



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CLEISON HERNANY COSTA

**UMA ANÁLISE DOS IMPACTOS DE HIPERPARÂMETROS APLICADOS AO
ALGORITMO KDE PARA A GERAÇÃO DE MAPAS DE HOTSPOTS CRIMINAIS**

QUIXADÁ

2022

CLEISON HERNANY COSTA

UMA ANÁLISE DOS IMPACTOS DE HIPERPARÂMETROS APLICADOS AO
ALGORITMO KDE PARA A GERAÇÃO DE MAPAS DE HOTSPOTS CRIMINAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Regis Pires Magalhães

Coorientador: Me. Francisco Carlos F. N.
Junior

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C871a Costa, Cleison Hernany.
Uma análise dos impactos de hiperparâmetros aplicados ao algoritmo kde para a geração de mapas de hotspots criminais / Cleison Hernany Costa. – 2022.
40 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Ciência da Computação, Quixadá, 2022.

Orientação: Prof. Dr. Regis Pires Magalhães.

Coorientação: Prof. Me. Francisco Carlos F. N. Junior.

1. Otimização. 2. Algoritmos. 3. Parâmetros. I. Título.

CDD 004

CLEISON HERNANY COSTA

UMA ANÁLISE DOS IMPACTOS DE HIPERPARÂMETROS APLICADOS AO
ALGORITMO KDE PARA A GERAÇÃO DE MAPAS DE HOTSPOTS CRIMINAIS

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da Computação
do Campus de Quixadá da Universidade Federal
do Ceará, como requisito parcial à obtenção do
grau de bacharel em Ciência da Computação.

Aprovada em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. Regis Pires Magalhães (Orientador)
Universidade Federal do Ceará (UFC)

Me. Francisco Carlos F. N. Junior (Coorientador)
Universidade Federal do Ceará (UFC)

Prof^ª. Ma. Livia Almada Cruz
Universidade Federal do Ceará (UFC)

Aos meus pais, minha irmã e minha avó.

AGRADECIMENTOS

Agradeço primeiramente a minha mãe, Raimunda Esmeralda Ricardo Costa, por sempre ter feito de tudo que estava ao seu alcance para me apoiar, por sempre ter acreditado em mim e me dado incentivo para ir em busca dos meus objetivos. Ao meu pai, José Maria da Costa, por todo o sacrifício que fez para me dar a base necessária para seguir em frente. A minha avó, Rita Pereira do Nascimento Ricardo, pelo cuidado e carinho que sempre teve comigo. A minha irmã, Maria Lucicleide da Costa, por sempre estar disposta a me ajudar quando eu precisei.

Aos meus amigos de longa data Pedro Ramylli da Silva, Beatriz Freitas de Medeiros e Stefane Adna dos Santos por todo o companheirismo, pelos bons momentos juntos e pela força nos momentos de incerteza.

Aos meus amigos que dividiram apartamento comigo em Quixadá, José Ediberto do Nascimento Junior, Wesley Oliveira Castro, Carlos Iury Santana Furtado, Jessé Veríssimo de Oliveira, José Ronaldo Guilherme Filho e Victor Ticiano Alves de Vasconcelos, vocês fizeram parte de um ciclo muito importante na minha vida, e mesmo com todas as dificuldades que passamos juntos, foi bom dividir esses momentos com vocês.

Meus mais sinceros agradecimentos aos meus nobres orientadores, Prof. Dr. Regis Pires Magalhães e Me. Francisco Carlos F. N. Junior, pelas conversas, conselhos, disponibilidade, e principalmente pela paciência e a compreensão que tiveram comigo durante a produção deste trabalho. Sou muito grato. Eu não poderia ter sido melhor orientado. Agradeço também a Prof^a. Ma. Livia Almada Cruz pela disponibilidade em participar da banca e pelas sugestões de melhoria do trabalho.

Agradeço aos demais professores e também aos servidores da UFC - Quixadá, por todo o conhecimento transmitido e por tantas experiências boas proporcionadas. Todos os aprendizados nesse período foram de extrema importância para minha formação, tanto acadêmica como profissional e também pessoal.

“Você se torna aquilo que faz repetidamente. Excelência não é um esforço, é um hábito.”

