Aprendizado Não Supervisionado Aplicado à Análise de Furtos de Cabos de Cobre em Belo Horizonte

Daniel E. Santos[1], **Carlos A. Silva**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG)

danielias.santos@gmail.com, carlos.silva@ifmg.edu.br

Abstract. *Traditional hotspot mapping methods, such as Kernel Density Estimation (KDE), fail to capture the temporal dynamics of criminal phenomena, thereby limiting the comprehension of their spatial evolution. This work proposes a dynamic spatio-temporal analysis methodology for the detection of cable theft hotspots in Belo Horizonte, for application at the Integrated Operations Center (COP-BH). The approach segments data into temporal windows, applying five clustering techniques (K-Means, K-Medoids, HAC, DBSCAN, and HDBSCAN) and evaluating their performance using internal metrics (Silhouette, Davies–Bouldin, Calinski–Harabasz, and Density-Based Clustering Validation) and the Predictive Accuracy Index (PAI). The results indicate that density-based methods, such as HDBSCAN, identify patterns of hotspot emergence, dissipation, and displacement more precisely than KDE. Furthermore, it was observed that the algorithms possess advantages and disadvantages relative to the requirements of COP-BH, highlighting the critical need for the appropriate application of each method. The proposed methodology contributes to the intelligent monitoring of critical infrastructure, providing support to COP-BH and establishing a solid foundation for future integrations with predictive and video analytics systems.*

Resumo. *Métodos tradicionais de mapeamento de hotspots, como a Estimativa de Densidade por Kernel (KDE), não capturam a dinâmica temporal de fenômenos criminais, limitando a compreensão de sua evolução espacial. Este trabalho propõe uma metodologia de análise espaço-temporal dinâmica para detecção de hotspots de furto de cabos em Belo Horizonte, com aplicação no Centro Integrado de Operações (COP-BH). A abordagem segmenta os dados em janelas temporais, aplicando cinco técnicas de clusterização (K-Means, K-Medoids, HAC, DBSCAN e HDBSCAN) e avaliando seu desempenho por métricas internas (Silhouette, Davies–Bouldin, Calinski–Harabasz e Density-Based Clustering Validation) e pelo Índice de Acurácia Preditiva (PAI). Os resultados indicam que métodos baseados em densidade, como o HDBSCAN, identificam padrões de surgimento, dissipação e deslocamento de hotspots de forma mais precisa que o KDE. Foi verificado, também, que os algoritmos possuem vantagens e desvantagens no atendimento à necessidade do COP-BH, sendo estritamente necessária a aplicação adequada de cada um deles. A metodologia proposta contribui para o monitoramento inteligente de infraestrutura crítica, fornecendo suporte ao COP-BH e criando uma base sólida para futuras integrações com sistemas preditivos e analíticos de vídeo.*

1. Introdução

A análise de dados geoespaciais é um elemento primordial para o desenvolvimento e aplicação de estratégias de políticas públicas baseadas em evidências. O Centro Integrado de Operações de Belo Horizonte[1] (COP-BH), um Centro de Gestão Integrada segundo a Carta Brasileira para Cidades Inteligentes[2], é a entidade municipal responsável pela integração de informações e da atuação das Instituições envolvidas na resposta a problemas públicos de Belo Horizonte, e que também analisa os dados gerados por meio dos seus processos de trabalho. O resultado desta análise é usado para retroalimentar e direcionar sua estratégia de atuação, conforme o seu Modelo de Gestão Integrada [COP-BH 2019]. O COP-BH viabiliza a integração entre as instituições e agências por meio de seis linhas de atuação: *Monitoramento da Cidade, Pronta Resposta, Gestão de Crises, Operações Integradas, Gestão de Eventos e Prevenção de Problemas.*

Destaca-se aqui a linha de atuação de Monitoramento da Cidade, que possui três vertentes: o monitoramento de sensores não especialistas, como câmeras; o monitoramento de sensores especialistas, tais como os meteorológicos; e o monitoramento de fontes de inteligência. O objetivo final desta linha de atuação é permitir que o COP-BH aja preventivamente a fim de evitar problemas públicos, ou responda adequadamente a estes, minorando as suas consequências. Uma das formas que utiliza os dados para orientar sua estratégia de atuação é identificar áreas de concentração de problemas públicos, ou *hotspots*, e utilizar isso como evidências que potencializem a atividade de Monitoramento da Cidade, bem como a alocação otimizada de recursos operacionais.

Um dos mais notórios problemas públicos que o COP-BH analisa é o furto de cabos de cobre, pois gera não apenas prejuízos financeiros diretos, mas também interrupções significativas em serviços essenciais para a população. Pode-se citar diversos problemas que o furto de cabos causa, tais como a interrupção de serviços básicos[3], do acesso à Internet[4], e de serviços de saúde[5], além de transtornos no tráfego e transporte público[6]. Além disso, há também o impacto econômico, pois o restabelecimento destes serviços demanda o sobregasto com reparos na infraestrutura[7]. Pequenos comerciantes e prestadores de serviços também sofrem impactos econômicos decorrentes da interrupção destes serviços essenciais ao funcionamento dos seus estabelecimentos[8]. Cita-se, ainda, a exposição a riscos à integridade física de pessoas que furtam cabos[9].

---

[1]https://prefeitura.pbh.gov.br/seguranca/copbh

[2]https://www.gov.br/cidades/pt-br/acesso-a-informacao/acoes-e-programas/desenvolvimento-urbano-e-metropolitano/projeto-andus/carta-brasileira-para-cidades-inteligentes

[3]https://www.cemig.com.br/release/furto-de-fios-de-cobre-provoca-interrupcao-da-oferta-de-servicos-publicos-causa-transtornos-e-gera-prejuizo-para-a-populacao

[4]https://noticias.r7.com/minas-gerais/balanco-geral-mg/videos/furtos-de-cabos-deixa-moradores-e-comerciantes-sem-internet-na-regiao-nordeste-de-bh-16092023/

[5]https://g1.globo.com/mg/minas-gerais/noticia/2022/03/02/apos-furto-de-cabos-na-santa-casa-de-bh-pacientes-tem-sessoes-de-radioterapia-desmarcadas.ghtml

[6]https://www.otempo.com.br/cidades/2024/7/29/metro-de-bh-tem-problemas-tecnicos-por-furto-de-cabos-e-atrasa-c

[7]https://www.em.com.br/gerais/2024/04/6828901-prejuizo-milionario-que-atrapalha-todo-mundo.html

[8]https://www.otempo.com.br/cidades/2025/6/25/furto-de-cabos-empresa-fica-sem-luz-duas-vezes-em-um-mes-no-serra-em-bh

[9]https://www.em.com.br/app/noticia/gerais/2023/03/04/interna_gerais,1464614/homem-tenta-furtar-fios-de-cobre-e-morre-eletrocutado-em-bh.shtml

Diante disso, pode-se reconhecer a relevância do problema do furto de cabos, e já existem esforços para resolvê-lo. O COP-BH desenvolveu um processo de análise de dados sobre problemas públicos de forma geral, que inclui o furto de cabos, e é usado para elaborar um roteiro de monitoramento. O roteiro foi concebido para apoiar as equipes de monitoramento, uma vez que a capacidade humana de processar fluxos contínuos de informação degrada-se com o tempo devido à fadiga e à perda de vigilância. Neste sentido, o roteiro tem o objetivo de sugerir quais os principais problemas públicos devem ser priorizados para monitoramento, os períodos de maior relevância para executar o monitoramento por câmeras, e quais as câmeras que deveriam ser monitoradas.

A partir deste roteiro, as equipes de monitoramento teriam um guia de referência que poderia aprimorar o seu trabalho no sentido da eficiência – pois conheceriam quais as câmeras que deveriam priorizar no monitoramento e quando deveriam ser monitoradas; e da eficácia – pois isso poderia aumentar a probabilidade de visualizar algum problema público durante o monitoramento. Portanto, a existência de um processo de trabalho no COP-BH que objetiva a elaboração de um roteiro de monitoramento de furtos de cabos justifica a importância do tema, pois demonstra que a instituição reconhece o impacto nocivo à cidade, e busca ativamente combatê-lo.

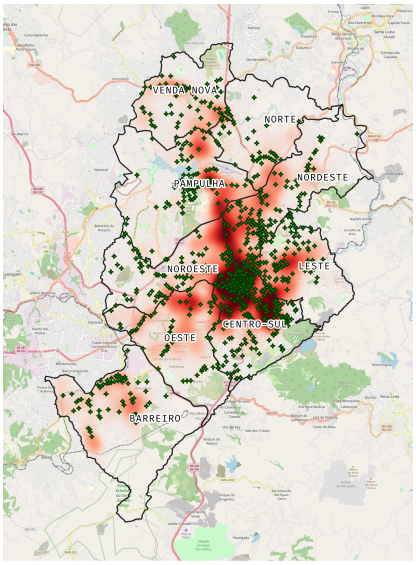


Figura 1. Roteiro de Monitoramento: Mapa de calor (KDE), de locais de furtos de cabos em Belo Horizonte, com sobreposição de câmeras públicas. Criado com o QGIS.

Contudo, a abordagem atual, usada para elaborar o roteiro de monitoramento, possui deficiências críticas que limitam sua eficácia. A metodologia atual, baseada na agregação de todo o histórico de dados em um único mapa de calor, demonstrado na Figura 1 (elaborado utilizando a ferramenta QGIS[10]) se mostra vulnerável a falácias estatísticas, como a falácia ecológica (*"Ecological Fallacy"*) [Robinson 2009]. A Estimativa de Densidade por Kernel (KDE) utilizada pelo COP-BH, não é estatisticamente significativa, além de depender de parâmetros de ajuste subjetivos para determinar a densidade dos dados [Kalinic and Krisp 2018]. Tais abordagens podem mascarar a natureza dinâmica dos

[10]https://qgis.org/project/overview/

dados, ocultando tendências, sazonalidades, mudanças estruturais nos dados e no padrão espacial das observações, que são essenciais à análise dos padrões de problemas públicos ora tratados [Chainey et al. 2008].

Acrescenta-se, ainda, o efeito de deslocamento geográfico (*"displacement"*) conceituado por Barr *et al.* [Barr and Pease 1990], que explorou as diferentes formas como o crime se move em resposta a ações de prevenção, conforme demonstrado na Figura 2. Além disso, a literatura aponta limitações em métodos que assumem formatos geométricos predefinidos para os *clusters*, uma vez que os padrões criminais frequentemente seguem a topologia irregular do ambiente urbano [Tavares 2009].

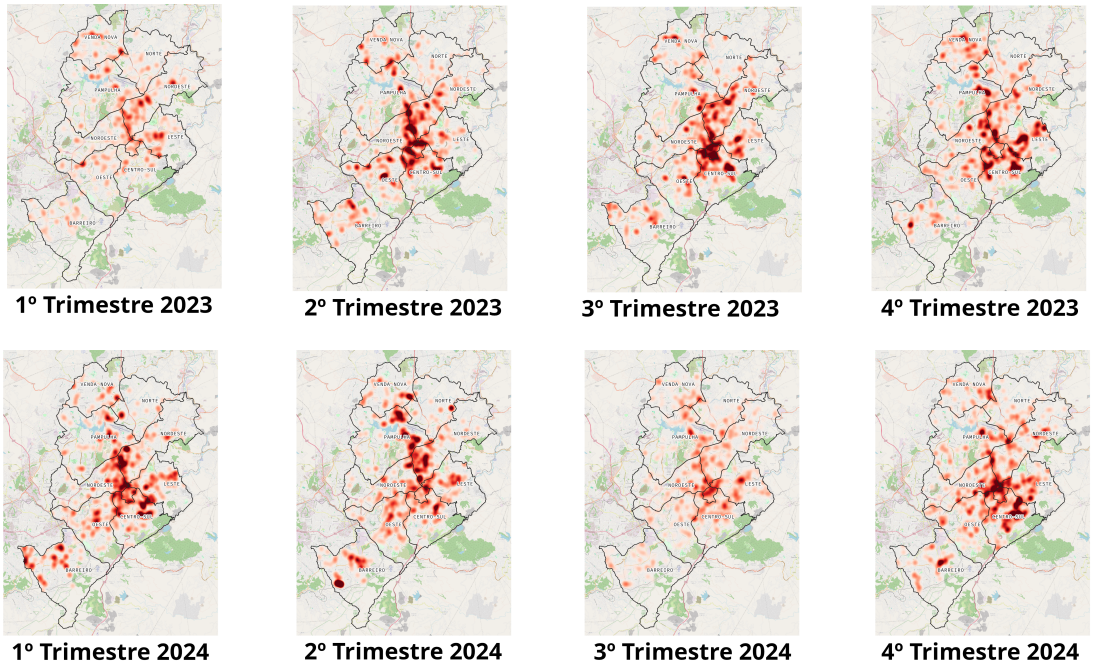


Figura 2. Comparação de mapas de calor (KDE), das observações de furtos de cabo em Belo Horizonte, dos anos de 2023 e 2024, por Trimestre. Criado com o QGIS.

Estes fatos corroboram o entendimento de que o método atualmente utilizado pelo COP-BH ainda é insuficiente para uma análise adequada do fenômeno. Assim, esse trabalho busca propor um novo método de análise, por meio do uso de ferramentas estatísticas sólidas e bem estabelecidas. E este intento introduz uma pergunta metodológica fundamental: Qual é, se existe, o algoritmo de clusterização universalmente superior para todos os tipos de dados geoespaciais? Conforme destacado por [Sadeghi 2025], a escolha de um método de clusterização e de suas respectivas métricas de avaliação é uma fonte chave de incerteza na análise de geodados, e a confiabilidade dos resultados depende de uma seleção criteriosa e justificada.

Diante deste cenário, o presente trabalho prestou-se a navegar nesta incerteza metodológica, aplicando o *framework* proposto por Sadeghi aos dados em análise. Foi realizada uma análise comparativa e sistemática de cinco abordagens de clusterização, repre-

sentando as principais famílias de algoritmos: particionamento (K-Means e K-Medoids), hierárquico (Hierarchical Agglomerative Clustering), e baseados em densidade (DBSCAN e HDBSCAN*). O objetivo foi avaliar a aplicabilidade e a performance de cada método na identificação de *hotspots* de furto de cabos de cobre em Belo Horizonte. Para cada algoritmo, foi verificada a adequação de um conjunto de métricas de seleção de modelo (e.g., Método do Cotovelo e Critério de Informação Bayesiana) e de avaliação de qualidade (e.g., Índices Silhouette, Davies-Bouldin e Calinski-Harabasz), demonstrando empiricamente as forças e fraquezas de cada combinação. Por fim, foi aplicada a métrica padrão Índice de Acurácia Preditiva (PAI - Predictive Accuracy Index) introduzido por [Chainey et al. 2008], para avaliar os *hotspots* pelos métodos de clusterização mencionados.

1.1. Trabalhos Relacionados

A necessidade de ferramentas robustas para a análise criminal dialoga diretamente com os princípios de avaliação e monitoramento de políticas públicas discutidos por [Tavares 2009]. Segundo o autor, a gestão eficiente de projetos públicos e sociais carece de mecanismos que evidenciem o impacto real das intervenções. Nesse sentido, a metodologia proposta neste trabalho preenche uma lacuna instrumental: ao identificar estatisticamente mudanças na dinâmica espaço-temporal de criminalidade, gestores públicos podem ter o indicador de efetividade necessário para validar ou reorientar as estratégias de segurança, transformando dados brutos em inteligência avaliativa.

A literatura sobre análise criminal tem evoluído da simples identificação espacial para abordagens que integram a dimensão temporal e a dinâmica do deslocamento criminal. [Chainey et al. 2008] destacam que o mapeamento de *hotspots* é uma ferramenta essencial para o direcionamento de recursos policiais, mas ressaltam a necessidade de métodos estatísticos rigorosos para evitar interpretações visuais subjetivas. Nesse contexto, [Kulldorff 1997] propôs a estatística de varredura espacial (*Spatial Scan Statistic*), um método baseado na razão de verossimilhança que permite identificar aglomerados geograficamente significativos sem a necessidade de definir fronteiras administrativas *a priori*, ajustando-se à população de base.

No entanto, a análise puramente espacial oferece uma visão estática. Para compreender a evolução dos *hotspots*, é necessário investigar mudanças em suas séries temporais. [Aminikhanghahi and Cook 2017] revisam diversos métodos de detecção de pontos de mudança, enfatizando a importância de identificar alterações abruptas no comportamento dos dados. Especificamente para modelos lineares, [Bai and Perron 1998] desenvolveram uma metodologia robusta para estimar múltiplas quebras estruturais em datas desconhecidas, permitindo identificar se e quando o nível médio de criminalidade em um *cluster* se alterou significativamente.

A interpretação dessas mudanças encontra respaldo na criminologia ambiental. Enquanto [Mohler et al. 2011] utilizam Processos de Hawkes para modelar o crime como um fenômeno de contágio autoexcitável para predição de curto prazo, [Barr and Pease 1990] discutem os conceitos de deslocamento (*displacement*) e deflexão. Segundo os autores, a intervenção em uma área pode prevenir o crime (deflexão) ou apenas movê-lo (deslocamento), sendo fundamental analisar a distribuição do crime para entender a eficácia das políticas de segurança. Este trabalho unifica essas perspectivas, ao

buscar a análise e classificação da dinâmica espaço-temporal de *clusters* identificados a partir dos dados.

Por outro lado, a Fatoração de Matrizes Não Negativas proposta por [Garcıa et al. 2019] busca decompor a matriz espaço-temporal para identificar padrões latentes e recorrentes de criminalidade em toda a malha urbana. Por sua vez, este trabalho adota uma perspectiva voltada para a detecção de anomalias e mudanças de regime. Diferentemente de Garcia, que foca na estrutura global e na redução de dimensionalidade para lidar com a esparsidade dos dados, a metodologia aqui apresentada isola geograficamente os aglomerados de risco e investiga a dinâmica temporal do fenômeno, oferecendo uma ferramenta diagnóstica para avaliação de intervenções pontuais.

Semelhantemente, [Mandalapu et al. 2023] conduziram uma revisão sistemática demonstrando a crescente predominância de técnicas de Aprendizado de Máquina (*Machine Learning*) e Aprendizado Profundo (*Deep Learning*) na análise criminal, motivadas pela capacidade de processar grandes volumes de dados para fins preditivos. No entanto, o mesmo estudo aponta desafios relacionados à complexidade computacional e à necessidade de abordagens que integrem efetivamente as dimensões espacial e temporal. Este trabalho se posiciona como um contraponto metodológico a essa tendência de 'caixa-preta': ao invés de buscar a predição complexa via redes neurais, opta-se por uma abordagem de inferência estatística robusta. Essa escolha garante não apenas a integração espaço-temporal apontada por Mandalapu, mas também assegura a interpretabilidade direta dos resultados (datas e locais precisos de mudança), atributo essencial para a tomada de decisão tática e estratégica por gestores de segurança pública.

No contexto específico de furtos de cabo de cobre, o trabalho de [Sidebottom et al. 2014] focou em testar a hipótese comum, mas pouco examinada, de que as flutuações no preço de metais estão associadas a mudanças no volume de roubo de metais. Especificamente, a pesquisa analisou a relação entre o preço do cobre e o número de furtos de cabos de cobre de energia elétrica registrados pela polícia na rede ferroviária britânica entre 2006 e 2012, utilizando análise de séries temporais. Eles encontraram apoio para a hipótese preço-roubo, concluindo que as mudanças no preço do cobre estavam positivamente associadas às variações no volume desse tipo de crime. Sustentaram que os padrões de roubo de cabos de cobre são melhor explicados por uma perspectiva de oportunidade criminal. Outros trabalhos, como o de [Pretorius 2012], [Govender 2013] e [Chachuli et al. 2016] investigaram o fenômeno do furto de cabos e o *modus operandi* dos criminosos em diversos países, examinando ações preventivas que instituições públicas e empresas privadas deveriam empreender para prevenir o problema. As soluções propostas vão desde melhorias estruturais e processuais, mediante o fortalecimento da capacidade policial e legal, ao controle estrito do mercado de sucata, incluindo o uso de sensores para identificar furtos em tempo real.

A Seção 2, a seguir, trata da Fundamentação Teórica deste trabalho, apresentando de forma não exaustiva as abordagens de clusterização citadas anteriormente, no contexto de aprendizagem não supervisionada, e os prós e contras de cada uma delas quando aplicadas a dados geoespaciais. Essa seção detalha, também, o *framework* "Navegando pela incerteza", usado neste trabalho para avaliar os algoritmos de clusterização testados. A Seção 3 apresenta a Metodologia do trabalho, que acrescenta ao *framework* de referência, a implementação e teste do algoritmo de clusterização HDBSCAN*. A Seção 4 apresenta

os resultados da análise, com o uso do PAI para validar a metodologia de monitoramento do furto de cabos de cobre ora proposta. Por fim, a Seção 5 apresenta as considerações finais do trabalho, vislumbrando desdobramentos e evoluções futuras.

2. Fundamentação Teórica

A análise de padrões de pontos é um conceito fundamental na análise de dados geoespaciais. Historicamente, os métodos de análise de padrões de pontos envolviam o cálculo de estatísticas resumidas básicas, como Contagem, Média e Desvio Padrão, mas são consideradas rudimentares e podem ocultar informações valiosas sobre os padrões observados. Em resposta a esta limitação, métodos mais sofisticados de análise de densidade e operações estatísticas avançadas foram introduzidos [Kalinic and Krisp 2018].

