ADRENALIN 2024

Load disaggregation Challenge: Energy use in buildings

## Report template

1. **Executive Summary**

* Objective: Disaggregate heating and cooling loads using smart meter data.
* Key Findings: Algorithm achieved X% accuracy.

1. **Introduction**

* Background on energy disaggregation.
* Objectives of the project.

1. **Methodology**

* Description of the algorithm.
* Data requirements and sources.
* Training and testing procedures.

1. **Data**

* Sources: Smart meters, weather data.
* Preparation: Cleaning, preprocessing steps.
* Characteristics: Data frequency, periods covered.

1. **Algorithm Implementation**

* Software used: Python, specific libraries.
* Code repository: Link to GitHub.
* Parameters: Default values, tuning details.

1. **Results and Evaluation**

* Performance metrics used: Accuracy, precision.
* Comparative analysis with other methods.
* Error analysis: Observed limitations.

1. **Reproducibility**

* Step-by-step instructions.
* Environment configuration: Hardware, OS, dependencies.
* Checklist for reproducibility.

1. **Discussion**

* Interpretation of results.
* Limitations of the study.
* Future work suggestions.

1. **Conclusion**

* Summary of findings.
* Recommendations based on results.

1. **References**

* List of cited works.

1. **Appendices**

* Additional supporting materials.

# ADRENALIN 2024

# Load disaggregation Challenge: Energy use in buildings

## 1. Executive Summary -ok

* **Objective:** This study aims to disaggregate the heating and/or cooling loads of buildings using smart meter data. The competition addresses a critical issue in the construction sector, which accounts for a significant portion of global energy consumption. Effective disaggregation of these loads is essential for achieving global carbon emission reduction targets, implementing targeted energy efficiency measures, and reducing operational costs for building owners and operators. The data provided in the competition includes main energy meter readings and weather data, while the sub-metering data for heating and cooling is kept confidential for the evaluation of results.
* **Key Findings:** The developed algorithm demonstrated high accuracy in disaggregating heating and cooling loads, achieving a Normalized Mean Absolute Error (NMAE) of 0.2557 across all analyzed buildings in public leaderboard of the competition. This result is well below the 0.5 threshold required for prize eligibility, showcasing the model's efficiency in most tested scenarios. The algorithm stands out for its simplicity, as it does not require additional temperature or weather data and needs minimal data cleaning. Additionally, the methodology was designed to be scalable, using reasonable computational resources and ensuring its reproducibility by domain experts, as defined by the competition's qualitative criteria.

## 2. Introdução -ok

* **Contexto sobre a desagregação de energia:** A desagregação de energia em edifícios é um campo crítico de pesquisa e desenvolvimento que busca identificar e quantificar o consumo de energia por serviços individuais dentro de um edifício, como aquecimento, refrigeração, ventilação, iluminação e cargas de aparelhos. Esse processo é essencial para entender melhor como a energia é consumida e para implementar estratégias de eficiência energética mais direcionadas. No entanto, muitos edifícios existentes são equipados apenas com medidores de energia principais que registram o consumo total de energia, sem detalhamento dos serviços específicos. Isso torna a desagregação um desafio significativo, especialmente em um contexto onde a instalação de sistemas de submedição detalhados é muitas vezes inviável devido a custos e complexidades logísticas.
* **Objetivos do projeto:** O objetivo principal deste projeto é desenvolver um algoritmo de desagregação de carga de energia que seja capaz de operar de forma não supervisionada e em várias resoluções de tempo, que pode variar de dados coletados a cada 5 minutos até dados coletados a cada hora, dependendo do edifício em questão. A finalidade é desagregar com precisão as cargas de aquecimento e/ou refrigeração dos dados do medidor principal, enfrentando assim os desafios associados à falta de medições específicas de dispositivos e à diversidade de práticas e padrões de construção dos edifícios. Este projeto busca também promover a generalização entre diferentes tipos de edifícios para melhorar a aplicabilidade e eficiência dos algoritmos desenvolvidos em um amplo espectro de cenários de construção.

## 3. Metodologia -ok

Para desagregar as cargas de aquecimento e refrigeração nos edifícios, foi desenvolvido um algoritmo STL Ajustado (Decomposição Sazonal-Tendência usando LOESS). Este método aborda os desafios associados às técnicas de decomposição tradicionais quando aplicadas a dados complexos e ruidosos de consumo de energia. O STL Ajustado combina a estabilidade da decomposição clássica com a adaptabilidade do método STL, resultando em previsões precisas e robustas.

### Abordagens Iniciais e Desafios -ok

*Decomposição Clássica:*

Inicialmente, empregamos métodos de decomposição clássica para analisar os dados de consumo de energia. Embora essa abordagem tenha determinado efetivamente a tendência geral (altura) dos dados, falhou em capturar com precisão o nível base. A linha de base ficou significativamente abaixo do ponto ideal, e o componente sazonal permaneceu constante, incapaz de se adaptar a períodos de sazonalidade ruidosa ou em rápida mudança (Figura 1). Essa rigidez dificultou o ajuste em situações onde os dados eram imperfeitos ou continham ruído, especialmente quando a sazonalidade era instável.

Gráfico, Histograma

Descrição gerada automaticamenteGráfico, Gráfico de barras

Descrição gerada automaticamente

Figura 1: Inflexibilidade da Decomposição Clássica

* + Demonstra como o método de decomposição clássica falha em se adaptar à sazonalidade em mudança, resultando em uma base desalinhada e padrões sazonais constantes.

*Modelos Multiplicativos:*

Também experimentamos modelos multiplicativos para abordar esses problemas. No entanto, esses modelos não produziram resultados satisfatórios, pois ainda não conseguiam lidar com as complexidades de padrões sazonais ruidosos e em mudança.

Ref1 – artigo que gerou a ideia inicial do modelo de decomposição e explica como unciona esse metodo com a decomposição clássica.

No apêndice B mostro como foi feito os cálculos da decomposição clássica e o método automático de ajuste do tamanho da altura e base que acabou sendo a ideia principal e culminou no stl ajustado.

O método de decomposição clássica ajudou muito a desenvolver o modelo final e em alguns prédios onde os dados estavam com muita qualidade eles inclusive geraram os melhores resultados, sendo interessante testar ele e inclusive utilizar dependendo da qualidade dos dados do prédio.

*Decomposição STL:*

Posteriormente, exploramos o método de decomposição STL padrão. A abordagem STL mostrou-se robusta na adaptação a padrões sazonais ruidosos e no tratamento de outliers. Forneceu um componente sazonal flexível, capaz de se ajustar a tendências sazonais complexas e em evolução. No entanto, quando a sazonalidade mudava drasticamente, o método STL tendia a amplificar excessivamente o nível base e os valores sazonais, levando a distorções na representação dos dados (Figura 2). Esse excesso de ajuste tornou as previsões pouco confiáveis, particularmente em cenários com variações significativas no consumo de energia.

Gráfico, Histograma

Descrição gerada automaticamenteGráfico, Gráfico de barras

Descrição gerada automaticamente

Figura 2: Excesso de Ajuste na Decomposição STL Padrão

* + Mostra a amplificação excessiva do nível base e dos valores sazonais ao usar a decomposição STL padrão em dados com sazonalidade em rápida mudança.

