UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
ENGENHARIA DE COMPUTAÇÃO

FELLIPE AUGUSTO SANTANA QUEIROZ

PREDIÇÃO DE CONSUMO ENERGÉTICO DE APLICAÇÕES OPENMP EM MÁQUINAS MULTI-CORE USANDO TÉCNICAS DE REGRESSÃO DE APRENDIZADO DE MÁQUINA

Tucuruí-PA
Dezembro de 2023

UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
ENGENHARIA DE COMPUTAÇÃO

FELLIPE AUGUSTO SANTANA QUEIROZ

PREDIÇÃO DE CONSUMO ENERGÉTICO DE APLICAÇÕES OPENMP EM MÁQUINAS MULTI-CORE USANDO TÉCNICAS DE REGRESSÃO DE APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso, apresentado à Faculdade de Engenharia de Computação, do Campus Universitário de Tucuruí, da Universidade Federal do Pará, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Marcos Túlio Amaris González

Tucuruí-PA
Dezembro de 2023

Queiroz, Fellipe
Predição de Consumo Energético de Aplicações OpenMP em Máquinas Multi-core usando Técnicas de Regressão de Aprendizado de Máquina / Fellipe Augusto Santana Queiroz . – Tucuruí-PA, Dezembro de 2023.
63 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Marcos Túlio Amaris González

Monografia – UNIVERSIDADE FEDERAL DO PARÁ
CAMPUS UNIVERSITÁRIO DE TUCURUÍ
, Dezembro de 2023.

1. OpenMP. 2. Green Computing. 3. Python. I. Título.

FELLIPE AUGUSTO SANTANA QUEIROZ

PREDIÇÃO DE CONSUMO ENERGÉTICO DE APLICAÇÕES OPENMP EM MÁQUINAS MULTI-CORE USANDO TÉCNICAS DE REGRESSÃO DE APRENDIZADO DE MÁQUINA

Trabalho de Conclusão de Curso, apresentado à Faculdade de Engenharia de Computação, do Campus Universitário de Tucuruí, da Universidade Federal do Pará, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação.

Data da Defesa: 13 de Dezembro de 2023
Conceito: Excelente

Banca Examinadora

Documento assinado digitalmente
gov.br
MARCOS TULIO AMARIS GONZALEZ
Data: 20/12/2023 11:56:07-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Marcos Túlio Amaris González
Faculdade de Engenharia de Computação - UFPA (CAMTUC)
Orientador

Documento assinado digitalmente
gov.br
OTAVIO NOURA TEIXEIRA
Data: 20/12/2023 18:07:01-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Otávio Noura Teixeira
Faculdade de Engenharia de Computação - UFPA (CAMTUC)
Membro da Banca

Documento assinado digitalmente
gov.br
DANIEL DA CONCEICAO PINHEIRO
Data: 20/12/2023 12:56:16-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Daniel da Conceição Pinheiro
Faculdade de Engenharia de Computação - UFPA (CAMTUC)
Membro da Banca

Tucuruí-PA
Dezembro de 2023

AGRADECIMENTOS

Manifesto aqui toda a minha gratidão à minha família, em especial aos meus pais, Dirlena C. Santana Queiroz e Antônio Augusto S. Queiroz, e aos meus irmãos, Bruna e Otávio, que sempre proporcionaram segurança e força na minha jornada durante a graduação e em minha vida, e a eles dedico este trabalho.

Estendo também a minha enorme gratidão a um irmão, que a faculdade me deu, que me ajudou muito mais do que ele imagina nessa minha passagem pela graduação, e que foi fundamental para a realização desse trabalho, como o de tantos outros. Meu muito obrigado, Erick Damasceno.

Deixo aqui meu grande agradecimento ao meu orientador, o Professor Dr. Marcos Amaris, cuja orientação possibilitou a execução e a finalização desta pesquisa.

Agradeço também a todos os professores que tive na Faculdade de Engenharia de Computação, UFPA- Tucuruí, pela excelente didática apresentada, e por contribuir com o meu crescimento pessoal e profissional.

Além disso, meus agradecimentos se estendem aos vários amigos e colegas, e a todos que, de alguma forma, colaboraram com a minha jornada. Suas contribuições foram essenciais para a finalização dessa etapa de minha vida.

“Sic Parvis Magna“
“Grandeza de Pequenos Começos...”
(Sir Francis Drake)

RESUMO

O campo de pesquisa em *Green Computing*, que visa tornar a computação mais sustentável e ecologicamente correta, tem sido impulsionado pela crescente integração de tecnologias de processamento e armazenamento de dados em larga escala. A complexidade crescente e o volume massivo de dados provenientes de diversas fontes têm desafiado as infraestruturas tradicionais, levando à exploração de plataformas Multi-core. Apesar do significativo aumento no desempenho e na eficiência energética com a utilização das máquinas multi-core, ainda assim, devido à crescente demanda de consumo, os gastos energéticos atingiu valores elevados. Neste estudo, descobrimos que modelos de regressão polinomial são mais eficazes que os lineares para prever o consumo de energia, especialmente em dados complexos. Além de que, a CPU é o maior consumidor de energia, sugerindo a necessidade de otimização ou uso de GPUs. Não encontramos uma correlação direta entre o tempo de execução e o consumo de energia, sugerindo que aplicações demoradas podem ter gastos menores, devido a otimizações. A análises de agrupamento dos *benchmarks* indicaram padrões de consumo semelhantes, úteis para otimizações futuras. A regressão polinomial de grau 3 foi eficiente em muitos casos, porém a eficácia varia com a quantidade de dados, e modelos personalizados de dados se mostraram mais eficientes do que abordagens unificadas.

ABSTRACT

The field of Green Computing research, which aims to make computing more sustainable and environmentally friendly, has been driven by the increasing integration of large-scale data processing and storage technologies. The growing complexity and massive volume of data from various sources has challenged traditional infrastructures, leading to the exploration of multi-core platforms. Despite the significant increase in performance and energy efficiency with the use of multi-core machines, nevertheless, due to the growing demand for consumption, energy expenditure has reached high figures. In this study, we found that polynomial regression models are more effective than linear ones for predicting energy consumption, especially in complex data. In addition, CPU is the largest energy consumer, suggesting the need for optimization or the use of GPUs. We did not find a direct correlation between execution time and energy consumption, suggesting that time-consuming applications may have lower costs due to optimizations. Cluster analysis of the benchmarks indicated similar consumption patterns, useful for future optimizations. Degree 3 polynomial regression was efficient in many cases, but effectiveness varies with the amount of data, and personalized data models proved more efficient than unified approaches.

Keywords: Multi-thread, OpenMP, Machine Learning, Green Computing, heterogeneous platforms.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

OpenMP	Open Multi-Processing, ou Multi-processamento aberto
ML	Machine Learning
CPU	Central Processing Unit, ou Unidade Central de Processamento
RAM	Random Access Memory, ou memória de acesso randômico
MAPE	Mean Absolute Percentage Error
IA	Inteligência Artificial
AM	Aprendizado de máquina
HPC	High-performance computing
BASH	Bourne-Again SHell
GPU	Graphics Processing Unit

SUMÁRIO

1 INTRODUÇÃO

Com o crescimento exponencial do mundo digital, marcado pela duplicação do número de usuários da internet e um aumento de 25 vezes no tráfego global de dados desde 2010, os *data centers* tornaram-se fundamentais para a eficiência da rede global (IEA, 2023). Essas infraestruturas são responsáveis por uma parcela significativa do consumo de energia, e representou entre 240 a 340 TWh em 2022, o que equivale a 1,3% do consumo global de eletricidade e 0,3% das emissões globais de CO2 (IEA, 2023).

O consumo energético dos *data centers* está aumentando rapidamente, com projeções que essas necessidades energéticas podem alcançar 752 TWh em 2030, ou seja, 2,13% da eletricidade global disponível (IEA, 2023). Esse aumento alarmante no consumo de energia levanta questões sobre a sustentabilidade dessas infraestruturas. Como resposta, a prática de *Green Computing* ganha destaque. Trata-se de uma área de pesquisa que busca fazer da computação um processo mais sustentável e ecológico (CORDEIRO et al., 2023). Inovações tecnológicas, como o desenvolvimento de máquinas multi-core e o uso de padrões de programação paralela como OpenMP, mostram-se promissoras para melhorar a eficiência energética (OPENMP API, 2020).

Contudo, novos desafios surgem com o avanço tecnológico, especialmente relacionados à otimização energética em sistemas com crescente paralelismo e heterogeneidade (JUDGE, 2022). Governos e as indústrias enfrentam a tarefa de intensificar os esforços em eficiência energética e pesquisa, em busca de alinhar o crescimento digital com objetivos de sustentabilidade e redução de emissões (Empresa de Pesquisa Energética (EPE), 2021). Portanto, o desafio consiste em desenvolver e implementar tecnologias que atendam à demanda por serviços digitais, mas que minimizem o impacto ambiental e promovam um futuro digital sustentável e energeticamente eficiente.

1.1 Problemática

Na era da digitalização, enfrenta-se o desafio de garantir a sustentabilidade ambiental da crescente infraestrutura digital. Com o uso da internet e o tráfego de dados em ascensão, as infraestruturas de *Big data* se destacam tanto pela importância quanto pelo elevado consumo de energia. A tendência é que este consumo aumente, e com isso surge a preocupação sobre as práticas atuais de gestão energética.

Neste contexto, o *Green Computing* surge como uma solução viável, ao propor a coexistência da computação com a sustentabilidade. Inovações como máquinas multi-core e a adoção de padrões como o OpenMP apontam para melhorias na eficiência energética e no desempenho. Contudo, equilibrar avanços tecnológicos com práticas sustentáveis apresenta desafios, especialmente na otimização de hardware que deve visar não apenas o desempenho, mas também a economia de energia.

A busca por soluções que aliam desempenho e sustentabilidade mobiliza diferentes setores, desde governos a indústrias, com o intuito de alcançar à eficiência energética e à inovação. O objetivo é atender à crescente demanda por serviços digitais de forma alinhada com metas de redução de emissões e sustentabilidade a longo prazo.

Neste contexto, compreender o funcionamento das tecnologias atuais e desenvolver estratégias para otimizá-las é essencial. A aplicação de técnicas de Aprendizado de Máquina desempenha um papel crucial nesse processo, já que oferece uma maneira de avançar na compreensão e na previsão do consumo de energia e do desempenho de aplicações em ambientes OpenMP em máquinas multi-core.

Prever o desempenho de uma aplicação ou seu consumo de energia é complexo devido à natureza frequentemente incompleta e não determinística das aplicações. No entanto, o Aprendizado de Máquina, com seu potencial para identificar relações complexas e não lineares nos dados, oferece uma solução promissora. Técnicas de Aprendizado de Máquina, como Regressão Linear, Support Vector Machine e Random Forest, podem ser aplicadas para melhorar a eficiência energética e o desempenho preditivo desses modelos.

Portanto, o *Green Computing*, juntamente com a aplicação estratégica de técnicas de Aprendizado de Máquina, representa um caminho para alcançar uma era digital mais verde e sustentável.

1.2 Objetivos do Estudo

1.2.1 Objetivo Geral

Este estudo visa compreender o comportamento de *hardwares* ao executar aplicações OpenMP em cenários de alta demanda computacional, com um foco especial na execução de *benchmarks*[1] do PolyBench/ACC. Além disso, busca-se abordar métodos e modelos que aprimoram a eficiência dessas aplicações, com vistas a criar um ambiente computacional mais sustentável.

O objetivo é promover práticas de computação mais sustentáveis e economicamente viáveis em termos de eficiência energética. Isso envolve a análise de padrões de consumo, bem como a coleta e avaliação de dados sobre o uso de energia. Através de técnicas de Aprendizado de Máquina, com modelos de regressão, busca-se desenvolver abordagens que não apenas otimizem o desempenho, mas também prevejam a eficiência energética das aplicações, e contribuir com a tomada de decisão em escalonadores de tarefas de computação de alto desempenho.

[1] É um processo de avaliação e medição do desempenho de hardware, software ou sistemas

1.2.2 Objetivos Específicos

Neste estudo, um olhar atento será voltado para os dados de desempenho e ao consumo energético das aplicações, a fim de compreender suas dinâmicas e características intrínsecas.

Os objetivos específicos seguintes estão direcionados a desvendar, por meio da análise de dados, pontos valiosos sobre como essas aplicações operam e consomem energia. Ao dissecar essas informações, a intenção é navegar através dos detalhes do funcionamento e demandas destas aplicações, com o intuito de explorar possíveis melhorias de eficiência e predições de gastos em execuções futuras.

Tendo em vista os preceitos estabelecidos e as informações preliminares exploradas, os objetivos específicos que serão perseguidos ao longo desta pesquisa englobam:

- **Coletar Dados Relacionados ao Consumo Energético:**

  A meta é realizar uma coleta de dados sistemática e precisa, com a finalidade de representar de forma autêntica o consumo de energia dessas aplicações, o que proporciona, dessa forma, uma fundamentação para as futuras análises.

- **Analisar do Comportamento das Aplicações:**

  Busca-se decifrar o modus operandi das aplicações OpenMP, com especial atenção às suas demandas e comportamento em relação à *CPU* e memória *RAM* durante execução de *benchmarks* do PolyBench.

- **Desenvolver Modelos de Predição de Consumo:**

  Este objetivo almeja aplicar técnicas de *Machine Learning* na criação de modelos preditivos que, amparados pelos dados coletados, possam antever o consumo energético futuro de maneira precisa.

- **Avaliar a Eficiência Energética das Aplicações:**

  Pondera-se realizar uma análise detalhada da eficiência energética das aplicações OpenMP na plataforma PolyBench, com objetivo de localizar e, posteriormente, explorar possibilidades de otimização energética.

1.3 Organização do Trabalho

A elaboração da monografia proposta, é estruturada de maneira sistemática, com objetivo de conduzir uma investigação detalhada e organizada sobre o tema em questão. Esta estrutura será desenvolvida para fornecer uma análise que seja ampla e informativa. A seguir, apresenta-se uma descrição da estrutura usada.

Capítulo 1: Destaca a problemática fundamental e os objetivos centrais da pesquisa, comentando sobre o básico das aplicações OpenMP, bem como em seu impacto energético e nos esforços para garantir a prática do *Green Computing*.

Capítulo 2: Proporciona uma fundamentação teórica para o estudo, e explora os temas e ferramentas que serão usados no desenvolvimento do trabalho. Serve como um alicerce, e explica de forma completa tudo o que foi usado no desenvolvimento da pesquisa.

Capítulo 3: Explora trabalhos que serviram como guia para a condução desta pesquisa, ao possibilitar a descoberta de abordagens para a elaboração deste trabalho.

Capítulo 4: Explora a metodologia do trabalho, de uma maneira detalhada, e proporciona uma visão dos métodos de pesquisa, estratégias de análise de dados e predições. Descreve os critérios para a mineração de dados de consumo e características das aplicações, as táticas para desenvolver os modelos de predição e as métricas de avaliação de desempenho usadas.

Capítulo 5: Discorre sobre os resultados obtidos e entra em uma discussão sobre suas implicações, desafios, e relevância. Destaca a aplicação e a eficácia dos modelos de predição de consumo desenvolvidos e avalia a eficiência energética das aplicações em testes. A seção também delibera sobre os conhecimentos obtidos na análise.

Capítulo 6: Encerra a monografia, com resumo das descobertas principais, suas implicações e os conhecimentos gerados pela pesquisa. Também são propostas sugestões para trabalhos futuros, com indicação de áreas que poderiam beneficiar-se de uma exploração mais aprofundada
.

Por fim, as referências listam todas as fontes acadêmicas, artigos e materiais que foram utilizados para fundamentar a pesquisa e discussões ao longo da monografia. Oferecer reconhecimento, fornece a possibilidade que leitores explorem as fontes originais para fins de pesquisa futura.

2 REFERENCIAL TEÓRICO

2.1 Green Computing

A Computação Verde, ou '*Green Computing*', refere-se ao uso eficiente e ambientalmente responsável de computadores e seus recursos. Esta abordagem abrange várias práticas e estratégias.

Com o aumento da disseminação dos meios computacionais, houve também a necessidade do aumento do poder computacional, o que por consequência aumentou também o consumo de energia e, por consequência, a emissão de dióxido de carbono (CO_2) no meio ambiente.

Os problemas ambientais gerados pelo aumento da emissão de CO_2 e o custo financeiro ocasionado pelo consumo de energia têm impulsionado estudos que visam o desenvolvimento de mecanismos e tecnologias que façam uso eficiente da energia. Esses mecanismos e tecnologias são denominados computação verde ou *Green Computing* (WILLIAMS; CURTIS, 2008).

Estes mecanismos abrangem várias práticas e estratégias, como:

- **Eficiência Energética**: Busca reduzir o consumo de energia dos dispositivos de computação, seja por meio de *hardware* mais eficiente ou de *software* otimizado.
- ***Design* Sustentável**: Criar produtos com menor impacto ambiental, como a utilização de materiais recicláveis ou biodegradáveis e a minimização de substâncias perigosas.
- **Gestão de Resíduos**: Disposição apropriada de equipamentos eletrônicos antigos e reciclagem de componentes.
- **Virtualização**: Utilizar tecnologias de virtualização para reduzir o número de servidores físicos necessários, assim gerando uma economia de energia e espaço.
- **Centros de Dados Verdes**: Projetar centros de dados para maximizar a eficiência energética, o que abrange o uso de refrigeração eficiente e natural e fontes de energia renováveis.
- ***Cloud Computing***: Utilizar serviços de nuvem para reduzir a necessidade de *hardware* local e melhorar a eficiência energética através de *data centers* centralizados e otimizados.