Uma aplicação crucial na área geoespacial é o Mapeamento de Crimes (*Crime Mapping*), um ramo dos Sistemas de Informação Geográfica (SIG) dedicado a explicar o comportamento espaço-temporal de crimes, enfatizando a importância da geografia local como determinante dos tipos e taxas de crimes. Um dos principais objetivos dessa abordagem é a capacidade de identificar e visualizar *hotspots* de crimes [Garcıa et al. 2019].

Neste sentido, um dos primeiros e mais frequentemente utilizado método para visualizar e analisar dados geoespaciais, para entender e potencialmente prever padrões de eventos, é a Estimativa de Densidade de Kernel (KDE) [Kalinic and Krisp 2018]. Na cartografia criminal, o KDE emergiu como uma abordagem comum para a identificação de *hotspots* (pontos de alta concentração de eventos), sendo frequentemente combinada com a Estatística *Spatial Scan* [Garcıa et al. 2019]. O KDE é particularmente útil porque realiza uma série de estimativas sobre uma grade colocada sobre o padrão de pontos, detectando os picos e vales das densidades. Mapas baseados em KDE têm sido muito procurados em campos como a análise de crimes, avaliações de risco, e planejamento de emergências [Kalinic and Krisp 2018].

No entanto, o KDE apresenta desafios notáveis. A correta definição do parâmetro de largura de banda (*bandwidth*) é complexa [Garcıa et al. 2019], e uma escolha inadequada pode resultar em superestimação ou desconsideração de *hotspots*, levando a resultados errôneos ou imprecisos [Kalinic and Krisp 2018]. Os *hotspots* resultantes de mapas KDE não possuem significância estatística, e a informação representada depende fortemente da escolha do *kernel* e do tamanho da grade de saída, exigindo experimentação [Kalinic and Krisp 2018]. Além disso, locais com atividades criminais regulares, mas não intensas o suficiente, são dificilmente identificados como *hotspots* por métodos baseados em KDE [Garcıa et al. 2019].

O agrupamento (*clustering*) é uma técnica fundamental n oaprendizado não supervisionada, cujo objetivo é agrupar pontos de dados com base em suas similaridades intrínsecas para descobrir estruturas e padrões subjacentes [Sadeghi 2025]. Em *geo-data science*, o agrupamento é essencial para análise exploratória de dados, detecção de anomalias e redução de dimensionalidade [Sadeghi 2025]. Diferentes métodos de agrupamento podem resultar em diferentes agrupamentos para um mesmo *dataset* [Herdiana et al. 2025], e devido à natureza heterogênea que é comum a dados geoespaciais – que frequentemente incluem ruído, *outliers* e *clusters* com formas irregulares ou densidades variáveis – a escolha do método é crucial para reduzir a incerteza na análise de dados e garantir resultados significativos [Sadeghi 2025].

Salienta-se que os conceitos de *cluster* e *hotspot* podem ser considerados similares em grande medida, pois ambos os termos se referem a aglomerações ou agrupamentos de dados, especialmente no contexto da análise espacial e de padrões. A similaridade essencial é que o agrupamento (*clustering*) é um método que visa agrupar pontos de dados de modo que aqueles no mesmo *cluster* tenham a máxima similaridade entre si. Um *hotspot* é uma manifestação específica desse agrupamento no espaço, sendo geralmente definido como uma área com um alto número de incidentes ou uma concentração de atividades relevantes. Portanto, a detecção de *hotspots* espaço-temporais frequentemente envolve *clustering* para agrupar eventos. A distinção reside na intensidade: um *hotspot* é tipicamente um *cluster* de alta densidade (alta incidência), já um *cluster* (no sentido estatístico) pode ser um agrupamento de baixa densidade (*cold spot*) ou um grupo autosemelhante [Garcıa et al. 2019] e [Herdiana et al. 2025]. A seguir são apresentadas as cinco abordagens analisadas neste trabalho.

2.1. Técnicas de Particionamento (K-Means e K-Medoids)

O **K-Means** é amplamente utilizado devido à sua simplicidade, velocidade e escalabilidade, sendo bem adequado para grandes conjuntos de dados [Herdiana et al. 2025]. O princípio do K-Means é minimizar a distância (geralmente Euclidiana) entre cada objeto e o centro do seu *cluster*, chamado centroide (*centroid*). Os centroides são recalculados como a média dos pontos de dados que pertencem àquele agrupamento, visando minimizar a distância entre os objetos e o seu centro [Herdiana et al. 2025]. No entanto, assume que os *clusters* são esféricos e é sensível a *outliers* e à escolha inicial dos centroides [Sadeghi 2025]. Um comparativo é apresentado na Tabela 1.

Tabela 1. Avaliação K-Means

Pontos positivos	Pontos negativos/limitações
• Simplicidade, rapidez na convergência, e eficiência no tratamento de grandes volumes de dados [Herdiana et al. 2025] e [Sadeghi 2025].	• Requer a determinação inicial do número de *clusters* (`k`). Métodos como o "Método do Cotovelo" podem ser subjetivos e imprecisos [Herdiana et al. 2025]. • Assume que os *clusters* são esféricos e uniformemente distribuídos (tamanhos iguais), o que raramente se alinha com a variabilidade natural em dados geocientíficos [Sadeghi 2025]. • É sensível a *outliers* (valores extremos) e à inicialização dos centroides [Sadeghi 2025].

O **K-Medoids** é uma alternativa similar ao K-Means, sendo mais robusto a *outliers* e ruídos. Ao invés de usar o centroide (que é um ponto médio), ele seleciona um ponto de dado real, o medoide (*medoid*), como o centro do *cluster*. O medoide é um ponto de dado real, que é selecionado para atuar como o centro de um *cluster* no Particionamento em

Torno de Medoides, o que o torna mais resiliente a *outliers* e ruídos do que um centroide médio calculado [Sadeghi 2025]. O algoritmo minimiza a soma das dissimilaridades (distâncias) entre os pontos e seus respectivos medoides [Sadeghi 2025], e isso o torna mais adequado para o uso de métricas de distância não-Euclidianas, embora possua um custo computacional mais elevado para grandes conjuntos de dados [Sadeghi 2025]. A Tabela 2 apresenta os prós e contras dessa alternativa.

Tabela 2. Avaliação K-Medoids

Pontos positivos	Pontos negativos/limitações
• Maior robustez contra *outliers* e ruídos em comparação ao K-Means [Sadeghi 2025]. • Adequado para conjuntos de dados com estruturas irregulares ou quando se utilizam distâncias não-Euclidianas [Sadeghi 2025].	• É computacionalmente mais exigente do que o K-Means, especialmente para grandes conjuntos de dados, pois calcula distâncias pareadas entre todos os pontos [Sadeghi 2025].

2.2. Técnica Hierárquica Aglomerativa (HAC - Hierarchical Agglomerative Clustering)

A clusterização hierárquica cria uma hierarquia de *clusters* por meio de uma abordagem de baixo para cima (aglomerativa) ou de cima para baixo (divisiva). O método aglomerativo (o *bottom-up*, que é o mais comum) começa com cada ponto sendo um *cluster* e itera mesclando os dois *clusters* mais próximos com base em critérios de ligação (*linkage criteria*), que são as regras ou funções que determinam como a distância ou dissimilaridade entre os *clusters* é calculada, para decidir quais pares serão mesclados ou divididos. O resultado é um dendrograma (ou árvore de *clusters*), um diagrama em forma de árvore (*tree-like diagram*), utilizado para examinar os *clusters* hierárquicos antes de determinar o número apropriado de agrupamentos [Sadeghi 2025]. A Tabela 3 compara os prós e contras dessa abordagem.

Tabela 3. Avaliação Hierarchical Agglomerative Clustering

Pontos positivos	Pontos negativos/limitações
• Oferece interpretabilidade significativa, pois o dendrograma visualiza a formação dos *clusters*, e é flexível e adaptável a diversos conjuntos de dados [Sadeghi 2025].	• Possui alta complexidade computacional: $O(n^3)$ para o tempo de execução e $O(n^2)$ em consumo de memória, para grandes conjuntos de dados [Baqir et al. 2020]. Pode ser sensível a ruídos [Sadeghi 2025].

2.3. Técnicas baseadas em Densidade (DBSCAN* e HDBSCAN*)

O *clustering* baseado em densidade é um paradigma que incorpora a estimação de densidade não paramétrica, permitindo modelar o ruído (objetos em regiões não densas) e encontrar *clusters* de formas variadas como componentes conexos dos conjuntos de nível de densidade. As técnicas baseadas em densidade são ideais para análises onde a densidade de dados é variável e as formas geométricas são arbitrárias, ou seja, não se limitam a formas circulares como o K-Means e o K-Medoids, características comuns em dados geoespaciais [Campello et al. 2015].

O **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) identifica *clusters* examinando a densidade de pontos em uma região, sendo capaz de isolar pontos de ruído. O algoritmo classifica os pontos como *core points* (com densidade suficiente), *border points* ou *noise points* (ruído) [Sadeghi 2025]. A Tabela 4 apresenta os pontos positivos e as limitações dessa abordagem.

Tabela 4. Avaliação *DBSCAN*

Pontos positivos	Pontos negativos/limitações
• Capacidade de identificar *clusters* de formas arbitrárias (não convexas), e é especialmente eficaz para conjuntos de dados que contêm ruído, e conjuntos de dados esparsos, o que é crucial em análises geoespaciais [Sadeghi 2025].	• O desempenho é altamente dependente da seleção cuidadosa de seus dois parâmetros: – O $epsilon$ (ϵ), que é a distância (raio de vizinhança) máxima entre dois pontos para serem considerados parte do mesmo *cluster* [Sadeghi 2025]; – O $minPts$, que define o número mínimo de pontos exigido para formar um novo *cluster* [Sadeghi 2025].

O algoritmo DBSCAN original incluía também os objetos de borda (*border objects*), que são objetos não centrais que estão dentro da vizinhança `epsilon` de um ou mais objetos centrais. Esses objetos de borda eram considerados e atribuídos a um *cluster* [Campello et al. 2015]. Contudo, novas definições foram usadas em DBSCAN*, definindo *clusters* com base apenas em objetos centrais (*core objects* ou *core points*, segundo Sadeghi). Um objeto central (*core object*) é um ponto de dados cuja vizinhança contém o número mínimo de objetos desejado (`mPts`): $|N_\epsilon(x_p)| \geq mPts$ [Campello et al. 2015]. Assim, o DBSCAN* (com a terminologia '*') é mais consistente com uma interpretação estatística dos *clusters* como componentes conectados de um conjunto de nível de densidade (*level set of a density*). Esta abordagem o torna mais fiel ao modelo de Hartigan (1975), de *clusters* de contorno de densidade e árvores de contorno de densidade [Campello et al. 2015]. Em termos práticos, mantém-se o uso da terminologia DBSCAN (sem o asterisco).

O **HDBSCAN*** (Hierarchical DBSCAN*), introduzido por Campello et al. é destinado à análise de agrupamentos baseados em densidade, detecção de *outliers* e visualização de dados. O HDBSCAN* é uma extensão e um aprimoramento do DBSCAN* (seu precursor), que o transforma em um algoritmo de clusterização hierárquica, herdando sua característica de aderência ao modelo estatístico de Hartigan de árvores de contorno de densidade (*density-contour trees*). Ele gera uma hierarquia de *clustering* completa, convertendo essencialmente o DBSCAN* em um método hierárquico, unindo o melhor das duas abordagens [Campello et al. 2015]. O processo de construção da hierarquia envolve:

- Cálculo da Distância de Mútua Acessibilidade: Primeiro, a distância *core* (distância mínima epsilon ϵ para que um ponto seja um *core object*) é calculada para todos os objetos. Em seguida, a distância de mútua acessibilidade (d_{mreach}) é definida como o máximo entre a distância *core* dos dois pontos e a distância euclidiana real entre eles [Campello et al. 2015].
- Construção da Árvore: Esta distância d_{mreach} é usada para construir o Grafo de Mútua Acessibilidade (G_{mpts}). Uma Árvore Geradora Mínima (MST) é calculada sobre este grafo. O MST é então estendido (MST_{ext}) ao adicionar para cada vértice uma "auto-aresta" (*self-edge*) com o peso igual à distância core do objeto correspondente [Campello et al. 2015].
- Extração da Hierarquia: A hierarquia HDBSCAN* é extraída do MST_{ext} removendo-se iterativamente todas as arestas em ordem decrescente de pesos. Este processo é equivalente a rodar o algoritmo *Single-Linkage* no espaço transformado das distâncias de mútua acessibilidade [Campello et al. 2015].

A hierarquia resultante encapsula todas as soluções possíveis do DBSCAN* para um alcance infinito de limiares de densidade. Para obter uma solução plana, o HDBSCAN* não realiza um corte horizontal único (que falharia em detectar *clusters* de densidades variadas), mas sim cortes locais através da árvore de *clusters* [Campello et al. 2015], resumindo (ou fundindo) a estrutura de *clusters* com base na quantidade mínima de pontos (parâmetro `mPts`) e extraindo *clusters* estáveis [Baqir et al. 2020]. A fusão ou seleção de *clusters* que o HDBSCAN* realiza para encontrar a estabilidade ocorre durante o processo de extração ótima da solução plana, que é formulado como um problema de otimização: maximizar a estabilidade geral agregada dos *clusters* extraídos [Campello et al. 2015].

Este processo é executado de baixo para cima (*bottom-up*, como o HAC) na árvore de *clusters* simplificada, usando uma estratégia de programação dinâmica. Em cada nó (*cluster* parente) C_i, o algoritmo compara o valor de estabilidade de C_i (como um *cluster* único e estável) com a soma das estabilidades ótimas de seus *subclusters* (seus filhos C_il e C_ir) [Campello et al. 2015]. Se a estabilidade do pai $S(C_i)$ for maior, o *cluster* pai é selecionado (o que pode ser visto como uma fusão implícita dos *subclusters* para formar a solução estável). Se a soma das estabilidades dos filhos for maior, os filhos são selecionados (o *cluster* é "dividido" na solução final) [Campello et al. 2015]. A estabilidade maximizada garante que o algoritmo selecione os *clusters* mais proeminentes que "sobrevivem" por mais tempo contra eventos de ruído e divisão [Campello et al. 2015].

O conceito tradicional de "tempo de vida" do *cluster* em dendrogramas hierárquicos é inadequado para o contexto baseado em densidade porque a remoção de um único objeto ruidoso não deve caracterizar a dissolução de um *cluster*. Portanto, o HDBSCAN*

define a estabilidade $(S(C_i))$ de um *cluster* C_i usando a medida de Excesso Relativo de Massa, que leva em consideração os perfis de densidade individuais dos objetos pertencentes ao *cluster* [Campello et al. 2015]. A estabilidade $S(C_i)$ é calculada pela soma das diferenças entre os níveis de densidade de nascimento e morte de cada objeto x_j dentro do *cluster* C_i [Campello et al. 2015]:

$$S(C_i) = \sum_{x_j \in C_i} (\lambda_{\max}(x_j, C_i) - \lambda_{\min}(C_i)) \tag{1}$$

Onde:

- $\lambda_{\min}(C_i)$ é o nível de densidade mínima no qual o cluster C_i existe (o nível de densidade no qual ele nasce ou se funde);
- $\lambda_{\max}(x_j, C_i)$ é o nível de densidade a partir do qual o objeto x_j deixa de pertencer a C_i (ou se torna ruído).

Como o nível de densidade λ está relacionado ao raio ϵ (o limiar de densidade) por $\lambda = \frac{1}{\epsilon}$, a fórmula também pode ser escrita em termos de $\epsilon_{\min}$ e $\epsilon_{\max}$. Esta informação de estabilidade é crucial para determinar qual *cluster* é o mais significativo dentro de uma ramificação da hierarquia. O cálculo da estabilidade pode ser realizado simultaneamente durante a construção da hierarquia HDBSCAN*, com pouquíssimo processamento adicional e sem alterar a complexidade assintótica do algoritmo [Campello et al. 2015]. Outra vantagem de destaque do HDBSCAN* é sua capacidade de identificar *clusters* de formato arbitrário [Sadeghi 2025], o que é especialmente importante levando-se em consideração as características do *dataset* analisado. Ao contrário dos métodos de "mínima variância", a abordagem baseada em densidade permite que estruturas mais complexas sejam encontradas nos dados [Campello et al. 2015].

Em relação ao tratamento de ruído, o DBSCAN original (e consequentemente o HDBSCAN*) é especialmente eficaz para conjuntos de dados que contêm ruído e *clusters* de densidades variadas. No HDBSCAN*, o conceito de *border points* (objetos de densidade baixa perto de *core objects*) é abandonado, e apenas os *core points* são considerados parte de um *cluster*, o que é mais consistente com os conceitos de conjuntos de níveis de densidade. A remoção de objetos de ruído (objetos que não são considerados *clusters*) durante o processamento hierárquico é tratada como um encolhimento do *cluster*, e não como uma divisão, facilitando a identificação de mudanças significativas na estrutura [Campello et al. 2015].

O HDBSCAN* é particularmente relevante para a *geo-data science*. Pesquisadores como Baqir et al. [Baqir et al. 2020] identificaram o HDBSCAN* como a técnica mais adequada em comparação com o HAC para a detecção de *hotspots* de crimes espaço-temporais em áreas urbanas. A análise comparativa mostrou que o HDBSCAN superou o HAC em termos de tempo e requisitos de memória ao processar dados numéricos. Em resumo, as características que ensejam a sua adequação ao *dataset* em análise, são:

- Supera o HAC (Hierarchical Agglomerative Clustering) em termos de tempo e requisitos de memória, processando grandes volumes de dados (e.g., 100 mil pontos) em significativamente menos tempo, em comparações empíricas [Baqir et al. 2020].

- Tem vantagem sobre o DBSCAN por não depender do parâmetro `epsilon` (raio de vizinhança), cuja definição é desafiadora, ao construir a hierarquia completa de possíveis *clusters* que existiriam em todos os níveis de densidade (todos os `eps`) [Campello et al. 2015].
- Mantém a capacidade de encontrar *clusters* de formas arbitrárias [Baqir et al. 2020].
- É eficaz para dados onde as densidades são muito variáveis e heterogêneas, pois transforma o espaço de acordo com a densidade antes de construir a hierarquia. Por sua vez, o DBSCAN utiliza um limiar de densidade global para discriminar entre objetos localizados em áreas densas *versus* não densas [Baqir et al. 2020].
- O HDBSCAN* é computacionalmente mais eficiente que o DBSCAN. No cenário em que o conjunto de dados X está disponível e a distância $d(\cdot, \cdot)$ é computada em tempo $O(a)$ (onde a é o número de atributos), a complexidade de tempo assintótica do algoritmo é $O(an^2)$. Além disso, é robusto e competitivo quando comparado a outros métodos de *clustering* baseados em densidade e métodos de detecção de *outliers* de última geração [Campello et al. 2015].

2.4. Escolha do Método de Clusterização mais adequado

Não obstante, a escolha do método de clusterização deve ser baseada nas características específicas do conjunto de dados, em vez de depender de uma abordagem universal [Sadeghi 2025]. Portanto, do ponto de vista teórico, em análises de dados geoespaciais onde a densidade é variável e as formas geométricas não devem ser restritivas, pode-se entender que:

- DBSCAN e HDBSCAN* são a escolha mais adequada, pois foram projetados para lidar com *clusters* de formas arbitrárias (não esféricas) e são eficazes na gestão de densidades variáveis e ruídos. Essa robustez é relevante, especialmente em conjuntos de dados esparsos [Sadeghi 2025].
- HAC é menos adequado por depender muito do critério de ligação (*linkage*) escolhido para definir o formato dos *clusters*. Os seus critérios de ligação tendem a funcionar melhor com *clusters* mais esféricos ou convexos, similar ao *K-Means* ou são extremamente sensíveis a ruído (efeito "*chaining*"), como o *single linkage*. Nenhum critério de ligação padrão do HAC é tão flexível quanto a abordagem por densidade para formas complexas e ruídos [Sadeghi 2025].
- K-Means e K-Medoids são os mais inadequados se a suposição de clusters esféricos e com densidades uniformes for violada, resultando em interpretações enganosas. O K-Medoids oferece maior robustez contra *outliers* do que o K-Means, mas ainda assume formas regulares [Sadeghi 2025].

Em última análise, a seleção do método ideal envolve um equilíbrio cuidadoso entre a incerteza do modelo e a validade dos *clusters*, usando métricas de avaliação como os Índices Silhouette (SI), Davies-Bouldin (DBI) e Calinski–Harabasz (CHI) para validar a qualidade da separação e compacidade dos *clusters*. O DBI, por exemplo, é mais confiável para agrupamentos compactos e bem separados, enquanto o SI é robusto para conjuntos de dados esparsos [Sadeghi 2025].

Portanto, a incerteza proveniente da escolha do método e das métricas de avaliação é um desafio significativo, especialmente em domínios da análise geoespaciais, onde

os resultados influenciam diretamente a tomada de decisões. O objetivo primordial é equilibrar a baixa incerteza com alta validade do *cluster* [Sadeghi 2025].

2.4.1. Métricas para avaliação da qualidade interna do *cluster* e comparação

As métricas de avaliação quantificam a qualidade dos *clusters* resultantes e são fundamentais para comparar o desempenho de diferentes algoritmos ou configurações [Sadeghi 2025]. Essas métricas avaliam a estrutura intrínseca dos *clusters* (compacidade e separação) e podem ser usadas tanto para selecionar o "$\mathtt{k}$ ótimo" quanto para validar a solução final [Sadeghi 2025].