### Desenvolvimento do Método STL Ajustado -ok

Para superar as limitações dos métodos anteriores, desenvolvemos o algoritmo STL Ajustado, que integra os componentes estáveis de base e tendência da decomposição clássica com o componente sazonal adaptável do método STL. A principal inovação reside em como os valores de decomposição são aplicados aos períodos de previsão, garantindo estabilidade e adaptabilidade sem distorcer os dados de base e tendência.

Gráfico, Histograma

Descrição gerada automaticamenteGráfico, Gráfico de barras

Descrição gerada automaticamente

Figura 3: Controle e Adaptabilidade do STL Ajustado

* A figura ilustra como o STL Ajustado combina eficazmente a adaptabilidade sazonal com o controle preciso da altura das previsões.

Principais Etapas do Pipeline do STL Ajustado

1. Pré-processamento e Limpeza de Dados:
   * Aplicamos técnicas de limpeza de dados para remover anomalias e garantir a qualidade dos dados.
   * O método STL Ajustado permanece robusto mesmo com dados imperfeitos ou ruidosos, reduzindo a necessidade de limpeza de dados rigorosa, ao contrario do modelo de decomposição clássica, que exige dados em qualidade quase perfeita para produzir resultados satisfatórios.
2. Seleção de Semanas de Referência Estáveis:
   * Identificamos "semanas de referência estáveis" onde não houve consumo de energia associado ao aquecimento ou resfriamento (Figura 4, esquerda).
   * Essas semanas servem como base para ajustar os padrões sazonais e de tendência em previsões futuras.
3. Seleção de Base e Altura:
   * Calculamos a mediana da base e da altura a partir das semanas de referência.
   * Estabelecemos níveis de base ideais e alturas de tendência para previsões precisas.
4. Seleção de Multiplicador para Aquecimento e Resfriamento:
   * Em edifícios com sistemas de aquecimento e resfriamento, identificamos a necessidade de um fator multiplicativo global para ajustar a discrepância entre as semanas de referência e o consumo real, pois as semanas de referência não capturavam adequadamente o consumo desses sistemas.
   * Para isso, aplicamos técnicas de identificação de picos e vales em séries temporais de alta granularidade (5 ou 15 minutos) para isolar pontos específicos de ligamento e desligamento dos sistemas HVAC, aprimorando a desagregação do consumo de energia.
5. Ajuste de Base e Altura Usando Multiplicadores:
   * Aplicamos os multiplicadores para ajustar os níveis de base e altura nas previsões no caso de ser aquecimento e resfriamento.
6. Decomposição STL:
   * Realizamos a decomposição STL nos dados de consumo de energia.
   * Extraímos os componentes sazonal, tendência e residual.
7. Método de Previsão para Plugs e Eletricidade:
   * Utilizamos a decomposição STL ajustada para prever as cargas de plugues e componentes elétricos.
   * Aplicamos normalização e ajustamos a tendência sazonal, base e altura conforme necessário (Figura 4, centro).
8. Método de Previsão para Consumo de Energia Dependente da Temperatura:
   * Desagregamos o consumo de energia dependente da temperatura comparando o consumo total com as previsões ajustadas.
   * Isolamos a energia usada para aquecimento e resfriamento (Figura 4, direita).

Interface gráfica do usuário, Gráfico, Histograma

Descrição gerada automaticamente

* Figura 4: Previsões Usando o Método STL Ajustado
  + *Esquerda (Ajuste Fino da Previsão):* Ilustra o uso de semanas de referência estáveis para ajustar finamente as previsões, ajustando os componentes sazonais e de tendência dinamicamente com base nos quantis semanais.
  + *Centro (Previsão a partir da Decomposição):* Compara o consumo total de energia com as previsões para plugues e componentes elétricos, destacando a capacidade do modelo de se adaptar à dinâmica dos dados enquanto controla os tamanhos das previsões.
  + *Direita (Previsão de Desagregação do Consumo Dependente da Temperatura):* Apresenta a desagregação do consumo de energia dependente da temperatura, mostrando o isolamento preciso das cargas de aquecimento e resfriamento.

Vantagens do Método STL Ajustado

* Estabilidade na Base e Tendência: Mantém um nível de base estável e altura de tendência, prevenindo distorções e garantindo previsões consistentes.
* Adaptabilidade à Sazonalidade Complexa: Adapta-se de forma flexível a padrões sazonais complexos e ruidosos, ajustando o componente sazonal conforme necessário.
* Robustez a Imperfeições nos Dados: Desempenha-se efetivamente mesmo com dados imperfeitos ou ruidosos, reduzindo a dependência de limpeza de dados extensiva.
* Desempenho Equilibrado: Oferece um equilíbrio ideal entre adaptabilidade e estabilidade, proporcionando previsões precisas e confiáveis em diversos cenários.

Conclusão

O algoritmo STL Ajustado aborda efetivamente as deficiências da decomposição clássica e dos métodos STL padrão ao combinar seus pontos fortes. Garante previsões precisas, estáveis e robustas para a desagregação do consumo de energia, particularmente em conjuntos de dados com padrões complexos ou ruidosos. Este método não apenas melhora a adaptabilidade a tendências sazonais em mudança, mas também mantém o controle necessário sobre os componentes de base e tendência, tornando-o altamente adequado para aplicações práticas na análise de consumo de energia.

### Requisitos e Fontes de Dados -ok

O modelo utilizou dados de consumo total de energia (kw\_total) obtidos a partir de medidores principais e baterias fotovoltaicas instaladas em diversos edifícios. Esses dados, registrados em intervalos regulares, serviram como a série temporal principal para a decomposição. O kw\_total abrange tanto o consumo das fontes principais quanto o uso de energia proveniente das baterias fotovoltaicas.

data\_energy['kw\_total'] = data\_energy['main\_meter(kW)'] + data\_energy['PV\_battery\_system(kW)']

Uma vantagem significativa do algoritmo STL Ajustado é sua independência de variáveis externas, como temperatura ou clima. Isso permite sua aplicação em cenários onde tais dados não estão disponíveis ou onde o consumo energético sazonal não depende exclusivamente de variações climáticas, simplificando sua implementação.

As principais variáveis utilizadas no algoritmo incluem:

* **Consumo total de energia (kw\_total):** Base para a análise e decomposição.
* **Dados agrupados por semanas e anos:** Garantem a sincronização temporal adequada.
* **Quantis semanais superiores e inferiores (week\_upper\_quantile e week\_lower\_quantile):** Utilizados para criar o week\_median e selecionar as semanas de referencia.

Esses dados permitiram identificar semanas de referência estáveis—períodos em que a carga de aquecimento/resfriamento não era utilizada ou era estava em seu estado mínimo na serie temporal —facilitando a desagregação das cargas associadas a aparelhos elétricos e iluminação. Essa identificação foi realizada de forma robusta, aproveitando as características intrínsecas dos dados, sem a necessidade de limpeza rigorosa.

O principal requisito para o algoritmo é que os dados de séries temporais estejam sincronizados sem dados nulos de energia. Essa simplicidade, aliada à robustez do método, permite ajustar eficientemente os períodos de referência sem depender de informações externas adicionais, como dados meteorológicos.