(Paulo Muzy)

RESUMO

O uso de aprendizado de máquina vem crescendo a cada dia e conta com aplicações em diversas áreas do cotidiano. Um exemplo do uso de aprendizado de máquina está na área de segurança pública, na qual existe uma necessidade de determinar locais de risco para que sejam direcionados reforços de patrulhamento. Diversos métodos já foram propostos para resolver esse problema. Em sua grande maioria, essas técnicas precisam que sejam feitas manualmente a configuração de parâmetros para que possam obter melhores resultados. Nessa necessidade, encontra-se o problema de otimização de hiperparâmetros. Este trabalho busca analisar os impactos de diferentes *hiperparâmetros* no resultado do modelo, para isso, são utilizadas as técnicas de otimização de hiperparâmetros *grid search* e *random search*, tendo como caso de uso o problema de geração de mapas de *hotspots* de crimes utilizando a técnica KDE. Além disso, este trabalho propõe uma extensão do KDE, combinando funções de kernel no espaço e no tempo como hiperparâmetros de otimização para melhorar os resultados. Os experimentos demonstram que com o uso do *random search* obteve-se resultados tendo avaliação até 7,63% melhor do que é obtido usando o *grid search*, devido principalmente a utilização de um *bandwidth* menor (318m). Também é observado que adicionar ao KDE uma função de kernel no tempo que diminui a relevância dos eventos mais antigos, resulta em uma melhora de 4,31% nos resultados do *grid search*.

Palavras-chave: Otimização. Algoritmos. Parâmetros.

ABSTRACT

The use of machine learning is growing every day and has applications in several areas of everyday life. An example of the use of machine learning is in the area of public safety, where there is a need to determine risky locations so that patrolling reinforcements can be directed. Several methods have already been proposed to solve this problem. For the most part, these techniques need to be manually configured parameters so that they can obtain better results. In this need, there is the problem of hyperparameter optimization. This work seeks to analyze the impacts of different hyperparameters on the model result, for this, the grid search and random search hyperparameter optimization techniques are used, having the problem as a use case. for generating crime hotspot maps using the KDE technique. Furthermore, this work proposes an extension of KDE, combining kernel functions in space and time as optimization hyperparameters to improve results. The experiments demonstrate that with the use of random search, results were obtained with an evaluation up to 7.63% better than what is obtained using the grid search, mainly due to the use of a lower bandwidth ($318m$). It is also observed that adding a kernel time function to KDE that decreases the relevance of older events results in a 4.31% improvement in grid search results.

Keywords: Optimization. Algorithms. Parameters.

LISTA DE ILUSTRAÇÕES

Figura 1 – Regiões dentro do espaço de busca	19
Figura 2 – Seleção de parâmetros com <i>grid search</i> e <i>random search</i>	20
Figura 3 – Fluxo de atividades	26
Figura 4 – Visualização dos eventos na cidade de São Francisco.	30
Figura 5 – Visualização da grade de células de $100m^2$ sobre a cidade de São Francisco.	31
Figura 6 – Divisões dos conjuntos de dados para treino e teste.	32
Figura 7 – Kernel no tempo aplicado ao KDE.	36
Figura 8 – Visualização da função de decaimento exponencial com diferentes <i>bandwidths</i>	36

LISTA DE TABELAS

Tabela 1 – Resultados do <i>grid search</i> aplicado ao KDE avaliado com a métrica HR. . .	33
Tabela 2 – Resultados do <i>grid search</i> aplicado ao KDE avaliado com a métrica PAI. . .	33
Tabela 3 – Melhores parâmetros obtidos no <i>random search</i> aplicado ao KDE avaliado com a métrica HR.	34
Tabela 4 – Melhores parâmetros obtidos no <i>random search</i> aplicado ao KDE avaliado com a métrica PAI.	34
Tabela 5 – Resultados do <i>grid search</i> aplicado ao KDE e kernel no tempo avaliado com a métrica HR.	37
Tabela 6 – Resultados do <i>grid search</i> aplicado ao KDE e kernel no tempo avaliado com a métrica PAI.	37

LISTA DE QUADROS

Quadro 1 – Comparação com os trabalhos relacionados.	25
Quadro 2 – Hiperparâmetros utilizados no KDE com grade de células.	33