Índice de Silhueta (Silhouette Index - SI) : Avalia o quão bem um ponto de dado está agrupado, comparando sua coesão (similaridade a pontos no mesmo *cluster*) com sua separação (dissimilaridade a pontos no *cluster* vizinho mais próximo) [Sadeghi 2025]. O seu *score* (pontuação) varia de -1 a $+1$. Valores próximos de $+1$ indicam que os pontos estão bem agrupados, próximos de 0 sugerem *clusters* sobrepostos, e próximos de -1 indicam agrupamento incorreto. O SI demonstrou ser robusto, por exemplo, em conjuntos de dados esparsos [Sadeghi 2025].

Índice de Davies-Bouldin (Davies-Bouldin Index - DBI) : É uma métrica que mede a qualidade da clusterização avaliando a separação e a compacidade dos *clusters* [Sadeghi 2025]. O resultado do DBI é também um *score*, onde valores mais baixos indicam melhor desempenho de clusterização. É eficaz em conjuntos de dados densos com *clusters* compactos e bem separados [Sadeghi 2025].

Índice de Calinski-Harabasz (Calinski-Harabasz Index - CHI) : Conhecido como Critério de Razão de Variância, avalia a qualidade comparando a dispersão dentro dos *clusters* ($\mathtt{trace}(W_k)$) com a dispersão entre os *clusters* ($\mathtt{trace}(B_k)$) [Sadeghi 2025]. Os valores de CHI mais altos indicam *clusters* mais bem definidos e separados. O CHI é computacionalmente eficiente, mas tende a favorecer *clusters* esféricos e uniformemente distribuídos [Sadeghi 2025].

Estas métricas, o SI, DBI e CHI assumem que os *clusters* são esféricos e distribuídos uniformemente. Essa suposição alinha-se com os modelos de *clustering* do tipo particionamento, como o K-Means e K-Medoids, que, por sua natureza, assume *clusters* esféricos [Sadeghi 2025]. Portanto, não seriam adequados à análise de algoritmos baseados em densidade. Não obstante, Sadeghi os utilizou para avaliar qualitativamente também o HAC e o DBSCAN, que são modelos hierárquico e baseado em densidade, respectivamente, embora tenha pontuado que a análise visual da formação de *clusters* em um dendrograma seja a forma ideal para identificar o número ótimo de *clusters* no HAC [Sadeghi 2025]. No caso do HDBSCAN*, a seleção do *cluster* não se baseia em um único limiar global, mas sim em uma medida de estabilidade derivada da hierarquia de densidade, por meio de duas técnicas principais [Campello et al. 2015]:

Árvore de Clusters Condensada (*Condensed Cluster Tree*) Gráficos de Acessibilidade (*Reachability Plots*) : É a ferramenta diagnóstica nativa e principal para avaliar o HDBSCAN*. Consiste de uma árvore hierárquica completa de todos os *clusters* possíveis em todos os níveis de densidade. Em essência, é uma evolução que automatiza a detecção de "vales", regiões de alta densidade (*clusters*), usando os mesmos conceitos do OPTICS, algoritmo que foi um predecessor conceitual importante para o HDBSCAN* [Campello et al. 2015]. Esse gráfico mostra a hierarquia dos *clusters*, e onde cada um deles "nasce" (em que nível de densidade/distância), e onde ele "morre" (se funde com outro) [Campello et al. 2015]. Apresenta, também, a estabilidade de cada *cluster* (geralmente pela área ou largura do ramo da árvore). O HDBSCAN* "corta" essa árvore automaticamente, selecionando os ramos (*clusters*) com maior estabilidade [Campello et al. 2015].

Validação de clusterização baseada em densidade (DBCV - Density-Based Clustering Validation) : O DBCV é um índice (um único número, como o Silhouette Score médio) que avalia a qualidade de uma partição de *cluster* baseada em densidade. Um *score* DBCV mais alto (próximo de 1.0) indica uma clusterização de alta qualidade (*clusters* densos e bem separados por ruído). Um *score* baixo (próximo de 0.0 ou negativo) indica *clusters* ruins ou sobrepostos em densidade [Campello et al. 2015]. Autores como Moulavi et al. [Moulavi et al. 2014] e Campello et al. [Campello et al. 2015] demonstram a eficácia do DBCV comparando-o com outras medidas tradicionais (como SI e DBI) em partições geradas por algoritmos baseados em densidade, incluindo HDBSCAN*. Em experimentos com *datasets* 2D, onde os *clusters* possuem formas arbitrárias, o DBCV foi o único a ser capaz de reconhecer a verdadeira estrutura presente nos dados, superando amplamente os concorrentes. Ele mede:

- Coesão (Densidade Interna): A densidade dentro de cada *cluster* (quão compactos eles são em termos de densidade) [Campello et al. 2015].
- Separação (Densidade Externa): A densidade entre os *clusters* (quão bem eles são separados por regiões de baixa densidade/ruído) [Campello et al. 2015].

2.4.2. Métricas de seleção (Determinação do "k ótimo" e parâmetros)

As métricas de seleção, frequentemente referidas como métodos para a determinação do número ótimo de *clusters* ($\mathtt{k}$), são cruciais em algoritmos como K-Means e K-Medoids, que exigem a determinação inicial de $\mathtt{k}$ [Sadeghi 2025].

Método do Cotovelo (Elbow Method) : É uma heurística popular devido à sua simplicidade e interpretabilidade [Sadeghi 2025]. Consiste em plotar a Soma dos Quadrados dos Erros Intra-Cluster (Within-Cluster Sum of Squares - WCSS) em função do número de *clusters* [Herdiana et al. 2025]. O seu funcionamento consiste da determinação do valor "$\mathtt{k}$ ótimo" pelo ponto que forma um "cotovelo", indicando que a adição de mais *clusters* não resulta em uma redução significativa do WCSS [Herdiana et al. 2025]. O WCSS é calculado como a soma dos quadrados das distâncias entre cada ponto X_i e o centroide C_{S_j} do seu *cluster* S_j [Herdiana et al. 2025]. A principal crítica ao Método do Cotovelo

reside na sua dependência da interpretação visual subjetiva, o que pode levar à escolha imprecisa do ponto [Sadeghi 2025]. Neste trabalho foi utilizada a biblioteca **Kneed**, criada por Kevin Arvai, para automatizar a seleção do ponto do "cotovelo" [Arvai 2020].

Critério de Informação Bayesiana (BIC - *Bayesian Information Criterion*) : É um critério de seleção de modelo utilizado para avaliar a qualidade de modelos estatísticos, incluindo algoritmos de clusterização, visando determinar o número ótimo de *clusters* [Sadeghi 2025]. Tem o seu funcionamento na busca por equilibrar o ajuste do modelo com a sua complexidade, ou seja, penaliza modelos com mais parâmetros para evitar o *overfitting*. É aplicado em conjunto com a clusterização para garantir que a estrutura subjacente dos dados seja capturada de forma robusta [Sadeghi 2025].

2.4.3. *Framework* "Navegando pela Incerteza"

O *framework* "Navegando pela Incerteza" enfatiza a necessidade de uma abordagem informada e específica ao contexto para a clusterização. Em lugar de depender de métodos convencionais isolados (como K-Means e o Método do Cotovelo), ele propõe a avaliação sistemática de múltiplos algoritmos e métricas para quantificar a incerteza [Sadeghi 2025]. A incerteza é quantificada pela análise da variabilidade no desempenho dos métodos de clusterização em diversas métricas de avaliação. O método mais confiável é aquele que apresenta baixa incerteza estatística (consistência e estabilidade) e alta validade (relevância geométrica ou geológica, dependendo da aplicação) [Sadeghi 2025].

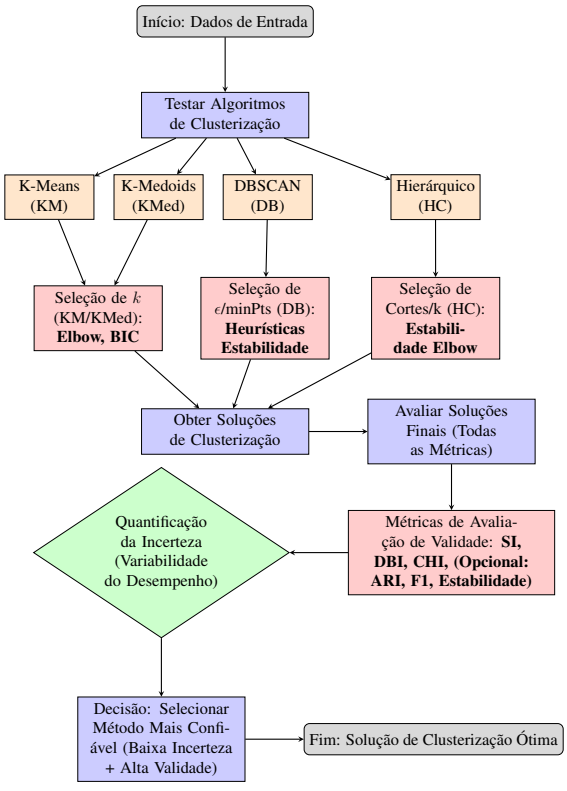


Figura 3. Diagrama de Fluxo do Framework "Navegando pela Incerteza", proposto por Sadeghi et al. [Sadeghi 2025].

O diagrama de fluxo da Figura 3 ilustra as etapas deste *framework* comparativo, proposto por Sadeghi [Sadeghi 2025], onde algoritmos como K-Means (KM), K-Medoids (KMed), DBSCAN (DB) e Hierárquico (HAC) são testados em conjunto com métricas de seleção (Elbow, BIC) e métricas de avaliação (SI, DBI, CHI) para minimizar a incerteza.

2.4.4. Índice de Acurácia Preditiva (PAI - Predictive Accuracy Index)

A métrica padrão Índice de Acurácia Preditiva foi aplicada neste experimento com o objetivo de avaliar os *hotspots* determinados como "ótimos" pelo *framework* descrito na subseção 2.4.3. O índice PAI foi utilizado para medir a eficiência de um mapa de *hotspots* em prever a localização de eventos futuros, e foi introduzido por [Chainey et al. 2008]:

$$\frac{\left(\frac{n}{N}\right) \times 100}{\left(\frac{a}{A}\right) \times 100} = \frac{HitRate}{AreaPercentage} = PAI \tag{2}$$

Onde:

- n – O número de eventos criminais futuros que ocorreram dentro das áreas onde os crimes foram previstos;
- N – O número total de eventos criminais no período de medição dentro de toda a área de estudo;
- a – A área geográfica (por exemplo, em km^2) das áreas onde os crimes foram previstos (os *hotspots*).
- A – A área geográfica (por exemplo, em km^2) de toda a área de estudo.

3. Metodologia

O presente estudo adota o *framework* metodológico "navegar pela incerteza", que se refere à seleção de algoritmos de clusterização e suas métricas de validação, conforme proposto por [Sadeghi 2025] para a análise de geodados. O objetivo principal não é apenas aplicar um método de clusterização, mas realizar uma avaliação comparativa sistemática de diferentes abordagens para identificar a mais confiável na detecção de *hotspots* de furto de cabos em Belo Horizonte. Em seguida avalia a eficácia do "melhor algoritmo" de clusterização identificado, comparando-o com o método atualmente utilizado no COP-BH, o KDE. A metodologia compreende três fases principais: (A) Pré-processamento dos dados, (B) Análise Metodológica Comparativa dos algoritmos de clusterização, e (C) Análise Dinâmica e Validação. A Figura 4 ilustra todos os passos utilizados.

3.1. Pré-processamento dos Dados (Fase A)

O *dataset* utilizado consiste em registros georreferenciados de ocorrências de furto de cabos em Belo Horizonte, fornecidos pelo COP-BH. Os dados brutos foram submetidos a um processo de limpeza e tratamento, seguido pela adição de colunas temporais para permitir análises em diferentes granularidades (semanal, mensal, bimestral e trimestral). O *dataset* contém 18.590 observações sobre o problema de furto de cabos, no período de 03/2018 a 05/2025, e consiste dos seguintes atributos:

- `origem`, que apresenta de qual sistema os dados se originaram;

- `data_hora`, que apresenta a data e hora em que o furto ocorreu;
- `mes_ano`, que apresenta o mês e ano da ocorrência (dados adicionados para facilitar a análise);
- `trimestre_ano`, que apresenta o trimestre e ano da ocorrência (dados adicionados para facilitar a análise);
- `latitude` e `longitude`, que apresentam a localização exata da ocorrência; e
- `endereco` e `regional`, que apresentam o endereço físico aproximado da ocorrência.

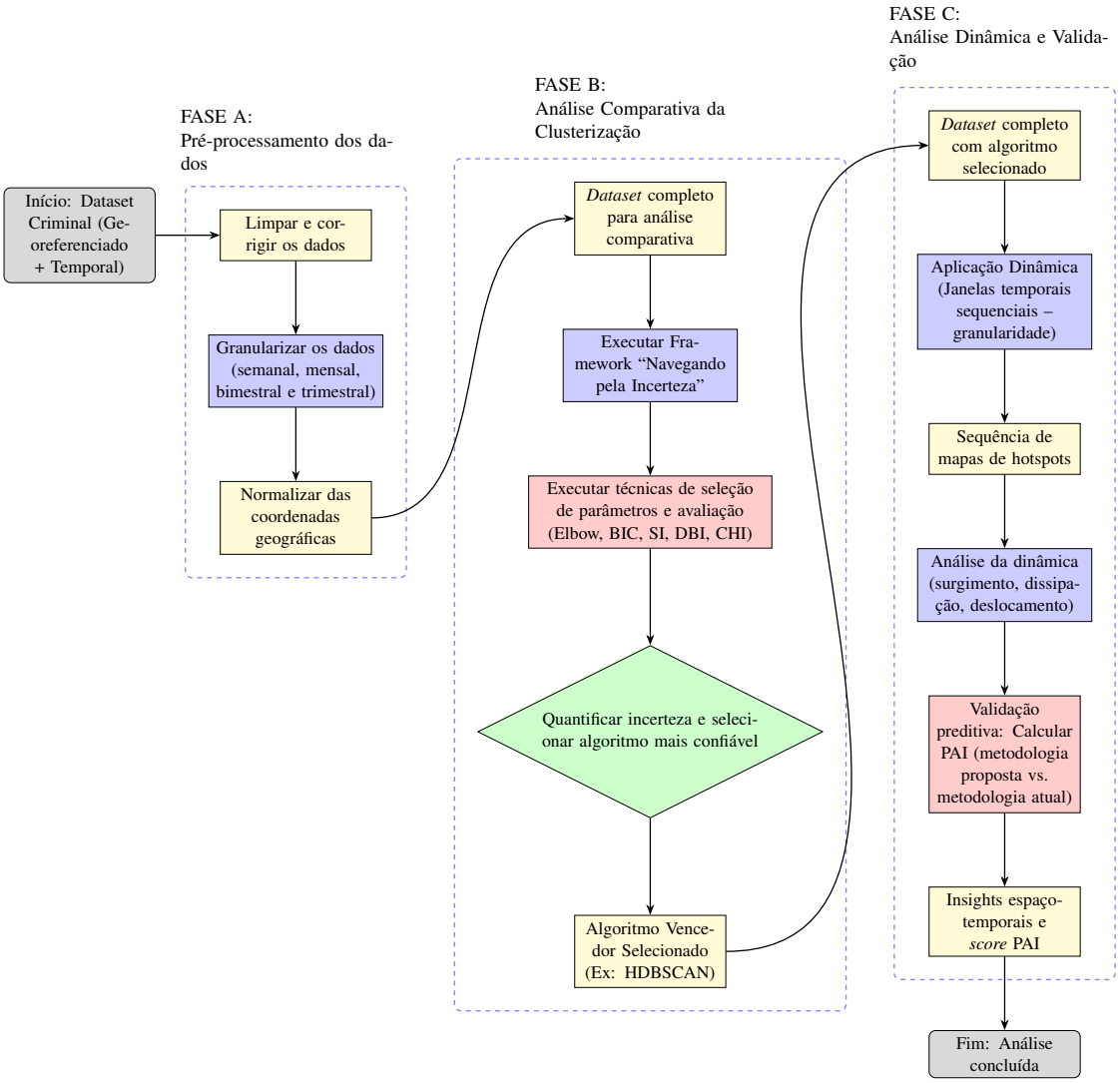


Figura 4. Fluxograma da metodologia de seleção do "melhor" algoritmo e validação comparativa dos métodos.

3.2. Análise Comparativa de Clusterização (Fase B)

Cinco algoritmos, representando as principais famílias de métodos de clusterização, foram implementados e avaliados sistematicamente. Para cada algoritmo, foi aplicado um processo estruturado de seleção de parâmetros e avaliação de qualidade, utili-

zando um conjunto consistente de métricas para permitir a comparação e a quantificação da incerteza.

3.2.1. K-Means (Particionamento por Centroides)

- Algoritmo: K-Means (`sklearn.cluster.KMeans`).
- Seleção de parâmetros (número de *clusters*: `k`):
  - Método do Cotovelo (Elbow Method): Foi calculado o WCSS para `k` de 2 a 15 *clusters*. O cotovelo foi identificado automaticamente pela biblioteca `kneed.KneedLocator`.
  - Critério de Informação Bayesiana (BIC): Calculado por meio da biblioteca `sklearn.mixture.GaussianMixture` para `k` de 2 a 15. O "`k` ótimo" é o que minimiza o BIC.
- Avaliação de qualidade: Para cada `k` foram calculados os índices Silhouette (SI), Davies-Bouldin (DBI) e Calinski-Harabasz (CHI).
- Decisão final (`k_otimo`): Comparação da qualidade dos *clusters* (SI, DBI e CHI) para os valores de `k` sugeridos pelo Elbow e BIC. O `k` que resultou na melhor combinação de *scores* foi escolhido.

3.2.2. K-Medoids (Particionamento por Medoides)

- Algoritmo: K-Medoids (`sklearn_extra.cluster.KMedoids`, método `alternate` para robustez de convergência);
- Seleção e Avaliação: Processo idêntico ao K-Means (Elbow e BIC para seleção; SI, DBI e CHI para avaliação e decisão final).

3.2.3. Hierarchical Agglomerative Clustering (Clusterização Hierárquica)

- Algoritmo: `scipy.cluster.hierarchy.linkage` (método `ward`) seguido por `sklearn.cluster.AgglomerativeClustering`.
- Seleção de Parâmetros:
  - Método Tradicional: Inspeção visual do Dendrograma gerado para encontrar o número de *clusters* ótimo (`k`).
  - Método Quantitativo (*Framework*): Cálculo de WCSS (Elbow) e BIC (via GMM) para `k` de 2 a 15, para fins de comparação.
- Avaliação de Qualidade: Cálculo dos índices SI, DBI e CHI para cada `k`.
- Decisão final (`k_otimo`): Comparação da qualidade dos *clusters* (SI, DBI e CHI) para os valores de `k` sugeridos pelo Elbow e BIC. O `k` que resultou na melhor combinação de *scores* foi escolhido.

3.2.4. DBSCAN (Densidade Fixa)

- Algoritmo: `sklearn.cluster.DBSCAN`.
- Seleção de Parâmetros (número mínimo de pontos (`minPts`), raio (`eps`)):

  - `minPts`: Foi testado para um *range* de valores, a fim de identificar qual valor teria aplicações mais adequadas ao COP-BH.
  - `eps`: Estimado automaticamente por meio da heurística recomendada por [Schubert et al. 2017] do gráfico *k-distance* (*sorted k-dist graph*) e `kneed.KneedLocator`.
- Avaliação de Qualidade:
  - Demonstração da Inadequação: Cálculo do SI excluindo e incluindo ruído para demonstrar empiricamente a invalidade de métricas de partição para algoritmos de densidade.
  - Avaliação Primária: Análise visual do mapa de *clusters* e métricas descritivas (número de *clusters*, percentual de ruído).

3.2.5. HDBSCAN* (Densidade Hierárquica)

- Algoritmo: `hdbscan.HDBSCAN` (Configuração: `metric='haversine'`, `algorithm='ball_tree'`).
- Seleção de parâmetros (número mínimo de pontos (`mPts`)):
  - Execução do algoritmo para um *range* de `mPts` (de 10 a 30). Plotagem do número de *clusters* e percentuais de ruído em função do `mPts`.
  - Decisão final: Escolha do `mPts` em uma região de estabilidade no gráfico de sensibilidade (heurísticamente, maximizando *clusters* e minimizando ruídos).
- Avaliação de Qualidade: Avaliação Primária – Análise visual do gráfico de sensibilidade e análise visual do mapa de *clusters* (considerando a probabilidade de pertencimento).

A etapa final da metodologia consiste na análise comparativa dos resultados obtidos. A "incerteza" é quantificada observando a variabilidade:

- No número ótimo de *clusters* (`k` ou equivalente) sugerido por diferentes métricas para um mesmo algoritmo.
- Na qualidade dos *clusters* (scores SI, DBI, CHI) obtida por diferentes algoritmos para um k comparável.
- Na robustez visual e na interpretabilidade dos *clusters* gerados por cada método (especialmente para DBSCAN e HDBSCAN).

Com base na análise destes três pontos, o algoritmo considerado o mais 'confiável' foi aquele que apresentou resultados mais estáveis, robustos à parametrização e consistentes com a natureza esperada de *hotspots* criminais (formatos irregulares, presença de ruído) – foi selecionado para a Fase C (Análise Dinâmica e Validação).

3.3. Análise Dinâmica e Validação (Fase C)

Para validar a eficácia preditiva da metodologia final desenvolvida, foi utilizado o Índice de Acurácia Preditiva (PAI), proposto por Chainey et al. O PAI mede a proporção de crimes em um período futuro (`t+1`) que ocorrem dentro das áreas identificadas como *hotspots* no período atual (`t`), ajustada pela proporção da área total coberta pelos *hotspots* [Chainey et al. 2008]. Serão calculados e comparados os *scores* PAI para:

- Os mapas *hotspots* com granularidades menores, gerados pela metodologia proposta (usando o algoritmo selecionado).
- Um mapa anual de *hotspots*, gerado por um método de linha de base (e.g., KDE agregado), simulando a abordagem tradicional.