### Procedimentos de Treinamento e Teste -ok

O treinamento do algoritmo STL Ajustado foi realizado se utilizando da otimização inicial do tamanho das semanas pela decomposição, clássica, procedimento descrito no apêndice b. Logo após foi feito um ajuste manual utilizando a visualização das series temporais de cada prédio para encontrar os melhores pontos dos dados a serem selecionados para as semanas de referência estáveis—períodos sem consumo de energia relacionado ao aquecimento ou resfriamento ou os períodos de menor consumo possível relacionado a temperatura.

Durante o treinamento, foram otimizados os parâmetros chave do modelo:

* **median\_height e median\_base:** Controlam a altura e a base dos componentes de plugues e iluminação no gráfico de energia total. A otimização desses parâmetros visou encontrar o tamanho ideal para permitir uma desagregação ideal dos componentes sazonais usando o STL.

Após o ajuste dos parâmetros, a fase de testes envolveu a comparação das previsões do modelo com resultados obtidos no laderboard público de treinamento da competição. Essa validação prática permitiu refinamentos adicionais, assegurando a melhor desagregação possível das cargas de energia e a precisão das previsões.

## 4. Dados -ok

**Fontes de Dados**: Os dados utilizados foram os disponibilizados pela competição para treinamento, que incluem os medidores principais de energia dos edifícios (main meters) e dados meteorológicos.

**Agregação dos Dados**: Inicialmente, os dados foram agregados por meio de uma função aggregate\_data, que uniu todos os dados em um único DataFrame e realizou a fusão com os dados temporais correspondentes a cada edifício.

**Processo de Limpeza e Preparação dos Dados**: Para cada edifício, identificado pelo seu id\_real (por exemplo, L14.B04\_1H), foi realizado o seguinte pipeline de limpeza dos dados:

1. **Carregamento dos Dados**: Os dados são carregados no DataFrame agregado através da função load\_dataframe. Uma sincronização é realizada, se necessária, utilizando a função check\_sync\_and\_fill\_gaps.
2. **Remoção de Intervalos e Feriados**: As variáveis de limpeza, como intervalos, feriados, anos e informações locais (country\_pt), são coletadas nos parâmetros. A função remove\_intervals\_and\_holidays é aplicada para remover os intervalos e feriados selecionados, transformando esses dados em valores nulos.
3. **Imputação dos Dados**: Os valores nulos são então imputados usando a função fill\_na, que agrupa a série temporal por horário e dia da semana, permitindo a imputação dos dados de maneira a respeitar a sazonalidade e os dados mais próximos.
4. **Verificação dos Resultados**: Os dados são visualizados para conferência manual do resultado final.
5. **Armazenamento dos Dados**: São salvos dois DataFrames: um com os dados limpos e outro com os dados originais, organizados por cada prédio e granularidade.

A função preprocessing\_and\_load executa o pipeline:

Texto

Descrição gerada automaticamente

**Características dos Dados**: Os dados disponíveis são detalhados na tabela anexa, que ressalta a diversidade dos tipos de edifícios e as variadas resoluções temporais de coleta dos dados, que vão de 5 minutos a 1 hora. Essas resoluções cobrem uma ampla gama de edifícios, incluindo escritórios, escolas, centros comerciais e residências de idosos, cada qual com características e demandas energéticas específicas. A complexidade é ainda aumentada pela presença de sistemas de bateria ou fotovoltaicos (PV) em alguns edifícios, adicionando uma camada extra na análise da carga energética. Ao todo, são 14 datasets, cada um representando um edifício diferente ou uma granularidade distinta.

Tabela

Descrição gerada automaticamente

A sazonalidade emerge como a característica mais relevante dos dados, manifestando-se principalmente por hora e por dia da semana. Por exemplo, em um gráfico de uma hora, a sazonalidade corresponderia a 168 linhas. Portanto, é crucial que os dados estejam sincronizados e livres de nulos para evitar distorções nas previsões derivadas dessa sazonalidade ampliada.

Outro aspecto significativo é a diferenciação entre dias de trabalho e horários laborais. Muitos edifícios apresentam variações temporais distintas, com períodos de ocupação e desocupação que podem ser claramente demarcados. Uma técnica que se mostrou eficaz nessa separação foi o uso de Hidden Markov Models, que conseguiu distinguir de forma precisa os períodos trabalhados dos não trabalhados. Isso possibilitou uma abordagem mais complexa para a decomposição dos dados, embora esta não tenha sido adotada em função de proporcionar resultados inferiores aos obtidos com a decomposição semanal.

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Para a Figura 2, que ilustra um exemplo de aplicação do modelo Hidden Markov Model (HMM) separando os dados por ocupação e desocupação de um edifício.

## 5. Implementação do Algoritmo -ok

**Software Utilizado**

* **Python**: A implementação utiliza a versão 3.12.4 de Python, escolhida por sua estabilidade e suporte a bibliotecas científicas.
* **Bibliotecas Específicas**: As bibliotecas necessárias estão listadas no arquivo requirements.txt. Este arquivo inclui pacotes como numpy, pandas, matplotlib, scipy, e statsmodels, que são fundamentais para a manipulação de dados, análise estatística e visualização gráfica.

**Repositório do projeto:**

* O código fonte do projeto está hospedado em um repositório privado no GitHub. Para acessar o repositório https://github.com/rafaelsudbrackzimmermann/REP\_ADRENALIN, os interessados devem solicitar acesso enviando um e-mail para [rafaelszimmermann@gmail.com](mailto:rafaelszimmermann@gmail.com).

**Acesso ao Código Fonte**

* O código fonte do projeto está hospedado em um repositório privado no GitHub. Para acessar o repositório [REP\_ADRENALIN](https://github.com/rafaelsudbrackzimmermann/REP_ADRENALIN), os interessados devem solicitar acesso enviando um e-mail para [rafaelszimmermann@gmail.com](mailto:rafaelszimmermann@gmail.com).
* <https://github.com/rafaelsudbrackzimmermann/REP_ADRENALIN>

**Parâmetros de Otimização**

* **Ajuste dos Dados:** Os parâmetros utilizados são especificamente para otimização de ajuste dos dados, visando aprimorar a análise em cada fase dos dados. A maioria dos ajustes foi realizada manualmente, buscando a melhor adaptação e interpretação possível dos dados.

## 6. Resultados e Avaliação -ok

**Métrica de Desempenho**: O algoritmo empregado é um método de desagregação não supervisionado. Por isso, a métrica de avaliação de treinamento adotada é a Normalised Mean Absolute Error (NMAE), que é utilizada na competição consolidando todos os datasets em um único resultado de erro. No caso do modelo STL ajustado implementado nesta submissão, o resultado de treinamento foi de 0.255.

**Análise Comparativa com Outros Métodos**: O desempenho do algoritmo foi comparado com outras técnicas tradicionais para a desagregação de energia em edifícios, tais como a decomposição clássica e a decomposição utilizando STL ajustado.