LISTA DE ABREVIATURAS E SIGLAS

KNN	<i>K-Nearest Neighbor</i>
SMBO	<i>Sequential Model-Based Optimization</i>
SVM	<i>Support Vector Machines</i>
BRT	<i>Boosted Regression Trees</i>
KDE	<i>Kernel Density Estimation</i>
HR	<i>Hit Rate</i>
PAI	<i>Predictive Accuracy Index</i>
RRI	<i>Recapture Rate Index</i>
NAS	<i>Neural Architecture Search</i>
RF	<i>Random Forest</i>
STKDE	<i>Space Time Kernel Density</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	17
1.1.1	<i>Objetivo Geral</i>	17
1.1.2	<i>Objetivos Específicos</i>	17
1.1.3	<i>Estrutura do trabalho</i>	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	Otimização de hiperparâmetros	18
2.2	Espaço de busca	18
2.3	Técnicas de otimização de hiperparâmetros	19
2.3.1	<i>Grid Search</i>	19
2.3.2	<i>Random Search</i>	20
2.4	<i>Hotspot Mapping</i>	21
2.5	<i>Kernel Density Estimation</i>	21
2.6	Métricas de avaliação	22
2.6.1	<i>Hit Rate - HR</i>	22
2.6.2	<i>Predictive Accuracy Index - PAI</i>	22
2.6.3	<i>Recapture Rate Index - RRI</i>	23
3	TRABALHOS RELACIONADOS	24
3.1	<i>Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS</i>	24
3.2	<i>Random Search for Hyper-Parameter Optimization</i>	24
3.3	<i>Hyperparameter Tuning and Performance Assessment of Statistical and Machine-Learning Algorithms Using Spatial Data</i>	25
4	METODOLOGIA	26
4.1	Escolha e obtenção do conjunto de dados	26
4.2	Pré-processamento dos dados	27
4.3	Seleção de técnicas de otimização de hiperparâmetros	27
4.4	Definição de cenários para realização dos experimentos	27
4.5	Aplicação das técnicas para obtenção de resultados e seleção dos melhores hiperparâmetros	28

4.6	Avaliação dos resultados	28
5	EXPERIMENTOS E RESULTADOS	29
5.1	Conjunto de dados	29
5.1.1	<i>Pré-processamento</i>	29
5.2	Geração dos mapas de <i>hotspots</i>	30
5.2.1	<i>Conjuntos de treino e teste</i>	31
5.2.2	<i>Execução e Avaliação</i>	32
5.2.2.1	<i>Grid Search</i>	32
5.2.2.2	<i>Random Search</i>	33
5.2.2.3	<i>KDE e Kernel no tempo</i>	34
6	CONCLUSÕES E TRABALHOS FUTUROS	38
	REFERÊNCIAS	40

1 INTRODUÇÃO

Aprendizado de máquina é a ciência de programar computadores para que eles possam aprender a partir dos dados. Um algoritmo de aprendizado de máquina, também chamado de modelo de aprendizado de máquina, é um processo computacional que usa dados de entrada para realizar uma tarefa sem ser explicitamente programado para produzir um resultado particular (NAQA; MURPHY, 2015). Esses modelos têm aplicações práticas em diversas áreas, tais como: reconhecimento de padrões, visão computacional, finanças, segurança pública, entre outras.

Modelos de aprendizado de máquina, em geral, são parametrizados, ou seja, possuem parâmetros de configuração que são usados no processo de treino. Existem dois tipos de parâmetros para esses modelos: os parâmetros de primeiro nível, ou simplesmente parâmetros, e os de segundo nível, também chamados de hiperparâmetros (PROBST *et al.*, 2019). A forma como são definidos os valores e o seu papel no processo de treino são diferentes, entender essa diferença e saber aplicar a técnica de otimização mais adequada poderá melhorar significativamente a performance do modelo tanto em tempo de processamento, quanto na sua acurácia.

Os parâmetros são internos ao modelo e os seus valores são estimados com base nos dados a medida que ocorre o processo de treino. Esses valores são fundamentais para a execução do algoritmo e fazem parte da estrutura interna de como a técnica funciona, entre alguns exemplos de parâmetros, podemos citar os coeficientes em uma regressão linear ou os pesos dos neurônios em uma rede neural.