Além disso, após a seleção do algoritmo de clusterização mais confiável, foi gerada uma sequência temporal de mapas de *hotspots*. O objetivo é analisar a dinâmica espacial do fenômeno (surgimento, dissipação e deslocamento). Espera-se que esta comparação permita quantificar objetivamente o ganho em capacidade preditiva oferecido pela análise dinâmica espaço-temporal em relação aos métodos tradicionais, validando a relevância operacional da metodologia proposta para o monitoramento proativo do COP-BH.

4. Resultados e Discussão

Esta seção apresenta os resultados da aplicação da metodologia (3) de definição do algoritmo mais adequado à análise dos dados de furto de cabos em Belo Horizonte. Cada subseção detalha o comportamento de cada um dos cinco algoritmos analisados, focando na seleção de parâmetros, na avaliação da qualidade dos *clusters* e nas limitações inerentes de cada método. As análises foram realizadas utilizando o *dataset* completo, com todas as 18.590 observações, para verificar o desempenho de cada um deles com essa quantidade de pontos. Para cada uma das análises foram geradas visualizações gráficas das métricas utilizadas.

Antes da aplicação dos algoritmos baseados em distância (K-Means, K-Medoids e HAC), as coordenadas (latitude e longitude) foram normalizadas utilizando a biblioteca `StandardScaler`, para garantir que ambas as dimensões contribuíssem igualmente para o cálculo das distâncias. Para os algoritmos baseados em densidade (DBSCAN, HDBSCAN), que requerem distâncias geodésicas, as coordenadas foram transformadas em radianos com a métrica `Haversine`.

O Método do Cotovelo (Elbow - WCSS/Inércia) não avalia os dados brutos (coordenadas normalizadas), mas sim a compactação interna da partição que o algoritmo criou. Portanto, esta métrica é calculada após o algoritmo (K-Means, K-Medoids e HAC) ter sido executado para um `k` específico. Ela precisa tanto dos dados normalizados quanto dos *labels* gerados para medir a soma das distâncias quadráticas de cada ponto ao centro do seu próprio *cluster*.

Por sua vez, o BIC (Bayesian Information Criterion) é aplicado diretamente aos dados brutos normalizados (coordenadas normalizadas). Ele testa os diferentes valores de `k` e calcula um *score* para cada um, tentando encontrar o `k` que melhor explica a distribuição dos dados, enquanto penaliza a complexidade (um `k` maior).

Já os índices SI (Silhouette), DBI (Davies-Bouldin) e CHI (Calinski-Harabasz) são aplicados após a execução do algoritmo (K-Means, K-Medoids, HAC e DBSCAN). Elas recebem como entrada principal os *labels* gerados pelo algoritmo e os dados normalizados (coordenadas normalizadas). Então calculam um *score* que quantifica, para aquela partição específica, quão bem os *clusters* estão formados (compactos e coesos) e quão bem eles estão separados uns dos outros.

4.1. Resultados do K-Means

O K-Means, primeiro método avaliado, é um algoritmo de particionamento baseado em centroides, e sua avaliação seguiu as fases de seleção de `k` e avaliação da qualidade (3.2.1), conforme a metodologia "navegando pela incerteza" de [Sadeghi 2025]. Devido à incerteza na seleção do número ótimo de *clusters* (`k`), foi testado um *range* de valores, variando de 2 a 15. A Figura 5 apresenta os valores ótimos encontrados para `k`, selecionados a partir das métricas de seleção (2.4.2) Elbow e BIC. Como citado anteriormente, o valor de `k` ótimo pela métrica Elbow foi determinado pela biblioteca `kneed.KneedLocator`. O tempo de execução da clusterização e avaliação do K-Means foi de `42.37` segundos.

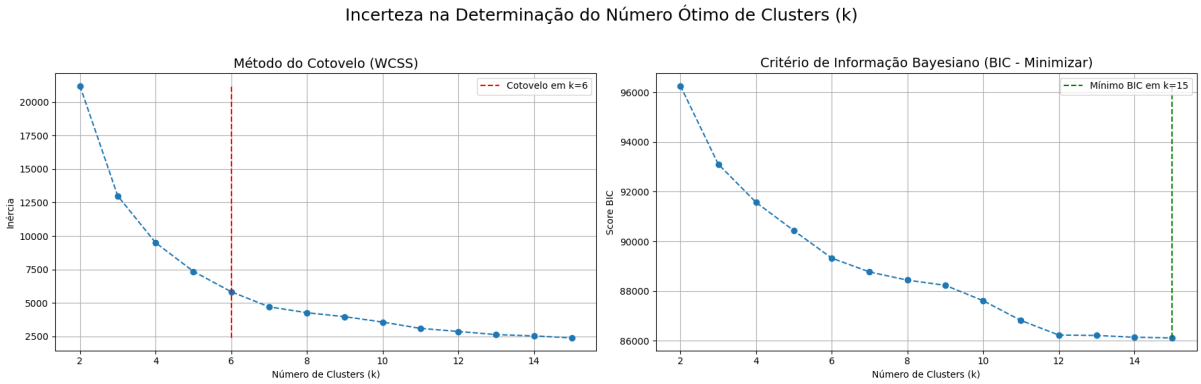


Figura 5. K-Means: Valores candidatos para `k`, a partir das métricas Elbow e BIC.

Método Elbow (Cotovelo - WCSS) : A curva WCSS exibe a esperada queda acentuada inicial, seguida por uma diminuição mais gradual. A detecção automática do cotovelo, utilizando o algoritmo KneedLocator sobre a curva convexa e decrescente, identificou `k=6` como o ponto ótimo, onde a adição de novos *clusters* passa a oferecer ganhos marginais decrescentes na redução da variância *intra-clusters*.

Método Bayesiano (BIC) : O gráfico do BIC, que penaliza a complexidade do modelo, atingiu seu valor mínimo em `k=15`. Este resultado sugere que, considerando o *trade-off* entre ajuste e número de parâmetros, um modelo com 15 *clusters* seria preferível.

A divergência entre os valores ótimos candidatos sugeridos pelas duas métricas (`k_elbow` vs. `k_bic`) evidencia a incerteza inerente à seleção de `k` mesmo dentro de uma única família de métricas, como apontado por [Sadeghi 2025]. Portanto, a partir desses valores de `k` identificados, foram aplicadas as métricas de avaliação da qualidade (2.4.1) Silhouette (SI), Davies–Bouldin (DBI) e Calinski–Harabasz (CHI). A Figura 6(a) mostra os *scores* das métricas de avaliação de qualidade para cada valor de `k` testado.

Índice Silhouette (SI) : O *score* de `k_elbow=6` foi de `0.402454`. Já o *score* de `k_bic=15` foi `0.381093`. Segundo esta métrica, estas foram as partições que apresentaram o melhor equilíbrio entre coesão *intra-cluster* e separação *inter-cluster*.

Índice Davies-Bouldin (DBI) : O *score* de `k_elbow=6` foi de `0.804272`, enquanto o *score* de `k_bic=15` foi de `0.839709`. Neste caso, estes foram os *scores* que indicaram melhor separação relativa.

Índice Calinski-Harabasz (CHI) : O *score* de `k_elbow=6` foi de `19988.1`, e o `score` de `k_bic=15` foi de `19365.7`. Estes são os valores que indicam *clusters* mais bem definidos e separados, segundo esta métrica.

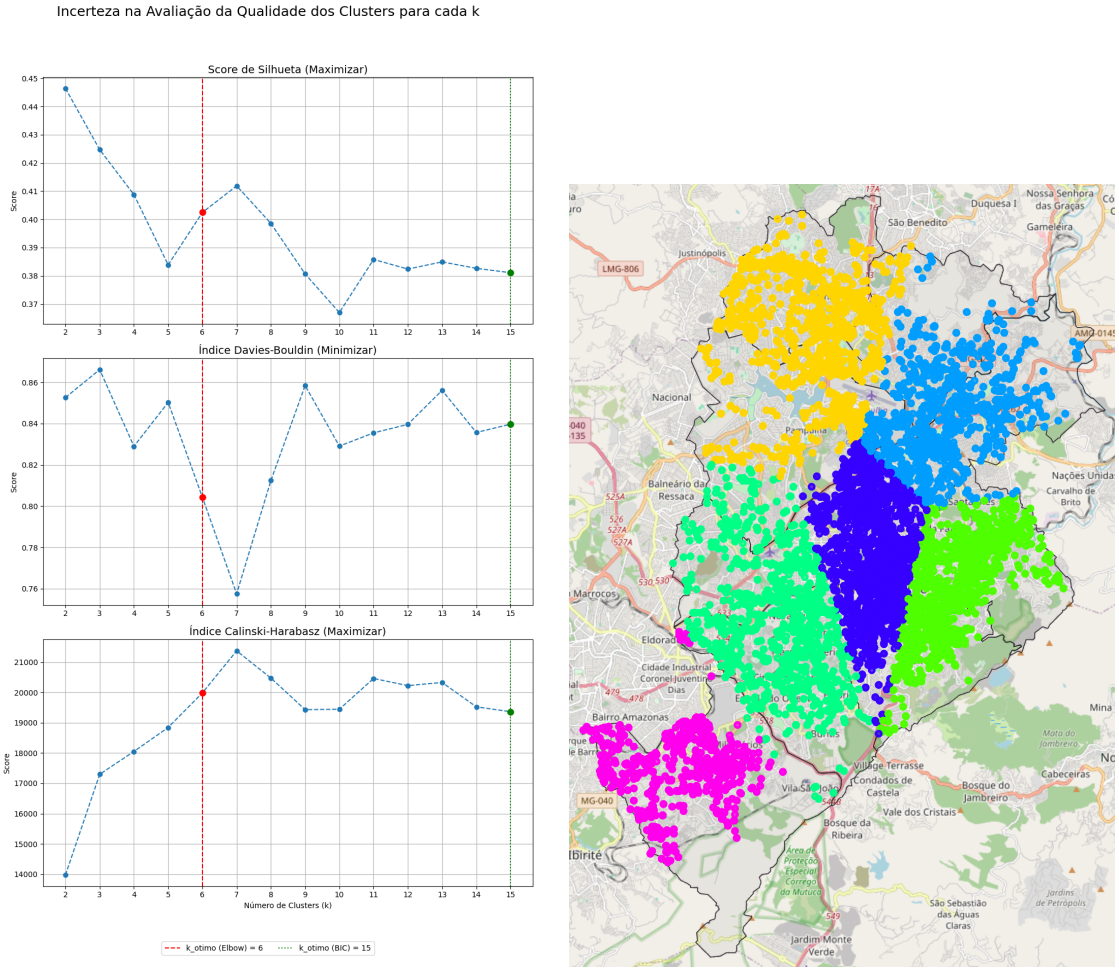
Novamente, observa-se uma falta de consenso entre as métricas de avaliação sobre qual valor de `k` produz a "melhor" partição geométrica. Portanto, a decisão final da análise qualitativa, seguindo o *framework* de validação cruzada, é comparada a qualidade dos *clusters* para os candidatos `k_elbow = 6` e `k_bic = 15`. A Tabela 5, a seguir, sumariza os *scores* de SI, DBI e CHI para ambos.

Tabela 5. Tabela comparativa de scores

Métrica	**k**	**Silhouette** (SI) (Alto é Melhor)	**Davies-Bouldin** (DBI) (Baixo é Melhor)	**Calinski-Harabasz** (CHI) (Alto é Melhor)
Elbow	6	0.402454	0.804272	19988.1
BIC	15	0.381093	0.839709	19365.7

Nesta análise, o candidato `k_elbow=6` obteve a melhor pontuação combinada nas métricas de avaliação, pois foi o que obteve o maior *score* no Índice SI, o menor no DBI e o maior no CHI. Portanto, segundo o *framework* "navegando pela incerteza", o `k_otimo=6` foi selecionado para a clusterização final. A Figura 6(b) apresenta a visualização dos *clusters* resultantes.

Observa-se que, mesmo com a seleção criteriosa de `k`, a maioria dos *clusters* gerados pelo K-Means tendem a ter formatos arredondados e abrangem áreas extensas, com pontos de diferentes *clusters* frequentemente interpostos espacialmente. Isso reflete as limitações intrínsecas do K-Means, tal como já ponderado por Sadeghi (2.1): sua incapacidade de lidar com ruído (forçando todos os pontos a pertencerem a um *cluster*) e sua dificuldade em se adaptar a padrões espaciais de formato irregular, comuns em geodados. Embora o K-Means forneça uma partição inicial, sua aplicabilidade direta para a delimitação precisa de *hotspots* operacionais mostra-se limitada neste contexto.



(a) K-Means: Avaliação da qualidade de *clusters* para os valores candidatos para `k` encontrados.

(b) Visualização dos *clusters* identificados pelo K-Means, para `k_otimo = 6`.

Figura 6. Resultados da análise de clusterização K-Means.

4.2. Resultados do K-Medoids

O K-Medoids também é um algoritmo de particionamento, mas que se distingue por ser baseado em medoides. Nesse sentido, a implementação e avaliação dos dois algoritmos é similar, e segue as fases de seleção de `k` e avaliação da qualidade (3.2.2), conforme a metodologia "navegando pela incerteza" de Sadeghi. Ademais, foram testados o mesmo *range* de valores para seleção do número ótimo de clusters (`k`), variando de 2 a 15. A Figura 7 apresenta os valores ótimos encontrados para `k`, selecionados a partir das métricas de seleção (2.4.2) Elbow e BIC. Tal como o *K-means*, o valor de `k` ótimo pela métrica Elbow foi determinado pela biblioteca `kneed.KneedLocator`. O tempo de execução da clusterização e avaliação do K-Medoids foi de `160.01` segundos, o que converge com a análise de que tem maior custo computacional que o K-Means (2.1).

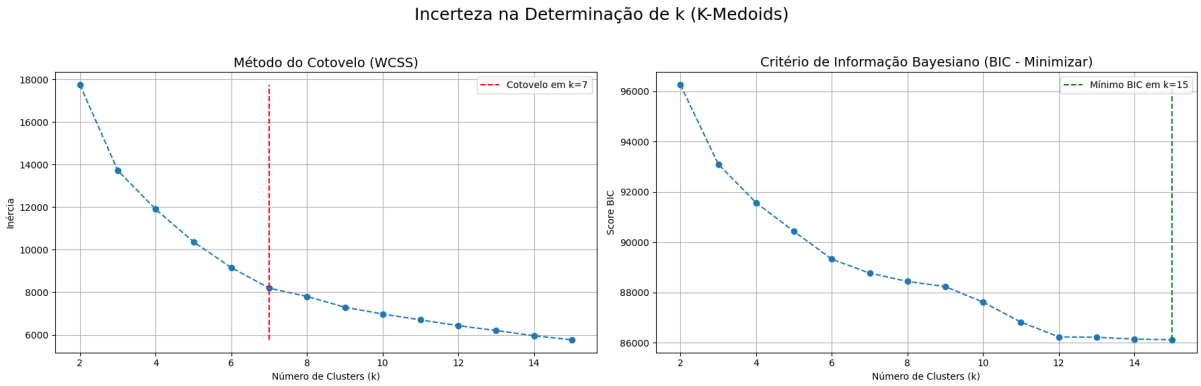
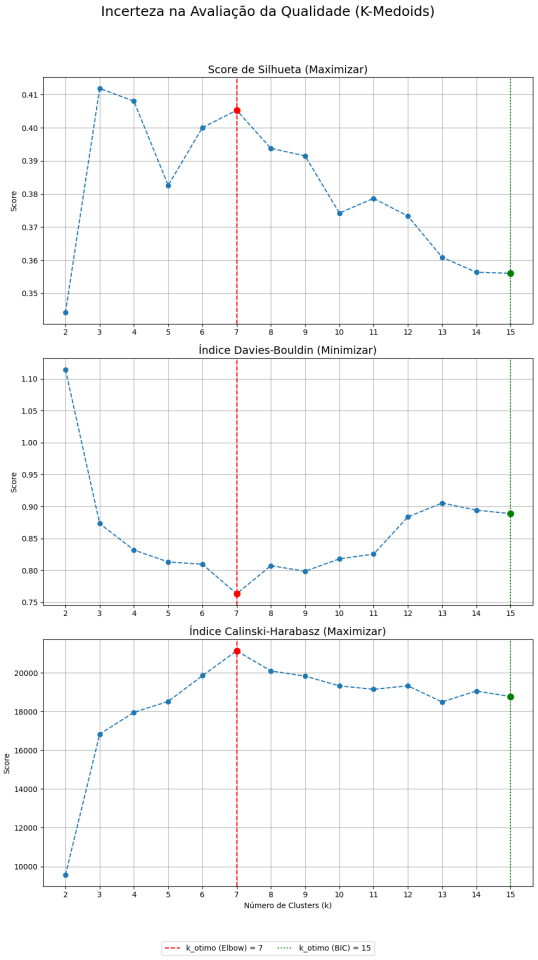


Figura 7. K-Medoids: Valores candidatos para `k`, a partir das métricas Elbow e BIC.

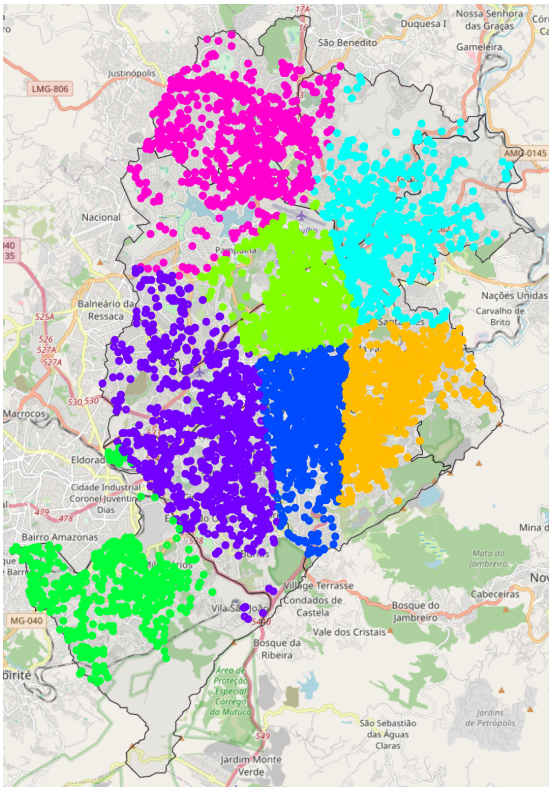
Método Elbow (Cotovelo - WCSS) : O algoritmo KneedLocator identificou `k_elbow=7`, sobre a curva de inércia WCSS convexa e decrescente, como o ponto ótimo, onde a adição de novos *clusters* passa a oferecer ganhos marginais decrescentes na redução da variância intra-cluster. Isso demonstra que o próprio método de clusterização influencia a determinação do `k_elbow`, pois no caso do K-Means, o KneedLocator indicou `k_elbow=6`, enquanto no K-Medoids, indicou `k_elbow=7`, aplicado ao mesmo conjunto de dados. Isso ocorre porque a definição de 'centro' (centroide vs. medoide) altera a partição de *clusters* gerada e, consequentemente, a forma da curva de inércia, levando a diferentes pontos de cotovelo matemáticos.

Método Bayesiano (BIC) : No caso do BIC o `k` encontrado foi o mesmo que o do K-Means, `k_bic=15`. O K-Medoids é um algoritmo iterativo, assim como o K-means. Portanto, ele realiza várias iterações para encontrar os medoides ideais, movendo-os até que a configuração do *cluster* se torne estável. Para evitar que o algoritmo seja executado indefinidamente, o número padrão de iterações máximas é configurável, que é de 300 (`max_iter=300`). Significa que o algoritmo itera 300 vezes antes de conseguir encontrar uma solução perfeitamente estável, ou seja, um estado onde nenhum ponto muda de *cluster* entre uma iteração e outra. Durante a execução foi verificado que este valor de 300 iterações era insuficiente para encontrar a estabilidade, então foram testados valores maiores (500, 1000, 5000, 10000, 50000 e 100000), mas sem sucesso na estabilização dos *clusters*.

Portanto, foi feito outro teste alterando o parâmetro de configuração `method`, de `pam` (algoritmo "*Partitioning Around Medoids*", que é o método clássico e mais exato, mas que pode ter dificuldades de convergência) para `alternate` (`method='alternate'`), que é a implementação de uma heurística mais rápida. Ela troca iterativamente os medoides com não-medoides para tentar otimizar a solução, um processo que a permite "quebrar" os ciclos de oscilação e convergir mais rapidamente. Apesar de ser uma solução muito ligeiramente sub-ótima se comparada ao `pam`, a estabilidade do *cluster* foi encontrada mais rapidamente, com `max_iter=1000`. A Figura 8(a) mostra os *scores* das métricas de avaliação de qualidade para cada valor de `k` testado.



(a) K-Medoids: Avaliação da qualidade de clusters para os valores candidatos para `k` encontrados.



(b) Visualização dos *Clusters* identificados pelo K-Medoids, para `k_otimo = 7`.

Figura 8. Resultados da análise de seleção de parâmetros do K-Medoids.

Índice Silhouette (SI) : O *score* de `k_elbow=7` foi de `0.4053`. Já o *score* de `k_bic=15` foi `0.3561`. Segundo esta métrica, estas foram as partições que apresentaram o melhor equilíbrio entre coesão *intra-cluster* e separação *inter-cluster*.