* **Decomposição Clássica**: Nos prédios onde a análise envolvia apenas o aquecimento, a decomposição clássica ajustada mostrou-se bastante promissora, conforme apresentado no apêndice B, alcançando um NMAE de 0.320 nas submissões de treinamento da competição. Contudo, quando aplicada a prédios que necessitavam de análise de aquecimento e resfriamento, os resultados foram apenas aceitáveis.
* **Decomposição STL**: Esta técnica apresentou resultados insatisfatórios na maioria dos prédios, principalmente devido à distorção que ocorria ao tentar aplicar a mesma metodologia da decomposição clássica. Os resultados foram particularmente ruins, com NMAE superiores a 0.400.
* **STL Ajustado**: O modelo que utilizou o STL ajustado superou as expectativas, conseguindo harmonizar o comportamento dos dados sem distorção nas previsões. Este modelo alcançou um resultado de 0.255. Em alguns prédios, variações do STL ajustado e da decomposição clássica foram combinadas e alcançaram um NMAE de 2.430 em treinamento, o que sugere uma possível submissão futura que integre ambos os métodos onde houve sucesso.

Um filtro que ajusta as previsões aos dados de temperatura foi desenvolvido, proporcionando uma melhoria nos resultados ao reduzir o melhor NMAE para 0.240. Dependendo dos resultados das submissões finais, este método poderá ser adicionado em futuras submissões. Contudo, sua complexidade e o aumento da necessidade de dados são vistos como desvantagens, especialmente considerando a pequena redução no score.

Além disso, diversos outros métodos foram testados, incluindo regressão e várias abordagens não supervisionadas de NILM e agrupamento de dados. No entanto, nenhum deles proporcionou resultados melhores que o STL ajustado e, na maioria dos casos, adicionaram complexidade excessiva ao processo, levando à sua exclusão.

**Análise de Erros e Limitações Observadas**: Uma limitação significativa encontrada é a necessidade de dados estarem limpos, sincronizados e imputados para uma decomposição dos dados poder ser feita sem erros ou distorções. Esse processo demanda um trabalho de limpeza ou, como foi escolhido, a seleção das semanas de referência, que, na minha opinião, faz mais sentido e é mais vantajoso do que a limpeza dos dados. Dependendo do prédio e do cenário específico, optar pela limpeza dos dados ou pela seleção das semanas de referência pode resultar em desempenhos melhores ou piores, porem na grande maioria dos casos a seleção das semanas de referencia vence essa disputa.

Neste estudo, optou-se pelo pipeline que oferece os melhores resultados para cada prédio, apesar de ser mais trabalhoso e poder necessitar de ajustes adicionais. Contudo, esse pipeline poderia ser totalmente automatizado, mantendo consistência nos resultados, especialmente se os dados forem de alta qualidade e estiverem bem sincronizados e limpos. Esta é a principal limitação do algoritmo: quando essas condições não são atendidas, torna-se necessário um processo de limpeza ou ajuste das semanas de referência para cada prédio com parâmetros específicos.

## 7. Reprodutibilidade -ok

* **Instruções Passo a Passo:**
  1. **Clone o Repositório:** Inicie clonando o repositório do GitHub onde o código do projeto está hospedado. O link pode ser encontrado na seção 'Implementação do Algoritmo' deste relatório.
  2. **Instalação de Dependências:** Instale todas as bibliotecas necessárias usando o comando pip install -r requirements.txt, que está incluído no repositório. Este arquivo contém todas as bibliotecas Python necessárias para executar o projeto.
  3. **Configuração de Dados:** Baixe os dados conforme descrito na seção 'Dados' e coloque-os na pasta data/train\_public.
  4. **Ajuste de variáveis:** Caso seja necessário ajuste a variável PATH no arquivo code/\_4\_main.py
  5. **Execução do Script:** Execute o script do arquivo \_4\_main.py
  6. **Avaliação dos Resultados:** Use os gráficos gerados para avaliar a decomposição de cada prédio baseada em sua serie temporal, plugs e temperatura dependente.
* **Configuração do Ambiente:**
  1. **Hardware:** Idealmente, um computador com pelo menos 16 GB de RAM e um processador moderno (i5 ou superior) para processamento eficiente.
  2. **Sistema Operacional:** O código é compatível com Windows, macOS e Linux.
  3. **Dependências:** Python 3.8 ou superior, Pandas, NumPy, Statsmodels, Bokeh, e Metpy. Detalhes específicos e versões estão listados no arquivo requirements.txt ou no arquivo environment.yml.
* **Lista de Verificação para Reprodutibilidade:**
  1. Código fonte clonado do repositório GitHub.
  2. Todas as dependências instaladas conforme requirements.txt.
  3. Dados baixados e armazenados na localização correta.
  4. Scripts executados na ordem correta.
  5. Ambiente de hardware e software configurado conforme as especificações.
  6. Resultados avaliados analisando os gráficos de decomposição.

## 8. Discussão -ok

* **Interpretação dos Resultados:** Os resultados demonstram que o modelo STL ajustado proporciona desagregações de energia consistentes e robustas. A capacidade do STL de se ajustar às tendências e sazonalidades dos dados permite uma desagregação eficaz mesmo quando a qualidade dos dados não é ideal. Isto é particularmente verdadeiro em casos onde apenas o aquecimento precisa ser desagregado, e as estações do ano têm um papel crucial, identificando semanas em que o aquecimento é predominantemente desligado, servindo como excelentes referências. A precisão alcançada, com um NMAE de 0.255, ressalta a eficácia do modelo em capturar padrões complexos de consumo, apesar das limitações dos dados. Esse sucesso reforça o potencial da modelagem não supervisionada como uma ferramenta valiosa na gestão de energia em edifícios, mesmo quando apenas dados de medidores principais estão disponíveis.
* **Limitações do Estudo:** Apesar dos resultados promissores, o estudo enfrentou algumas limitações que poderiam impactar a generalização dos resultados:
  + O modelo pode encontrar dificuldades principalmente se o prédio tiver um sistema que fica ligado em estações que não necessitam de resfriamento ou aquecimento. Nesses casos, é mais desafiador realizar a desagregação das semanas de referência, necessitando de uma nova etapa, descrita no Apêndice C, que envolve uma desagregação global usando multiplicadores das semanas de referência para encontrar o valor ideal de desagregação dessas semanas, aumentando a complexidade do algoritmo.
  + Se os dados forem de baixa qualidade, especialmente se não estiverem sincronizados ou apresentarem semanas de comportamento atípico, como total ausência de ocupação durante férias, isso pode distorcer as semanas de referência e prejudicar substancialmente a desagregação do algoritmo. Nestes casos, torna-se necessário uma nova etapa para ignorar esses dados atípicos.
* **Sugestões para Trabalhos Futuros:** Em futuras investigações, seria benéfico tentar incorporar dados de condições climáticas na desagregação. Também seria produtivo desenvolver um modelo automático de avaliação da qualidade dos dados, especialmente para identificar semanas atípicas, pois isso poderia melhorar significativamente os resultados do algoritmo em cenários de automação completa do pipeline. Investir mais tempo na análise de sazonalidade, tentando dividir os dados entre ocupação e desocupação e realizar desagregações separadas para cada cenário, é outra área promissora. Embora tentativas anteriores não tenham sido frutíferas, persistir nessa abordagem pode valer a pena e potencialmente elevar a qualidade do algoritmo. Por fim, um teste com um maior número de edifícios seria interessante para determinar em que cenários o modelo se ajusta melhor e identificar onde são necessárias melhorias, como as sugeridas no Apêndice C.