2.2 Programação Paralela

O desenvolvimento de modelos de Computação Paralela tem por objetivo oferecer APIs (Application Programming Interface) e demais ferramentas computacionais de modo a padronizar e simplificar a utilização dos recursos do *hardware* para processamento paralelo (KIRK; WEN-MEI, 2013).

2.3 OpenMP

A API OpenMP é usada para programação *Multi-thread* em ambientes de memoria compartilhados para plataformas UNIX e Microsoft NT (PENHA; CORRÊA; MARTINS, 2002). Sua evolução desde 1997 reflete a crescente necessidade de abstrair a complexidade do paralelismo em *hardware* diversificado. Por meio de um modelo de programação *Fork-join*, o OpenMP permite que um *thread* mestre se divida em múltiplos *threads* para executar tarefas simultaneamente, com a posterior reunião para sincronização. O uso de diretivas de compilação, que são linhas de código interpretadas pelo compilador, é uma característica distintiva do OpenMP, o que faculta aos programadores paralelizar aplicações com facilidade, ao adicionar simples anotações ao código.

Entre as vantagens desta API, pode-se destacar a padronização e a portabilidade (PENHA; CORRÊA; MARTINS, 2002). A portabilidade é o benefício, que suporta uma variedade de plataformas e arquiteturas de *hardware*, ao facilitar a execução de programas em diferentes sistemas sem necessidade de reescrever o código. No entanto, apesar dessas vantagens, o OpenMP enfrenta desafios, como limitações em escalar para um número alto de núcleos de processamento e a complexidade inerente ao manter a eficiência em sistemas de grande escala.

2.4 PolyBench

O PolyBench é uma suíte de *benchmarks*, desenvolvido pela polyhedral community (YUKI, 2014) . Essencial na otimização de compiladores e na avaliação de sistemas de computação de alto desempenho. Ele oferece uma variedade de *kernels* de computação e programas que simulam comportamentos em domínios como álgebra linear e dinâmica de fluidos computacional. Esses *benchmarks* são cruciais para testar novas técnicas de otimização de compiladores, arquiteturas de *hardware* e sistemas de gerenciamento de memória.

O *benchmarking* com o PolyBench permite a medição do impacto de otimizações em desempenho e consumo de energia. Isso é realizado por meio de métricas quantitativas que avaliam o desempenho sob diferentes cargas de trabalho, o que viabiliza análises, como a eficácia de um novo algoritmo de otimização ou como uma arquitetura de CPU gerencia tarefas paralelas.

2.5 Jupyter

Jupyter é o sistema mais usado para programação literária interativa (SHEN, 2014). O uso de *Jupyter Notebooks* para publicar pesquisas reprodutíveis é defendido por combinar texto de relatório com código executável (KLUYVER et al., 2016). Apesar do formato em si não garantir a reprodutibilidade (PIMENTEL et al., 2019), a documentação de processos com resultados pode ser bem valiosa para esse propósito, e seu formato possibilita e facilita a análise exploratória de dados (PERKEL, 2018).

A interatividade do *Jupyter* permite que este paradigma seja usado em tempo real para análises de dados, com o processo sendo documentado durante o desenvolvimento, resultados sendo exibidos de forma instantânea e discutidos imediatamente em linguagem natural (PIMENTEL et al., 2021).

2.6 Aprendizado de Máquina

Aprendizado de Máquina (AM) é uma área de IA (Inteligência Artificial) cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática (MONARD; BARANAUSKAS, 2003).

Utiliza algoritmos para identificar padrões e capacitar computadores a executar tarefas. Diante de desafios como a necessidade de grandes conjuntos de dados precisos e questões de viés e ética, o Aprendizado de Máquina cresce em prevalênciae impulsiona inovações em diversos setores.

Os algoritmos de aprendizado de máquina se dividem em supervisionado, não-supervisionado e por reforço, como mostra a Figura1 .

Figura 1 – Diagrama dos tipos de aprendizado em machine learning.

Fonte:(ALMEIDA; CARVALHO; MENINO, 2017)

Na primeira, os algoritmos ajustam parâmetros de um modelo a partir do erro medido entre respostas obtidas e esperadas. Na segunda, os parâmetros de um modelo são ajustados com

base na maximização de medidas de qualidade das respostas obtidas. A terceira é caracterizada pelo uso de algoritmos híbridos, que fazem uso dos recursos de correção de erro e de maximização de medidas de qualidade, conforme necessário (BRUNIALTI et al., 2015).

2.7 Correlação de Pearson

Para medir a interação entre as aplicações, calcula-se o coeficiente de correlação entre pares. A correlação entre duas características resulta em um grau de relação entre elas. Esta relação pode ser linear ou não linear. Para esta pesquisa, foi utilizada a correlação linear, que calcula o coeficiente de correlação de Pearson (ρ).

A correlação de Pearson, é uma medida de associação bivariada do grau de relacionamento entre duas variáveis (GARSON, 2013). Para (MOORE, 1996), "A correlação mensura a direção e o grau da relação linear entre duas variáveis quantitativas" . Em uma frase: o coeficiente de correlação de Pearson (r) é uma medida de associação linear entre variáveis (FILHO; JÚNIOR, 2009).

Dito isto, entende-se que a correlação de Pearson (r) é uma medida estatística usada para avaliar a relação entre duas variáveis. Quando é feita a afirmação que duas variáveis estão associadas, significa que elas têm semelhanças na distribuição de seus valores. Em termos de correlação de Pearson, essa associação é quantificada através da variância compartilhada entre as duas variáveis.

Este tipo de correlação assume um modelo linear, o que implica que mudanças em uma variável estão linearmente relacionadas a mudanças na outra. Por exemplo, um aumento em uma unidade na variável X poderia estar relacionado a um aumento ou diminuição proporcional na variável Y. Sua fórmula é a seguinte:

$$r = \frac{1}{n-1} \sum \left(\frac{x_i - \overline{x}}{s_x} \right) \left(\frac{y_i - \overline{y}}{s_y} \right) \tag{2.1}$$

O coeficiente de correlação Pearson (r) varia de -1 a 1. O sinal indica direção positiva ou negativa do relacionamento e o valor sugere a força da relação entre as variáveis. Uma correlação perfeita (-1 ou 1) indica que o escore de uma variável pode ser determinado exatamente ao se saber o escore da outra. No outro oposto, uma correlação de valor zero indica que não há relação linear entre as variáveis (FILHO; JÚNIOR, 2009).

Para (COHEN, 2013), valores entre 0,10 e 0,29 podem ser considerados pequenos; escores entre 0,30 e 0,49 podem ser considerados como médios; e valores entre 0,50 e 1 podem ser interpretados como grandes.

Em resumo, a correlação de Pearson (r) varia de -1 a +1, onde valores próximos a 1, independente do sinal, indicam uma forte relação linear entre as variáveis, e valores próximos a

0 indicam uma fraca relação linear.

2.8 Modelos de Regressão

Neste estudo, o foco dos modelos de regressão é desenvolver uma equação que estabeleça de forma precisa a relação entre um preditor x, e o consumo energético y, que é a variável dependente, com o uso dos dados coletados. Esta seção fornece a base teórica sobre os métodos de regressão adotados, e oferece uma ilustração de como esses métodos são aplicados na prática, por meio do emprego de ferramentas computacionais, para predizer o consumo de energia.

2.8.1 Regressão Linear

Os modelos de regressão são fundamentais para entender as dinâmicas entre variáveis, permitindo classificar e quantificar suas relações tanto qualitativa quanto quantitativamente. Eles são cruciais para analisar um espaço amostral e, potencialmente, prever ocorrências futuras em situações ainda não observadas. Em particular, o modelo de Regressão Linear simples é um método preditivo elementar que relaciona uma variável dependente Y com uma variável independente X, por meio da representação de uma reta (CHEIN, 2019). Embora possa indicar uma relação de causa e efeito, é importante notar que a Regressão Linear simples por si só não prova a causalidade.

Para um conjunto de dados com n observações, temos n equações como a Equação 2.2:

$$Y_i = \beta_0 + \beta_1 X_i + E_i \tag{2.2}$$

Aqui, E_i representa os termos de erro aleatórios e independentes com média zero. Os parâmetros desejados são β_0 e β_1, onde β_0 é o coeficiente linear, que indica onde a linha intercepta o eixo Y, e β_1 é o coeficiente angular, que é a tangente do ângulo que a reta faz com o eixo X representa os termos de erro aleatórios e independentes (BASTOS; GUIMARÃES; SEVERO, 2015).

Para cada valor de *Xi*, *Yi* é uma variável aleatória cuja média μ_{Y_i} é igual a $\beta_0 + \beta_1 X_i$. Assim, a Equação 2.3 pode ser reescrita como:

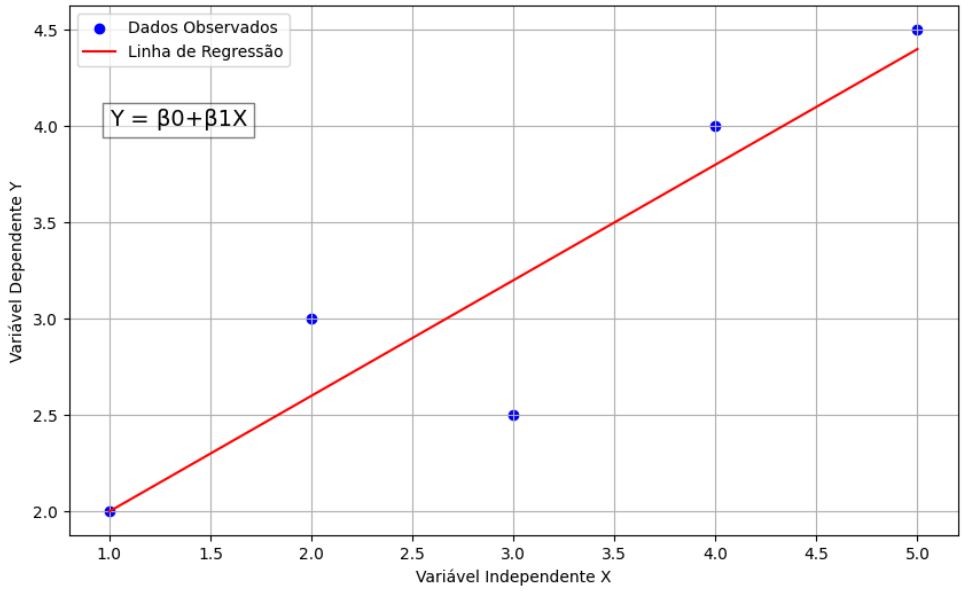
$$Y_i = \mu_{Y_i} + E_i \tag{2.3}$$

Após calcular as médias e estimar os parâmetros, obtém-se à forma final da equação de Regressão Linear simples:

$$Y = \alpha + \beta X \tag{2.4}$$

Aqui está a representação visual com um gráfico que demonstra um exemplo de uma Regressão Linear simples com dados hipotéticos:

Figura 2 – Exemplo de uma Regressão Linear simples.



Fonte: Autoria própria.

2.8.2 Regressão Polinomial

A Regressão Polinomial é uma forma de Regressão Linear onde a relação entre a variável independente X e a variável dependente Y é modelada como um polinômio de grau n. Diferente do modelo linear simples, que só pode representar relações lineares, a Regressão Polinomial é capaz de capturar relações mais complexas e curvilíneas entre as variáveis.

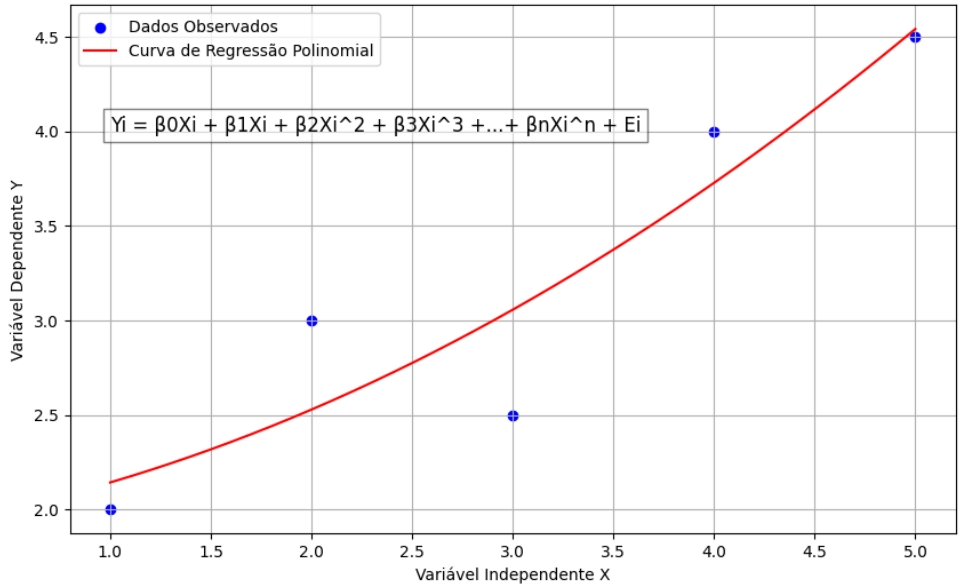
Na Regressão Polinomial, a equação geral é representada por:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_n X_i^n + E_i \tag{2.5}$$

No modelo de Regressão Polinomial, β_0 continua a ser o intercepto com o eixo Y, mas agora temos múltiplos coeficientes $(\beta_1, \beta_2, \ldots, \beta_n)$, que representam os efeitos de X elevado a diferentes potências. O termo E_i representa o erro aleatório com média zero. A inclusão de potências mais altas de X permite que o modelo ajuste curvas mais complexas aos dados, o que pode levar a um ajuste mais preciso.

Na figura 3, está a representação visual que demonstra um exemplo de uma Regressão Polinomial com dados hipotéticos:

Figura 3 – Exemplo de uma Regressão Polinomial.



Fonte: Autoria própria.

2.9 Métricas de Performance

Para garantir a validade dos modelos preditivos de consumo energético desenvolvidos neste estudo, é crucial aplicar métodos matemáticos que quantifiquem a acurácia com que os modelos de regressão se alinham aos dados observados.

Dessa forma, o indicador estatístico utilizado para avaliar a performance dos modelos é o Erro Percentual Médio Absoluto (MAPE). Esta é métrica detalhada e discutida na seção 2.9.1, oferecendo uma base sólida para aferir a confiabilidade das previsões de consumo energético realizadas pelos modelos propostos.

2.9.1 Erro Percentual Médio Absoluto (MAPE)

O MAPE (*Mean Absolute Percentage Error*), ou Erro Percentual Médio Absoluto, é uma métrica usada para medir a precisão de previsões estatísticas e modelagem preditiva. Basicamente, calcula a média dos erros percentuais absolutos entre os valores previstos e os valores reais, o que o torna útil para entender o quão precisas são as previsões em termos relativos.

A fórmula do MAPE é expressa como a média dos valores absolutos da diferença entre o valor previsto e o valor real, dividida pelo valor real, tudo isso multiplicado por 100 para obter uma porcentagem. Matematicamente, é representado como:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{A_i - P_i}{A_i} \right| \tag{2.6}$$

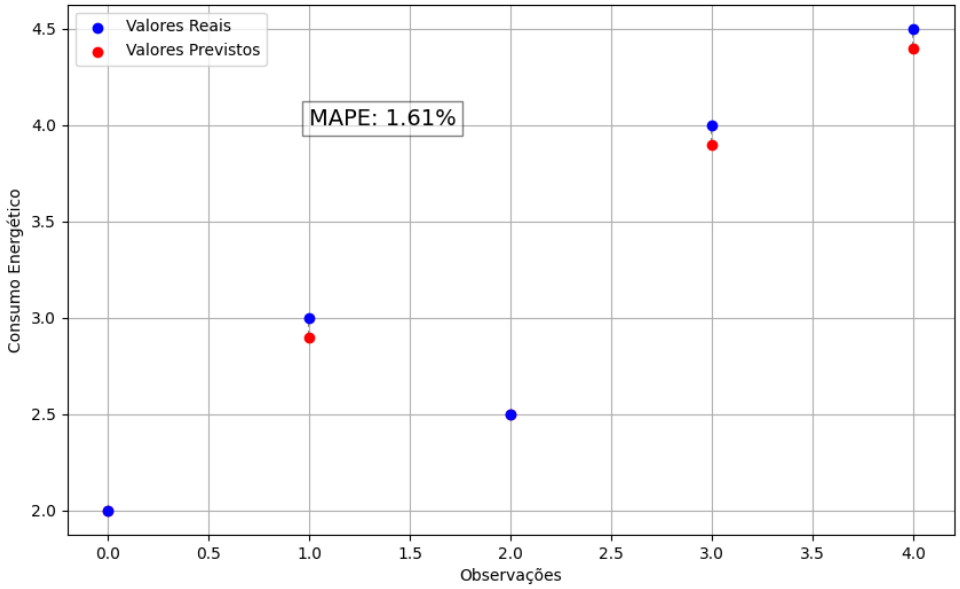
Nesta equação, A_i é o valor real, P_i é o valor previsto e n é o número de pontos de dados.

A ideia é calcular a diferença percentual para cada ponto de dado individual e, em seguida, tirar a média dessas diferenças para obter o resultado de precisão.

O MAPE é particularmente útil porque fornece uma medida de erro em termos percentuais, o que o torna independente da escala do dado. Isso é especialmente valioso em situações em que há a comparação entre precisão de previsões em diferentes escalas de dados.