Índice Davies-Bouldin (DBI) : O *score* de `k_elbow=7` foi de `0.7633`, enquanto o *score* de `k_bic=15` foi de `0.8888`. Neste caso, estes foram os *scores* que indicaram melhor separação relativa.

Índice Calinski-Harabasz (CHI) : O *score* de `k_elbow=7` foi de `21137.6383`, e o `score` de `k_bic=15` foi de `18773.5128`. Estes são os valores que indicam *clusters* mais bem definidos e separados, segundo esta métrica.

Mais uma vez observam-se as divergências entre as métricas de avaliação. Seguindo o *framework* de validação cruzada, compara-se a qualidade dos *clusters* dos candidatos para finalmente decidir o `k_otimo` (Tabela 6):

Tabela 6. Comparando a qualidade dos *clusters* para os candidatos a `k`

Métrica	k	SI (Maximizar)	DBI (Minimizar)	CHI (Maximizar)
Elbow	7.0000	**0.4053**	**0.7633**	**21137.6383**
BIC	15.0000	0.3561	0.8888	18773.5128

Uma vez mais o candidato `k_elbow=7` obteve a melhor pontuação combinada nas métricas de avaliação: obteve o maior *score* no Índice SI, o menor no DBI e o maior no CHI. A Figura 8(b) apresenta a visualização dos *clusters* resultantes para `k_otimo=7`. Observa-se que, tal como o K-Means, o K-Medoids mostra-se limitado ao contexto de uso de geodados, pelos mesmos motivos: sua incapacidade de lidar com ruído e sua dificuldade em se adaptar a padrões espaciais arbitrários, o que é típico a algoritmos de particionamento segundo Sadeghi (2.1). Os extensos *clusters* gerados pelo K-Medoids não teriam aplicabilidade ao COP-BH para identificar *hostpots* específicos, bem delimitados, sobre os quais se poderia atuar.

4.3. Resultados HAC - Hierarchical Agglomerative Clustering

A Clusterização Hierárquica Aglomerativa foi avaliada como representante da família de algoritmos hierárquicos. O uso do Elbow e BIC por [Sadeghi 2025] falhou ao determinar o `k_otimo`, porque estes métodos resultaram em valores nulos (NaN) no processamento do *dataset* analisado. Não obstante, os índices SI, DBI e CHI foram usados para avaliar a qualidade do HAC, apesar de terem limitações conhecidas nessa aplicação. Conforme Sadeghi, o SI, DBI e CHI assumem que os *clusters* são esféricos e uniformemente distribuídos, o que pode não se alinhar com a estrutura dos dados geocientíficos. Apesar disso, Sadeghi justificou a tentativa de usá-los no contexto de uma avaliação crítica e multi-métrica para reduzir a incerteza. Neste experimento, foi utilizado o mesmo processo que para os algoritmos anteriores: uso do Elbow e BIC para seleção dos valores de `k` candidatos, e determinação do `k_otimo` através da comparação dos resultados dos Índices SI, DBI e CHI. A Figura 9 apresenta os valores candidatos encontrados para `k`:

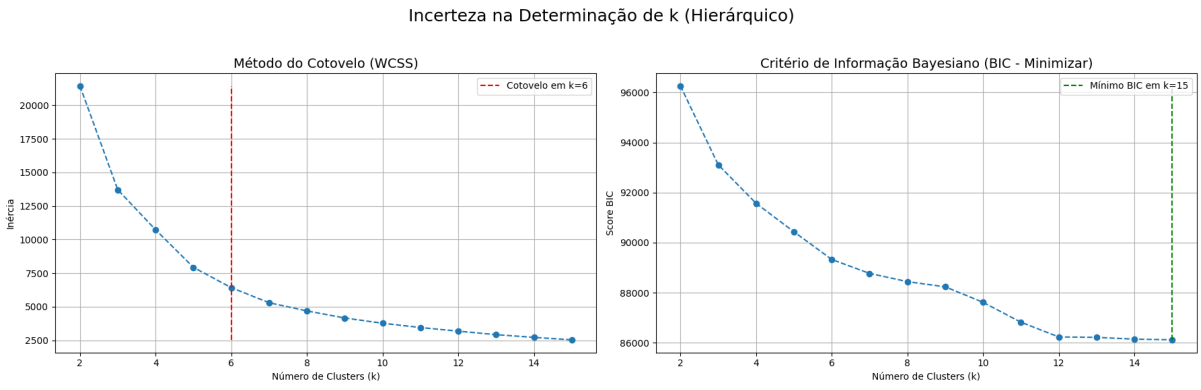


Figura 9. HAC: Valores candidatos para `k`, a partir das métricas Elbow e BIC.

Método Elbow (Cotovelo - WCSS) : O algoritmo KneedLocator identificou `k_elbow=6`, sobre a curva de inércia WCSS convexa e decrescente, como o seu candidato. Coincidentemente, este foi o mesmo valor de `k` encontrado no K-Means. Na implementação do K-Means foi utilizada a inércia (`kmeans.inertia_`), que é o WCSS da partição ideal. O K-Means é projetado para minimizar o WCSS. Já no HAC, o WCSS é calculado após a formação dos *clusters* pelo algoritmo, formados pelo critério de ligação `ward` (`linkage=ward`), escolhido devido ao uso de distâncias euclidianas. Este critério de ligação tem um objetivo similar ao do K-Means: a cada iteração, ele une os dois *clusters* que resultam no menor aumento da variância interna total, ou seja, o WCSS. Como ambos os algoritmos (K-Means e HAC com `ward`) estão tentando otimizar a mesma métrica WCSS, é altamente plausível que as duas curvas WCSS resultantes (curva ótima de inércia e curva "gulosa/*bottom-up*" do `ward`) sejam muito semelhantes. Então, o KneedLocator, ao analisar essas duas curvas de formas similares, encontrou o ponto de máxima curvatura (cotovelo) no mesmo ponto (`k_elbow=6`).

Método Bayesiano (BIC) : Para o BIC, que penaliza a complexidade do modelo, o `k` encontrado foi o `k_bic=15`. Este resultado foi o mesmo encontrado para os algoritmos K-Means e K-Medoids. A forma usada para implementar o cálculo BIC para todos os algoritmos é a mesma, através do `GaussianMixture` (GMM), portanto é esperado que os valores sejam o mesmos.

Seguindo o *framework*, a partir desses valores de `k` identificados, foram aplicadas as mesmas métricas de avaliação da qualidade (SI, DBI e CHI). A Figura 10(a) mostra os *scores* das métricas de avaliação de qualidade para cada valor de `k` testado.

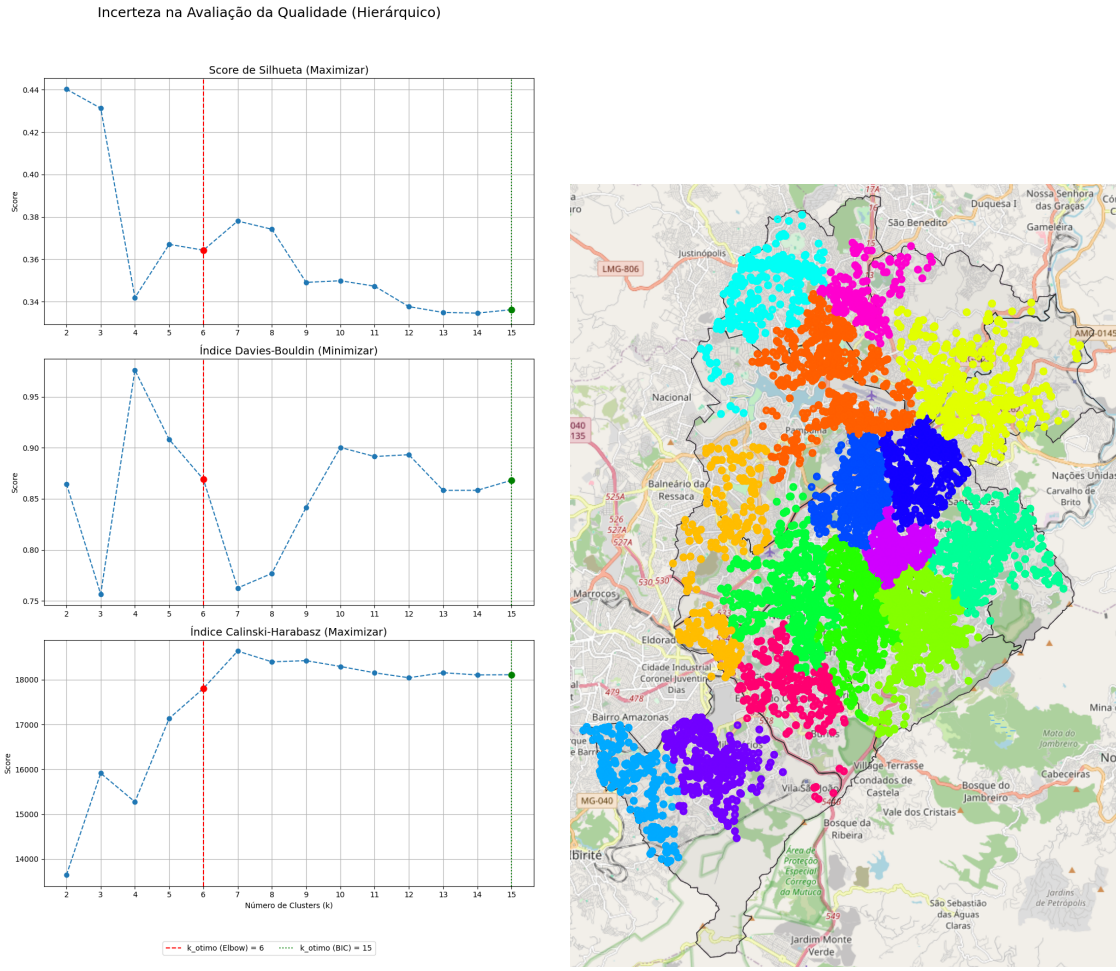
A seguir, foi realizada a validação cruzada, na qual se compara a qualidade dos *clusters* para os candidatos `k_elbow=6` e `k_bic=15`, para finalmente decidir o `k_otimo`. A Tabela 7, a seguir, sumariza os *scores* de SI, DBI e CHI para ambos:

Tabela 7. Tabela comparativa de scores

Métrica	k	SI (Maximizar)	DBI (Minimizar)	CHI (Maximizar)
Elbow	6	0.36428	0.86911	17804.5
BIC	15	0.336235	**0.86819**	**18110**

De forma distinta do K-Means e K-Medoids, o HAC encontrou o seu `k_otimo=15` no BIC, que obteve maior pontuação combinada nas métricas de avaliação (menor DBI e maior CHI). Isso porque a lógica de funcionamento do K-Means (*Top-Down*/Iterativo) e do K-Medoids (Particionamento/Iterativo), tenta encontrar a partição globalmente ótima (ou próxima disso) para `k` *clusters*. Ao tentar criar 15 *clusters*, ele pode ter encontrado *scores* de avaliação (SI, DBI, CHI) ruins.

Já o HAC (*Bottom-Up*/Guloso), constrói *clusters* fundindo os mais próximos. Ao testar o valor de `k_bic=15`, o algoritmo encontrou *clusters* naturais e coesos, com melhores *scores* de avaliação. Valores de `k_bic` menores podem ter fundido *clusters* que eram naturalmente distintos, resultando em *clusters* grandes, disformes e com *scores* de



(a) HAC: Avaliação da qualidade de *clusters* para os valores candidatos para `k` encontrados.

(b) Visualização dos *clusters* identificados pelo HAC, para `k_otimo = 15`.

Figura 10. Resultados da análise de clusterização Hierárquica (Euclidean).

avaliação (SI, DBI, CHI) piores. A partir dessa conclusão, a Figura 10(b) apresenta a visualização dos *clusters* resultantes para `k_otimo=15`.

O resultado final da clusterização gerada pelo HAC é mais adequado operacionalmente para o COP-BH que as duas clusterizações anteriores geradas pelo K-Means e K-Medoids. O número de *clusters* é maior, com menor tamanho, e as formas começam a se tornar mais arbitrárias. Contudo, ainda não é o ideal para atender à necessidade. Neste sentido, foi experimentado o uso da técnica tradicional de análise do HAC, que é a análise visual do dendrograma.

O Agrupamento Hierárquico (HAC) produz naturalmente um dendrograma, um diagrama em forma de árvore que "visualiza a formação de *clusters* e permite identificar o número ideal de *clusters* ao cortar a árvore em um nível escolhido" [Sadeghi 2025]. A análise visual do dendrograma para determinar o ponto de corte é a abordagem tradicional de avaliação, podendo ser, portanto, subjetivo. [Campello et al. 2015] e [Sadeghi 2025] criticam essa abordagem ao destacar suas limitações, embora seja um método comum. Campello et al. afirmam que "um corte horizontal corresponde a um único limiar de

densidade global que pode não detectar simultaneamente *clusters* com densidades locais amplamente variáveis". Por sua vez, Sadeghi argumenta que confiar apenas em métodos convencionais (como o Elbow, que é visualmente similar à interpretação de um dendrograma) pode introduzir incerteza considerável na análise.

Assim, para entender o comportamento desta abordagem tradicional do HAC sobre o *dataset* analisado, fizemos um segundo experimento. Neste, optou-se por utilizar coordenadas geodésicas (latitude e longitude), tendo em vista que o *dataset* trata de pontos no espaço geográfico que representam os furtos de cabo. Neste procedimento, as coordenadas foram transformadas para radianos, e a métrica utilizada foi `Haversine` (`metric='haversine'`). O método de ligação da matriz de distâncias foi *Average* (`method='average'`). Os outros valores para este atributo poderiam ser `'complete'`, que usa a distância máxima entre as observações de dois *clusters* – no qual os *clusters* tenderiam à forma esférica – e `'single'`, que usa a distância mínima entre as observações de dois *clusters* e tem como desvantagem o 'efeito de encadeamento', que é extremamente sensível a ruídos e pode causar a fusão incontida de *clusters*. O método *Average* foi usado, portanto, como um meio termo entre os dois.

Uma heurística de avaliação visual do dendrograma, que pode ser deduzida de Hastie et al. [Hastie et al. 2009] e Campello et al. [Campello et al. 2015], é a identificação do maior salto no eixo `y`. A linha vertical representa a distância de um *cluster* para o próximo, e quanto mais longa (maior salto), mais distante é o *cluster* do próximo. Neste ponto, do maior salto, é aplicada uma linha de corte horizontal, e `k` será a quantidade de vezes em que ela interceptar linhas verticais. A Figura 11 apresenta o dendrograma gerado para o *dataset* analisado, com a linha de corte definida em `k_otimo=3`, seguindo a heurística.

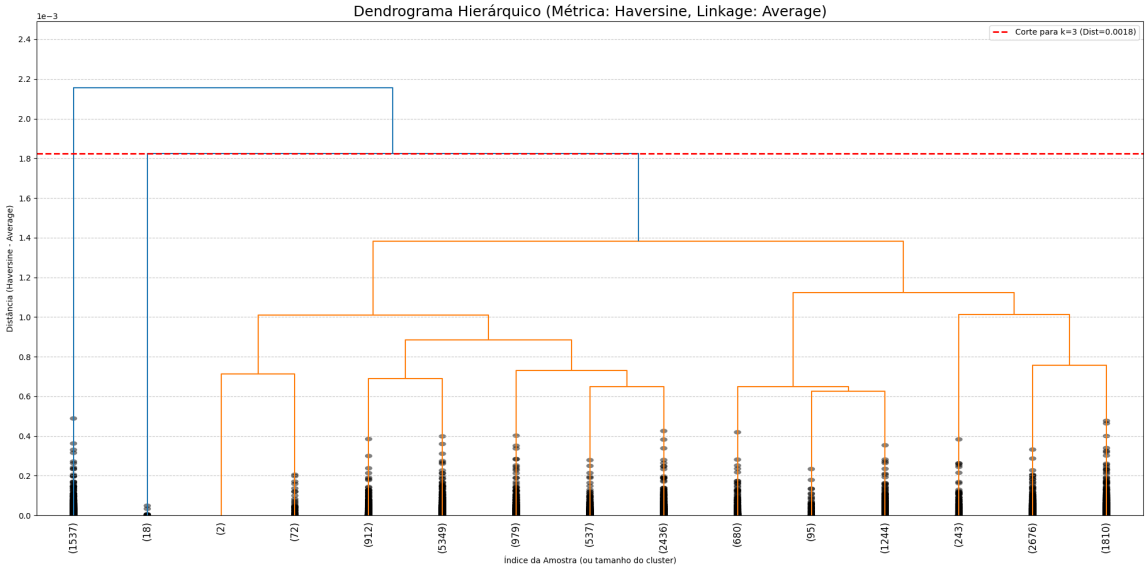
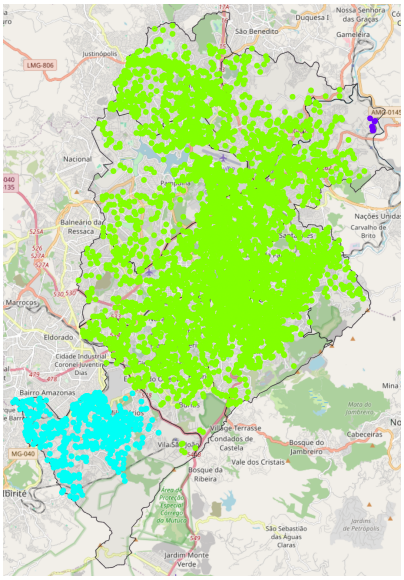
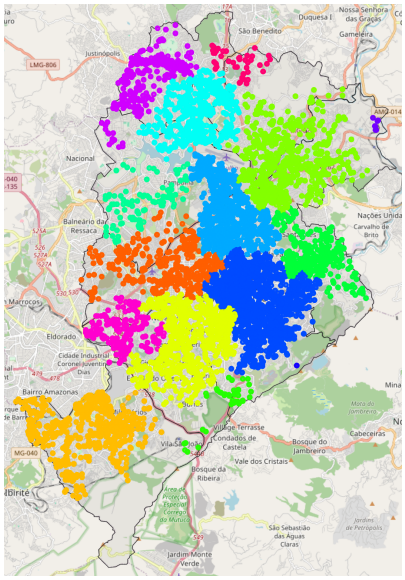


Figura 11. Visualização do dendrograma do HAC, com o corte definido para `k_otimo=3`.



(a) Visualização dos *clusters* identificados pelo HAC, com `k_otimo=3` a partir de análise do dendrograma.



(b) Visualização dos *clusters* identificados pelo HAC, com `k_otimo=15` a partir de análise do dendrograma.

Figura 12. Resultados da análise de clusterização HAC com a métrica *Haversine* com diferentes `k`.

A Figura 12(a) apresenta a visualização dos *clusters* resultantes para essa abordagem, ou seja, os que foram determinados a partir da inspeção visual do dendrograma. Como se pode verificar, o resultado não foi satisfatório, ficando aquém do resultado obtido utilizando o *framework* de Sadeghi. *Clusters* tão extensos, ainda que com formatos arbitrários, não têm significado operacional para o COP-BH. A Figura 12(b) foi gerada arbitrando-se o valor de `k_otimo=15`. O resultado é significativamente melhor, e verifica-se que o formato dos *clusters* não é exatamente o mesmo que os *clusters* gerados com parâmetros determinados pelo *framework* "navegando pela incerteza". Quanto ao uso de coordenadas geodésicas para calcular distâncias, pode-se perceber que é mais adequado metodologicamente, em lugar de utilizar distâncias Euclidianas. Todavia, o HAC ainda apresenta limitações, principalmente quanto à subjetividade em sua parametrização.

4.4. Resultados do DBSCAN - Density-Based Spatial Clustering of Applications with Noise

O DBSCAN foi analisado como um dos representantes dos algoritmos de clusterização baseados em densidade. Sadeghi concluiu que o DBSCAN emergiu consistentemente como o método de agrupamento mais confiável nos estudos de caso que ele conduziu, que envolviam dados com diferentes níveis de densidade, tendo sido particularmente eficaz para conjuntos de dados que continham ruído e *clusters* de densidades variadas [Sadeghi 2025]. Ele se destaca na detecção de *clusters* não convexos (seja, de formatos arbitrários) e na gestão de conjuntos de dados com ruído significativo, abordando desafios que outros algoritmos de agrupamento não conseguem lidar eficazmente. Sua capacidade de lidar com formatos irregulares e ruído o torna especialmente adequado para análise de dados espaciais. Essas características o tornam o mais adequado, até o

momento, para analisar adequadamente o *dataset* utilizado neste trabalho, que trata de dados geoespaciais.

O algoritmo DBSCAN baseia-se em dois parâmetros principais: epsilon (ϵ), que define o raio da distância máxima, e `minPts`, o número mínimo de pontos necessários para formar uma região densa. Os pontos são classificados em pontos centrais (*core points*), pontos de fronteira (*border points*) e pontos de ruído (*noise points*). No contexto do *framework* "navegando pela incerteza", Sadeghi não utilizou os métodos de seleção Cotovelo (Elbow) e Bayesiano (BIC), tendo em vista que o DBSCAN não se vale do parâmetro `k` para determinar o número de *clusters*. Além disso, não há clareza sobre como estes dois parâmetros (`epsilon` e `minPts`) foram determinados em sua análise. De fato, Sadeghi apenas o analisou com os métodos de avaliação Silhouette (SI), Davies-Bouldin (DBI) e Calinski-Harabasz (CHI), usando-os para validar e demonstrar o desempenho superior do DBSCAN por sua capacidade de lidar com formas arbitrárias e ruído em análises na área de *geo-data science*. Concluiu que, dentre eles, o DBI e o SI apresentaram resultados melhores, quando comparados ao uso da tradicional combinação K-Means + Elbow, por exemplo.