Os hiperparâmetros por sua vez são externos ao modelo, ou seja, os valores não podem ser computados de forma implícita com base no conjunto de dados. Isso demanda que haja um esforço prévio para defini-los. Um exemplo simples é o valor de K para o algoritmo *K-Nearest Neighbor* (KNN), o qual determina a quantidade de amostras que serão consideradas para realizar a classificação de um novo dado de entrada. É necessário utilizar técnicas bem estabelecidas na literatura para definir esse tipo de parâmetro, pois ele tem um impacto direto nos resultados do modelo.

Em muitos casos a escolha dos hiperparâmetros é feita de forma manual, seja baseada na própria experiência de quem está utilizando a técnica ou simplesmente por tentativa e erro. Em ambas as formas, esse tipo de abordagem é falha, pois os hiperparâmetros ideais podem variar de acordo com os dados utilizados, então mesmo que haja expertise em aprendizagem de máquina, não há como saber com certeza quais valores escolher (KOCH *et al.*, 2018).

Nesse cenário, diversos trabalhos já foram realizados no intuito de solucionar esse problema. Akiba *et al.* (2019) e Koch *et al.* (2018) são exemplos de frameworks estado da arte para a otimização de hiperparâmetros. Em Bergstra e Bengio (2012) é dada ênfase a técnica clássica *random search*, além disso são feitas algumas definições teóricas sobre o que é a otimização de hiperparâmetros. Em suma, a otimização de hiperparâmetros não tem uma dependência direta do modelo de aprendizado de máquina que se está tentando otimizar. Liashchynskiy e Liashchynskiy (2019) por exemplo, aplicam técnicas de otimização de hiperparâmetros ao problema de arquitetura de redes neurais, já em Schratz *et al.* (2019) a técnica *Sequential Model-Based Optimization* (SMBO) é utilizada para otimizar os hiperparâmetros de diversos algoritmos bem conhecidos na literatura, tais como: KNN, *Support Vector Machines* (SVM) e *Boosted Regression Trees* (BRT).

No presente trabalho, o problema de geração de mapas de *hotspots* foi escolhido como caso de uso. *Hotspots* são delimitações geográficas que indicam a grande incidência de eventos em determinado local durante um período de tempo. Em criminologia, seu uso é comum para prever áreas de risco com altos índices criminais, baseado em dados históricos, gerando análises que auxiliam no planejamento, para que forças policiais e órgãos de segurança possam atuar de forma a prevenir ou diminuir a ocorrência dos delitos. Existem diversos algoritmos para a geração de hotspots. Em Chainey *et al.* (2008) alguns deles são demonstrados e uma comparação é feita entre as técnicas, tendo como destaque o *Kernel Density Estimation* (KDE), que obteve os melhores resultados nos experimentos.

As contribuições deste trabalho são direcionadas às pessoas que necessitam definir hiperparâmetros para criar um modelo de geração de *hotspots* utilizando o KDE. Visto que normalmente esse processo de configuração do modelo é feito manualmente e que não há trabalhos que abordem especificamente esse problema, espera-se fornecer insumos para auxiliar no processo de otimização de hiperparâmetros. Dessa forma, desenvolvedores e pesquisadores terão uma base sobre como os hiperparâmetros influenciam no resultado do modelo, além de qual técnica de otimização consegue obter melhores resultados aplicada ao KDE.

1.1 Objetivos

1.1.1 *Objetivo Geral*

O objetivo deste trabalho é realizar uma análise do impacto dos hiperparâmetros nos resultados do KDE para a geração de mapas de *hotspots*.