Considere um exemplo simples onde temos dados reais e previsões para cinco pontos no tempo:

Figura 4 – Visualização do MAPE.



Fonte: Autoria própria.

A interpretação dos valores de MAPE geralmente segue uma escala em que números menores indicam previsões mais precisas (LEWIS, 1982). A tabela a seguir fornece uma orientação para interpretar os valores típicos de MAPE:

Tabela 1 – Interpretação dos valores de MAPE.

MAPE	Interpretação
<10	Previsão altamente precisa
10-20	Previsão boa
20-50	Previsão razoável
>50	Previsão imprecisa

- **<10**: Indica uma previsão altamente precisa. Valores de MAPE abaixo de 10% são considerados de alta precisão, o que significa que o modelo de previsão está muito alinhado com os valores reais;
- **10-20**: Representa uma boa previsão. Valores de MAPE nessa faixa indicam que o modelo tem um bom desempenho, com um nível de erro aceitável;

- **20-50**: É interpretado como uma previsão razoável. Nesse intervalo, o modelo ainda é útil, mas os erros são mais significativos, o que pode exigir aprimoramentos no modelo;
- **>50**: Significa uma previsão imprecisa. Um MAPE acima de 50% indica que as previsões estão muito distantes dos valores reais, e o modelo pode não ser adequado para a tomada de decisão.

O MAPE é uma ferramenta estatística valiosa e quando interpretado corretamente, pode fornecer validações significativas sobre a confiabilidade de um modelo preditivo. É importante notar que enquanto o MAPE oferece uma visão clara e intuitiva do desempenho do modelo em termos percentuais, ele deve ser utilizado em conjunto com outras métricas e considerações do domínio específico para uma avaliação compreensiva.

2.10 Bash

Bash, uma abreviação de "*Bourne Again Shell*", é um interpretador de comandos de *software* livre e uma linguagem de *script*. É amplamente utilizado em sistemas operacionais baseados em Unix, como Linux e macOS.

Esses interpretadores são programas feitos para intermediar o usuário e seu sistema. O usuário digita um comando e o interpretador o executa no sistema. Pode-se dizer que o *shell* [1] é o intermediário entre o usuário e o *kernel* [2]. Diversas pessoas utilizam-se desta linguagem para facilitar a realização de inúmeras tarefas administrativas no Linux, ou até mesmo criar seus próprios programas (FRANK; SEIBT, 2008).

Além de executar comandos do sistema, o *shell* também tem seus próprios comandos, como IF, FOR e WHILE, e também possui variáveis e funções. Tudo isso para tornar um pouco mais "espertas"e flexíveis essas chamadas de comandos feitas pelo usuário. Como estas são as características de uma linguagem de programação, o *shell* é uma ferramenta muito poderosa para desenvolver scripts e programas rápidos para automatizar tarefas (FRANK; SEIBT, 2008).

Uma das principais vantagens dos *shell scripts*, como os escritos para o Bash, é que eles não precisam passar por um processo de compilação. Em vez disso, basta criar um arquivo de texto simples e adicionar comandos a ele. Isso facilita a escrita, a edição e a execução de *scripts*, que os torna ferramentas poderosas.

[1] O shell é uma interface de linha de comando que facilita a interação entre o usuário e o sistema operacional

[2] O kernel é o núcleo do sistema operacional responsável por gerenciar recursos do hardware, fornecer uma interface entre software e hardware.

3 REVISÃO LITERÁRIA

Alguns estudos e trabalhos que já exploraram temas semelhantes a este projeto e que nos ajudarão a entender melhor o contexto no qual este trabalho esta inseridos. Essas obras previamente publicadas oferecem um caminho sobre as diversas abordagens relacionadas ao foco da pesquisa e são apresentadas a seguir.

Princípios e Tendências em Green Cloud Computing (WESTPHALL; VILLARREAL, 2013), o artigo discute a importância crescente da computação em nuvem e da TI verde, com foco no conceito de "green cloud computing", que combina flexibilidade e eficiência. A pesquisa explora as tecnologias atuais que apoiam a computação em nuvem verde, com a referencia de contribuições recentes de vários autores e oferece uma visão geral que serve como base para futuras pesquisas técnico-científicas na área.

A Preliminary Study of the Impact of Software Engineering on GreenIT (NOUREDDINE et al., 2012), neste trabalho, os autores desenvolveram um *framework*[1] para monitoramento de energia em tempo real, projetado para reportar de maneira eficiente o consumo energético de processos do sistema. Foi utilizada a biblioteca POWERAPI, para medir o consumo de energia dos processos, como CPU e rede. Com um objetivo de apontar possíveis "vazamentos"de energia.

Evaluating execution time predictions on GPU kernels using an analytical model and machine learning techniques (AMARIS et al., 2023), discutem várias abordagens para predição de tempo de consumo de funções CUDA executadas em placas gráficas de propósito geral (GPU). Explicam que em modelo analítico, há aplicações regulares onde modelos analíticos podem ser usados para prever desempenho, e outras mais irregulares onde técnicas de aprendizado de máquina obtém melhores predições.

PolyBench: The First Benchmark for Polystores (KARIMOV; RABL; MARKL, 2018), o estudo em questão foca nos desafios da Inteligência de Negócios moderna, que lida com o processamento de dados em diversos domínios e paradigmas, como modelos relacionais, de fluxo e gráficos. Uma dificuldade chave é a limitação dos sistemas em lidar apenas com alguns modelos de dados. Como solução, os "Polystores"são propostos, para integrar diferentes sistemas de processamento de dados para manipulação eficiente de variados modelos de dados. No entanto, faltam padrões para avaliar o desempenho desses *Polystores*. Portanto, o principal objetivo do trabalho é desenvolver o primeiro *benchmark* para *Polystores*, projetado para testar uma ampla variedade de soluções.

PolyBench/Python: Benchmarking Python Environments with Polyhedral Optimizations (ABELLA-GONZALEZ et al., 2021), "PolyBench/Python"implementa 30 *kernels* do "PolyBench/C"em *Python*, com o objetivo do explorar e analisar o desempenho do *Python* em códigos numéricos regulares, especialmente ao compará-los com implementações nativas em

[1] É um conjunto organizado de ferramentas e convenções que simplifica o desenvolvimento de software

C. O trabalho destaca a conveniência e desafios do uso da linguagem para códigos numericamente intensivos, além de apresentar uma avaliação de desempenho do *benchmark* em diversos interpretadores da linguagem e estabelecer uma comparação crítica com sua versão em C, que evidenciou as limitações de desempenho da linguagem *Python*, para essas aplicações.

Data Center e Eficiência Energética (BRITO; MATAI; SANTOS, 2023), a pesquisa busca entender como a eficiência energética em *data centers* é avaliada e aprimorada, conforme explorado em 75 artigos da base de dados *ScienceDirect*. Dentre as principais estratégias identificadas para melhorar a eficiência energética e minimizar emissões de carbono, destacam-se: aprimoramento dos sistemas de ventilação e refrigeração, uso de barramentos CC supercondutores, implementação de algoritmos para reduzir o consumo de energia e gerenciar tarefas, e aplicação de modelos que contemplam dispositivos de TI e refrigeração. Também foram consideradas soluções como otimização da distribuição do fluxo de ar, integração de sistemas de resfriamento, aquecimento e energia, e a utilização de energias renováveis.

Plataforma para medir el consumo energético de algoritmos (URIBE, 2022), a plataforma "PowerTester"foi desenvolvida para medir e analisar o consumo energético e o desempenho de programas escritos em C++ por meio da tecnologia Intel RAPL [2] e da ferramenta de análise "*Perf*"[3] do Linux. Os dados coletados dos diferentes medidores são enviados a um servidor, organizados e disponibilizados para os usuários através de gráficos e arquivos CSV para análise online. A validação da plataforma foi realizada com sucesso mediante experimentos com voluntários e a usabilidade foi avaliada através da escala SUS, e identifica oportunidades para futuras melhorias e trabalhos.

Análise Comparativa do Uso de Multi-Thread e OpenMP Aplicados a Operações de Convolução de Imagem (PENHA; CORRÊA; MARTINS, 2002), este estudo realiza uma análise comparativa com foco em métodos de implementações paralelas em sistemas de memória compartilhada em operação de convolução de imagem, e utiliza os padrões WinThread e OpenMP. No que se trata de programabilidade e desempenho, foi observado que a implementação com uso do OpenMP ofereceu uma abordagem mais simplificada, devido aos seus métodos menos explícitos de programação paralela. Em matéria de desempenho, ambas as implementações paralelas demonstraram um aprimoramento significativo em comparação com suas contrapartes sequenciais.

Mean Absolute Percentage Error for regression models (MYTTENAERE et al., 2016) aborda as implicações do uso do Erro Percentual Médio Absoluto (MAPE) como medida de qualidade para modelos de regressão. O foco é na investigação da existência de um modelo MAPE ótimo e na demonstração da consistência universal da Minimização do Risco Empírico baseada no MAPE. Além disso, o estudo revela que encontrar o melhor modelo sob o MAPE

[2] Intel RAPL (Running Average Power Limit) é uma tecnologia que monitora e controla o consumo de energia em tempo real em processadores Intel

[3] É uma ferramenta de profiling (perfilamento) que permite a análise de desempenho de um sistema

equivale a realizar uma regressão ponderada pelo Erro Médio Absoluto (MAE), sendo essa estratégia de ponderação aplicada à regressão *kernel*.

A literatura existente sobre medidas de qualidade para modelos de regressão oferece uma variedade de abordagens, e a escolha da métrica apropriada pode ser crucial para avaliar o desempenho de modelos preditivos. Trabalhos relacionados nesta área podem ter explorado diferentes métricas de avaliação de modelos, suas propriedades teóricas e implicações práticas. Ao revisar esses trabalhos, é possível ganhar uma compreensão mais ampla das considerações teóricas e práticas associadas à escolha de métricas de qualidade, bem como comparar os resultados e conclusões alcançados por diferentes pesquisadores. Isso enriquecerá a discussão sobre a seleção e interpretação apropriadas de métricas de avaliação de modelos de regressão.

4 METODOLOGIA

Com a tecnologia atual, tem-se que a maioria das pessoas e empresas usufruem de alguma forma da internet. Essa informatização gera um volume de dados cada vez maior, que precisam ser armazenados de forma que possam também ser analisados. Por conta disso, a popularização dos Bancos de Dados deu origem ao termo "*Big Data*", o qual resume uma complexa gama de assuntos e metodologias para o tratamento e processamento de grandes volumes de dados (CALDAS; SILVA, 2016).

Este projeto empregou uma abordagem para criar um conjunto de dados, que visa analisar o consumo de energia em aplicações de Big Data com o uso da programação paralela. A metodologia adotada pode ser detalhada nas seguintes etapas:

4.1 Seleção de Ferramentas e *benchmarks*

4.1.1 Ferramenta de Hardware e *Software*

A Pesquisa, fez o uso de um único sistema para realizar os experimentos. As configurações do sistema incluíam um processador *Intel Core i5-7200U* com 2 núcleos físicos e 4 núcleos lógicos, uma frequência base de 2.50 GHz (que pode atingir até 3.10 GHz em modo turbo), e uma memória cache composta por 128 KB de cache L1, 512 KB de cache L2 e 3 MB de cache L3. Com capacidade de memória do sistema de 12 GB de memória DDR4 a 2133 MHz.

Além disso, o sistema estava equipado com gráficos integrados Intel® HD Graphics 620, com uma frequência de base gráfica de 300 MHz e uma máxima de 1.00 GHz.

A maquina trabalha com o sistema operacional *Linux Ubuntu* versão 22.4 e utilizaram o compilador *GCC* na versão 11.4.0.

4.1.2 Ferramenta de Monitoramento e coleta

A *Jouleit*, parte integrante da *PowerApi*, é uma ferramenta de monitoramento de consumo de energia, usada no projeto para medir o consumo de energia dos programas durante sua execução. O *Jouleit*[1] usa a tecnologia Intel *"Running Average Power Limit"* (RAPL) que monitora e registra o consumo de energia da CPU, memória RAM e GPU integrada. Esta tecnologia está disponível em CPUs Intel desde a geração Sandy (BELGAID; ROUVOY; SEINTURIER, 2020).

A ferramenta simplifica o processo de coleta de dados, além de torna-lo mais eficiente e menos propenso a erros. Devido a isso, a *Jouleit* é eficaz na captura de todas as métricas relevantes, e garante que os dados coletados sejam abrangentes.

[1] Um repositório de *scripts* que podem ser usados para monitorar o consumo de energia de programas

Um aspecto distintivo da *Jouleit* é sua habilidade de separar dados de consumo por componentes de hardware específicos, como CPU, DRAM e IGP [2]. Além de outras informações, como tempo de execução e possíveis erros de execução.

Além disso, a ferramenta também oferece recursos para o armazenamento de dados. A ferramenta é equipada com funcionalidades que permitem a criação de arquivos CSV. Esta funcionalidade abrange desde a geração de cabeçalhos até o registro dos dados de consumo, o que simplifica significativamente a organização e o armazenamento das informações coletadas.

4.1.3 Seleção do suite e dos *benchmarks*

O *PolyBenchACC* foi selecionado para a pesquisa devido à sua habilidade em maximizar a eficiência da paralelização em múltiplos núcleos de processamento. PolyBench-ACC deriva do conjunto de *benchmark* PolyBench/C e fornece implementações OpenMP, OpenACC, CUDA, OpenCL e HMPP (CHATARASI et al., 2017).

Um ponto de destaque é a incorporação do OpenMP, evidenciada pelas diretivas presentes nos códigos das aplicações. Em contraste, o PolyBench/C não possui estas diretivas do OpenMP, e sua execução é projetada de forma sequencial.

Dentre os 33 *benchmarks* disponíveis na plataforma, 12 foram escolhidos, e representam uma diversidade de funções que vão desde o cálculo de correlações e covariâncias até a Decomposição de Cholesky [3] e a solução de sistemas de equações lineares. Esta seleção foi feita com o objetivo de abranger varias aplicações, como multiplicação de matrizes, decomposição de matrizes e cálculos estatísticos.

Os *benchmarks* selecionados foram:

Tabela 2 – *benchmarks* selecionados e suas descrições

Benchmark	**Descrição**
2mm	Dupla multiplicações de matrizes sequenciais.
3mm	Tripla multiplicações de matrizes sequenciais.
Trmm	Multiplicação de uma matriz geral por uma matriz triangular
Symm	Multiplicação de matrizes simétricas.
Gemm	Multiplicação de matrizes gerais, e resultado escalada por fator.
Syr2k	Soma de duas multiplicações de matrizes, adiciona o resultado a uma matriz simétrica.
Syrk	Multiplica uma matriz por sua transposta e adiciona o resultado a uma matriz simétrica.
Gramschmidt	Método de Gram-Schmidt para ortonormalizar um conjunto de vetores.
Correlation	Cálculo da correlação entre conjuntos de dados.
Covariance	Cálculo da covariância entre conjuntos de dados.
LU	Decomposição LU de uma matriz.
Cholesky	Decomposição de Cholesky de uma matriz simétrica positiva definida.

Fonte: De autoria própria.

[2] *Integrated Graphics Processor* é uma placa gráfica integrada no CPU.

[3] Método matemático para fatorar uma matriz simétrica positiva em um produto de uma matriz triangular inferior e sua transposta.

A metodologia do uso desses *benchmarks* consiste na execução e coleta de dados de problemas densos de Álgebra Linear e *Data Mining*, que são frequentes em Computação Científica.

As entradas nos *benchmarks* são representadas por notações como *N*, *NxN*, até N^5, e servem como indicadores do tamanho do conjunto de dados (*Dataset*[4]) de cada problema específico. Essas notações são responsáveis por determinar a escala das operações computacionais a serem executadas. Cada *benchmark* possui suas próprias variações específicas de entradas, adequadas aos desafios e complexidades que ele visa explorar.

Por exemplo, no contexto de Álgebra Linear, uma entrada *NxN* pode representar a dimensão de uma matriz a ser manipulada. À medida que o valor de *N* aumenta, o tamanho do problema e, consequentemente, a carga computacional também aumenta. Essas entradas variáveis permitem uma análise mais abrangente do desempenho dos algoritmos e da eficiência dos sistemas em diferentes cenários de complexidade.

4.2 Criação de *Script*s de Coleta de Dados

A metodologia adotada para a coleta de dados envolveu a utilização de um *script* em Bash para automatizar a execução dos *benchmarks* selecionados. Cada *benchmark* foi executado 101 vezes, com uma variação na complexidade da execução para cobrir diversos cenários. As entradas variaram de 800 a 4000, com o aumento crescente em incrementos de 32 em 32. Essa abordagem garante a repetibilidade dos testes e possibilita uma análise de diferentes tamanhos de problemas.

O *script* compila e executa cada *benchmark* em sequência, com o ajuste gradual dos valores de entrada para aumentar a carga de trabalho conforme a capacidade do sistema. E em paralelo, executa a ferramenta *Jouleit*, para realizar o monitoramento e coleta das informações de consumo dos *benchmarks*. Com a ferramenta *Jouleit* os dados coletados foram organizados em arquivos CSV, categorizados com base em atributos como CORE, CPU, DRAM, DURATION, UNCORE e INPUT.

O algoritmo a seguir detalha, um exemplo do processo de coleta de dados do *benchmark* 3mm, que é executado por meio de um *script* em Bash:

[4] São conjuntos de dados organizados, geralmente em formato tabular, que fornecem informações específicas para análise estatística, machine learning ou pesquisa em diversas áreas.

