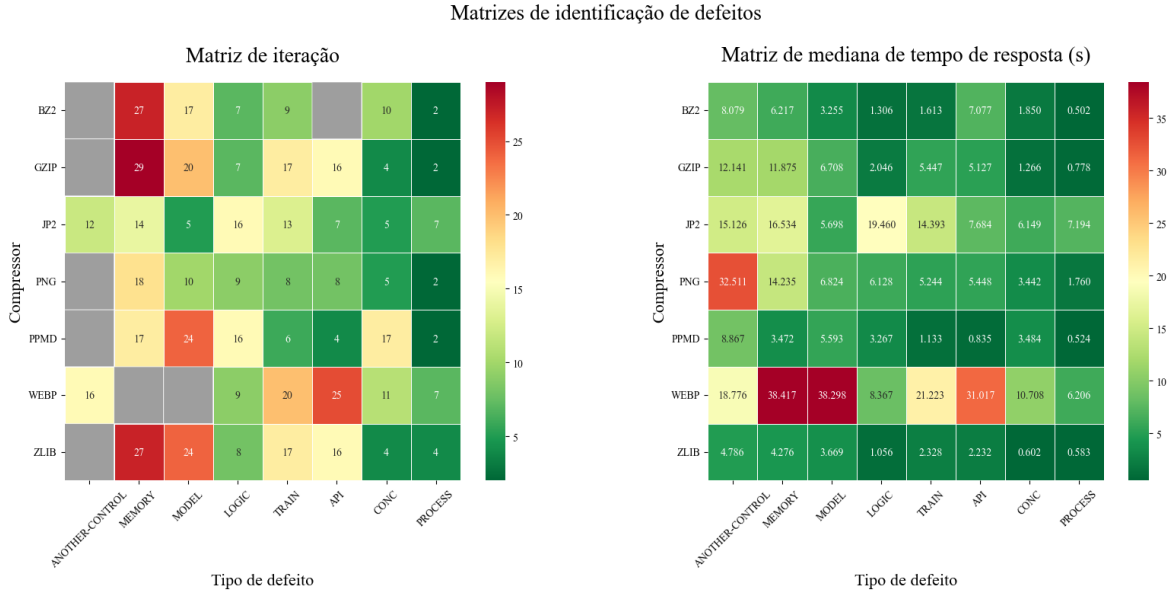
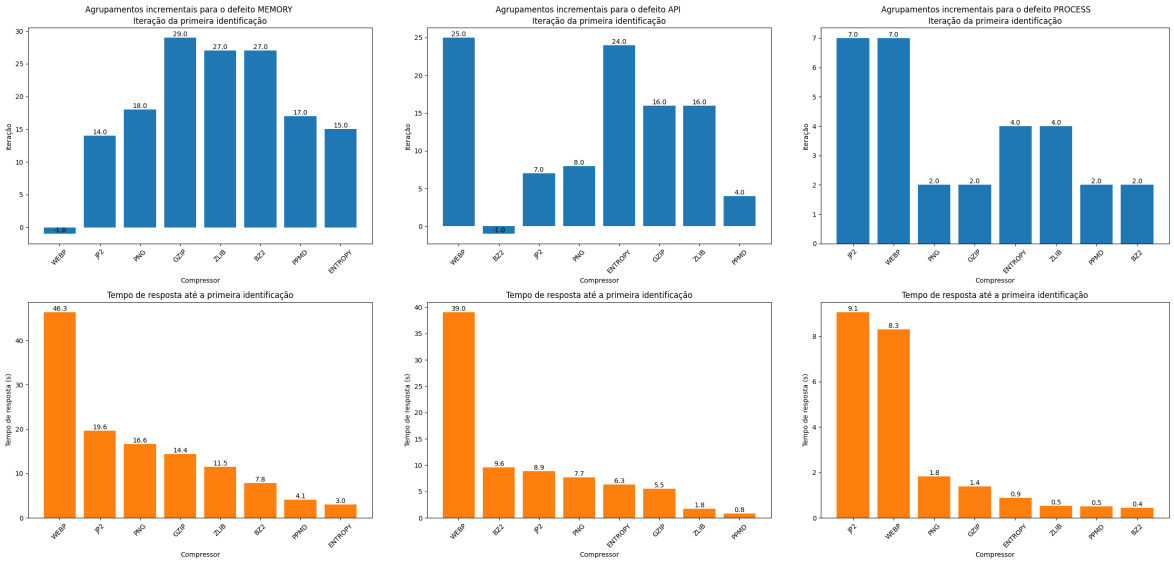


Figura 3: matriz de identificação de anomalias na carga *CANCER-PHON*.