1.1.2 *Objetivos Específicos*

- Utilizar de diferentes tamanhos de *bandwidths* e funções de kernel
- Avaliar intervalos temporais distintos no período de treino
- Utilizar diferentes métricas na avaliação
- Adicionar um kernel de tempo ao KDE

1.1.3 *Estrutura do trabalho*

O trabalho segue a seguinte estrutura: no Capítulo 2, é apresentada a fundamentação teórica, na qual são explicados conceitos importantes que servem de base para a realização do trabalho; no Capítulo 3, são apresentados os trabalhos relacionados, trazendo uma descrição da temática central de cada um deles, os resultados obtidos e fazendo uma comparação com o presente trabalho; no Capítulo 4, são descritos os procedimentos metodológicos, detalhando os passos relacionados a utilização e avaliação das técnicas de seleção de hiperparâmetros; no Capítulo 5, são apresentados os resultados obtidos com a execução dos procedimentos metodológicos; no Capítulo 6, é apresentada a conclusão do trabalho, abordando uma visão geral sobre os resultados e citando sugestões de melhorias e possibilidades de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados conceitos que são base para o desenvolvimento deste trabalho. Na Seção 2.1 é apresentado o conceito de otimização de hiperparâmetros, descrevendo o problema, explicando a terminologia e sua utilidade para algoritmos de aprendizado de máquina. Na Seção 2.2 alguns conceitos sobre espaço de busca e a sua relevância no processo de otimização são detalhados. Na Seção 2.3 algumas das técnicas para a otimização de hiperparâmetros são apresentadas. Na Seção 2.4 e 2.5 são apresentados o conceito de *Hotspot Mapping* e KDE, respectivamente. E por fim, na Seção 2.6, são apresentadas as métricas para avaliar os *hotspots*.

2.1 Otimização de hiperparâmetros

Os parâmetros do modelo têm seus valores definidos durante o processo de treinamento com base nos dados que são fornecidos ao modelo, não podendo ser definidos manualmente. Em contraste, os hiperparâmetros do modelo não aprendem seus valores a partir dos dados. Portanto, é preciso que sejam configurados manualmente. Os hiperparâmetros têm a capacidade de alterar significativamente o resultado do modelo, por isso a importância de serem escolhidos adequadamente.

É muito comum que, mesmo os modelos simples, tenham dois ou mais hiperparâmetros. Portanto, deve-se considerar todos eles em conjunto para encontrar os valores ideais para cada um. O processo de encontrar a combinação ideal de hiperparâmetros do modelo é chamado de otimização de hiperparâmetros. Esse processo muitas vezes é feito de forma manual, por tentativa e erro, ou baseado em experiências anteriores (KOCH *et al.*, 2018).

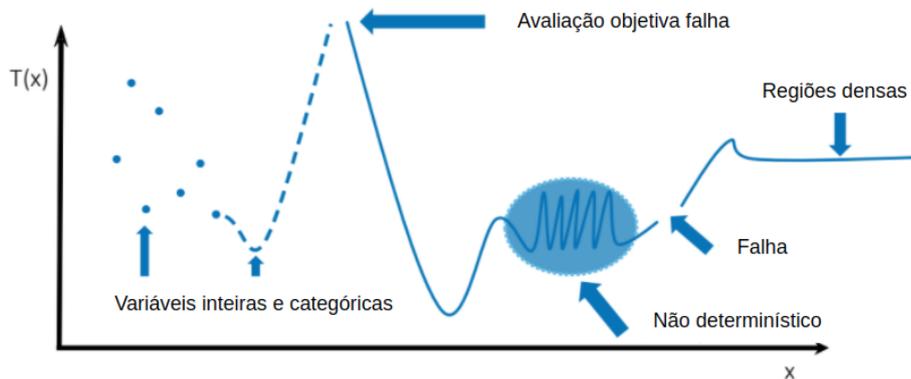
A abordagem manual é falha, pois os hiperparâmetros ideais podem variar de acordo com os dados utilizados. Além disso, para modelos com muitos hiperparâmetros acaba sendo muito complexo testar todos de forma manual para obter bons resultados. Devido a esse problema várias técnicas surgiram para automatizar essa tarefa.

2.2 Espaço de busca

O espaço de busca é o que define quais possíveis valores os hiperparâmetros de um modelo podem assumir. Se houver 3 hiperparâmetros diferentes, o espaço de busca será tridimensional. O espaço de busca pode ser altamente dimensional conforme o número de

hiperparâmetros aumenta. Podem existir diversas regiões dentro do espaço de busca, a Figura 1 ilustra esses cenários.