Algoritmo 1: Coleta de Dados do *benchmark* 3mm

Entrada : intervalo de valores N
Saída : Arquivo CSV com os dados de consumo e tempo de execução

Dados: Valores N, *scripit Jouleit* para monitoramento de energia
Resultado: Arquivo output_3mm.csv com resultados

```
Mudar para o diretório do benchmark 3mm;
Executar Jouleit para obter o cabeçalho do CSV ;
Adiciona ao cabeçalho do CSV a coluna INPUT ;
Inicializar arquivo CSV com dados das leituras iniciais;
for cada valor de N no intervalo de 800 a 4000, com incremento de 32 em 32 do
    Compilar o benchmark 3mm com parâmetros de tamanho N^5;
    Executar Jouleit com o benchmark compilado;
    Armazenar e formatar os resultados de saída;
    Adicionar resultados ao arquivo CSV;
end
Mover o arquivo CSV para o diretório de dados;
Retornar ao diretório original;
```

A adição da coluna "INPUT"ao início do arquivo CSV representa os valores de entrada N. Dessa forma, é possível observar como o consumo de energia varia com o aumento da Complexidade Computacional, refletido pelo aumento progressivo do valor de N.

Além disso, a utilização do *script* em Bash, juntamente com o *Jouleit*, demonstrou uma integração eficaz de ferramentas de *software*, para a realização de experimentos em computação. Esta perspectiva prática e automatizada não só facilita a coleta de grandes volumes de dados, mas também assegura que sejam consistentes e confiáveis, o que é essencial para uma análise aprofundada das aplicações em um contexto de computação intensiva.

Nos conjuntos de dados criados, as métricas de consumo de energia em CORE, CPU, DRAM, UNCORE e PSYS foram registradas em microjoules e a métrica DURATION em microssegundos. Em sistemas de computação modernos, o consumo de energia pode ser muito baixo. Medir em microjoules permitiu capturar variações finas do consumo .

Para complementar a coleta, uma avaliação estática de código foi realizada nas aplicações. Esta análise teve como objetivo extrair informações individuais de cada *benchmark*, com base na estrutura e características do código. Dentre os dados obtidos nesta análise, estão:

- **INPUTS**: Representa a quantidade de valores de entrada N para cada *benchmark*. Este dado demonstra a carga de trabalho imposta à aplicação e como isso pode influenciar no consumo energético;
- **1D**: Indica o uso (ou não) de arrays de uma dimensão nas aplicações. Arrays são estruturas de dados comuns em operações de computação intensiva, e a maneira como são utilizados pode ter um impacto significativo;

- **ARRAYS**: Refere-se ao número de arrays utilizados no código. Este dado ajuda a avaliar a complexidade das coleções de dados manipulados e como isso pode afetar a eficiência energética;
- **LOOPS**: Quantidade de laços de repetição presentes no código. Loops são fundamentais em operações iterativas e seu uso eficiente é um fator chave na otimização do gasto energético;
- **MULT (Multiplicações)**, **SOM (Somas)**, **SUB (Subtrações)**, **DIV (Divisões)**: Representam a quantidade de operações matemáticas básicas no código. Estas métricas são indicativas da intensidade computacional das aplicações e proporcionam uma visão direta dos processos que mais demandam recursos de processamento, e consequentemente, energia.

Com isso, os *dataset* obtidos ficaram estruturados da como representados na imagem a seguir:

Figura 5 – Exemplo de Organização inicial dos *datasets*.

	CORE	CPU	DRAM	DURATION	PSYS	UNCORE	EXIT_CODE	INPUT	INPUTS	1D	ARRAYS	LOOPS	MULT	SUB	DIV	SOM
0	2054071	2339472	226379	188381	859251	0	0	800	1	0	2	8	5	0	2	2
1	2299371	2633355	277160	210400	984189	3052	0	832	1	0	2	8	5	0	2	2
2	2551080	2913200	314086	234566	1095457	0	0	864	1	0	2	8	5	0	2	2
3	2857109	3277885	366027	260317	1226620	0	0	896	1	0	2	8	5	0	2	2
4	3733327	4240772	421142	323573	1510617	1954	0	928	1	0	2	8	5	0	2	2

Fonte: Autoria própria.

4.3 Pré-processamento dos Dados

A qualidade dos dados é uma das principais preocupações em Aprendizado de Máquina (AM), onde os algoritmos são frequentemente utilizados para extrair conhecimento. Uma vez que a maioria dos algoritmos de Aprendizado de Máquina induz conhecimento a partir de dados, a qualidade do conhecimento extraído é amplamente determinada pela qualidade dos dados de entrada (BATISTA, 2003).

Neste projeto, o pré-processamento foi conduzido com a linguagem de programação *Python*, mais especificamente através do *software Jupyter Notebook*[5]. O primeiro passo nesse processo foi a limpeza e organização dos dados. A importância do pré-processamento na categorização de documentos é significativa, pois este influencia diretamente nos resultados dos classificadores (ALMEIDA; LEITE; RAMOS, 2019). A limpeza de dados envolve a identificação

[5] *Jupyter Notebooks* são plataformas de código aberto que integram código, visualizações e documentação em um ambiente interativo.

e correção de erros comuns, tais como dados faltantes, incorretos, ou inconsistências na grafia. Este processo assegura que os dados que serão analisados são precisos e confiáveis.

O processo também envolveu uma verificação de erros de execução, como indicado na análise da coluna 'EXIT_CODE'. Em conjuntos de dados relacionados a processos computacionais ou experimentos, a presença de códigos de saída específicos pode indicar falhas ou interrupções que podem impactar a integridade dos dados. Os erros dos *Datasets* foram identificados e corrigidos.

Posteriormente,os dados das métricas coletadas foram convertidas para joules e segundos, respectivamente, com objetivo de uma representação visual mais clara do sistema. Esta padronização foi idealizada para facilitar a análise e interpretação dos dados no contexto do Aprendizado de Máquina.

As métricas, EXIT_CODE (que indicava falhas na coleta), CORE(que indicava o uso apenas dos núcleos de processamento) e PSYS(que representa o consumo total da máquina), foram consideradas irrelevantes para a análise, e foram descartadas.

A métrica EXIT_CODE foi removida após o processamento de erros por ser desnecessária, enquanto a métrica CORE, que representa apenas uma parte do consumo da CPU, foi considerada redundante e, portanto, excluída. Além disso, a métrica PSYS foi descartada devido à inconsistência dos valores apresentados, que não correspondiam às expectativas para a métrica. Estas decisões foram tomadas para garantir a precisão e relevância dos dados coletados.

Além disso, foi criada uma nova métrica denominada 'ENERGY'. Esta coluna tem como objetivo representar o consumo total de energia das aplicações, sendo calculada pela soma dos valores dos elementos CPU, DRAM e UNCORE, de modo a propor assim, um indicador único e mais concreto do consumo de energia. Ao final dessa etapa, o *datasets* utilizados ficaram da seguinte forma:

Figura 6 – Exemplo de Organização final dos *datasets*.

	CPU	DRAM	DURATION	UNCORE	INPUT	INPUTS	1D	ARRAYS	LOOPS	MULT	SUB	DIV	SOM	ENERGY
0	2.34	0.23	0.19	0.0	800	1	0	2	8	5	0	2	2	2.57
1	2.63	0.28	0.21	0.0	832	1	0	2	8	5	0	2	2	2.91
2	2.91	0.31	0.23	0.0	864	1	0	2	8	5	0	2	2	3.22
3	3.28	0.37	0.26	0.0	896	1	0	2	8	5	0	2	2	3.65
4	4.24	0.42	0.32	0.0	928	1	0	2	8	5	0	2	2	4.66

Fonte: Autoria própria.

4.4 Mineração de dados

Para a manipulação dos dados e desenvolvimento dos códigos de Mineração de dados, adotou-se a linguagem *Python*, através do *software Jupyter Notebook*. No textitsoftware , o projeto foi elaborado com o uso de *Notebooks*, que facilitam a execução de segmentos de código em um único arquivo.

Além disso, houve o uso de três bibliotecas *Python*: *Pandas, NumPy e Matplotlib*. *Pandas*, será empregado para tarefas de manipulação e limpeza de dados, que incluem filtragem, agrupamento e transformações. *NumPy* será aplicado em operações numéricas e cálculos estatísticos, devido à sua eficiência em processamento matemático. *Matplotlib*, por sua vez, será usado para a criação de gráficos, essenciais na visualização de tendências, padrões e discrepâncias nos dados.

Ao explorar os dados obtidos, buscou-se entender o comportamento energético das aplicações em diferentes cenários e também identificar padrões e tendências que podem informar o desenvolvimento de práticas de computação mais eficientes e ecologicamente sustentáveis. Esta análise é uma forma de obter informações de gargalos nas aplicações, para que possam ser feitas melhorias no desempenho computacional.

Então, a mineração foi estruturada em 6 pontos principais, detalhados a seguir:

4.4.1 Verificação de *Outliers*

Identificar outliers, que são valores que se diferenciam significativamente da maioria dos dados, é necessário realizar uma inspeção dos dados fornecidos. Um *outlier* pode ser identificado como um ponto de dados que não segue a tendência geral ou o padrão estabelecido pelos outros valores. O analista precisa observar a existência desses valores juntamente com os dados como um todo, e identificar se estes precisam ser realmente retirados, pois afetam o modelo do Aprendizado de Máquina, ou se eles podem ser significativos para base.(FIORETO, 2023)

4.4.2 Consumo em Comparação com os Valores de Entradas

Esta análise explora a relação entre o tamanho dos conjuntos de dados de entrada e o consumo energético correspondente. Investigar essa relação permite identificar padrões e tendências que indicam ineficiências nas aplicações em diferentes escalas de dados, e oferece informações para melhorias futuras.

4.4.3 Consumo de Cada Componente de *Hardware*

O foco no consumo energético de componentes individuais de hardware, como CPU[6], memória RAM[7] e GPU[8] integrada. Analisar o consumo por componente é crucial para compreender como diferentes partes do sistema influenciam o consumo total de energia. Esta abordagem identifica componentes mais exigidos em diversas cargas de trabalho, com indicações de gargalos e oportunidades de otimização. Além disso, é fundamental para desenvolver estratégias de gerenciamento de energia eficientes e para o *design* de sistemas computacionais sustentáveis e ambientalmente corretos.

4.4.4 Consumo Total de Cada *Benchmark*

Ao avaliar o consumo total de energia para cada *benchmark*, pode ser obtida uma visão geral da eficiência energética das várias operações e processos computacionais. Esta análise é relevante ao comparar o desempenho energético entre diferentes *benchmarks* e entender como variações nas operações computacionais impactam no consumo de energia. Isso pode indicar *benchmarks* particularmente eficientes ou ineficientes, e direcionar futuras pesquisas e o aprimoramento de algoritmos e técnicas de programação.

4.4.5 Tempo Total de Execução de Cada *Benchmark*

O tempo de execução está diretamente relacionado à eficiência computacional e energética. Avaliar o tempo necessário para completar cada *benchmark* fornece uma perspectiva importante sobre a relação entre desempenho e consumo de energia.

Embora *benchmarks* mais longos possam sugerir maior consumo de energia, isso não se aplica necessariamente a sistemas otimizados para eficiência em tarefas prolongadas. Compreender essa relação é essencial para balancear a necessidade de desempenho com a sustentabilidade energética.

4.4.6 Mapas de calor de correlação com cluster hierárquico

O método de agrupamento hierárquico será utilizado para analisar a correlação de Pearson, abordada na seção 2.7, entre os *benchmarks* com base em seus consumos energéticos. Este método é uma técnica de análise de dados que permite identificar e organizar padrões dentro de um conjunto de dados, com a criação de grupos ou *clusters* [9] baseados em semelhanças.

[6] CPU (Unidade Central de Processamento) é o componente principal de um computador responsável pela execução de instruções, processamento de dados e controle das operações do sistema.

[7] RAM (Memória de Acesso Aleatório) é um tipo de memória utilizada por computadores para armazenar temporariamente dados e instruções em uso

[8] GPU (Unidade de Processamento Gráfico), é um componente especializado em processamento gráfico, acelerando o processamento de imagens e gráficos em computadores.

[9] É uma técnica de agrupamento de dados que organiza elementos similares em conjuntos distintos.

O agrupamento hierárquico funciona com a construção de uma hierarquia de *clusters*, onde inicialmente cada elemento (neste caso, cada *benchmark*) é considerado um *cluster* individual. À medida que o processo avança, os elementos mais similares são agrupados juntos, e formam clusters maiores. Esta semelhança é geralmente determinada por uma medida de distância, que neste estudo será baseada no consumo energético dos *benchmarks*.

Um aspecto importante do agrupamento hierárquico é o Dendrograma, que é um diagrama em forma de árvore, que demostra a ordem e a distância em que os elementos foram agrupados. Nesse contexto, o Dendrograma será útil para visualizar as relações entre diferentes *benchmarks* com base em seus perfis de consumo energético.

Ao aplicar o agrupamento hierárquico aos dados de consumo energético dos *benchmarks*, se torna possível identificar grupos que têm comportamentos similares . Isso pode revelar padrões sobre quais aplicações são mais eficientes energeticamente, quais consomem mais energia e como eles se relacionam entre si. Essa abordagem ajuda a compreender melhor o impacto do consumo das aplicações de HPC e fornece direções para melhorias.

4.5 Aprendizado de Máquina para Predição de Consumo energético

Com a base de dados já preparada e estruturada, foi possível executar a etapa de Modelagem. Foram aplicados 2 modelos de Aprendizado de Máquina, sendo eles Regressão Linear, Regressão Polinomial, citados nas seções 2.8.1 e 2.8.2.

Para cada modelo, foram adotados procedimentos padronizados que utilizou os métodos da biblioteca *Scikit-learn*.

Inicialmente, um objeto correspondente ao modelo escolhido é instanciado e, em seguida, treinado com os conjuntos de dados X_{train} e y_{train}, através do método `fit()`. Após o treinamento, emprega-se o modelo para realizar predições da variável dependente y a partir do conjunto X_{test}, por meio do método `predict()`.(FIORETO, 2023)

Nos modelos , foram utilizados os parâmetros padrão fornecidos pela biblioteca *Scikit-learn*. A única alteração realizada foi, no ajuste dos graus do polinômio conforme necessário.

- Regressão Linear:

```
grau = 1
poly_features = PolynomialFeatures(degree=grau)
X_train_poly = poly_features.fit_transform(X_train)
```

- Regressão Polinomial:

```
grau = N
poly_features = PolynomialFeatures(degree=grau)
X_train_poly = poly_features.fit_transform(X_train)
```

, onde N é o grau do polinômio desejado.

Além disso, o trabalho tem duas abordagens distintas dos dados para a predição do consumo, ambas centradas na aplicação de modelos de regressão, para prever o consumo energético.

Abordagem 1: Predição do *benchmark* por *Datasets* Próprios

- Modelos Utilizados: Modelos de Regressão Linear e polinomial;
- Dados: Dados de *datasets* específicos do *benchmark*;
- Treinamento: 60 a 80% dos dados do *dataset* para treinamento dos modelos, com graus de polinômio variando de 1 a 3;
- Previsão: Objetivo de prever os 10 últimos valores do *dataset*;
- Avaliação de Eficiência: Uso do MAPE (*Mean Absolute Percentage Error*) para comparar as previsões dos modelos com os valores reais e avaliar a eficiência;
- Objetivo: Comparar se a carga de dados utilizados na predição, impacta diretamente na eficiência dos modelos regressivos.

Abordagem 2: União de Múltiplos *Datasets*

- Modelos Utilizados: Regressão linear e Polinomial;
- Dados: Combinação de 11 dos 12 *datasets* disponíveis, com reserva do 12º para testes;
- Treinamento: Os 11 *datasets* concatenados foram utilizados para treinar o modelo;
- Teste: O conjunto de dados reservado foi utilizado para testar a eficácia do modelo;
- Avaliação de Eficiência: O MAPE também foi usado nesta abordagem para avaliar a precisão das previsões em relação aos dados reais do *dataset* de teste;
- Objetivo: Comparar se através de parâmetros compartilhados entre *benchmarks*, há a possibilidade de uma previsão eficiente com uso dos modelos regressivos.

Através desta abordagem, o objetivo é compreender o comportamento atual do consumo energético e prever tendências futuras. Além de avaliar a eficácia dos modelos por meio do MAPE (Erro Médio Percentual Absoluto), para garantir uma medida precisa do desempenho dos modelos.

5 RESULTADOS

Neste capítulo, irão ser abordados os resultados obtidos dos experimentos apresentados no capítulo anterior. Com a apresentação dos resultados obtidos, na Mineração de Dados e o uso de modelos de regressão para a predição de consumo energético.