O desempenho em anomalias específicas é detalhado na Figura 4. Na detecção de

MEMORY, **ENTROPY** foi o mais rápido (3s em 15 iterações), enquanto **WEBP** falhou. Para a anomalia API, **PPMD** teve a melhor performance (0,8s em 4 iterações), e **WEBP** novamente a pior (39s em 25 iterações). Finalmente, na anomalia *PROCESS*, um grupo de compressores (**BZ2**, **PPMD**, **GZIP**, **PNG**) detectou a falha em 2 iterações, com **BZ2** sendo o mais veloz (0,4s). Em contrapartida, **JP2** apresentou o maior tempo de resposta (9,1s em 7 iterações).



(a) (b) (c)

Figura 4: Resultado dos agrupamentos incrementais para os tipos *MEMORY*, *API* e *PROCESS* na carga *CANCER-PHON*.

4.7 Análise Crítica dos Resultados

Para validar estatisticamente a eficiência dos compressores, a distribuição dos dados foi primeiramente verificada com o teste de Shapiro-Wilk. Os resultados (Tabela 5) indicam que apenas as amostras dos compressores **JP2** e **BZ2** seguem uma distribuição normal.

Compressor	Defeito	Valor-P	Dist. Normal
JP2	CONC	0.1084	SIM
BZ2	PROCESS	0.0571	SIM

Tabela 5: compressores com tempos de execução com distribuição normal pelo teste de Shapiro-Wilk.

Devido à não normalidade da maioria dos dados, o teste não paramétrico de Wilcoxon foi empregado para comparar o desempenho entre os compressores. A Ta-

bela 6 aponta a única comparação cujo resultado não foi estatisticamente significativo (p>0,05).

Tabela 6: Diferenças de tempo entre compressores consideradas não significativas pelo teste de Wilcoxon (p > 0,05).

Comparação		**Defeito**	**Valor-P**
BZ2	JP2	API	0.1772

Dessa forma, a diferença de desempenho entre os compressores **BZ2** e **JP2** para a anomalia, do tipo *API*, não é significativa, sendo seus desempenhos considerados equivalentes neste caso. Em todas as demais comparações, as variações de performance observadas são estatisticamente relevantes.

Para consolidar a análise, a Figura 5 apresenta um mapa de desempenho que relaciona a eficácia e a eficiência de cada compressor na carga de trabalho estudada. Para esta visualização, a eficácia (eixo Y) foi definida como acurácia normalizada, enquanto a eficiência (eixo X) corresponde ao inverso da mediana do tempo de resposta, também normalizada.

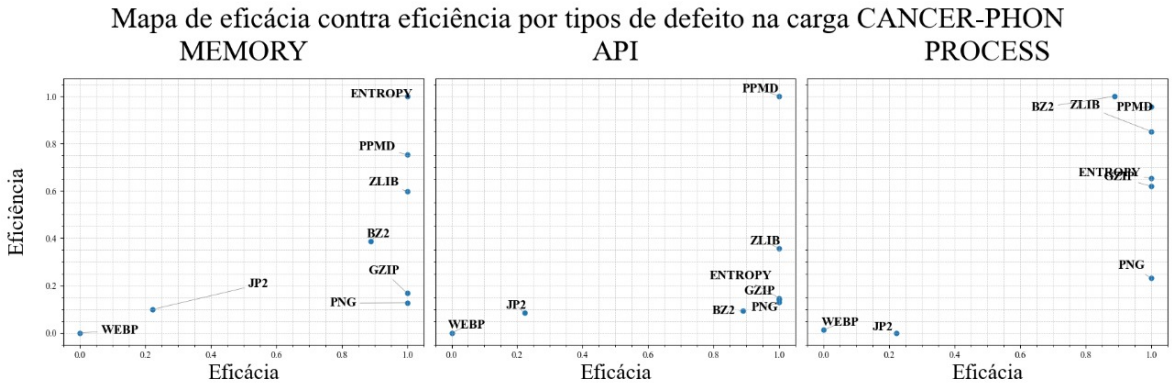


Figura 5: Mapa de eficácia por eficiência para três tipos de anomalias.