Neste sentido, neste trabalho foi utilizada a heurística do *k-distance*, para determinar o valor ótimo do parâmetro da distância do raio épsilon (ou `eps` – ϵ). [Schubert et al. 2017] mencionaram que os autores originais do DBSCAN propuseram essa heurística. A abordagem consiste em plotar o gráfico de *k-distance* ordenado (*sorted k-dist graph*), que representa as distâncias do k-ésimo vizinho mais próximo para cada ponto do conjunto de dados, plotadas da maior para a menor distância. Portanto, a heurística estabelece uma dependência entre a escolha do parâmetro `eps` e a definição prévia do parâmetro `minPts`. Nesta heurística, `k` é um valor auxiliar usado para encontrar o raio ideal, e corresponde a `minPts`-1, visto que a contagem de vizinhos mais próximos não inclui o ponto de consulta (`minPts`$-$`ponto_central`). O valor ideal do parâmetro `eps` é encontrado procurando visualmente por um *"valley, knee, or elbow"* (vale, joelho ou cotovelo) no gráfico *k-distance*. Neste caso, foi então utilizado o KneedLocator, seguindo os processos anteriores.

Quanto ao parâmetro `minPts`, Sander et al. citado por Schubert et al., sugerem usar o dobro da dimensionalidade como ponto de partida: `minPts` $= 2 \times D$, onde D seria a quantidade de dimensões do *dataset*. No caso em estudo, tendo em vista que se trabalha com latitude e longitude (2D, logo $D = 2$), o resultado seria `minPts` $= 4$. Schubert et al. defendem que seja mantido no valor padrão de `minPts=4` para qualquer dado bidimensional [Schubert et al. 2017]. Com objetivo de comparar os resultados de diferentes valores para `minPts`, neste trabalho foi utilizado um *range* de valores para este parâmetro, variando de 4 a 15. Nesta implementação, todas as computações de distância utilizam a métrica *Haversine*, que calcula a distância de arco na superfície de uma esfera. Os dados são, portanto, convertidos de graus para radianos antes da análise. A Tabela 8 demonstra os valores do parâmetro `eps` obtidos a partir de cada um dos valores de `minPts` testados, bem como o percentual de ruído identificado e os *scores* das métricas de avaliação de qualidade (SI, DBI e CHI).

Conforme o *framework* de Sadeghi, que realiza a competição entre os *scores* para definir o resultado ótimo, o valor de `minPts=13` (neste caso) foi encontrado como o melhor pontuado. Este valor correspondeu ao `eps` estimado de 0.00013910 em radianos,

determinado pelo Método do Cotovelo (Figura 13), com dois *clusters* e $13,99\%$ de ruído. Considerando o raio médio da circunferência terrestre, que é de 6.371km, a aplicação da fórmula do comprimento de arco ($s = r \cdot \theta$), onde r é raio médio e θ é o ângulo em radianos, resulta em exatos 886,5661 metros como a distância estimada máxima necessária para formar um *cluster* com o mínimo de 13 pontos.

Tabela 8. Resultados da análise do DBSCAN

`minPts`	`eps` (estimado)	# *clusters*	% Ruído	SI	DBI	CHI
4.00	0.00011230	9.00	2.1517%	-0.178316	0.665242	28.8160
5.00	0.00008662	14.00	13.4481%	-0.272568	0.587938	768.0110
6.00	0.00015179	3.00	0.0000%	0.090948	0.559090	49.0323
7.00	0.00011989	6.00	6.9930%	-0.104833	0.699408	35.8502
8.00	0.00010459	6.00	16.6756%	0.085703	0.570392	1920.0844
9.00	0.00014538	4.00	2.1517%	-0.044655	0.630732	34.6840
10.00	0.00013404	4.00	13.4481%	0.098597	0.645022	49.3736
11.00	0.00012195	5.00	20.9790%	0.110559	0.538279	2390.3733
12.00	0.00015664	2.00	6.9930%	0.342967	0.465856	65.6747
13.00	0.00013910	2.00	13.9860%	**0.343038**	**0.465702**	65.6865
14.00	0.00016221	2.00	11.2964%	0.343023	0.465750	65.6843
15.00	0.00015243	2.00	11.2964%	0.343023	0.465750	65.6843

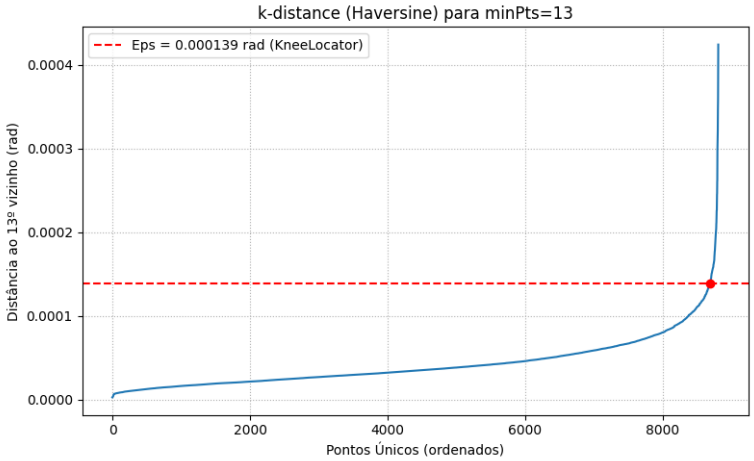
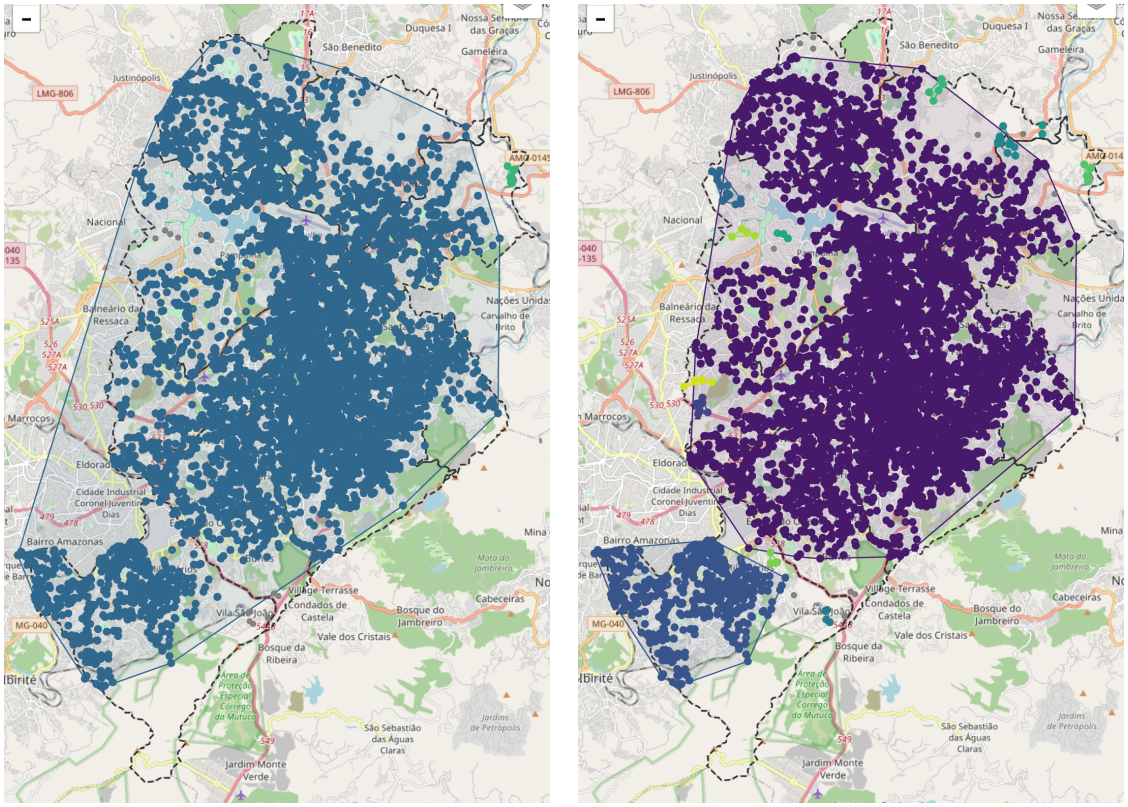


Figura 13. Visualização do `eps` estimado ótimo, para `minPts = 13`, usando o Método do Cotovelo (Elbow).

A Figura 14(a) apresenta a visualização do mapa de *clusters* resultantes para a análise baseada no *framework* de Sadeghi. Como se pode verificar, o resultado não foi satisfatório. Novamente, *clusters* muito extensos não têm significado, ou aplicabilidade operacional para o COP-BH. A análise dos *hotspots* precisa ser direcionada áreas geográficas onde se possa aplicar ações específica, particularizadas às características próprias da área. Retornando à Tabela 8, na busca de outros valores que possam fazer sentido,

verificou-se que, para o valor de `minPts=5`, o `eps` ótimo identificado pelo método Elbow foi notavelmente menor (0.00008662), correspondendo a 552 metros, e a quantidade de *clusters* significativamente maior (14), com percentual de ruído aceitável (13.4481%). A Figura 14(b) apresenta a visualização dos *clusters* encontrados com esses parâmetros.



(a) Visualização dos Clusters identificados pelo DBSCAN, usando a abordagem de Sadeghi, com minPts = 13 e eps = 0.00013910.

(b) Visualização dos Clusters identificados pelo DBSCAN, com referência na heurística k-distância, com minPts = 5 e eps = 0.000087.

Figura 14. Comparação de resultados do DBSCAN com diferentes parâmetros.

Mais uma vez, o resultado apresentado na Figura 14(b) é inadequado. Este comportamento do DBSCAN é, provavelmente, o que foi descrito por [Campello et al. 2015] e [Schubert et al. 2017], como o colapso da estrutura de *clustering* ("comportamento degenerativo"), relacionado a um valor excessivamente grande para o parâmetro do raio de distância (`eps` ϵ). Isso causa um efeito de encadeamento (*"Chaining Effect"*), que leva o DBSCAN a fundir a maioria dos pontos em um único *cluster*. Schubert et al. fornecem diretrizes para se evitar essa degeneração e garantir que o DBSCAN apresente resultados positivos. São elas:

- Regra do Ruído: O DBSCAN deve produzir uma quantidade desejável de ruído (pontos não agrupados), geralmente entre 1% e 30% (dependendo da qualidade dos dados), como um indicador de um épsilon adequado.
- Heurística *k-distance*: O método de k-distância é explicitamente sugerido para escolher o épsilon o menor possível antes que a curva comece a cair bruscamente, evitando assim um raio que seja grande demais e cause a fusão.

Todas essas condições foram atendidas neste experimento, mas ainda assim, o comportamento degenerativo ocorreu. Uma possível causa de o Método do Cotovelo ter encontrado valores tão altos para épsilon, mesmo utilizando a heurística *k-distance* é, talvez, a qualidade do *dataset* de furto de cabos. Como se pode verificar na Figura 13, foram usados pontos de dados únicos para avaliar a curva do cotovelo. O uso de dados únicos para analisar a curva do cotovelo, nesse caso, deveu-se ao fato de que o *dataset* é composto por muitos pontos sobrepostos, ou em que a distância entre eles é muito pequena. Isso reduziu a uma amostra de 8.805 pontos, no caso da análise do DBSCAN (o número de pontos sobrepostos ou com distância mínima é de 9.785 pontos, 52.64% do *dataset*). Numa experimentação com o *dataset* completo, ou seja, considerando também os pontos com distância zero ou com valores pequenos, o KneedLocator encontrou valores ótimos para `eps` ainda maiores, e para alguns valores de `minPts`, não foi possível calcular os índices SI, DBI e CHI, por ter sido gerado apenas um *cluster*, como se pode verificar na Tabela 9, abaixo:

Tabela 9. Resultados da análise do DBSCAN para o *dataset* completo

`minPts`	`eps` (estimado)	# *clusters*	% Ruído	SI	DBI	CHI
4.00	0.00010582	10.00	4.3034%	-0.036455	0.568760	1075.8383
5.00	0.00012330	7.00	4.3034%	-0.108953	0.665021	32.9295
6.00	0.00015757	3.00	0.0000%	0.090948	0.559090	49.0323
7.00	0.00019083	3.00	0.0000%	0.090948	0.559090	49.0323
8.00	0.00022556	3.00	0.0000%	0.090948	0.559090	49.0323
9.00	0.00022556	3.00	0.0000%	0.090948	0.559090	49.0323
10.00	0.00023784	1.00	0.0000%	NaN	NaN	NaN
11.00	0.00023784	1.00	0.0000%	NaN	NaN	NaN
12.00	0.00024077	1.00	0.0000%	NaN	NaN	NaN
13.00	0.00024484	1.00	0.0000%	NaN	NaN	NaN
14.00	0.00023794	1.00	0.0000%	NaN	NaN	NaN
15.00	0.00023794	1.00	0.0000%	NaN	NaN	NaN

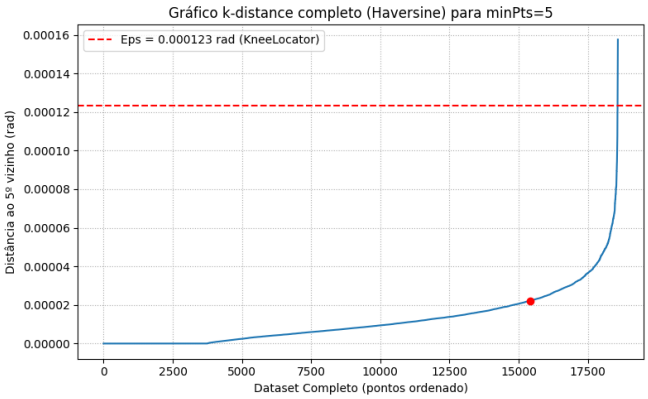
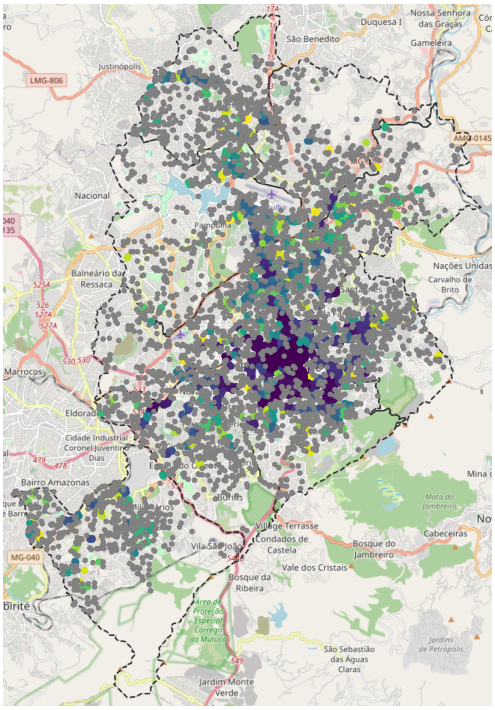


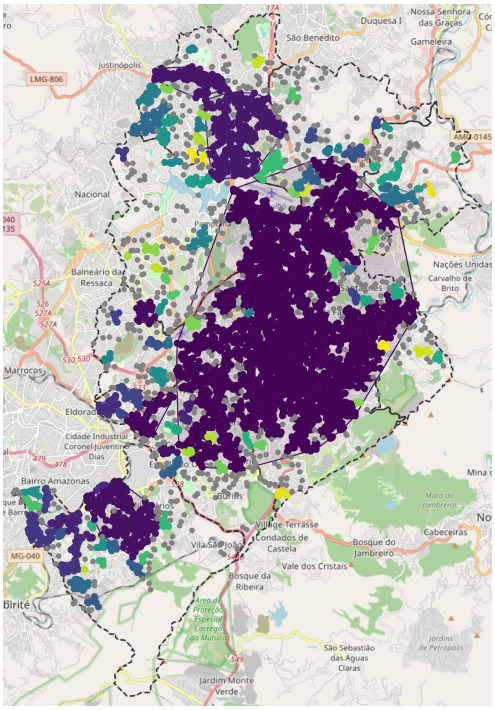
Figura 15. Visualização do gráfico *k-distance* com *dataset* completo, para `minPts = 5`.

O gráfico *k-distance* aplicado ao *dataset* completo gerou um gráfico com uma curva distorcida, com uma linha reta no eixo X, representando os pontos sobrepostos ou com distância zero, e depois a inflexão do cotovelo (Figura 15). O mapa de *clusters* gerado foi similar ao apresentado na Figura 14(a), quanto à forma e localização dos *clusters*. Portanto, verificou-se que utilizar o *dataset* completo para o gráfico *k-distance*, e após, usar o método do Cotovelo para identificar o `eps` ótimo, não ocasionou em bom resultado.

A sobreposição de pontos decorre de dois fatores operacionais: a recorrência de furto de cabos num mesmo local, e a forma com que a definição da coordenada geográfica do ponto em que ocorreu o furto de cabos se dá, na base de dados geográficos da cidade de Belo Horizonte (BDGC - Banco de Dados Geográfico Corporativo). A definição das coordenadas do ponto onde aconteceu o furto de cabos é aproximada, em muitos casos. Ocorre que o (BDGC) identifica a coordenada, primariamente, a partir de um endereço de logradouro. Contudo, o furto de cabos pode ocorrer em qualquer ponto da cidade, e não necessariamente próximo a um endereço de logradouro. Então, para se obter a coordenada geográfica, determina-se um endereço aproximado, e no caso de túneis, viadutos, e trevos (chamados "obras de arte" no BDGC), é calculada uma coordenada geográfica referente ao centroide dessas obras. Portanto, pode-se deduzir que a qualidade do *dataset* tenha prejudicado o uso do Método do Cotovelo, no caso do DBSCAN. Neste sentido, foi arbitrado um valor para `minPts` e `eps`, respectivamente, 10 e 0.0000157 radianos (aproximadamente 100 metros). O resultado foi o oposto: foram gerados 346 *clusters* com 42.7% de ruído, conforme Figura 16(a).



(a) Visualização dos Clusters identificados pelo DBSCAN, com minPts = 10 e eps = 0.0000157 (aproximadamente 100 metros).



(b) Visualização dos Clusters identificados pelo DBSCAN, com minPts = 10 e eps = 0.00003923 (aproximadamente 250 metros).

Figura 16. Comparação de resultados do DBSCAN com diferentes valores de eps.

Mais um experimento foi realizado, alterando o valor de `eps` para aproximadamente 250 metros (0.00003923), mantendo-se o valor de `minPts` em 10, com resultados significativamente melhores, mas ainda assim com a ocorrência do encadeamento (Figura 16(b)). Como se pode verificar, ainda que com a presença do efeito de encadeamento, os prováveis valores ótimos de `minPts` e `eps` estão na região de 10 e 0.00003923 radianos (250 metros). A redução de `minPts` (e.g. `minPts = 7`) e de `eps` (e.g. `eps = 0.00002354` – 150 metros) poderia aumentar o número de *clusters* e reduzir o *"Chaining Effect"*. Infelizmente, o COP-BH não dispõe de informações operacionais, ou empíricas, que possam ser utilizadas como referência para determinar valores adequados para estes parâmetros. Portanto, sua adoção dependeria do teste de várias combinações dos parâmetros, para verificar qual teria o melhor resultado.

4.5. Resultados do HDBSCAN* - Hierarchical Density-Based Spatial Clustering of Applications with Noise

O HDBSCAN* foi o segundo algoritmo baseado em densidade analisado neste trabalho. O objetivo foi encontrar uma abordagem que superasse a limitação principal do DBSCAN, que é a de exigir um único limiar (épsilon – ϵ) de densidade global. O HDBSCAN* trabalha com apenas o parâmetro de número mínimo de pontos (*minimum points* – `min_size` ou `mPts`), que representa a quantidade mínima de pontos em uma vizinhança para que um ponto seja considerado um ponto *core*, atuando como um fator de suavização (2.3). O HDBSCAN* não participou da análise de Sadeghi, em seu experimento "navegando pela incerteza". Assim, buscou-se identificar formas de avaliar a qualidade da sua clusterização, e o DBCV (2.4.1) surgiu como o ideal, tendo em vista que foi proposto explicitamente como um índice de validação relativa para *clusters* baseados em densidade e com formato arbitrário [Moulavi et al. 2014].

Contudo, a tentativa de implementá-lo ao *dataset* completo, usando a biblioteca `hdbscan.validity`, não logrou sucesso, provavelmente pelo mesmo motivo do DBSCAN (4.4): a presença de grande quantidade de pontos sobrepostos ou com distância zero em relação aos seus vizinhos. Ocorre que a matriz de distâncias autogerada calcula a densidade na forma $\frac{1}{distancia}$. Quando a distância foi 0.0, o resultado é um valor ∞. Os erros seguintes decorriam destes valores infinitos, calculados a partir da divisão por zero. Assim, foi feita uma tentativa de truncar pequenas distâncias (incluindo zero) ao mínimo de 10 centímetros ($1e-4$), e variando-o até 20 metros ($2e-2$), mas ainda sem sucesso. Ocorre que, ao realizar a inversão, o valor $\frac{1}{1e-4}$, o *array* de distância recebia valores muito pequenos ou muito grandes, gerando um erro de *broadcast*. Houve a tentativa de utilizar a função `NearestNeighbors` do `sklearn.neighbors` para calcular as distâncias para o *array* `core_distances`, mas com resultados semelhantes.