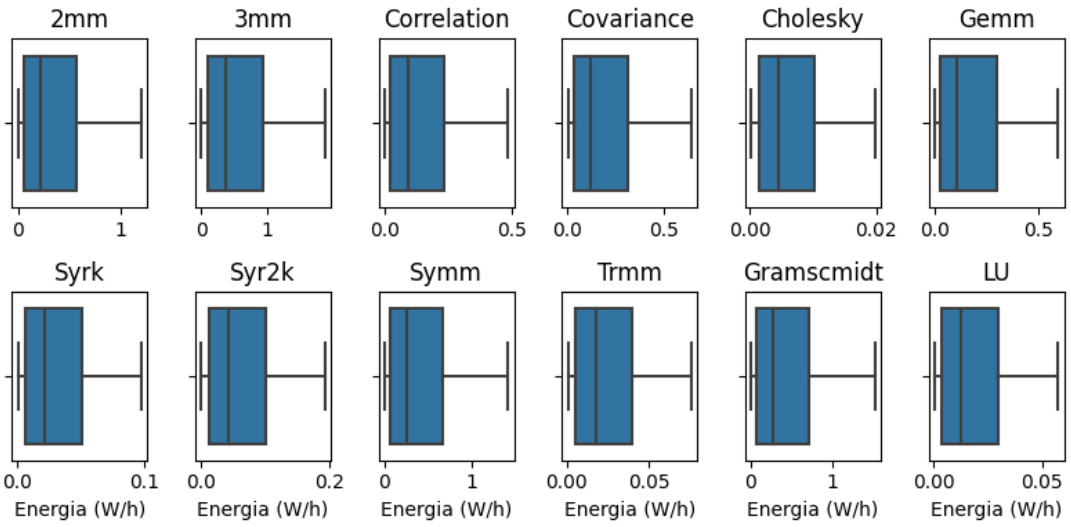
5.1 Resultados Mineração de Dados

5.1.1 Verificação de *Outliers*

Conforme abordado na seção 4.4.1, a verificação de *outliers* é essencial devido à possibilidade de existência de valores atípicos que divergem da tendência dos dados restantes. Esses valores anômalos podem surgir naturalmente, explicados por fatores que causam essa discrepância, ou artificialmente, devido a erros de medição ou coleta de dados incorreta. A identificação correta dos *outliers* é crucial, pois permite a análise precisa dos padrões de consumo de energia e o reconhecimento de pontos para otimização ou correção metodológica. O método adotado para identificar a presença de valores atípicos, foi o *Box Plot*, disponível na biblioteca de visualização *Seaborn*.

Esta ferramenta gráfica facilita a visualização dos quartis de cada variável e dos *outliers*, com isso proporciona uma análise comparativa intuitiva. O *Box Plot* é constituído pelas linhas extremas que indicam os valores máximo e mínimo, um retângulo que denota os quartis e a mediana representada pela linha central do retângulo. Os *outliers* são ilustrados por círculos ou losangos, a depender da plataforma de visualização utilizada.

Figura 7 – Análise de *outliers*.



Fonte: Autoria própria.

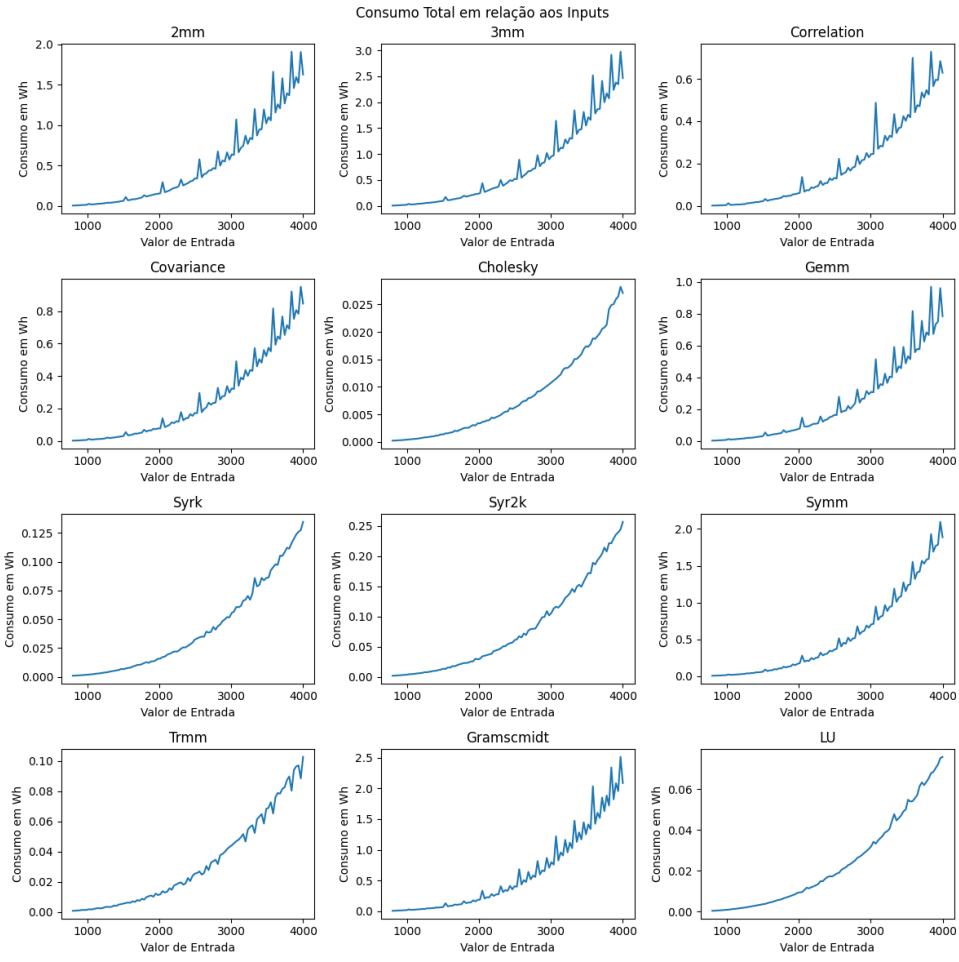
A análise dos *boxplot*, indica que os valores de consumo energético dos não apresentam

valores anormais. Portanto, não se faz necessário um tratamento específico para *outliers* nesses casos.

5.1.2 Consumo em Comparação com os Valores de Entradas

Como discutido na Seção 4.4.2, a análise investiga a correlação entre o consumo energético e o aumento da complexidade dos conjuntos de dados processados pelos *benchmarks*. Este estudo é relevante para identificar possíveis gargalos ou pontos de ineficiência ao longo das execuções. Utilizou-se o método *tightlayout* da biblioteca *Matplotlib* para a visualização de dados. Esta ferramenta ajusta automaticamente os parâmetros do *subplot*, portanto, otimiza a área da figura e minimiza sobreposições de elementos gráficos como títulos e rótulos dos eixos.

Figura 8 – Consumo x Valores de Entradas.



Fonte: Autoria própria.

Observa-se uma relação exponencial entre as variáveis, o que sugere que uma Regressão Linear pode não ser adequada para esta base de dados. Em contraste, uma Regressão Polinomial pode ser mais eficaz para previsões. Além disso, identificam-se pontos com maior consumo. Análises detalhadas revelam que estes consumos seguem um padrão específico.

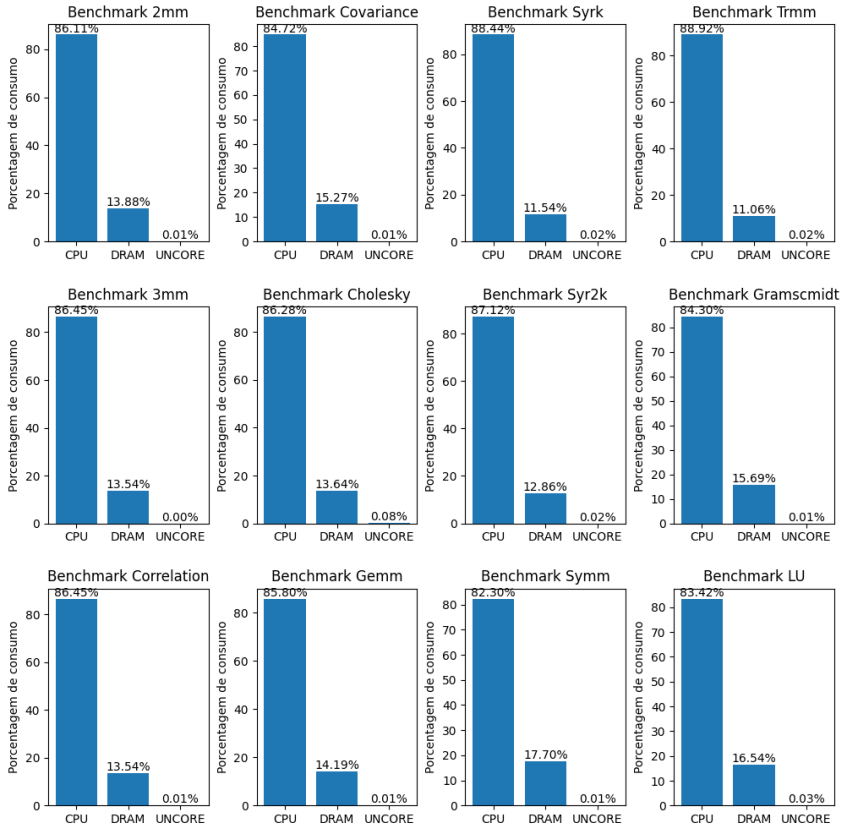
Entre os 12 *benchmarks* analisados, 7 exibem um padrão de consumo elevado para entradas divisíveis por 128. Para valores de entrada menos complexos, como 896 , 1024, o aumento do consumo é quase imperceptível. Contudo, com o aumento da complexidade, o impacto no consumo também cresce de forma exponencial. Uma hipótese para a causa desse fenômeno é a maneira como essas aplicações gerenciam a memória RAM.

Estruturas de dados maiores exigem mais memória, e um alinhamento de memória ineficiente pode aumentar o consumo de energia. A ocorrencia de, '*Cache misses*', onde dados precisam ser recuperados da RAM, eleva o consumo de energia. O *Thrashing* de memória, que ocorre quando há uma sobrecarga no sistema de gerenciamento de memória devido ao acesso excessivo e ineficiente à memória, também causa um consumo elevados. Estratégias de gerenciamento de energia variáveis da RAM podem interagir com padrões de entrada específicos,e afetar a eficiência energética.

5.1.3 Consumo de Cada Componente de Hardware

Na seção 4.4.3, explicou-se a abordagem de consumo por componente. Que tem como finalidade entender a influencia dos *hardwares* separadamente. Assim, sendo possível identificar componentes sob maior carga de trabalho, pontos relevantes e oportunidades de melhorias.

Figura 9 – Consumo por coponente de *hardware*.



Fonte: Autoria própria.

A figura 9, mostra o comparativo do consumo de energia dos Componentes, nomeadamente "CPU", "DRAM"e "UNCORE", constatou-se a CPU como principal consumidor de energia. Ao avaliar os *benchmarks*, revelou-se que a CPU foi responsável por um consumo que variou entre 82% e 88% do total de energia das aplicações analisadas.

No contexto desses resultados, o componente DRAM se posicionou com percentuais que oscilaram entre 11% a 17% do consumo total das aplicações. Enquanto a métrica "UNCORE", teve valores de consumo irrisórios, mas esse padrão já era esperado, pois não foi utilizado o processadores gráficos integrado ne execução dos *benchmarks*. Pois, poderia acarretar em um desgaste do hardware, um consumo elevado do processador e ineficiências de acesso a memoria, já que ambos são integrados e compartilham a memoria RAM da maquina.

A análise dos padrões de consumo energético dos componentes revela informações sobre a eficiência das aplicações de Big Data em ambientes heterogêneos. A predominância de alta energia na CPU enfatiza a possibilidade de otimizações para melhorar o desempenho e redução do consumo desse componente. Apesar da discrepância do consumo, esse comportamento também é algo esperado, já que a grande carga de trabalho fica de responsabilidade do processador. Mas isso abre precedentes para tentativas de otimização do uso do *hardware* ou até mesmo realizarem estudos de como o uso de processadores gráficos pode fazer essa carga de trabalho se dissipar, e se há a possibilidade da diminuição do consumo nesse componente.

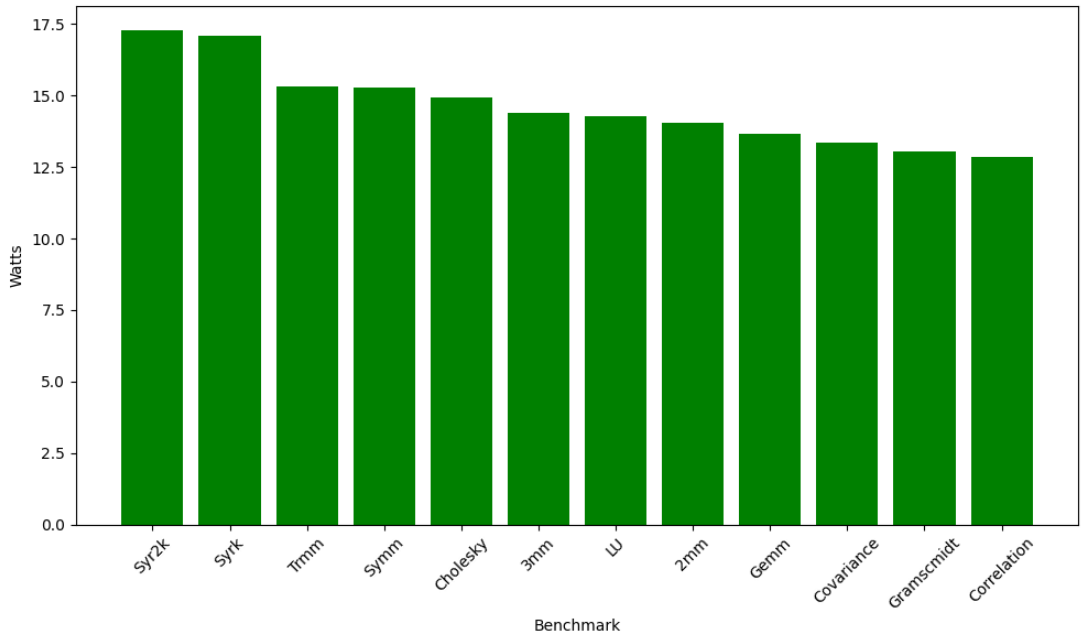
5.1.4 Consumo Total de Cada Benchmark

No tópico referenciado como 4.4.4, a observação foca no gasto energético total das aplicações para obter uma visão geral das execuções dos *benchmarks*. Esta análise é crucial para as discussões nas seções 5.1.5 e 5.1.6, onde é realizada uma investigação sobre a relação entre o consumo energético e o tempo de execução, bem como a correlação entre diferentes *benchmarks*. Esta abordagem fornecerá certos conhecimentos, sobre se o tempo de execução está diretamente relacionado ao consumo de energia e se *benchmarks* de tipos semelhantes apresentam padrões de consumo similares ou distintos.

No gráfico apresentado na figura 10, observa-se que os *benchmarks* com maior consumo energético incluem ***Syr2k*** e ***Syrk***, ambos relacionados a operações de Álgebra Linear. De acordo com a tabela 2, Syr2k e Syrk executam operações que envolvem multiplicação de matrizes e subsequente adição a uma matriz simétrica. A distinção entre os dois reside na quantidade de matrizes utilizadas na multiplicação. Portanto, devido às semelhanças nas tarefas que executam, pode-se deduzir que o consumo energético de Syr2k e Syrk seria semelhante.

Já os *benchmarks* que envolvem multiplicação de matrizes, ***Trmm, Symm, 3mm, 2mm, Gemm***, apresentaram resultados bastante próximos, com uma variação de 2w do maior para o menor consumo, essa variação deve ocorrer devido às diferenças entre matrizes e operações adicionais que diferem entre as aplicações. Para deduzir se há ou não espaço para melhorar a

Figura 10 – Consumo Total de Cada Benchmark



Fonte: Autoria própria.

eficiência entre os *benchmarks* envolvidos, se faz necessário uma análise mais estática de como cada um deles opera e executa.

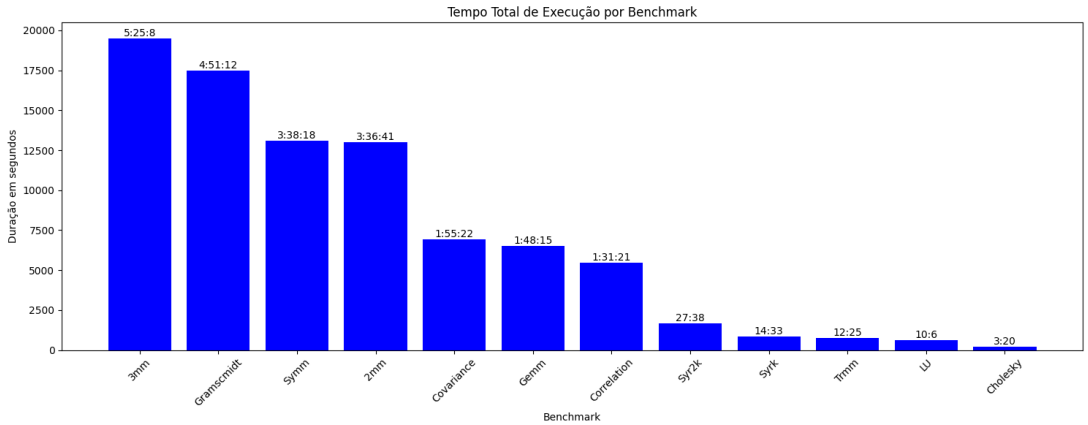
Cholesky e ***LU*** são *benchmarks* de decomposição de matrizes com consumos energéticos muito próximos, sendo aproximadamente 15 watts para Cholesky e 14,25 watts para LU. Apesar de serem aplicações do mesmo tipo, as diferenças nas execuções indicam a necessidade de uma análise mais profunda do código e da organização de cada uma dessas aplicações. Com base nas informações obtidas, não há a possibilidade de saber se a eficiência dos dois é a ideal , mas pode-se inferir que, devido à proximidade nos valores de consumo, a eficiência energética de ambos tem similaridades.