O primeiro, composto por **WEBP** e **JP2**, demonstrou desempenho consistentemente baixo, posicionando-se no canto inferior esquerdo dos gráficos. Isso indica que não apresentaram vantagens significativas em eficácia ou eficiência.

Em contrapartida, o segundo grupo, formado pelos compressores genéricos, **ENTROPY** e **PNG**, obteve os melhores resultados. Estes mantiveram 100% de eficácia na maioria dos cenários, com exceção do **BZ2**, cuja eficácia foi inferior a 90% nas anomalias *MEMORY* e *API*. Em termos de eficiência, o desempenho deste grupo foi variado; contudo, destaca-se negativamente o **PNG**, que permaneceu consistentemente abaixo de 50% de eficiência.

5 Publicações

Este projeto de Iniciação Científica resultou na submissão de dois artigos para eventos acadêmicos, detalhando os avanços da pesquisa.

Primeiramente, o resumo estendido **"Analisando Compressores sem Perdas de Informações na Detecção de Anomalias de Software com a TRICORDER"** [20] foi submetido ao Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP), na sua 33° edição. Este artigo apresentou uma análise comparativa da eficácia de diferentes compressores sem perdas.

Um segundo artigo, intitulado **"Efeitos da Compressão no Aprendizado Não Supervisionado para Detecção de Anomalias em Software"** [21], foi submetido ao Simpósio de Sistemas Computacionais de Alto Desempenho (SSCAD), porém ainda se encontra em fase de análise. Este estudo expandiu a análise ao investigar os efeitos de compressores com e sem perdas em algoritmos de aprendizado não supervisionado e foi desenvolvido em colaboração com Hugo Nakamura, unindo os resultados de ambas as pesquisas.

6 Conclusões

Este trabalho atingiu seu objetivo principal de realizar uma análise comparativa da eficácia e eficiência de diferentes compressores na ferramenta DAMICORE. A pesquisa validou o desempenho de algoritmos de referência, avaliou alternativas baseadas em formatos de imagem e, mais importante, revelou o potencial de um método baseado em entropia.

A análise confirmou a superioridade dos compressores genéricos de referência, como **ZLIB** e **PPMD**, que se estabeleceram como o padrão de alto desempenho. Dentre os compressores de imagem, o **PNG** destacou-se ao igualar a eficácia dos melhores compressores, embora tenha revelado um claro *trade-off*, com sua eficiência permanecendo consistentemente baixa. Em nítido contraste, os demais compressores de imagem, **JP2** e **WEBP**, mostraram-se inadequados para a tarefa.

A contribuição mais significativa deste estudo reside na análise do **ENTROPY**. Onde demonstrou uma eficácia de alto nível, comparável à dos melhores compressores genéricos. A disputa de desempenho, portanto, concentrou-se na eficiência, onde o **ENTROPY** se mostrou uma alternativa competitiva. Seu grande diferencial, porém, não reside apenas no desempenho quantitativo, mas em suas vantagens operacionais

(simplicidade, complexidade linear e ausência de dependências externas), que o tornam uma alternativa estratégica. Por fim, o estudo também validou metodologicamente que a configuração de máxima compressão é a ideal para otimizar os resultados da métrica NCD.

A produção e submissão de dois artigos científicos atendeu plenamente ao objetivo formal de disseminar os resultados desta Iniciação Científica. Mais importante, essa experiência prática consolidou um conjunto de habilidades essenciais para a pesquisa, incluindo a escrita científica, a análise estatística de dados e o aprofundamento teórico em aprendizado não supervisionado, capacitando o autor para futuros desafios acadêmicos.

Para garantir a transparência e a reprodutibilidade dos resultados, os *scripts*, dados e materiais que sustentam esta pesquisa estão disponíveis no repositório do projeto no GitHub: LosslessCompressors_IC [22].