## 9. Conclusão - ok

**Summary of findings**: Este estudo detalha o sucesso na aplicação do modelo STL para a desagregação de energia em edifícios, mesmo sob limitações de qualidade dos dados. A habilidade do modelo de se adaptar a tendências e sazonalidades diversas destacou-se como um elemento central para a obtenção de desagregações robustas e consistentes, particularmente eficazes em cenários onde apenas o aquecimento é desagregado e as variações sazonais desempenham um papel crucial. A precisão demonstrada pelo NMAE de 0.255 enfatiza a competência do modelo em capturar padrões complexos de consumo, oferecendo um método promissor para a gestão eficiente de energia.

Entretanto, algumas limitações foram identificadas, tais como dificuldades de desagregação em edifícios com sistemas que permanecem ligados em estações não dependentes de aquecimento ou resfriamento e a potencial distorção causada por dados de baixa qualidade ou semanas atípicas. Estes desafios sugerem a necessidade de ajustes nos algoritmos, como a introdução de multiplicadores das semanas de referência para melhorar a precisão da desagregação nessas condições particulares.

**Recommendations based on results**: Com base nos resultados obtidos, recomenda-se a incorporação de dados climáticos para aprimorar a precisão das desagregações e a criação de um modelo automático para avaliação da qualidade dos dados, focando na identificação de períodos atípicos. Além disso, sugere-se a persistência na exploração da divisão dos dados entre períodos de ocupação e desocupação, realizando desagregações específicas para cada cenário, o que poderia melhorar significativamente a eficácia do modelo.

Estudos adicionais envolvendo múltiplos prédios poderiam elucidar em quais cenários o modelo se mostra mais eficiente. Por exemplo, em edifícios que, durante estações como verão e inverno, possuem semanas de referência que representam bem o consumo de plugues e iluminação, o modelo pode realizar uma desagregação muito eficaz da energia de aquecimento ou resfriamento. Isso permitiria entender melhor quais configurações de edifício e de uso se beneficiam mais da aplicação do modelo STL.

Para cenários onde não se consegue obter semanas de referência claras, em que a energia de plugues e iluminação não são adequadamente desagregadas na decomposição, seria necessário o desenvolvimento de etapas adicionais no modelo. Uma dessas etapas poderia ser a aplicação de multiplicadores ajustados às características de consumo específicas de cada prédio, obtidos através de uma análise de picos e vales, ou de máximas e mínimas globais de consumo como detalhado no apêndice c.

## 10. Referências -ok

1. ACM Reference Format: Narges Zaeri, H. Burak Gunay, and Araz Ashouri. 2022. Unsupervised Energy Disaggregation Using Time Series Decomposition for Commercial Buildings. In 6th International Workshop on NonIntrusive Load Monitoring (NILM ’22), November 9–10, 2022, Boston, MA, USA. ACM, New York, NY, USA, 5 pages. [https://doi.org/10. 1145/3563357.3566155](https://doi.org/10.%201145/3563357.3566155)  
   https://nilmworkshop.org/2022/proceedings/nilm22-final97.pdf
2. Unsupervised Adaptive Event Detection for Building-Level Energy Disaggregation Karim Said Barsim, Roman Streubel, and Bin Yang Institute for Signal Processing and System Theory University of Stuttgart Email: {karim.barsim,roman.streubel,bin.yang}@iss.uni-stuttgart.de <https://www.researchgate.net/profile/Karim-Said-Barsim/publication/266613706_Unsupervised_Adaptive_Event_Detection_for_Building-Level_Energy_Disaggregation/links/55cc87b708aea2d9bdce4454/Unsupervised-Adaptive-Event-Detection-for-Building-Level-Energy-Disaggregation.pdf>
3. <https://sci-hub.se/https://doi.org/10.1109/TIM.2017.2700987> Event-Based Energy Disaggregation Algorithm for Activity Monitoring From a Single-Point Sensor José Alcalá, Jesús Ureña, Senior Member, IEEE, Álvaro Hernández, Senior Member, IEEE, David Gualda
4. <https://sci-hub.se/10.1109/ICPS51807.2021.9416607>  
   Joint Energy Disaggregation of Behind-the-Meter PV and Battery Storage: A Contextually Supervised Source Separation Approach
5. <https://sci-hub.se/https://doi.org/10.1016/j.enbuild.2015.03.048>  
   HVAC terminal hourly end-use disaggregation in commercial buildings with Fourier series model Ying Ji, Peng Xu∗, Yunyang Ye

## 11. Apêndices

Os apêndices servem para fornecer detalhes adicionais que são importantes para a compreensão completa do projeto, mas que podem ser muito extensos ou detalhados para incluir no corpo principal do relatório. Os materiais de suporte podem incluir:

* **Apêndice A: Dicionário de Dados**
  + Descrição detalhada de cada variável nos datasets utilizados, incluindo tipo de dados, significado e qualquer codificação ou transformação aplicada.
* **Apêndice B: Códigos e Scripts Completos**
  + Scripts de análise e modelagem que foram utilizados no projeto. Isso inclui códigos para limpeza de dados, análise exploratória, treinamento de modelos e avaliação de desempenho.
* **Apêndice C: Resultados Detalhados**
  + Tabelas e gráficos adicionais que mostram os resultados detalhados do modelo, incluindo desempenho em diferentes subsets de dados e análises de erro.
* **Apêndice D: Documentação de Configuração de Hardware e Software**
  + Instruções detalhadas sobre a configuração do ambiente de desenvolvimento e execução, incluindo especificações de hardware, versões de software e bibliotecas utilizadas.
* **Apêndice E: Revisão de Literatura Adicional**
  + Uma lista detalhada de estudos e artigos que forneceram base teórica para o desenvolvimento do projeto, que não foram diretamente citados no corpo do relatório.
* **Apêndice F: Formulários de Consentimento e Aprovação Ética**
  + Caso o projeto envolva dados humanos ou qualquer outra consideração ética, incluir cópias dos formulários de consentimento e aprovações éticas relacionadas.
* **Apêndice G: Correspondência com Entidades Envolvidas**
  + Cópias de qualquer correspondência relevante com stakeholders, patrocinadores do projeto ou outras entidades relacionadas ao desenvolvimento e implementação do projeto.

### Apêndice A - **Parâmetros Utilizados - ok**

**Parâmetros Utilizados para Configuração e Otimização do Modelo por Edifício: Descrição Detalhada**

Este apêndice tem como objetivo explicar em detalhes os parâmetros utilizados para configurar e otimizar o modelo aplicado a cada edifício.