Os três *benchmarks* com o menor consumo energético são ***Covariance***, ***Gramschmidt*** e ***Correlation***, com consumos aproximados de 12.3 watts, 13 watts e 12.85 watts, respectivamente. Essas aplicações têm relevância em Álgebra Linear e Análise de Dados, mas servem a diferentes propósitos. Portanto, não é possível estabelecer uma relação direta entre suas execuções e a eficiência no consumo de energia. No entanto, uma análise estática do código poderia ser realizada para examinar as estruturas de dados usadas em cada um dos códigos, em busca por semelhanças entre eles.

5.1.5 Tempo Total de Execução de Cada Benchmark

Em conformidade com o que foi abordado na seção 4.4.5, o objetivo desse tópico, se refere a entender se, o tempo de execução está diretamente relacionado à eficiência computacional e energética. Pois embora *benchmarks* com maior duração possam indicar um consumo de energia mais elevado, isso não é necessariamente verdadeiro para sistemas que são otimizados para eficiência em tarefas de longa duração.

Figura 11 – Tempo Total de Execução de Cada Benchmark.



Fonte: Autoria própria.

O gráfico 11, demonstra exatamente o que foi dito no paragrafo anterior, o consumo nem sempre esta diretamente ligado ao tempo de duração da execução da aplicação.

O comportamento dos *benchmarks* ***Syrk***, ***Syr2k*** e ***Gramschmidt*** ilustra claramente essa observação. ***Syrk*** e ***Syr2k*** registraram os maiores consumos de energia, porém com tempos de execução relativamente curtos, de 14 minutos e 33 segundos, e 27 minutos e 38 segundos, respectivamente. Em contraste, ***Gramschmidt*** apresentou um consumo de energia mais baixo, mas com uma duração de execução significativamente maior, atingindo 4 horas e 51 minutos.

Esta observação também se aplica a outras aplicações, onde existe potencial para melhorar a eficiência em relação ao tempo. Isso é evidente no caso do *benchmark* ***2mm***, que registrou um consumo de energia próximo ao do *benchmark* ***3mm***, mas com uma diferença de quase 2 horas na duração. Com a otimização da distribuição de tarefas em relação ao tempo, é provável que o consumo de energia desse *benchmark* possam ser reduzidos.

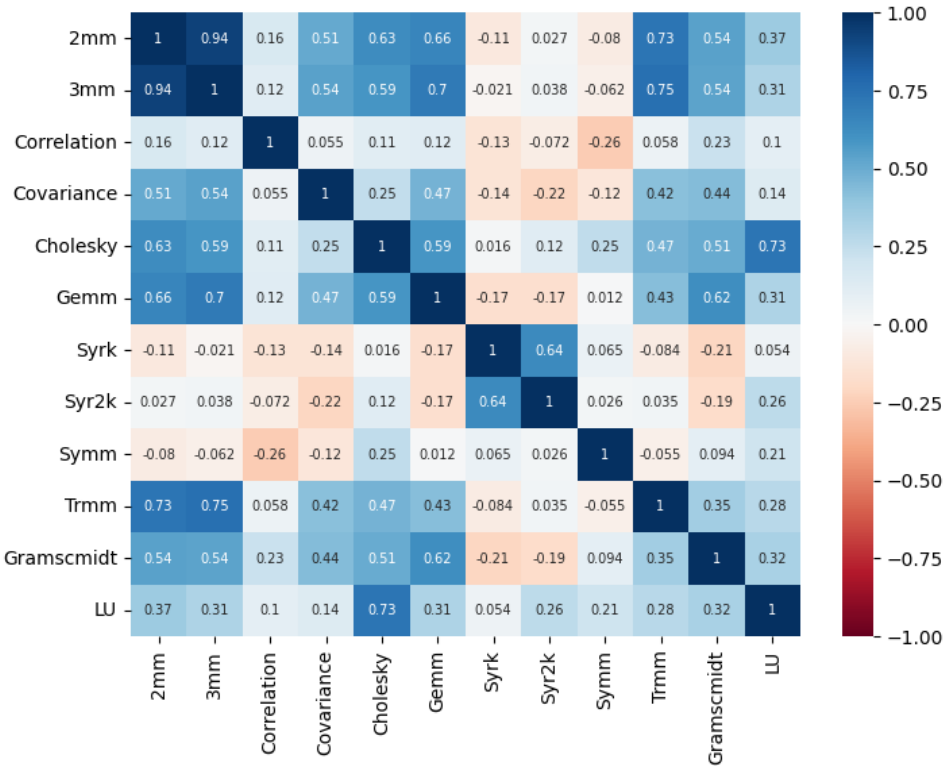
Portanto, evidencia-se que uma divisão mais eficiente de tarefas ao longo do tempo pode reduzir o consumo energético. Ao evitar a utilização excessiva dos recursos de *hardware* e distribuir a carga de trabalho de maneira mais equilibrada, mesmo com uma duração de execução mais longa, o sistema pode operar de forma mais eficiente, o que resulta em economia de energia.

5.1.6 Mapas de calor de correlação com cluster hierárquico

Conforme explicado anteriormente na seção 4.4.6, o objetivo da execução dessa visualização de dados é que, ao aplicar o Agrupamento Hierárquico aos dados de consumo energético, torna-se possível identificar grupos de aplicações com comportamentos similares. Esta similaridade é determinada com base em cada uma das execuções dos *benchmarks*, e não pelo consumo total, conforme foi analisado anteriormente na seção 5.1.4.

Parar realizar esse experimento foi usado a função *heatmap()*, da biblioteca *Seaborn*. Essa função, gera um mapa de calor, um gráfico que emprega cores para representar informações de maneira intuitiva. Valores próximos de 1 e -1 no *heatmap* indicam forte correlação entre as variáveis, enquanto valores perto de 0 sugerem uma baixa correlação. O resultado da correlação pode ser visto na Figura 12

Figura 12 – Mapas de calor de correlação.



Fonte: Autoria própria.

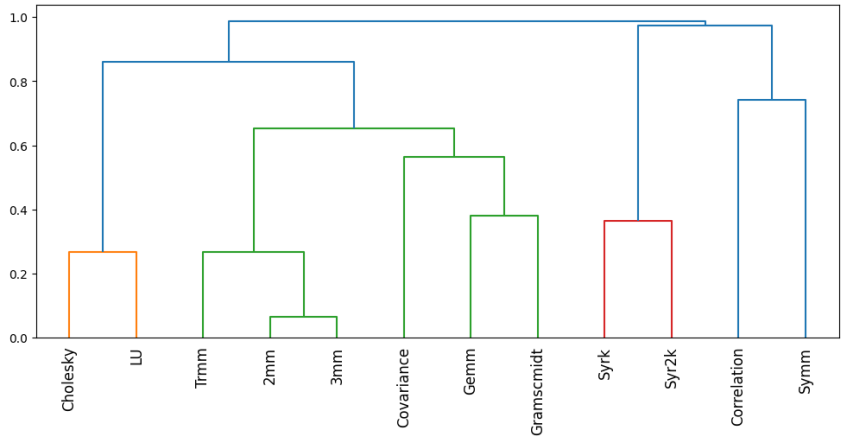
À primeira vista, são observadas algumas correlações positivas (em azul). No entanto, o mapa de calor pode apresentar certa complexidade visual devido à falta de uma organização ideal. Para otimizar a visualização, os recursos correlacionados serão agrupados por meio de *clustering* hierárquico.

O clustering hierárquico é um método empregado para estabelecer hierarquia nos dados, ao organiza-los em *clusters*. Nesse método, com o uso do Coeficiente de Correlação de Pearson, a matriz de distância é calculada de modo que correlações fortes (positivas ou negativas) resultem

em distâncias longe de zero, ao passo que a ausência de correlação resulta em uma distância aproximada de zero

Após o cálculo da matriz de distância, os dados são agrupados hierarquicamente com base em suas distâncias. Esse processo permite visualizar a relação entre os diferentes conjuntos de dados em um Dendograma, que é um diagrama em forma de árvore. Conforme pode ser visto na Figura 13.

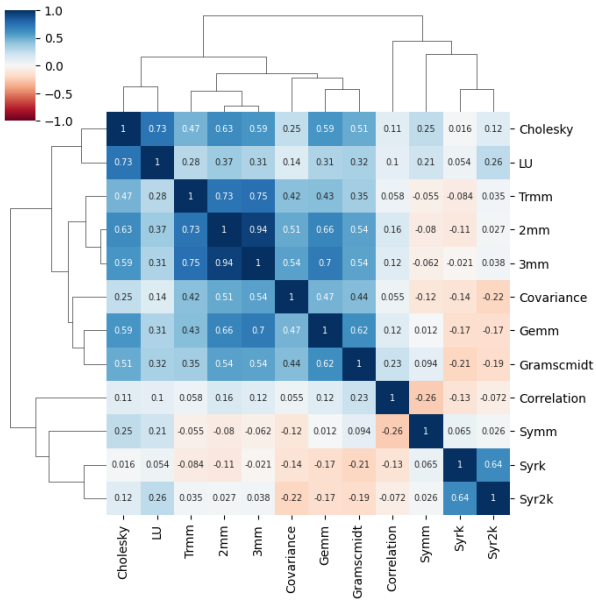
Figura 13 – Dendograma de Relação.



Fonte: Autoria própria.

O *Seaborn* oferece a função *clustermap()*, que permite traçar heatmaps de correlação juntamente com Dendogramas.

Figura 14 – Mapas de calor de correlação com cluster hierárquico.



Fonte: Autoria própria.

Com base no que é apresentado nas Figuras 13 e 14, podem-se observar algumas correlações interessantes, como a correlação entre os *benchmarks* ***Covariance***, ***Gemm***, e ***Gramscmidt***.

Outra relação que pode ser vista, é a entre os *benchmarks* ***Correlation*** e ***Symm***. O consumo energético representado neste *heatmap* está diretamente relacionado à correlação entre o gasto energético e a complexidade de cada execução dos *benchmarks*. Assim, o Dendograma ilustra a relação entre o consumo de energia e a crescente complexidade dos conjuntos de *benchmarks* executados, já que esses valores são iguais em todos.

A altura das linhas no Dendrograma reflete a dissimilaridade entre os *benchmarks*, onde linhas mais baixas indicam maior semelhança. Por exemplo, a maior distância entre ***Correlation*** e ***Symm*** sugere menor similaridade em comparação com a proximidade entre ***Syr2k*** e ***Syrk***. O Dendrograma também revela uma estrutura hierárquica, já que sugere que *benchmarks* agrupados em níveis mais altos podem compartilhar propriedades comuns.

benchmarks distintos como ***Cholesky*** e ***LU***, que aparecem isolados, podem ter características únicas ou uma correlação menor com os outros. A partir dessas observações, estratégias de otimização podem ser desenvolvidas, como a comparação de semelhantes para identificar e implementar reduções no consumo de energia, investigação dos picos de consumo para refinamento de eficiência e a aplicação de estratégias de otimização compartilhadas para *benchmarks* que exercem demandas similares no *hardware*.

Além disso, *benchmarks* que se destacam individualmente podem necessitar de abordagens de otimização dedicadas, enquanto que para aqueles agrupados, pode-se considerar o balanceamento de carga e a refatoração do código para melhorar a eficiência energética. Ajustes no *hardware* e configurações do sistema também podem ser generalizados para grupos de *benchmarks* com características semelhantes, com a otimização do consumo de energia em um espectro mais amplo. Essas medidas coletivas formam a base para um plano de otimização que visa aprimorar a eficiência energética de forma holística e específica.

5.2 Resultado Aprendizado de Máquina para Predição de Consumo energético

Conforme explicado na seção 4.5, a abordagem para esse tópico visa compreender o comportamento do consumo energético atual e prever tendências futuras. A eficácia dos modelos é avaliada com base no MAPE (Erro Médio Percentual Absoluto), o que assegura uma medida precisa do desempenho dos modelos.

A seguir, serão apresentados os resultados das abordagens utilizadas na predição do gasto energético, ambas centradas na aplicação de modelos de regressão para prever o consumo.

5.2.1 Abordagem 1: Predição do *Benchmark* por *Datasets* Próprios

A primeira abordagem foca na predição do desempenho de *benchmarks* com a utilização de seus próprios *datasets*. Para isso, aplicam-se modelos de Regressão Linear e Polinomial, que

aproveita dados específicos de cada *benchmark*. No processo de treinamento, são selecionados entre 50 a 80% dos dados para treinar os modelos, além do ajuste dos polinômios de graus que variam de 1 a 3. O objetivo principal é prever os 10 últimos valores do *dataset*.

Para avaliar a eficiência dessa abordagem, recorreu-se ao MAPE, que possibilitou a comparação das previsões geradas pelos modelos com os valores reais observados. Com isso, é possível medir quão próximas as previsões estão dos resultados reais e, consequentemente, avaliar a eficiência dos modelos.

O propósito central dessa metodologia é investigar a relação entre a quantidade de dados usados na predição e a eficiência dos modelos regressivos ao utilizar diferentes cargas de dados. Deseja-se compreender se o volume de dados utilizado tem um impacto direto na precisão das previsões feitas pelos modelos.

Para cada modelo, foi construída uma tabela que destaca a porcentagem de dados de treinamento que resulta no menor valor de MAPE. Além disso, foi criado um gráfico para facilitar a visualização e comparação desses valores. Conforme explicado na tabela 1, um valor mais baixo de MAPE sinaliza maior eficiência na predição.

5.2.1.1 Regressão Linear

Figura 15 – Tabela dos valores MAPE - Regressão Linear.

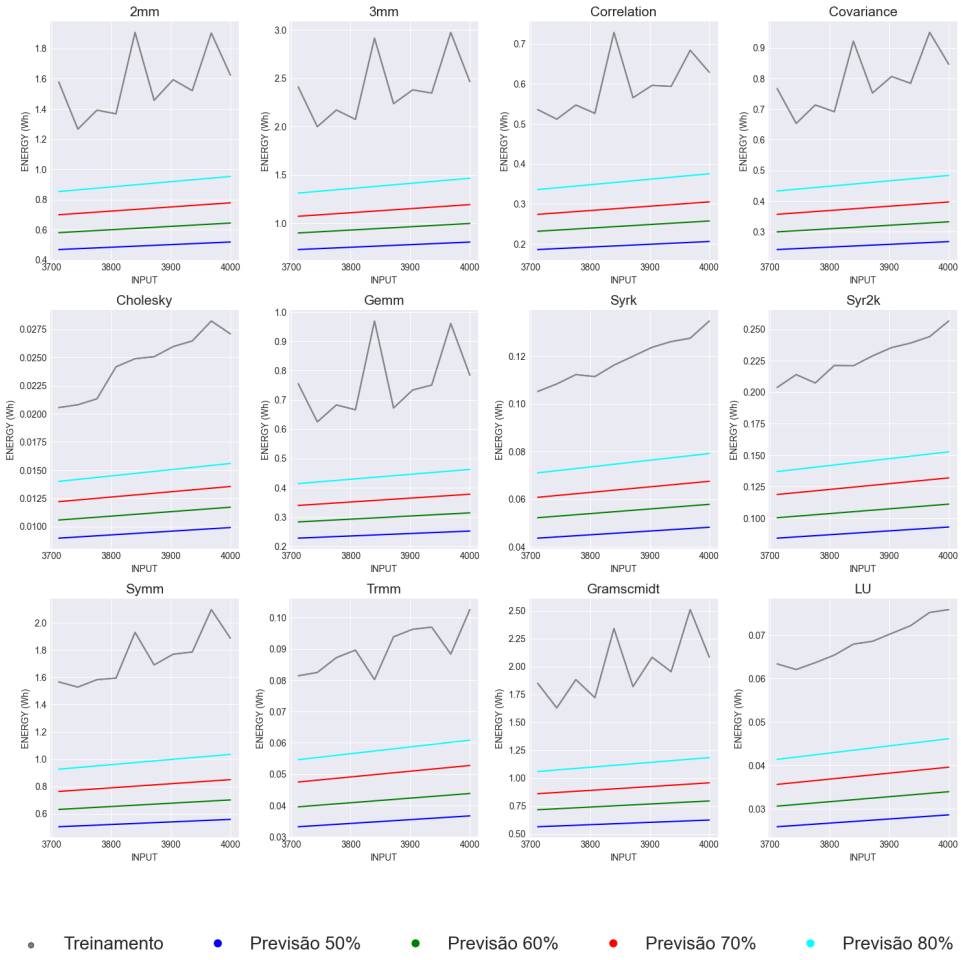
	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramscmid	LU
50	67.9	67.61	66.53	67.39	61.06	67.82	61.15	60.9	69.25	60.95	69.67	60.08
60	60.2	59.92	58.26	59.55	54.04	59.96	53.45	53.31	61.47	53.4	61.43	52.71
70	52.0	52.15	50.6	51.73	46.9	51.94	45.74	44.65	53.37	43.98	53.63	44.92
80	**41.35**	**41.31**	**39.32**	**41.31**	**38.98**	**41.28**	**36.47**	**36.06**	**43.31**	**35.49**	**42.8**	**35.91**

Fonte: Autoria própria.

Com inicio pela Regressão Linear, conforme mencionado nos resultados da seção 5.1.2, o modelo linear não apresenta uma grande taxa de acerto na predição dos gastos energéticos,com todos os seus valores na faixa entre 20% e 50%, que, de acordo com a tabela 1, são considerados razoáveis. Para uma melhor observação dessa predição, pode-se verificar a figura 16.