Então, optou-se por utilizar a mesma estratégia do DBSCAN: utilizar somente pontos únicos para avaliar a qualidade dos *clusters*. Neste sentido, foram identificados 8.805 pontos únicos, e que não tinham distância igual ou próximo de zero em relação à sua vizinhança. A implementação do DBCV se mostrou bastante complexa, em termos de custo computacional, devido à necessidade de tratar os dados de distâncias, o que, nesse caso, exigiu o recálculo da matriz de distância Haversine completa ($N \times N$) para todos os N pontos não-ruído a cada iteração. O cálculo do DBCV, por definição, baseia-se na densidade de cada ponto. Para saber a densidade, cada ponto precisa conhecer a distância para seus vizinhos. A forma mais simples de fazer isso é

calculando uma matriz de distância completa, comparando cada um dos N pontos com todos os outros N pontos. Esta é uma operação de complexidade quadrática ($O(N^2)$) [Moulavi et al. 2014]. Acrescenta-se à essa complexidade a necessidade de que, na implementação, a função `validity_index`, que calcula o DBCV, ter sido responsável por calcular a matriz de distância (com a complexidade $O(N^2)$) para todos os pontos únicos. Isso porque, devido a uma limitação desta mesma biblioteca, não foi possível pré-computar a matriz de distância, o que poderia ter diminuído a carga de processamento (`metric='precomputed'`). Assim, o algoritmo teve que recalcular do início os ≈77,4 milhões de distâncias Haversine. O tempo total deste processamento foi de `2650.57` segundos. Neste sentido, foram testados apenas os seguintes valores para `mPts`: $5, 10, 15, 20, 25e30$. A Tabela 10, a seguir, demonstra os resultados deste experimento.

Tabela 10. Tabela de Análise de Sensibilidade (DBCV)

mPts	num_clusters	dbcv_score	percent_ruido	tamanhos_clusters
5	2	-0.935325	4.5429%	0: 16, 1: 8785
10	2	-0.935325	4.5429%	0: 16, 1: 8785
15	2	-0.935325	4.5429%	0: 16, 1: 8785
20	2	-0.915650	39.7501%	0: 8737, 1: 33
25	2	-0.915650	39.7501%	0: 8737, 1: 33
30	2	-0.915650	39.7501%	0: 8737, 1: 33

Conforme informou [Moulavi et al. 2014], o DBCV gera resultados entre -1 e $+1$ como *scores*, sendo que os mais próximos de -1 são resultados ruins, e os mais próximos de $+1$, bons resultados. Como se pode verificar na tabela, todos os `mPts` foram avaliados como ruins. A quantidade de *clusters* não foi a ideal, mesmo para o reduzido *dataset* de pontos únicos. A própria composição dos *clusters* deixa clara essa estrutura disforme: um *cluster* com 16 pontos e outro com 8.785 pontos (para `mPts` de 5 a 15), por exemplo. Dentre eles, o `mPts=20` foi o considerado menos ruim, por ter sido o primeiro com *score* um pouco mais distante de -1. É bem provável que, caso tivesse sido possível avaliar o *dataset* completo, e este fosse isento de sobreposições, a avaliação através do DBCV tivesse melhores resultados, pois a densidade foi afetada pela redução de pontos.

Numa outra abordagem, a Figura 17 apresenta o Gráfico de Árvore de Clusters Condensada (*simplified cluster tree* ou *cluster tree*) gerada nativamente pelo algoritmo HDBSCAN*, sendo um elemento central do seu *framework*. Segundo Campello et al. [Campello et al. 2015], essa árvore é uma representação simplificada e pós-processada da hierarquia completa de *clusters* gerada pelo algoritmo, sendo descrita como uma ferramenta de análise exploratória essencial e um passo para a extração de soluções de *clustering* planas. O propósito dessa simplificação é extrair uma árvore resumida de apenas *clusters* mais "significativos". A Árvore de Clusters Condensada (ou simplificada) é obtida focando apenas nos níveis hierárquicos em que ocorrem as mudanças mais significativas na estrutura de *clustering* [Campello et al. 2015]. Especificamente, a hierarquia é simplificada ao focar em eventos onde um novo cluster surge através de uma "verdadeira" divisão (*true split*) de um *cluster* existente, ou um *cluster* desaparece completamente.

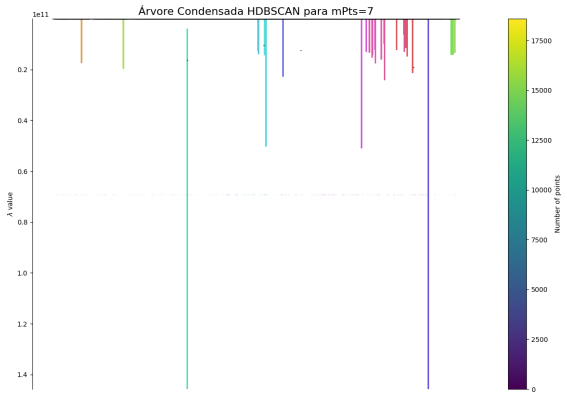


Figura 17. Gráfico de Árvore Condensada para `mPts = 7`.

Qualquer remoção de objetos de ruído de um *cluster* não é considerada uma "verdadeira" divisão, mas apenas um encolhimento desse *cluster*. Para estes casos de encolhimento, os objetos restantes mantêm o mesmo rótulo original. A ideia de simplificação pode ser generalizada usando um tamanho mínimo de *cluster* (`mclSize`). Este é um parâmetro comum para evitar que o algoritmo encontre *clusters* muito pequenos que podem surgir aleatoriamente devido a uma amostra finita de dados. Uma componente é considerada espúria se tiver menos objetos que o `mclSize`, e são rotuladas como ruído. Para simplificar o uso do HDBSCAN*, [Campello et al. 2015] sugerem frequentemente definir o tamanho mínimo do *cluster* (`mclSize`) igual ao fator de suavização de densidade (`mPts`), tornando-o um parâmetro único que controla o suavização das estimativas de densidade e o limiar explícito para o tamanho mínimo dos *clusters*. Conforme Campello et al., a árvore indica:

- Estrutura Hierárquica Significativa: Ela fornece uma descrição mais rica das estruturas de *clustering* do que os modelos "planos" (não hierárquicos).
- Facilidade de Visualização: É adequada para visualização e exploração interativa de dados (por exemplo, como dendrograma ou na forma compactada), embora possa ser difícil de interpretar em sua forma bruta para grandes conjuntos de dados "ruidosos".
- Base para Soluções Planas Otimizadas: A principal utilidade da árvore é permitir a extração de uma solução de *clustering* "plana" (não hierárquica) a partir de cortes locais através da árvore, maximizando uma medida de qualidade, como a estabilidade do *cluster*. Este é um processo de otimização que permite a detecção de *clusters* com densidades variadas, o que não é possível com um único corte horizontal em uma hierarquia tradicional (que usa um único limiar global de densidade).
- Estabilidade e Vida Útil do *Cluster*: A Árvore Condensada é usada para calcular a estabilidade individual de cada *cluster* (baseada no conceito de *relative excess of mass*), que quantifica o quão proeminente um *cluster* é e por quanto tempo ele "sobrevive" ao longo dos níveis de densidade da hierarquia.

Portanto, o Gráfico de Árvore Condensada apresenta, em suas linhas verticais, os *clusters* finais que o algoritmo retornou nos `labels` dos pontos. Linhas que começam

no topo do gráfico e descem significativamente, representam *clusters* de alta densidade e estabilidade. A escala de cores (*number of points*) representa todos os pontos do *dataset* ordenados pelo algoritmo usando uma "ordenação de folha", para que os pontos que estão hierarquicamente próximos (ou seja, que acabam no mesmo *cluster*) fiquem lado a lado. Neste sentido, o comprimento de cada linha vertical da árvore demonstra a estabilidade dos pontos: linhas curtas representam *clusters* pequenos e instáveis, ao passo que linhas longas representam *clusters* robustos e significativos. A Figura 17 apresenta essa Árvore Condensada, gerada para `mPts = 7`, determinado a partir dos valores descritos na Tabela 11.

Tabela 11. Resultados da análise HDBSCAN*

	mPts	num_clusters	percentual_ruido	mean_stability	std_dev_stability
0	5	1318	22.6520%	0.878558	0.249544
1	6	972	27.9666%	0.869458	0.245686
2	7	856	29.8117%	0.870398	0.249634
3	8	693	33.4535%	0.856872	0.256171
4	9	595	34.4648%	0.851240	0.253857
5	10	504	36.8693%	0.851786	0.250232
6	11	454	37.8860%	0.854480	0.244484
7	12	400	39.9677%	0.852404	0.246798
8	13	370	40.8768%	0.862183	0.233884
9	14	330	42.3722%	0.867586	0.230082
10	15	312	43.5180%	0.872142	0.226727
11	16	272	43.9753%	0.854947	0.248567
12	17	239	43.7924%	0.846490	0.256390
13	18	217	45.9172%	0.841552	0.253503
14	19	207	47.6385%	0.846063	0.248427
15	20	191	47.0791%	0.850149	0.238539
16	21	176	47.0737%	0.852839	0.230962
17	22	162	45.8472%	0.846129	0.237558
18	23	142	42.0925%	0.856131	0.222806
19	24	141	42.9693%	0.857038	0.213385
20	25	128	44.1259%	0.860196	0.207854
21	26	122	45.7773%	0.856201	0.213954
22	27	120	46.4658%	0.860508	0.212325
23	28	111	47.4502%	0.866509	0.203050
24	29	106	47.3050%	0.882674	0.190784
25	30	96	47.9290%	0.881527	0.184032

Os dados da tabela foram obtidos a partir do processamento *dataset* completo com o HDBSCAN*, com as 18.590 observações e com um range de `mPts`, variando de 5 a 30. Verifica-se que o resultado do processamento do *dataset* completo foi melhor que as avaliações geradas pelo DBCV, com maior quantidade de *clusters* e percentuais de ruído mais uniformes. Retomando a regra de [Schubert et al. 2017] para o DBSCAN, poderíamos selecionar o `mPts=7` como um candidato com vantagens para ser definido como

ótimo tendo em vista que, conforme exposto na Tabela 11, essa estrutura de *cluster* é a última da lista a ter ruído inferior a 30%. Ademais, grandes quantidades de *clusters* tornam-nos muito pulverizados, como para `mPts=5`, o que pode também não ser signficativo operacionalmente para o COP-BH. Por outro lado, quanto menor for a quantidade de *clusters*, maior será a quantidade de ruído, e isso pode ocasionar perdas na análise do problema. Destarte, considera-se adequada a escolha do `mPts=7` pois sua estrutura de *clusters* pode coincidir com um trecho de uma via, ou com algumas quadras da cidade, sobre as quais se poderá debruçar e pesquisar mais profundamente na busca por entender os motivos que fizeram com que aquela área se destacasse como um *hotspot*. Ainda assim, outras densidades de clusterização (eg. `mPts` entre 8 e 12) podem proporcionar perspectivas diferentes de análises e *insights* sobre o motivo da concentração dos pontos de furto de cabo, podendo contribuir, principalmente, para o monitoramento por câmeras, por abrangerem áreas maiores e ainda assim alcançáveis pelas imagens.

As colunas `mean_stability` e `std_dev_stability` da Tabela 11 apresentam, respectivamente, a média e o desvio padrão do atributo `clusterer.probabilities_`, uma saída nativa do algoritmo HDBSCAN. Consiste de um *array* que quantifica a "certeza" de pertencimento de cada ponto ao seu respectivo *cluster*. Os valores variam de 0.0 a 1.0: pontos de ruído (*noise*) têm probabilidade 0.0, pontos centrais (*core points*) de um *cluster* estável têm probabilidade 1.0 e pontos de borda (*border points*) têm probabilidade entre 0.0 e 1.0 (e.g. 0.5). No cálculo da estabilidade média (`mean_stability`) não foram considerados os pontos de ruído, para maximizar a qualidade geral e a certeza do pertencimento. Adicionalmente, o desvio padrão da estabilidade (`std_dev_stability`) dessas mesmas probabilidades pode ser usado como um indicador de análise para avaliar a consistência e a homogeneidade da certeza dos pontos dentro dos *clusters* encontrados. Embora essa heurística não tenha respaldo na literatura, pode ser razoável para analisar empiricamente a estabilidade.

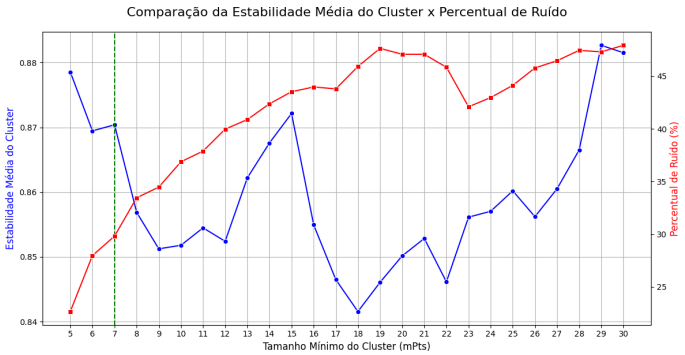
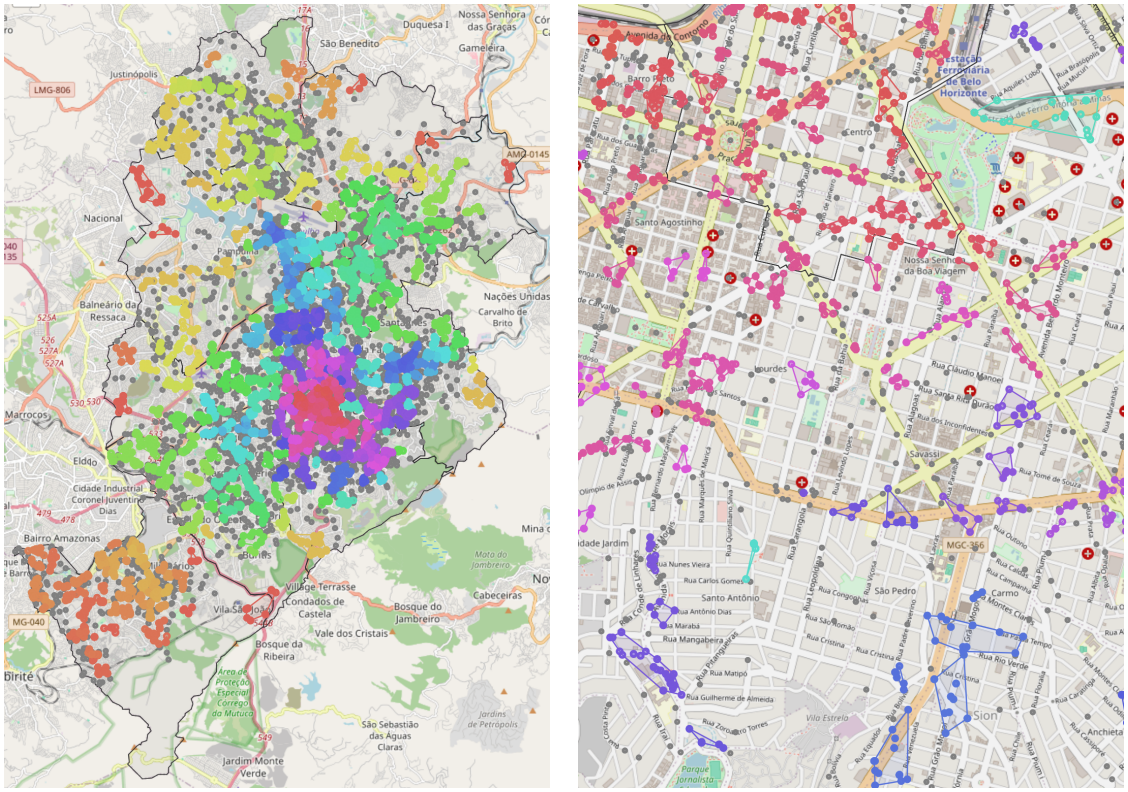


Figura 18. Gráfico comparativo da estabilidade média com o percentual de ruído, em função de `mPts`.

A Figura 18 apresenta um comparativo da estabilidade média com o percentual de ruído, para cada um dos `mPts` testados. Como se pode verificar no gráfico, `mPts=7` apresenta um equilíbrio entre estabilidade média (> 0.87) e percentual de ruído ($< 30\%$), se comparado aos demais valores para `mPts`. Por fim, o mapa de *clusters* apresentado na Figura 19(a) a seguir demonstra a distribuição da clusterização processada pelo

HDBSCAN*. A escala de cores utilizada foi a mesma do Gráfico de Árvore Condensada (Figura 17) apresentado anteriormente, correlacionando a estabilidade dos *clusters* indicada na árvore à sua localização geoespacial.

Numa visão mais aproximada da região central da cidade, onde sabidamente ocorre o maior número de furtos de cabo, a Figura 19(a) apresenta a adequação do HDBSCAN* às necessidades do COP-BH, pois os *clusters* identificados possuem formas arbitrárias, que são mais próximas das características geográficas e urbanas, a partir dos dados geoespaciais. Vê-se, também, que a maior parte dos pontos determinados como ruído estão distanciados dos *clusters*, o que indica o bom tratamento que o algoritmo dá a esses pontos. A partir desta visualização, como ponderado anteriormente, a análise do ponto de vista operacional para entender os motivos que causam o *hotspot*, e a respectiva estratégia de atuação e resposta, é aprimorada. Logicamente, há que se testar futuramente, na prática e *in loco*, se esse resultado é relevante, então se poderá avaliar empiricamente o valor de `mPts=7` e outros.



(a) Visualização dos *Clusters* identificados pelo HDBSCAN, para `mPts = 7`.

(b) Visualização aproximada dos *Clusters* identificados pelo HDBSCAN, para `mPts = 7`.

Figura 19. Resultados da análise de clusterização HDBSCAN.

4.6. Resultados da Avaliação Dinâmica e Validação

Tal como descrito na Metodologia (3.3), esta etapa tem o objetivo de validar a eficácia dos experimentos realizados, sendo o último passo da tentativa de comprovar o valor prático dessa abordagem. Conforme a subseção anterior (4.5), o HDBSCAN* surgiu como algoritmo de clusterização mais robusto para atender à necessidade do COP-BH,

por suas características de lidar melhor com formas arbitrárias de *clusters*, densidades variáveis e ruído. Neste sentido, ele foi selecionado para o teste final, de avaliação dinâmica, estática e validação. Esta experimentação tem o objetivo de responder às seguintes perguntas:

(a) O algoritmo de clusterização HDBSCAN*, identificado como o mais adequado às necessidades do COP-BH, supera a técnica do KDE utilizada tradicionalmente no COP-BH para a identificação preditiva de *hotspots* de furtos de cabo de cobre?
(b) A análise dos dados em granularidades menores (semanal, mensal, bimestral e trimestral) é melhor que a análise dos dados históricos completos?
(c) A aplicação do HDBSCAN* permitirá a identificação da dinâmica espacial do fenômeno, a saber: o surgimento, a dissipação ou o deslocamento dos *hotspots* de furtos de cabo?

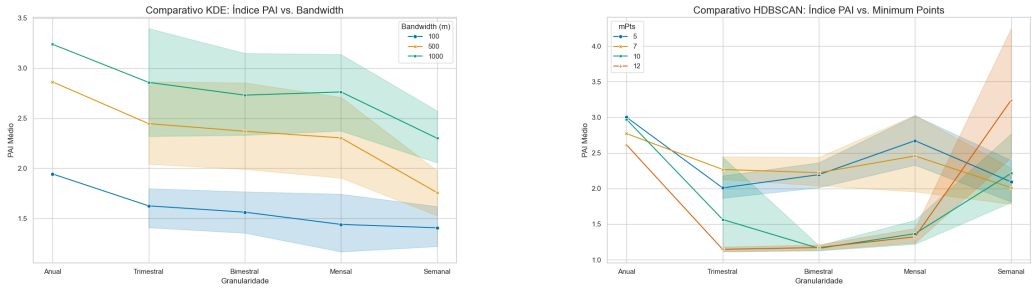
Para tal, o *dataset* foi processado e analisado pelas técnicas HDBSCAN* e KDE. Foram utilizados os dados dos anos de 2023 e 2024 por estarem completos, sendo que o ano de 2023 foi usado como treinamento e o ano de 2024 como teste. A métrica PAI - *Prediction Accuracy Index* (2.4.4) foi utilizada para calcular a eficiência preditiva de *hotspots* pelos algoritmos HDBSCAN* e KDE, tendo sido aplicadas a um *range* de valores:

- Parâmetro *minimum points* (número mínimo de pontos) do HDBSCAN*: `mPts = [5, 7, 10, 12]`, selecionados a partir do experimento demonstrado na Subseção 4.5, como valores representativos devido ao nível de ruído que apresentaram;
- Parâmetro *bandwidth* (largura de banda) do KDE: `bandwidth = [100, 500, 1000]`, determinados a partir do método tradicional (empírico) utilizado no COP-BH, sem qualquer otimização, para geração de *hotspots*. Além deste parâmetro, foram considerados os valores de 50m (metros) para o *grid cell size* e, como limiar de *hotspot* (a), o valor de 5%, ou seja, os 5% da área da cidade com maior densidade de crimes. Estes valores foram assim definidos por ausência de referência, portanto foi utilizado o mesmo padrão do experimento de [Chainey et al. 2008].

O processamento do KDE iniciou com a reprojeção de todos os pontos de crime (EPSG:4326) para o sistema métrico da área de estudo (EPSG:31983), permitindo a aplicação dos *bandwidths* de 1000m, 500m e 100m. Para cada granularidade temporal (semanal, mensal, bimestral e trimestral) e parâmetro, os dados de treinamento (2023) foram usados para gerar uma superfície de densidade sobre um *grid* (50x50m). O hotspot (a) foi então definido de forma controlada, selecionando o limiar de densidade que correspondia a uma porcentagem fixa da área total do município (5%). Esta área a foi utilizada para calcular a taxa de acerto (n) dos dados de teste (2024) e, subsequentemente, o PAI.