Referências

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

[2] A. Sanches, J. M. P. Cardoso e A. C. B. Delbem, "Identifying Merge-Beneficial Software Kernels for Hardware Implementation," em *2011 International Conference on Reconfigurable Computing and FPGAs*, San José, Costa Rica: IEEE, 2011.

[3] M. Li, X. Chen, X. Li, B. Ma e P. M. Vitányi, "The Similarity Metric," *IEEE Transactions on Information Theory*, v. 50, n. 12, pp. 3250–3264, dez. de 2004. DOI: 10.1109/TIT.2004.838101.

[4] R. Cilibrasi e P. M. Vitányi, "Clustering by Compression," *IEEE Transactions on Information Theory*, v. 51, n. 4, pp. 1523–1545, abr. de 2005. DOI: 10.1109/TIT.2005.844059.

[5] S. Santos, B. Silveira, V. Durelli, R. Durelli, S. Souza e M. Delamaro, "On Using Decision Tree Coverage Criteria for Testing Machine Learning Models," em *SAST 2021*, Joinville, Brazil: ACM, 2021, pp. 1–9.

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

[7] B. K. M. Cesar, "Estudo e extensão da metodologia DAMICORE para tarefas de classificação (in Portuguese)," Master Dissertation, USP, 2016.

[8] zlib, *zlib Home Site*, https://zlib.net/, Acessado em: 2025-08-29, 2025.

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

[10] J. Ziv e A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, v. 23, n. 3, pp. 337–343, mai. de 1977.

[11] D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," *Proceedings of the IRE*, v. 40, n. 9, pp. 1098–1101, set. de 1952.

[12] Free Software Foundation, *Gzip - GNU Project - Free Software Foundation*, https://www.gnu.org/software/gzip/, Acessado em: 2025-08-29, 2025.

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

[14] A. Moffat, "Implementing the PPM Data Compression Scheme," *IEEE Transactions on Communications*, v. 38, n. 11, pp. 1917–1921, 1990.

[15] M. Burrows, D. J. W. D. I. G. I. T. A. L, R. W. Taylor, D. J. Wheeler e D. Wheeler, "A Block-sorting Lossless Data Compression Algorithm," 1994. endereço: https://api.semanticscholar.org/CorpusID:2167441.

[16] T. Boutell, T. Lane e et al., "PNG (Portable Network Graphics) Specification 1.0," Internet Engineering Task Force (IETF), Request for Comments RFC 2083, jan. de 1997.

[17] ISO/IEC-JTC-1/SC-29, "Information technology – JPEG 2000 image coding system: Core coding system," Int. Org. for Standardization, Standard ISO/IEC 15444-1:2004, 2004.

[18] J. Bankoski, J. Koleszar, L. Quillio, J. Salonen, P. Wilkins e Y. Xu, "VP8 Data Format and Decoding Guide," Internet Engineering Task Force (IETF), Request for Comments RFC 6386, mai. de 2011. endereço: https://www.rfc-editor.org/rfc/rfc6386.

[19] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal*, v. 27, n. 3, pp. 379–423, jul. de 1948.

[20] I. S. Soares, H. H. Nakamura, C. G. Herrera, S. do Rocio Senger de Souza e P. S. L. de Souza, "Analisando Compressores sem Perdas de Informações na Detecção de Anomalias de Software com a TRICORDER," em *33ª Simpósio Internacional de Iniciação Científica e Tecnológica da USP (SIICUSP)*, [SUBMETIDO], São Carlos, SP: Universidade de São Paulo, 2025.

[21] H. H. Nakamura, I. S. Soares, C. G. Herrera, S. do Rocio Senger de Souza e P. S. L. de Souza, "Efeitos da Compressão no Aprendizado Não Supervisionado para Detecção de Anomalias em Software," em *Simpósio em Sistemas Computacionais de Alto Desempenho (SSCAD)*, [SUBMETIDO], Brasil: Sociedade Brasileira de Computação, 2025.

[22] I. S. Soares, *LosslessCompressors_IC*, https://github.com/ISS2718/LosslessCompressors_IC, 2025.