No gráfico 16, observa-se uma distância considerável entre os dados preditos e os dados reais. Além disso, nota-se que a maior quantidade de dados utilizada (80%) para treinamento foi a que mais se aproximou dos valores reais, algo que será verificado em modelos subsequentes. Os valores de MAPE obtidos foram altos, e apesar de levemente melhores do que o esperado, o modelo atual não se mostra adequado para as previsões que busca-se realizar.

Figura 16 – Gráfico dos 10 valores previstos x os 10 valores reais - Regressão Linear.



Fonte: Autoria própria.

5.2.1.2 Regressão Polinomial de Grau 2

Figura 17 – Tabela dos valores MAPE - Regressão Polinomio de Grau 2°.

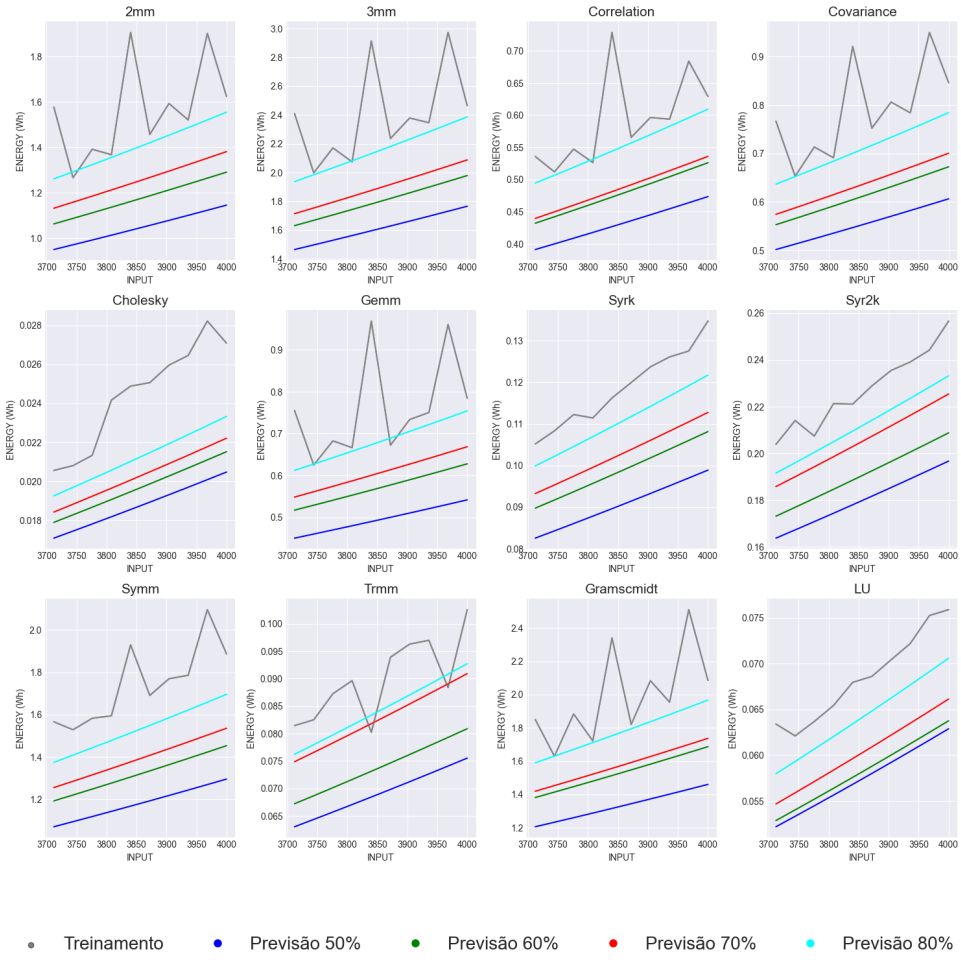
	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramscmid	LU
50	32.13	31.9	26.49	29.23	22.89	33.79	23.47	20.69	31.9	22.85	32.27	16.05
60	23.79	23.92	18.56	21.74	19.09	23.59	16.53	15.98	23.86	17.55	22.08	14.92
70	18.64	19.89	17.09	18.61	16.6	18.8	13.13	9.56	19.69	8.28	19.85	11.88
80	**9.29**	**9.0**	**6.41**	**9.31**	**12.6**	**9.42**	**6.57**	**6.6**	**11.68**	**7.3**	**9.72**	**6.25**

Fonte: Autoria própria.

Como esperado, a Regressão Polinomial exibiu uma eficiência superior. O modelo com um polinômio de segundo grau apresentou, na maioria dos casos, valores de MAPE próximos ou inferiores a 10, que indica uma previsão de alta precisão e, em alguns casos, uma boa previsão .

No gráfico a seguir, pode-se observar melhor essas informações.

Figura 18 – Gráfico dos 10 valores previstos x os 10 valores reais - Regressão Polinômio de Grau 2°



Fonte: Autoria própria.

Na análise de Regressão Polinomial, o modelo de polinômio de grau 2 provou ser eficiente. Conforme evidenciado no gráfico 18, os valores previstos por este modelo estão em estreita concordância com os valores reais, e indica uma alta precisão na previsão. A eficácia do modelo de grau 2 destaca-se pela sua capacidade de capturar as tendências dos dados de maneira equilibrada, e por evitar tanto o excesso (*overfitting*[1]) quanto a falta (*underfitting*[2]) de ajuste.

[1] Ocorre quando um modelo de Aprendizado de Maquina se ajusta excessivamente aos dados de treinamento, perdendo a capacidade de generalizar para novos dados.

[2] Ocorre quando um modelo de Aprendizado de Maquina é muito simples para capturar a complexidade dos dados de treinamento, que resulta em baixo desempenho.

5.2.1.3 Regressão Polinomial de Grau 3

Figura 19 – Tabela dos valores MAPE - Regressão Polinomio de Grau 3.

	2mm	3mm	Correlation	Covariance	Cholesky	Gemm	Syrk	Syr2k	Symm	Trmm	Gramscmid	LU
50	14.88	14.15	36.06	8.35	17.57	25.95	11.34	1.89	23.02	8.94	8.1	7.48
60	8.32	**7.69**	**5.3**	**7.56**	8.54	10.55	**1.44**	**1.38**	**4.41**	8.19	13.89	7.68
70	**7.79**	8.58	9.84	7.68	9.6	**8.53**	1.96	5.83	6.74	11.53	**8.01**	4.44
80	12.28	12.86	9.63	9.67	**6.06**	13.89	5.16	1.55	5.5	**4.05**	10.77	**3.82**

Fonte: Autoria própria.

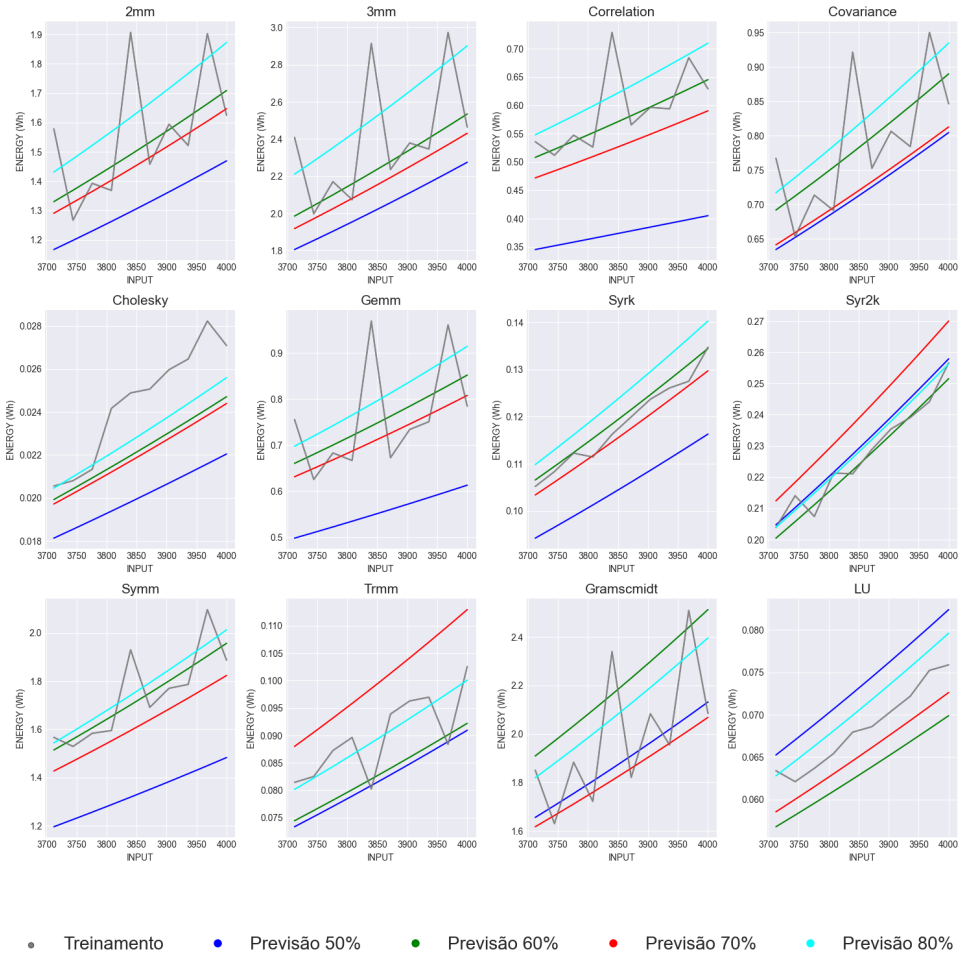
O modelo de polinômio de grau 3 destacou-se como o mais eficiente em nossa análise, ao oferecer valores consistentemente baixos de Erro Percentual Médio Absoluto (MAPE), com o alcance em alguns casos de valores próximos a 1%. Este desempenho notável sugere que o aumento da complexidade do modelo, neste caso para um polinômio de terceiro grau, pode ser altamente benéfico na previsão de dados, especialmente em contextos do presente trabalho.

Interessantemente, observou-se que *benchmarks* como ***3mm***, ***Correlation***, ***Covariance***, ***Syr*** e ***Syr2k*** alcançaram seus melhores valores de MAPE com apenas 60% dos dados disponíveis. Isso indica que, para esses casos específicos, uma quantidade menor de dados foi suficiente para treinar o modelo eficientemente. Por outro lado, *benchmarks* como ***2mm***, ***Gemm*** e ***Gram-Schmidt*** mostraram a necessidade de uma carga de dados ligeiramente maior, com suas melhores previsões em cerca de 70% dos dados.

Finalmente, *benchmarks* como ***Cholesky***, ***Trmm*** e ***LU*** destacaram-se por necessitar de uma quantidade ainda maior de dados para alcançar a otimização, com seus melhores resultados sendo obtidos com 80% da carga de dados. Esta variação na quantidade de dados necessária para otimizar o modelo em diferentes *benchmarks* ilustra a importância de considerar a especificidade de cada conjunto de dados ao aplicar modelos de Regressão Polinomial, especialmente em contextos de alto volume de dados.

No gráfico subsequente, tem-se a oportunidade de observar e analisar essas informações com mais detalhes.

Figura 20 – Tabela dos valores MAPE - Regressão Polinomio de Grau 3.



Fonte: Autoria própria.

Pode-se observar que as linhas de previsão seguem de perto os pontos de treinamento em todos os gráficos, o que sugere que o modelo possui uma ótima precisão. Esta precisão é particularmente notável nos *benchmarks* onde as previsões com 60% dos dados já proporcionam uma aproximação muito próxima aos valores reais, como nos casos de ***3mm***, ***Correlation*** e ***Covariance***.

A consistência entre as previsões e os dados reais em diferentes proporções de treinamento indica que a Regressão Polinomial de grau 3 é bem adaptada para esta base de dados. Ela supera outros modelos testados, sendo a mais ideal para analisar e prever o consumo energético entre as abordagens consideradas. Isso reforça a ideia de que a escolha correta do grau do polinômio é fundamental para capturar a complexidade dos padrões nos dados, e preservar assim a eficiência do modelo.

5.2.2 Abordagem 2: União de Múltiplos *Dataset*s

Na Abordagem, conforme explicado na seção 4.5, foi escolhida uma estratégia integrada, unindo múltiplos *datasets* para potencializar o modelo analítico. Foram utilizados modelos de Regressão Linear e Polinomial, com a seleção do modelo mais eficiente com base nos resultados de experimentos, e serão discutidos sobre sua precisão para cada *benchmark* isoladamente.

Para esta análise, foram combinados 11 dos 12 *dataset*s disponíveis, co a reserva do décimo segundo como um conjunto de dados de teste independente. O processo de treinamento foi realizado com os 11 *dataset*s concatenados, o que nos permitiu aproveitar uma vasta quantidade de dados e características comuns entre os *benchmarks*. Essas características, são as os dados oriundos da análise estática do código, feito citado na seção 4.2.

Os modelos foram então aplicados a esse conjunto de dados unificado, com objetivo de maximizar a precisão das previsões. Para testar a eficácia do modelo treinado, o *dataset* reservado foi empregado, e fornecer uma oportunidade de avaliar a performance do modelo em condições reais e inéditas.

A precisão dessas previsões foi avaliada com o uso do MAPE, o Erro Médio Percentual Absoluto, que nos ofereceu uma medida confiável e precisa do desempenho dos modelos em relação aos dados reais do conjunto de testes.

O objetivo central desta abordagem é duplo: primeiro, compara-se se a fusão de parâmetros comuns entre diferentes *benchmarks* resultaria em uma previsão eficiente com os métodos regressivos; segundo, busca-se compreender o comportamento atual do consumo energético e prever as tendências futuras.

Com os dados dos valores de MAPE obtidos dos experimentos, foi possível criar a seguinte tabela:

Tabela 3 – Valores de MAPE para diferentes modelos de regressão

Benchmark	Regressão Linear	Regressão Polinomial Grau 2	Regressão Polinomial Grau 3
2mm	3.04%	1.86%	7.55%
3mm	6.37%	5.71%	7.06%
Correlation	8.96%	46.01%	33.42%
Covariance	62.15%	15.41%	15.90%
Cholesky	20.13%	30.26%	14.92%
Gemm	8.22%	2.81%	8.39%
Syrk	20.30%	7.61%	8.33%
Syr2k	16.23%	6.00%	4.12%
Symm	46.82%	39.33%	30.75%
Trmm	4.12%	5.87%	7.26%
Gramscmidt	41.96%	29.41%	59.52%
LU	8.11%	9.09%	4.26%

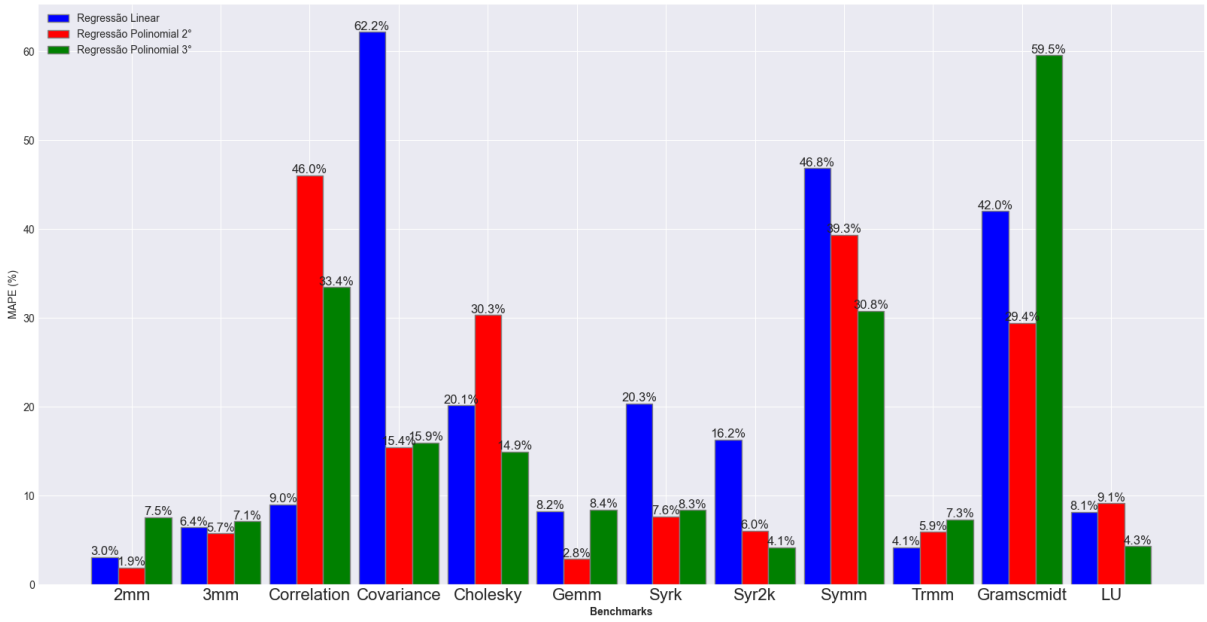
Os dados da tabela revelam uma dispersão significativa nos melhores valores de MAPE

entre os diferentes modelos de regressão. Cada *benchmark* apresenta um comportamento distinto que pode ser melhor capturado por diferentes graus de modelos polinomiais ou até mesmo pela Regressão Linear. Além disso, ao comparar os resultados com a abordagem anterior, foi observada uma variação no desempenho, que em alguns casos indica uma diminuição na eficácia dos modelos atuais.

Esta variação é particularmente notável em *benchmarks* como o ***3mm*** e o ***Syrk***, onde a Regressão Polinomial de grau 3 mostrou um aumento no MAPE, que indica uma menor precisão em comparação com a Regressão Linear ou polinomial de grau 2. A Regressão Linear, por sua vez, mostrou-se notavelmente eficaz para o *benchmarks* ***2mm***, pois demonstrou que a relação entre os dados de entrada e o consumo energético para este *benchmark* pode ser mais linear.