**1. Parâmetros de Decomposição de Séries Temporais**

Os seguintes parâmetros são empregados para decompor as séries temporais e determinar o melhor ajuste para os dados:

Texto

Descrição gerada automaticamente

* **Seasonality (Sazonalidade) (Int 1-9999):**
  + **Descrição:** Define o período sazonal considerado na análise.
  + **Como Configurar:** Selecionar o período que corresponde à sazonalidade esperada (e.g., 168 para semanal, 24 para diária).
* **Multiplier (Multiplicador) (Float 0-1):**
  + **Descrição:** Aplica um fator de multiplicação para ajustar a amplitude do tamanho dos dados, visando encontrar o fator ideal para componentes específicos como plugs e eletricidade.
  + **Como Configurar:** Caso seja necessário, valores entre 0.5, a 0.7, caso não seja usado multiplica-se por 1.
* **STL (Decomposição STL) (String ‘stl’, ‘stl\_ajusted’, ‘classic’):**
  + **Descrição:** Especifica o tipo de decomposição a ser aplicado aos dados.
  + **Como Configurar:** Selecionar o método mais adequado a partir de estudos de eda.
* **Parâmetros do STL:**
  + **Descrição:** Permite a personalização detalhada da decomposição STL através de configurações específicas.
    - **Suavização de Tendência (stl\_trend) (Int 1-9999):** Controla o grau de suavização aplicado ao componente de tendência.
    - **Sazonalidade (stl\_seasonal) (Int 1-9999):** Ajusta a janela de suavização para o componente sazonal.
    - **Filtro Passa-Baixa (stl\_low\_pass) (Int 1-9999):** Determina a filtragem adicional para separar componentes de alta frequência.
    - **Robustez (stl\_robust)(Bool):** Habilita a decomposição robusta para reduzir o impacto de outliers.
* **Seleção de Encaixe Semanal (Week Plug Selection):**

Texto

Descrição gerada automaticamente Gráfico, Diagrama

Descrição gerada automaticamente

A Seleção de Encaixe Semanal é um conjunto de parâmetros utilizados no algoritmo de decomposição para selecionar semanas de referência de forma robusta. Esses parâmetros permitem lidar eficazmente com outliers e ruídos nas séries temporais, como períodos de férias, sem a necessidade de um tratamento de dados exaustivo.

O algoritmo utiliza seis parâmetros principais, fornecidos em formato de lista, que controlam a forma como as semanas de referência são selecionadas e como os dados são normalizados.

**Parâmetro 0 – Quantil Mínimo (week\_lower\_quantile) (Float 0 a 1)**

* + - **Descrição**: Define o quantil utilizado para determinar o valor mínimo considerado em cada semana de referência. Por exemplo, um valor de 0,01 indica que será utilizado o quantil de 1% dos dados agrupados por ano e semana do prédio.
    - **Finalidade**: Ajustar o cálculo do mínimo semanal para reduzir o impacto de outliers inferiores.
  + **Parâmetro 1 – Quantil Máximo (week\_upper\_quantile) (Float 0 a 1)**
    - **Descrição**: Define o quantil utilizado para determinar o valor máximo considerado em cada semana de referência. Por exemplo, um valor de 0,99 indica que será utilizado o quantil de 99% dos dados agrupados por ano e semana do prédio.
    - **Finalidade**: Similar ao Parâmetro 0, mas focado em outliers superiores, ajustando o máximo semanal para refletir com maior precisão a distribuição dos dados.
  + **Cálculo da Mediana Semanal (week\_median):** A week\_median é uma medida central ajustada que considera tanto o quantil superior quanto o inferior. Vai ser usada no cálculo dos parâmetros 2 e 3.  
    week\_median = (week\_upper\_quantile + week\_lower\_quantile)/2
  + **Parâmetro 2 – Quantil Mínimo para Seleção de Semanas (Float 0 a 1)**
    - **Descrição:** Estabelece o limiar mínimo do week\_median para que uma semana seja incluída como referência. Por exemplo, um valor de **0,05** seleciona semanas cujo week\_median está acima do quantil de 5%.
    - **Finalidade:** Filtrar semanas com valores anormalmente baixos, possivelmente decorrentes de ruídos ou períodos atípicos.
  + **Parâmetro 3 – Quantil Máximo para Seleção de Semanas (Float 0 a 1)**
    - **Descrição:** Define o limiar máximo do week\_median para a inclusão de semanas de referência. Um valor de **0,25** inclui semanas com week\_median abaixo do quantil de 25%.
    - **Finalidade:** Excluir semanas com valores superiores aos típicos para uma semana de referência, selecionando apenas aquelas com pouco ou nenhum uso de energia para ajustes de temperatura.
  + **Observação:** A configuração dos Parâmetros 2 e 3 deve ser ajustada conforme o comportamento específico dos dados de cada prédio. Recomenda-se a realização de uma Análise Exploratória de Dados (EDA) e ajustes visuais para determinar os quantis mais apropriados.
  + **Parâmetro 4 – Quantil Mínimo para Normalização (Float 0 a 1)**
    - **Descrição:** Define como os outliers serão tratados durante a normalização das previsões, especificamente o valor mínimo considerado em cada semana de previsão. Um valor de **0,01** indica que o quantil de 1% será utilizado como mínimo.
    - **Finalidade:** Auxiliar na normalização de dados altamente ruidosos ou com outliers inferiores, tornando o processo mais eficaz.
  + **Parâmetro 5 – Quantil Máximo para Normalização (Float 0 a 1)**
    - **Descrição:** Similar ao Parâmetro 4, mas referente ao valor máximo na normalização das previsões. Um valor de **0,99** utiliza o quantil de 99% como máximo.
    - **Finalidade:** Ajustar a normalização na presença de outliers superiores, garantindo consistência na escala dos dados.
  + **Considerações Práticas:** Embora seja possível utilizar valores padrão (por exemplo, 0,01 e 0,25 nos parâmetros 2 e 3 e 0.01 e 0.99 nos parâmetros 0,1,4,5) com bons resultados gerais, é interessante adaptar os parâmetros às características específicas dos dados disponíveis para otimizar o desempenho do algoritmo.
* **Seasonal Trend Power (Potência da Tendência Sazonal):**

**** Gráfico, Histograma

Descrição gerada automaticamente

A **Potência da Tendência Sazonal** é um componente essencial do nosso modelo que ajusta a influência da sazonalidade nos dados analisados. Este parâmetro permite uma adaptação mais precisa da altura da sazonalidade aos dados observados e modifica a forma como a base de cálculo é estabelecida, otimizando a representação das flutuações sazonais nas previsões.

O ajuste da potência da tendência sazonal é realizado por meio de dois parâmetros principais que configuram a adaptação e a base para o cálculo da normalização dos dados.

* + **Parâmetro 0 – Valor de adaptação da tendencia (Float 0 a 1)**
    - **Descrição**: Define a intensidade com que a altura dos dados se adapta em relação às semanas que apresentam maiores flutuações que as semanas de referência. Por exemplo, um valor de **0.1** significa que será adicionado 10% da intensidade das semanas mais expressivas aos valores máximos das semanas de referência.
    - **Finalidade**: Este parâmetro permite que o modelo ajuste de maneira flexível a altura da tendência sazonal de forma suavizada.
  + **Parâmetro 1 – Tipo de base dos dados que vai utilizar (String ‘norm\_min’ or ‘Base\_min’)**
    - **Descrição**: Especifica o tipo de base mínima que será utilizada para a normalização das semanas de referência na construção das previsões.
      * **norm\_min**: Usa a mínima de cada semana como sua base.
      * **base\_min**: Usa a mínima das semanas de referência.
    - **Finalidade**: A escolha do tipo de base permite adaptar as previsões de forma a refletir de maneira mais precisa e adaptativa a mínima da normalização a variação dos dados.