Para o HDBSCAN, o processamento compartilhou a mesma etapa inicial de reprojeção para o sistema métrico (EPSG:31983), essencial para o cálculo de distâncias. A iteração de parâmetros foi realizada sobre os *minimum points* (`mPts`) de 5, 7, 10 e 12. A definição do hotspot (a) foi o diferencial metodológico: em vez de uma porcentagem de área fixa, o *hotspot* foi definido de forma *data-driven*, ou seja, a própria estrutura e distribuição espacial dos pontos no conjunto de dados (os crimes de 2023) foi o que determinou a forma, o tamanho e a localização dos *hotspots*. Isso porque o HDBSCAN* não parte de

uma área pré-definida, ele examina as relações de densidade entre cada ponto e identifica agrupamentos "naturais" (os *clusters*) que são estatisticamente estáveis e se separam de ocorrências isoladas (ruído). Dessa forma, a área de *hotspot* (a) deixa de ser um parâmetro de entrada arbitrário (como os 5% da área total no KDE), e passa a ser um resultado direto da análise. O *hotspot* é, literalmente, a área geométrica real dos *clusters* que o algoritmo encontrou, e esse valor a varia em cada período de tempo, refletindo a verdadeira aglomeração dos dados daquele período. Executamos, então, o HDBSCAN* sobre dados de treino (2023), identificamos os *clusters* estáveis (excluindo ruído), e definimos a como a área geométrica real desses *clusters* (calculada via função `Convex Hull`). Esta área a, que é baseada na estrutura dos dados e pode variar, foi então usada para o cálculo do PAI contra os dados de teste (2024). A Figura 20 demonstra o desempenho PAI dos dois algoritmos, levando-se em consideração os valores dos parâmetros testados e as granularidades propostas:



(a) KDE: PAI por `bandwidth` e Granularidade. (b) HDBSCAN*: PAI por `mPts` e Granularidade.

Figura 20. Desempenho do Índice PAI por parâmetros dos algoritmos KDE e HDBSCAN*.

Como se pode verificar, o KDE obteve melhor desempenho geral com o parâmetro `bandwidth` igual a 1000m, inclusive para a granularidade maior, anual, que é a mais utilizada pelo COP-BH. Para a granularidade semanal o desempenho foi o menor, alcançando o PAI médio de aproximadamente 2.3, provavelmente devido à característica contínua dos *hotspots* gerados por ele, que desfavorecem visualizações em escalas pequenas. Por sua vez, o HDBSCAN teve desempenhos variados, em grande parte, inferiores ao KDE. O `mPts=5` teve desempenho inesperadamente melhor que o `mPts=7`, contrariando a percepção da avaliação do HDBSCAN* (4.5), provavelmente devido ao menor percentual de ruído e maior quantidade de *clusters*. Por sua vez, `mPts=12` destaca-se pelo seu bom desempenho na análise semanal dos dados, ou seja, de baixa granularidade. Pode-se concluir que as duas abordagens têm vantagens e desvantagens devido às suas características, quanto à análise da dinâmica temporal dos dados. Cada uma delas terá uma aplicação mais adequada, que poderá contribuir com melhores análises, dependendo da necessidade operacional do COP-BH.

Noutra perspectiva, a Tabela 12 a seguir sumariza os resultados médios encontrados, por granularidade e algoritmo, e a Figura 21 apresenta o comportamento da curva média desses resultados.

Tabela 12. Comparativo de Métricas PAI por Algoritmo e Granularidade

Granularidade	HDBSCAN			KDE		
	PAI	Hit Rate (%)	Área (%)	PAI	Hit Rate (%)	Área (%)
Anual	2.8389	20.5025	7.2953	2.6820	13.4102	5%
Trimestral	1.7470	49.9595	40.2832	2.3097	11.5488	5%
Bimestral	1.6872	55.0496	44.1460	2.2210	11.1050	5%
Mensal	1.9540	52.2464	38.1101	2.1686	10.8433	5%
Semanal	2.3899	33.1991	18.8627	1.8209	9.1048	5%

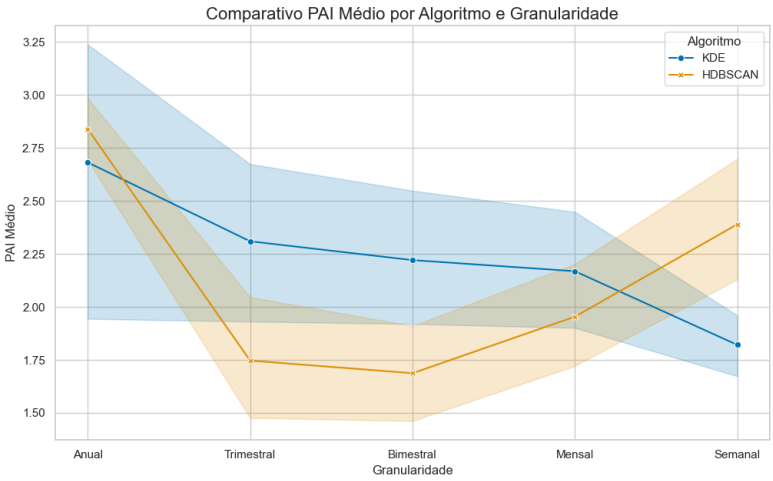
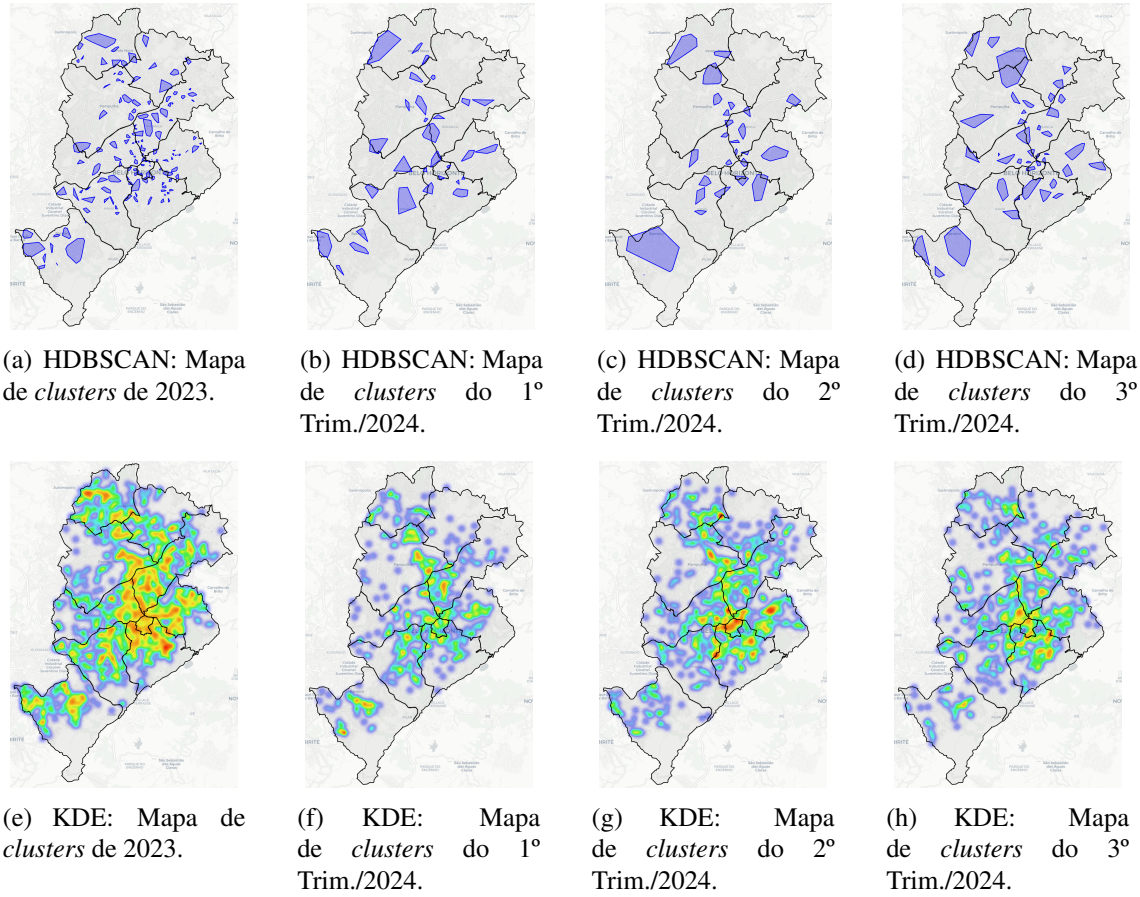


Figura 21. Gráfico comparativo do desempenho PAI por Algoritmo.

Analisando o gráfico apresentado na Figura 21, pode-se perceber que os dois algoritmos tiveram desempenho aproximado, com ligeira vantagem para o KDE. O HDBSCAN* se sobressaiu nos extremos, sendo que na granularidade Semanal com maior vantagem. Deduz-se que seja melhor para análises detalhadas, de curto prazo, e com pequena vantagem na análise de longo prazo. Quanto ao médio prazo, o KDE tem expressiva superioridade. Contudo, vale ressaltar que o HDBSCAN* tem a capacidade de lidar com *clusters* de densidades variadas e áreas também variadas, tal como demonstrado na Tabela 12. Já o KDE é limitado a uma determinada área de análise global, o que pode indicar maior precisão no acerto das previsões feitas pelo HDBSCAN*, em comparação com o KDE. Mais uma vez, o KDE pode ter sido favorecido pela sua característica de área continua, ao passo que o HDBSCAN* delimita áreas discretas e com formas arbitrárias.

Por fim, a Figura 22, a seguir, demonstra a capacidade que o HDBSCAN tem para identificar a dinâmica espacial do fenômeno, justamente pela última vantagem citada. Para fins exemplificativos, foi utilizado o mapa de *clusters* do período de 2023 como base comparativa, e contraposto às granularidades Trimestral de 2024. Como se pode verificar, é possível ver, mais claramente nos mapas gerados pelo HDBSCAN* a dinâmica espacial do furto de cabos, o que é útil para análises de curto prazo. Observou-se que, embora *hotspots* menores de surgimento e dissipação ocorram mensalmente, os *clusters* de maior

volume demonstraram alta permanência. Isso corrobora os achados da análise preditiva, que apontaram os modelos anuais e trimestrais como os mais robustos, indicando que os fatores de oportunidade que geram os *hotspots* principais deste delito são estruturalmente estáveis no longo prazo.



(a) HDBSCAN: Mapa de *clusters* de 2023.

(b) HDBSCAN: Mapa de *clusters* do 1º Trim./2024.

(c) HDBSCAN: Mapa de *clusters* do 2º Trim./2024.

(d) HDBSCAN: Mapa de *clusters* do 3º Trim./2024.

(e) KDE: Mapa de *clusters* de 2023.

(f) KDE: Mapa de *clusters* do 1º Trim./2024.

(g) KDE: Mapa de *clusters* do 2º Trim./2024.

(h) KDE: Mapa de *clusters* do 3º Trim./2024.

Figura 22. Comparativos da visualização da dinâmica espacial de furtos de cabo.

5. Considerações Finais

Este trabalho propôs e validou uma metodologia automatizada para a análise espaço-temporal de *hotspots* de furto de cabos em Belo Horizonte. Ao aplicar o *framework* "Navegando pela Incerteza", foi possível quantificar o desempenho e selecionar o HDBSCAN* como a ferramenta mais robusta para o agrupamento dos dados e análise da dinâmica geoespacial. Foi demonstrado que a abordagem tradicional de KDE estático, atualmente em uso pelo COP-BH, tem vantagens e desvantagens com relação ao HDBSCAN*, que surgiu como ferramenta que provê especialização à análise do fenômeno. Os resultados da análise de eficiência (medidos pelo PAI) revelam que, para modelos preditivos, o KDE ainda possui vantagens, o que leva à conclusão de que estas ferramentas devem ser aplicadas em conjunto para atender a necessidades específicas, tanto operacionais quanto analíticas, pelo COP-BH.

A principal contribuição desta pesquisa é a entrega de um método validado que pode ser diretamente aplicado pelo COP-BH para otimizar o monitoramento da infraes-

trutura crítica da cidade. No entanto, reconhece-se que a análise de sensibilidade dos algoritmos (quanto às parametrizações do HDBSCAN* e do KDE) é crucial e impacta significativamente os resultados, exigindo uma calibração cuidadosa para cada granularidade. Conforme discutido, este trabalho teve o seu foco na comparação dos métodos para entender as respectivas aplicabilidade, não adentrando à exploração completa das calibragens dos algoritmos. Cabe ressaltar que métodos como o GridSearchCV podem robustecer a parametrização do KDE, levando-o a resultados ainda melhores. Por sua vez, o HDBSCAN* já implementa e otimiza nativamente a clusterização dos dados, mas ainda há a possibilidade de otimizar a determinação do número mínimo de pontos, seja por testes empíricos feitos no COP-BH, ou pelo DBCV, com o tratamento adequado do *dataset*. Neste sentido, a equipe de tecnologia do COP-BH deverá se responsabilizar por essa calibragem, para garantir a eficácia do método. Vislumbram-se como aplicações operacionais da metodologia de clusterização espacial e a granularização temporal dos dados, no combate ao furto de cabos:

1. Identificação de manchas criminais com geometria real (HDBSCAN*): Diferentemente do mapa de calor tradicional usado no COP-BH, ou varreduras circulares, o uso do HDBSCAN* permite identificar *clusters* de formatos irregulares e arbitrários. Isso é crucial para o furto de cabos, pois o crime segue a infraestrutura de rede (linhas de transmissão, avenidas, etc.). Ação prática:
    - O policiamento não deve ser despachado para um "raio" ou "bairro", mas sim para o traçado específico do *cluster* identificado (e.g. "toda a extensão da Avenida A, entre as ruas A e B"), otimizando a rota de patrulha para cobrir a infraestrutura atacada e não apenas uma área geográfica genérica.
2. Filtragem de ruído e foco operacional: Uma característica nativa do HDBSCAN* é a capacidade de classificar pontos isolados como "ruído", separando-os dos *clusters* de alta densidade. Exemplo prático:
    - A gestão operacional pode ignorar eventos dispersos (ruído) que não representam tendências organizadas, focando recursos escassos apenas nos "pontos nucleares" (*core points*) dos *clusters* validados pelo algoritmo. Isso aumenta a eficiência do despacho de viaturas e recursos operacionais, evitando o desperdício de efetivo em ocorrências isoladas que não possuem padrão de repetição.
3. Alocação dinâmica por janelas de tempo (Granularização): A técnica de granularização revelou que a densidade dos *clusters* varia drasticamente conforme a janela temporal.
    - A escala de trabalho das equipes de monitoramento e das rondas da equipe de campo pode ser ajustada conforme os picos revelados pela granularização. Se a análise mostra que um *cluster* específico na região Centro-Sul ocorre somente na janela da madrugada de dias úteis, o monitoramento deve ser intensificado especificamente nessa faixa de tempo, permitindo que o monitoramento das câmeras opere em "modo de alerta" nesses horários críticos.
4. Mitigação da fadiga no videomonitoramento: A combinação da localização exata (HDBSCAN*) com o tempo exato (Granularização) atua como um filtro cognitivo para os operadores. Aplicação prática:
    - Em vez de vigiar aleatoriamente centenas de câmeras, o operador pode receber alertas priorizados: "Câmera X (dentro do Cluster 1) está no horário

crítico identificado pela granularização". Isso reduz a carga cognitiva e direciona a atenção humana para onde o risco estatístico é máximo.

Portanto, a etapa mais premente é a validação empírica da metodologia ora proposta, por meio da verificação prática da sua eficácia, tanto no entendimento da dinâmica espaço-temporal do fenômeno, quanto na aferição da sua eficácia enquanto ferramenta integrante da estratégia de combate ao crime de furtos de cabo de cobre. Esta validação empírica se dará por meio do uso das informações, analisadas e geradas a partir da metodologia proposta, nas atividades operacionais cotidianas do COP-BH, com o uso de um Roteiro de Monitoramento remodelado e conhecimento de campo, para entender como as informações geradas podem contribuir com novas percepções e perspectivas sobre o fenômeno no ambiente real urbano.

O próximo passo será transformar a metodologia em uma aplicação completa, que execute a ingestão e tratamento automatizado dos dados, o processamento, e a apresentação dos resultados em uma página Web, que permita a interação pela equipe operacional do COP-BH com os mapas dinâmicos e granularidades temporais. O passo seguinte será o acréscimo de outros dados, como o volume de cabos furtados por evento, informações econômicas, como o valor de mercado do cobre, e sociológicos como a existência de moradores em situação de rua, por exemplo, para ampliar a capacidade de análise sobre influências externas sobre o fenômeno.

Além disso, futuramente, a aplicação deste *framework* comparativo proposto pode servir à análise de outros tipos de problemas públicos monitorados pelo centro de operações relacionados ao conceito de "Desordem Física e Social", tais como deposição clandestina de inservíveis, acidentes de trânsito, vandalismo e barulho excessivo, que são tratados pela Linha de Atuação de Prevenção de Problemas, com distintas abordagens de tratamento. Por fim, o método serve como alicerce para futuras integrações, conforme vislumbrado na introdução, tal como o sistema de vídeomonitoramento utilizado no COP-BH, para automatização da execução de modelos de Visão Computacional desenvolvidos para detectar problemas públicos.

Referências

[Aminikhanghahi and Cook 2017] Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowledge and information systems*, 51(2):339–367.

[Arvai 2020] Arvai, K. (2020). Kneed documentation. GitHub.

[Bai and Perron 1998] Bai, J. and Perron, P. (1998). Estimating and testing models with multiple structural changes. *Econometrica*, 66:47–78.

[Baqir et al. 2020] Baqir, A., ul Rehman, S., Malik, S., ul Mustafa, F., and Ahmad, U. (2020). Evaluating the performance of hierarchical clustering algorithms to detect spatio-temporal crime hot-spots. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5.

[Barr and Pease 1990] Barr, R. J. and Pease, K. (1990). Crime placement, displacement, and deflection. *Miscellaneous*, 12:277–318.

[Campello et al. 2015] Campello, R. J. G. B., Moulavi, D., Zimek, A., and Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Trans. Knowl. Discov. Data*, 10(1).

[Chachuli et al. 2016] Chachuli, S. M., Nazri, S. M., Yusop, N., and Mohamad, N. (2016). Cable theft monitoring system $ctms$ using gsm modem. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 2(1):57–66.

[Chainey et al. 2008] Chainey, S., Tompson, L., and Uhlig, S. (2008). The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21:4–28.

[COP-BH 2019] COP-BH (2019). Modelo de gestão integrada do cop-bh.

[Garcıa et al. 2019] Garcıa, G., Silveira, J., Poco, J., Paiva, A., Nery, M. B., Silva, C. T., Adorno, S., and Nonato, L. G. (2019). Crimanalyzer: Understanding crime patterns in são paulo. *IEEE transactions on visualization and computer graphics*, 27(4):2313–2328.

[Govender 2013] Govender, R. (2013). *Investigation towards the prevention of cable theft from Eskom*. PhD thesis, University of Zululand.

[Hastie et al. 2009] Hastie, T., Tibshirani, R., Friedman, J., et al. (2009). *The elements of statistical learning*. Springer series in statistics New-York.

[Herdiana et al. 2025] Herdiana, I., Kamal, M. A., Estri, M. N., et al. (2025). A more precise elbow method for optimum k-means clustering. *arXiv preprint arXiv:2502.00851*.

[Kalinic and Krisp 2018] Kalinic, M. and Krisp, J. M. (2018). Kernel density estimation (kde) vs. hot-spot analysis–detecting criminal hot spots in the city of san francisco. *Lund, Sweden*.

[Kulldorff 1997] Kulldorff, M. (1997). A spatial scan statistic. *Communications in Statistics - Theory and Methods*, 26:1481–1496.

[Mandalapu et al. 2023] Mandalapu, V., Elluri, L., Vyas, P., and Roy, N. (2023). Crime prediction using machine learning and deep learning: A systematic review and future directions. *IEEE Access*, 11:60153–60170.

[Mohler et al. 2011] Mohler, G., Short, M., Brantingham, P., Schoenberg, F., and Tita, G. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106:100–108.

[Moulavi et al. 2014] Moulavi, D., Jaskowiak, P. A., Campello, R. J., Zimek, A., and Sander, J. (2014). Density-based clustering validation. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 839–847. SIAM.

[Pretorius 2012] Pretorius, W. L. (2012). *A criminological analysis of copper cable theft in Gauteng*. University of South Africa.

[Robinson 2009] Robinson, W. S. (2009). Ecological correlations and the behavior of individuals*. *International Journal of Epidemiology*, 38(2):337–341.

[Sadeghi 2025] Sadeghi, B. (2025). Clustering in geo-data science: Navigating uncertainty to select the most reliable method. *Ore Geology Reviews*, page 106591.

[Schubert et al. 2017] Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

[Sidebottom et al. 2014] Sidebottom, A., Ashby, M., and Johnson, S. D. (2014). Copper cable theft: Revisiting the price–theft hypothesis. *Journal of Research in Crime and Delinquency*, 51(5):684–700.

[Tavares 2009] Tavares, R. (2009). *Extensões da Estatística Scan Espacial utilizando Técnicas de Otimização Multiobjetivo*. PhD thesis, Universidade Federal de Minas Gerais $UFMG$.