Essa diferença de desempenho entre os modelos destaca a complexidade dos dados e a importância de selecionar o modelo apropriado para cada conjunto de dados específico. Para facilitar a visualização e a compreensão dessas discrepâncias, o gráfico comparativo, a seguir foi criado.

Figura 21 – Comparativo dos valores de MAPE entre os modelos.



Fonte: Autoria própria.

Ao analisar o gráfico, pode-se notar que não há um modelo que se destaque uniformemente em todos os *benchmarks*, em vez disso, cada modelo parece ser mais adequado para determinados tipos de dados.

Por exemplo, a Regressão Linear (azul) mostra o menor MAPE para o *benchmark* ***Trmm***, o que aponta que, para este caso específico, a relação entre as variáveis de entrada e o consumo energético é mais próxima de uma função linear. Em contrapartida, para o *benchmark* ***3mm***, a Regressão Polinomial de grau 2 (vermelha) apresenta o menor MAPE, então indica que um

modelo ligeiramente mais complexo captura melhor a relação entre os dados.

Interessante observar que, para *benchmarks* como ***Syr2k*** e ***LU***, a Regressão Polinomial de grau 3 (verde) exibe os maiores valores de MAPE, o que poderia ser interpretado como um indicativo de *overfitting*, onde o modelo se ajusta excessivamente aos dados de treinamento e falha em generalizar bem para novos dados.

E a comparação entre a eficiência dos modelos utilizados, pode-se chegar ao ponto que o modelo que melhor obteve resultados foi o de grau 2, já que 6 aplicações tiveram sua melhor eficiência com esse modelo, em seguida vem o de grau 3 com 4 e por ultimo o linear com 2 aplicações.

A abordagem de unificação dos *benchmarks*, apresenta valores de MAPE relativamente baixos, e demonstrou a eficiência esperada, especialmente quando contrastada com os resultados obtidos na primeira experiência. Naquela inicial, a modelagem específica para cada *benchmark* individual apresentou uma precisão notavelmente boa, que remete que a estratégia de modelagem personalizada para cada tipo de dado é eficaz.

Apesar dos valores ligeiramente diferentes entre as abordagens, as porcentagens de MAPE para a predição unificada dos *dataset*s tiveram em alguns casos, resultados melhores do que na abordagem individual. Esses casos foram ***2mm***, ***3mm*** e ***Gemm***. Isso se da devido a grande correlacionabilidade presente nas aplicações de manipulação de matrizes, como foi apresentado na figura 13, sobre o *cluster* hierárquico.

Para os demais *benchmarks*, 7 deles tiveram uma eficiência melhor com o uso da abordagem individual, ***Correlacion***, ***Covariance***, ***Cholesy***, ***Syrk***, ***Syr2k***, ***Symm*** e ***Gramscmidt***. Eles juntos representam mais da metade de todos os *benchmarks* testados. O que leva a duvidar da eficiência do modelo unificado quando comparado com o individual. Além disso, 2 *benchmarks*, tiveram desempenhos aproximadamente iguais em ambos os experimentos. São eles ***Trmm*** e ***LU***.

Uma possível explicação para essa diferença significativa de desempenho é que a união dos *benchmarks* pode ter diluído características únicas , que são cruciais para a precisão das previsões. A complexidade inerente aos diferentes tipos de *benchmarks* pode exigir modelos que se ajustem de maneira mais fina às suas especificidades, algo que uma abordagem unificada pode não ser capaz de capturar de forma adequada.

Além disso, para que a abordagem de união de *dataset*s tenha sucesso, pode ser necessário um volume maior de dados. Neste estudo, foram utilizados dados de apenas 12 *benchmarks* do PolyBench, de um total de 33 disponíveis. A incorporação de dados de todos os 33 *benchmarks* poderia potencialmente introduzir uma variedade maior de padrões de consumo e correlações, o que enriqueceria o conjunto de treinamento e poderia melhorar a precisão dos modelos.

Portanto, a adição de mais dados de outros *benchmarks*, que possam ter consumos energéticos e correlações mais representativas ou variadas, pode ser uma estratégia para aprimorar

a modelagem. Esse aumento na densidade de dados poderia proporcionar aos modelos regressivos uma base mais robusta para treino, que resultaria em previsões mais precisas e eficientes.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, realizou-se a análise e a predição de consumo energético de *benchmarks* executados em Máquina Multi-core. Os resultados obtidos fornecem informações sobre as relações entre o consumo de energia, a complexidade dos conjuntos de dados de entrada e os componentes de *hardware* envolvidos. Foram abordadas várias questões importantes, como a identificação de *outliers*, a análise do consumo em comparação com os valores de entrada, a análise do consumo de cada componente de *hardware*, o consumo total de cada *benchmark* e a relação entre o tempo de execução e o consumo de energia.

Um dos resultados notável foi a identificação de padrões de consumo exponenciais em relação à complexidade dos conjuntos de dados de entrada. Isso sugere que abordagens de Regressão Polinomial podem ser mais adequadas para prever o consumo de energia nesses casos, em oposição a modelos lineares tradicionais. O que foi observado através dos experimentos executados. Além disso, concluiu-se que a CPU é o componente de *hardware* dominante em termos de consumo de energia, o que assinala a necessidade de otimizações específicas para esse componente, ou até mesmo a necessidade de divisão de trabalho através do uso de processadores gráficos (GPUs) para a redução significativa deste componente.

Outra descoberta importante foi a falta de correlação direta entre o consumo de energia e o tempo de execução. *Benchmarks* com duração mais longa nem sempre resultaram em maior consumo de energia, o que destaca a importância da otimização de tarefas ao longo do tempo para reduzir o consumo energético.

O agrupamento hierárquico dos *benchmarks* com base na correlação do consumo energético revelou conhecimentos adicionais. Foram identificados grupos de *benchmarks* que compartilham padrões de consumo semelhantes, bem como *benchmarks* que se destacam individualmente. Essa identificação fornece uma base para estratégias de otimização futuras, que podem ser direcionadas a grupos semelhantes ou a *benchmarks* individuais, dependendo das características de consumo de energia.

Os resultados obtidos revelaram compreensões interessantes sobre a complexidade dos dados e a importância da escolha do modelo apropriado. Na primeira abordagem, em que foram aplicados modelos de Regressão Linear e Polinomial a cada *benchmark* individualmente, observou-se que a Regressão Polinomial de grau 3 se destacou como o modelo mais eficiente em muitos casos, ja que oferece previsões altamente precisas. No entanto, também ficou claro que a eficácia do modelo estava diretamente relacionada à quantidade de dados usada para treinamento, com mudanças de um *benchmark* para o outro.

Por outro lado, na segunda abordagem, que uniu múltiplos *datasets* para criar um modelo analítico abrangente, encontrou-se uma variação significativa no desempenho dos modelos de regressão. Cada *benchmark* apresentou um comportamento distinto que exigia diferentes tipos de

modelos. Com os dados dessa pesquisa, a abordagem de união dos *benchmarks* não demonstrou a eficiência esperada, o que sugere que a modelagem personalizada para cada tipo de dado é mais eficaz do que uma abordagem unificada.

Portanto, nossos resultados destacam a importância de considerar a especificidade de cada conjunto de dados ao aplicar modelos de Aprendizado de Máquina para previsão de consumo energético. Além disso, sugerem que a adição de mais dados de outros *benchmarks* pode ser uma estratégia para aprimorar a modelagem unificada e melhorar a precisão das previsões nessa abordagem.

6.1 Trabalhos Futuros

Como trabalho futuro, recomenda-se:

Primeiramente, sugere-se a expansão do estudo para incluir dados de todos os 33 *benchmarks* do PolyBench, ao invés dos 12 utilizados. Isso permitiria capturar uma maior diversidade de padrões de consumos e correlações, e enriqueceria o conjunto de dados de treinamento e potencialmente melhorando a precisão dos modelos de regressão.

Além disso, é recomendável explorar outros modelos avançados de Aprendizado de Máquina, como Redes Neurais Profundas, além das técnicas de Regressão Linear e Polinomial já utilizadas. Isso pode proporcionar conhecimentos adicionais e aprimorar a precisão na predição do consumo energético.

Uma análise mais detalhada da correlação entre diferentes *benchmarks* também é um possível tema a se abordar, pois visa uma compreensão mais aprofundada das características únicas de cada *benchmark*. A clusterização desses *benchmarks* com base em seus padrões de consumo pode revelar grupos com comportamentos similares, e revelar abordagens de modelagem mais específicas.

A análise detalhada do impacto de diferentes componentes de *hardware*, como CPUs, GPUs e outros dispositivos, também é um tema fundamental. Essa análise ajudaria a identificar estratégias de otimização de *hardware* para melhorar a eficiência energética.

Por fim, a realização de uma análise da influência de outros fatores, como configurações de sistema operacional, práticas de programação e eficiência de algoritmos, no consumo de energia dos *benchmarks*. Essa abordagem holística pode oferecer uma visão mais completa sobre os gastos energéticos em ambientes computacionais.

A pesquisa dessas recomendações tem o potencial de impulsionar inovações em eficiência energética, e contribuir significativamente para práticas de computação mais sustentáveis e responsáveis.

REFERÊNCIAS

ABELLA-GONZALEZ, M. A. et al. Polybench/python: Benchmarking python environments with polyhedral optimizations. In: **ACM SIGPLAN International Conference on Compiler Construction (CC)**. Seul: [s.n.], 2021. p. 59–70. Disponível em: <https://hal.archives-ouvertes.fr/hal-03153351>.

ALMEIDA, A.; CARVALHO, F.; MENINO, F. **Introdução ao machine learning: de alunos para alunos**. [S.l.]: Dataat, 2017. <https://dataat.github.io/introducao-ao-machine-learning/index.html#licen%C3%A7a>.

ALMEIDA, A. M. G. D.; LEITE, N. A. B.; RAMOS, R. F. Recuperação da informação e a importância do pré-processamento. **RETEC-Revista de Tecnologias**, v. 12, n. 2, 2019.

AMARIS, M. et al. Evaluating execution time predictions on gpu kernels using an analytical model and machine learning techniques. **JPDC**, Elsevier, v. 171, p. 66–78, 2023.

BASTOS, E. V. P.; GUIMARÃES, J. C. F. d.; SEVERO, E. A. Linear regression model for investment analysis of an oil company. **Revista Produção e Desenvolvimento**, v. 1, n. 1, p. 77–88, 2015. Disponível em: <https://revistas.cefetrj.br/index.php/producaoedesenvolvimento/article/view/e62>.

BATISTA, G. E. d. A. P. A. **Pré-processamento de dados em aprendizado de máquina supervisionado**. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, University of São Paulo, São Carlos, 2003.

BELGAID, M. C.; ROUVOY, R.; SEINTURIER, L. **Jouleit: a tool that can be used to monitor energy consumption for any Linux program**. 2020. Disponível em: <https://github.com/powerapi-ng/jouleit/>.

BRITO, J. D.; MATAI, P. H. L. d. S.; SANTOS, M. R. dos. Data center e eficiência energética: Data center and energy efficiency. **Brazilian Journal of Business**, v. 5, n. 2, p. 786–795, 2023.

BRUNIALTI, L. et al. Aprendizado de máquina em sistemas de recomendação baseados em conteúdo textual: Uma revisão sistemática. In: **Anais do XI Simpósio Brasileiro de Sistemas de Informação**. Porto Alegre, RS, Brasil: SBC, 2015. p. 203–210. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/sbsi/article/view/5818>.

CALDAS, M. S.; SILVA, E. C. C. Fundamentos e aplicação do big data: como tratar informações em uma sociedade de yottabytes. **Bibliotecas Universitárias: pesquisas, experiências e perspectivas**, v. 3, n. 1, abr. 2016. Disponível em: <https://periodicos.ufmg.br/index.php/revistarbu/article/view/3086>.

CHATARASI, P. et al. An extended polyhedral model for spmd programs and its use in static data race detection. In: **Languages and Compilers for Parallel Computing: 29th International Workshop, LCPC 2017**. Rochester, NY, USA: Springer, 2017. p. 106–120.

CHEIN, F. **Introdução aos modelos de regressão linear: um passo inicial para compreensão da econometria como uma ferramenta de avaliação de políticas públicas**. Brasília: ENAP, 2019.

COHEN, J. **Statistical power analysis for the behavioral sciences**. [S.l.]: Academic press, 2013.

CORDEIRO, D. et al. Green cloud computing: Challenges and opportunities. In: **Anais do XIX SBSI**. Maceió/AL: SBC, 2023. p. 129–131.

Empresa de Pesquisa Energética (EPE). **Anuário Estatístico de Energia Elétrica 2021**. 2021. Rio de Janeiro: EPE. Disponível em: <https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/PublicacoesArquivos/publicacao-160/topico-168/EPEFactSheetAnuario2021.pdf>.

FILHO, D. B. F.; JÚNIOR, J. A. S. Desvendando os mistérios do coeficiente de correlação de pearson (r). **Revista Política Hoje**, v. 18, n. 1, p. 115–146, 2009.

FIORETO, V. D. L. Aplicação de modelos de aprendizado de máquina para a previsão da temperatura de um motor elétrico. Universidade Estadual Paulista (Unesp), 2023.

FRANK, D. R.; SEIBT, L. Shell script. Av. Oscar Martins Rangel, 4500 - Taquara - RS - Brasil, 2008.

GARSON, G. D. **Hierarchical linear modeling: Guide and applications**. [S.l.]: Sage, 2013.

IEA. **Tracking Clean Energy Progress**. 2023. <https://www.iea.org/reports/tracking-clean-energy-progress-2023>. Licença: CC BY 4.0.

JUDGE, P. **Data Centers e o grande problema com as estimativas de energia**. 2022. DataCenterDynamics. Disponível em: <https://www.datacenterdynamics.com/br/anÃąlises/data-centers-e-o-grande-problema-com-as-estimativas-de-energia/>.

KARIMOV, J.; RABL, T.; MARKL, V. Polybench: The first benchmark for polystores. In: NAMBIAR, R.; POESS, M. (Ed.). **Performance Evaluation and Benchmarking for the Era of Artificial Intelligence - 10th TPC Technology Conference, TPCTC 2018**. Rio de Janeiro, Brazil: Springer, 2018. (Lecture Notes in Computer Science, v. 11135), p. 24–41. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-030-11404-6_3>.

KIRK, D. B.; WEN-MEI, W. H. **Peogramming Massively Parallel Processors: A Hands-on Approach**. [S.l.]: China Machine Press, 2013.

KLUYVER, T. et al. Jupyter notebooks - a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. IOS Press, 2016. p. 87–90. Disponível em: <https://eprints.soton.ac.uk/403913/>.

LEWIS, C. D. **Industrial and Business Forecasting Methods**. [S.l.]: Butterworths, 1982.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas inteligentes-Fundamentos e aplicações**, v. 1, n. 1, p. 32, 2003.

MOORE, L. M. **The basic practice of statistics**. [S.l.]: Taylor & Francis, 1996.

MYTTENAERE, A. D. et al. Mean absolute percentage error for regression models. **Neurocomputing**, Elsevier, v. 192, p. 38–48, 2016.

NOUREDDINE, A. et al. A preliminary study of the impact of software engineering on greenit. In: **First International Workshop on Green and Sustainable Software**. Zurique, Suíça: INRIA, 2012. p. 21–27. Disponível em: <https://inria.hal.science/hal-00681560>.

OPENMP API. **OpenMP 5.2 API Syntax Reference Guide**. [S.l.], 2020. [PDF]. Disponível em: <https://www.openmp.org/specifications/>.

PENHA, D.; CORRÊA, J.; MARTINS, C. Análise comparativa do uso de multi-thread e openmp aplicados a operações de convolução de imagem. In: **Anais do III Workshop em Sistemas Computacionais de Alto Desempenho**. Porto Alegre, RS, Brasil: SBC, 2002. p. 118–125. ISSN 0000-0000. Disponível em: <https://sol.sbc.org.br/index.php/wscad/article/view/20770>.

PERKEL, J. Why jupyter is data scientists' computational notebook of choice. **Nature**, v. 563, p. 145–146, 11 2018.

PIMENTEL, J. F. et al. A large-scale study about quality and reproducibility of jupyter notebooks. In: **2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)**. [S.l.: s.n.], 2019. p. 507–517.

PIMENTEL, J. F. et al. Ciência de dados com reprodutibilidade usando jupyter. In: **Sociedade Brasileira de Computação**. [S.l.: s.n.], 2021.

SHEN, H. Interactive notebooks: Sharing the code. **Nature**, v. 515, n. 7525, p. 151–152, 2014.

URIBE, D. C. **Plataforma para medir el consumo energético de algoritmos**. 2022. Universidad de Concepción. Disponível em: <http://www.inf.udec.cl/~jfuentess/files/theses/ugrad_thesis_DC.pdf>.

WESTPHALL, C.; VILLARREAL, S. R. PrincÍpios e tendÊncias em green cloud computing. **Revista Eletrônica de Sistemas de Informação**, v. 12, n. 1, p. 1–16, 2013.

WILLIAMS, J.; CURTIS, L. Green: The new computing coat of arms? **IT Professional Magazine**, IEEE Computer Society, v. 10, n. 1, p. 12, 2008.

YUKI, T. Understanding polybench/c 3.2 kernels. In: **International workshop on polyhedral compilation techniques (IMPACT)**. [S.l.: s.n.], 2014. p. 1–5.