**2. Limpeza e Preparação dos Dados**

Texto

Descrição gerada automaticamente

Para garantir que os dados sejam adequados para a decomposição e análise, os seguintes parâmetros de limpeza são utilizados:

* **Country (País) (String do pais):**
  + Especifica o contexto geográfico dos dados na função de limpeza de feriados.
* **Ano dos Dados (lista dos anos):**
  + Indica os anos a serem usados nas funções de limpeza de feriados e intervalos.
* **Clear Holidays (Bool):**
  + Usa ou não a função de limpeza de feriados da biblioteca holidays.
* **Interval (Intervalo) (Lista de tuplas):**
  + Tuplas de intervalos a serem removidos na série temporal.
* **Holidays (Feriados Específicos) (Lista de tuplas):**
  + Tuplas com feriados a serem removidos dos dados.

### Apêndice B - **Decomposição clássica -ok**

Este apêndice tem como objetivo explicar a formula utilizada na decomposição clássica e o método utilizado para otimizar a altura e base da decomposição.

1 - Decomposição clássica:

O modelo de decomposição clássico é descrito em detalhes no artigo ref1, basicamente o modelo faz uma decomposição da serie temporal e usa a sazonalidade + a tendencia mínima para gerar a previsão de plugs e luzes eletricidade.

plugs = sazonalidade + tendencia.min()

Decomposição = energia total – plugs

Gráfico, Histograma

Descrição gerada automaticamente

Fig 1: Decomposição clássica sem nenhum ajuste de altura e base

Esse modelo gera resultados interessantes, porém o fato de usar a tendencia mínima, faz com que os dados tenham de estar perfeitos em relação a outliers e com muita qualidade, senão é necessário um ajuste manual da altura da sazonalidade e principalmente do local da base das previsões, como mostra a imagem acima.

2- Modelo de otimização e ajuste da altura e base da otimização clássica:

A primeira ideia para resolver esse problema foi um sistema de otimização, que selecionava as semanas de referência e então ajustava a altura e a base dos dados a essa semana de referência usando a otimização.

Semana de referencia = Dados das semanas entre o quartil 0 a 0.25 dos dados.

**Otimização que gera o ajuste da altura e base:**

*error\_exponent = 10*

*error = np.mean(np.abs((*Semana de referencia) *- prediction) \*\* (1 / error\_exponent))*

Utilizando esse algoritmo descrito no código por essa função (optimize\_parameters) é feita a primeira ideia de otimização usando as semanas de referencia, que depois foi evoluído para o sistema que faz a normalização e ajuste de tamanho dos dados da decomposição stl, dentro do modelo do stl ajustado.

Gráfico, Histograma

Descrição gerada automaticamente

Fig 2: Decomposição clássica com otimização que encontra melhores valores de ajuste para altura e base das previsões baseado na semana de referencia.

Nesta imagem podemos ver o procedimento de de otimização que ajusta o tamnanho das predições ao tamanho dos dados na reference week diminuindo o erro e gerando previsões de plugs muito bem ajustadas as dados e também um desagregação bonita e elegante da energia usada para temperatura.

Tela de computador

Descrição gerada automaticamente

Fig 3: Mostra a otimização da decomposição clássica a partir das semanas de referencia para encontrar o melhor ajuste.

Esta imagem mostra a otimização acontecendo e seu resultado final que encontra um fator de multiplicação de ajuste dos dados em 0.6999, e um ajuste na altura que usa o quntile 0.5801 dos dados que é mostrado na prediction from decomposition dos plugs na imagem do meio da figura.

Esse método se mostrou extremamente eficiente em situações em que os dados são de ótima qualidade e a sazonalidade estáveis, também se mostrou uma ótima forma de descobrir e otimizar o tamanho ideal de altura e base dos dados, para depois aplicar o stl ajustado com parâmetros similares de altura e base.

Um exemplo da versão final deste prédio utilizando o stl ajustado, baseado nos tamnhos otimizados pelo modelo.

Gráfico, Histograma

Descrição gerada automaticamente

Fig 4: Mostra a decomposição usando stl ajustado, muito similar a decomposição clássica, porque usa seus parâmetros de ajuste para encontrar os pontos ideias da altura e base na week reference.

Perceba como a altura da reference week e a base tem valores bem parecidos, o que gera previsões muito similares a previsão da decomposição clássica, porem no modelo do stl ajustado temos maior controle da forma como a sazonalidade vai ser usada nas previsões.

O calculo que gera a previsão dos plugs se encontra na função (calculate\_prediction) usando a decomposição clássica.

Texto

Descrição gerada automaticamente

Fig 5: Mostra como os dados de ajuste de tendencia e quantile vao ser implementados nos dados de decomposição clássica para gerar as previsões de plugs.

### Apêndice C - Seleção de Multiplicador -ok para Aquecimento e Resfriamento

#### Tipo de dia – ocupação e desocupação

A primeira parte do algoritmo faz uma separação importante entre períodos de ocupação e desocupação usando o hhm. Abaixo, podemos ver as imagens da separação horária em ocupação e desocupação dos prédios L10 e L09.

Essa ideia foi sugerida no artigo de referência 5, em que é proposto que, se forem separados os dados por dias de ocupação/trabalho e dias de desocupação/fim de semana, ao invés de usar a sazonalidade semanal, poderia ser usada a sazonalidade diária. Essa ideia realmente faz muito sentido. Eu cheguei a trabalhar nela, porém os resultados semanais ainda foram melhores em meus estudos. Contudo, ainda tenho minhas dúvidas se, com maior empenho nessa ideia, não poderia encontrar uma forma de melhorar os resultados ou criar novas formas de resolver o problema.

Algumas tentativas de aplicar essa abordagem são detalhadas abaixo. Este método de separação pode revelar várias maneiras de realizar a desagregação, fornecendo insights sobre o valor ideal do multiplicador global de desagregação em contextos complexos de aquecimento e resfriamento. Um exemplo disso são as semanas de referência nos prédios L09 e L10, que apresentam sistemas potencialmente mais complexos. Devido à falta de informações completas, posso apenas especular que esses prédios operam com um sistema HVAC que permanece ligado de forma intermitente, inclusive sob condições ideais de temperatura. Vale ressaltar que essa suposição é baseada em especulação, já que dados específicos sobre a operação dos sistemas não foram fornecidos. Essa hipótese sugere que os sistemas estão frequentemente ativos, integrados à ventilação e a outras demandas energéticas, conforme descrito no artigo de referência 5, que detalha os componentes de um terminal HVAC. Esta análise segue minha linha de raciocínio ao explorar as possíveis dinâmicas de consumo energético nos prédios mencionados.

Gráfico, Gráfico de dispersão

Descrição gerada automaticamenteGráfico, Gráfico de dispersão

Descrição gerada automaticamente

Fig1: Observa-se que a separação feita pelo hhm considerou os **horários de ocupação**, o que, neste cenário, mostrou-se mais adequado do que a separação por dias de ocupação. Por outro lado, testei um algoritmo que adicionava uma etapa extra, **separando os dias de ocupação** dos dias de desocupação. Contudo, essa abordagem não foi tão eficaz quanto a separação por horários de ocupação. Além disso, separar por dias pode introduzir erros em feriados e dias atípicos, nos quais o padrão de ocupação não segue o convencional, comprometendo a precisão do modelo.

#### Modelo 1 - Busca pelos valores de multiplicador usando o quantils do consumo de energia altamente correlacionados com a temperatura do ar

Um modelo que se mostrou logicamente viável para encontrar um valor de multiplicador global de desagregação, ou seja, uma média de desagregação global do uso de plugues para posterior análise do uso de aquecimento ou resfriamento, foi a estratégia de separar os dados por horário, estação do ano e também por ano. A seguir, verificava-se se, naquele horário e estação, havia uma alta correlação com a temperatura e a energia total consumida. Caso houvesse essa correlação, poderíamos pegar a amplitude desses dados determinando assim a porcentagem de energia usada para plugues e iluminação.

No GitHub, há um arquivo chamado apêndice c que demonstra exatamente como o código foi aplicado.

Agora, farei uma breve explicação sobre a lógica e como o modelo é calculado:

1. Esperamos que um determinado horário tenha uma alta correlação com a temperatura e pegamos seu valor mínimo e máximo para gerar a amplitude.
2. Depois, dividimos esse valor pela amplitude total e subtraímos um para chegar ao valor de plugues e iluminação.
3. Foram usados quantis, ao invés de máximas e mínimas, para determinar os valores extremos, considerando-se as distorções causadas por outliers.
4. A fórmula utilizada foi: Plugs = 1-((quantil09 - quantil01)/quantil09)

O procedimento foi aplicado tanto para o prédio L9 quanto para o prédio L10, os valores de Plugues variaram entre 0.5 e 0.6, sugerindo que, em média, o sistema de aquecimento e resfriamento consome de 40% a 50% da energia, dependendo do horário, ocupação e prédio.

Reconhecemos que essa não é a forma ideal de realizar o cálculo e que, por exemplo, durante períodos de desocupação ou noturnos, esses valores provavelmente são menores. No entanto, o objetivo aqui é encontrar uma média para todos os dados e não um modelo completo de desagregação, embora seja possível tentar criar um modelo completo de desagregação a partir dessa lógica no futuro. Fiz algumas tentativas, mas elas não produziram resultados tão bons quanto a desagregação a partir da decomposição stl ajustada.

L09

Interface gráfica do usuário, Gráfico, Aplicativo

Descrição gerada automaticamente

Por exemplo, no prédio L9, podemos observar que somente alguns horários alcançaram uma correlação maior que 70%. Para esses momentos, a análise média revelou um valor de plug de 0.60, com desvio padrão de 0.03, um valor mínimo de 0.55 e um máximo de 0.65. Grosseiramente, poderíamos afirmar que, durante períodos de ocupação, cerca de 60% do consumo de energia é utilizado por plugs e iluminação, variando entre 55% e 65%.

L10

Interface gráfica do usuário, Gráfico

Descrição gerada automaticamente

No prédio L10, conseguimos identificar um número significativamente maior de horários com correlação elevada com a temperatura, sendo que a maioria dessas correlações superou 0.8. Em média, o valor para plugs e iluminação foi de 0.50, com um desvio padrão de 0.09, um valor mínimo de 0.37 e um máximo de 0.63. Embora tenhamos uma correlação maior, também observamos um desvio padrão mais elevado e uma variação maior entre os valores mínimo e máximo. Poderíamos dizer que, em média, os plugs e a iluminação desse prédio estão na faixa de 40% a 60%. No entanto, o valor que se mostrou realmente efetivo foi a média de 0.50.

#### Modelo 2 - Procura por picos e vales para tentar encontrar o valor dos plugs e iluminação

Outra abordagem para encontrar um valor mediano para semanas de referência, quando estamos trabalhando com um sistema HVAC, é a procura por eventos a partir de picos e vales. Na minha opinião, essa é uma técnica interessante para procurar por momentos em que é possível que algum equipamento ou sistema seja ligado ou desligado. Por exemplo, é útil para identificar momentos rápidos em que o sistema HVAC é desligado ou quando algum plug importante ou alguma iluminação é desativada. No entanto, fica difícil saber exatamente que tipo de equipamento está sendo ligado ou desligado numa frequência de 5 minutos ou 15 minutos, o que demanda um trabalho cuidadoso com esse tipo de modelo para garantir resultados mais consistentes.

Com isso em mente, tanto no edifício L9 quanto no edifício L10, procuramos por padrões que pudessem indicar momentos de ligamento ou desligamento de algum sistema importante. Preferencialmente, busquei pelos sistemas de aquecimento ou resfriamento, mas não exclusivamente, pois não é possível afirmar com certeza o que está sendo ligado ou desligado, principalmente trabalhando com frequências tão baixas. Cada um dos edifícios apresenta situações distintas; por exemplo, quando há desocupação no prédio L9, existem vales no gráfico de 5 minutos que podem indicar um desligamento de um sistema importante.

Os cálculos estão detalhados no código do apêndice C no GitHub. Abaixo, vou mostrar algumas imagens e demonstrar basicamente como os cálculos são feitos.

L09

Gráfico

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente

Como mencionado anteriormente, aqui podemos observar os vales que, na verdade, estão representados como picos. Isso porque a função scipy.signal.find\_peaks só detecta picos, então foi necessário inverter os vales para possibilitar a identificação. Este é um padrão interessante que, aparentemente, parece indicar um desligamento rápido de um sistema importante. Busquei encontrar todos esses vales e, a partir deles, gerar uma média de amplitude muito similar ao Modelo 1, mas agora utilizando esses momentos de desligamento rápido. No final do cálculo, chegamos a uma mediana dos plugs de 0.6000256065552783.

A fórmula utilizada para o cálculo dos plugs é a seguinte:

properties['plugs']= properties['prominences']/((properties['peak\_heights'] - properties['prominences'])\*-1)

L10

Gráfico, Histograma

Descrição gerada automaticamente

Gráfico

Descrição gerada automaticamente

No prédio L10, observamos um padrão distinto de picos que sugere momentos de rápido ligamento e desligamento de um sistema importante. Por exemplo, em dias de desocupação, como fins de semana ou horários noturnos, percebemos um pico significativo de 15 minutos, seguido pelo retorno rápido dos valores de energia ao normal. Esse pico sugere um momento em que o algum sistema é ativado e, por algum motivo, rapidamente desativado, embora seja difícil especular as causas sem informações adicionais.

Neste cenário, é possível calcular a amplitude desses picos para entender melhor a variação de energia. A mediana dos plugs, calculada para esses eventos, foi de 0.5025378015940775. A fórmula utilizada para calcular os plugs foi:

properties['plugs']= properties['prominences']/(properties['peak\_heights'] - properties['prominences'])