Figura 1 – Regiões dentro do espaço de busca



Fonte: adaptado de Koch *et al.* (2018)

Conforme mostrado em Bergstra e Bengio (2012), o espaço de busca tem um impacto significativo na técnica de otimização de hiperparâmetros. Uma alta dimensionalidade aumenta a complexidade da busca, além de aumentar os recursos computacionais necessários.

Bergstra e Bengio (2012) demonstraram que nem todos os hiperparâmetros no espaço de busca têm a mesma importância. Por esse motivo, algumas técnicas utilizam apenas um subconjunto do espaço de busca e conseguem obter bons resultados utilizando menos recursos computacionais, por não ter a necessidade de analisar completamente o espaço de busca.

2.3 Técnicas de otimização de hiperparâmetros

Atualmente existem diversas técnicas e frameworks para esse problema. A seguir serão detalhadas as duas técnicas mais amplamente utilizadas nessa área: *grid search* e *random search*.

2.3.1 Grid Search

Grid search é a técnica mais popular para o problema de otimização de hiperparâmetros (LIASHCHYNSKYI; LIASHCHYNSKYI, 2019). Muito utilizada em conjunto com a seleção manual, tem uma fácil implementação e um conceito simples de ser interpretado.

Basicamente, essa técnica recebe como entrada um conjunto de combinações dos hiperparâmetros para o modelo e itera sobre ele, a cada iteração um elemento desse conjunto é

escolhido para dar origem a um novo modelo. Os dados de entrada ficam totalmente a critério do usuário e não são alterados durante o processo de geração dos modelos. Nessa técnica, o critério de parada é somente quando todas as possibilidades do espaço de busca são testadas.

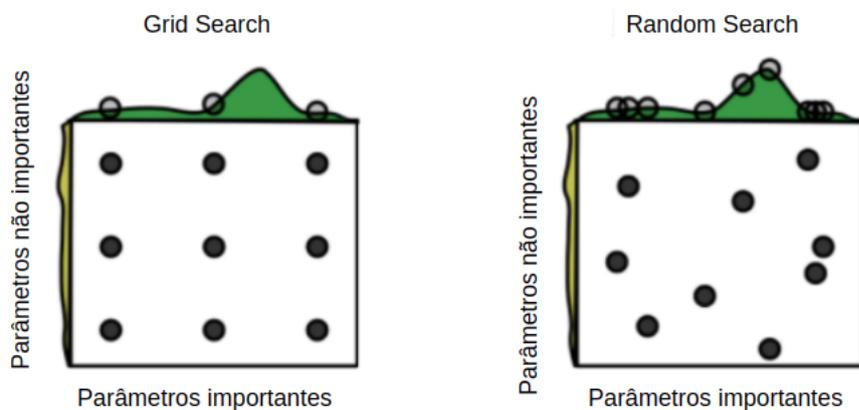
Iterar sobre todas as combinações de hiperparâmetros diferentes definidas pelo usuário pode ser demorado e computacionalmente custoso, especialmente se o espaço de busca for grande. Sendo assim, essa técnica é mais indicada para ser utilizada em modelos com uma baixa dimensionalidade, mas também é possível aplicar esse método em modelos com muitos hiperparâmetros, devido sua implementação ser bastante simples e os valores de hiperparâmetros serem independentes, dessa forma, paralelizar a solução para obter uma melhor performance torna-se trivial (BERGSTRA; BENGIO, 2012).

2.3.2 *Random Search*

A técnica *random search* se difere do *grid search* principalmente por não percorrer necessariamente todo o espaço de busca. Essa técnica recebe como entrada uma distribuição de parâmetros, ou uma faixa de valores onde estão os possíveis hiperparâmetros. Os modelos são gerados escolhendo de forma randômica uma combinação desses valores. Para essa técnica, o critério de parada é um número de iterações definido pelo próprio usuário.

A Figura 2 ilustra a distribuição dos hiperparâmetros nas duas técnicas. É possível observar que o *random search* explora mais regiões do espaço de busca se comparado com o *grid search*.

Figura 2 – Seleção de parâmetros com *grid search* e *random search*



Fonte: adaptado de Bergstra e Bengio (2012)

2.4 *Hotspot Mapping*

Em criminologia, *hotspot* é uma localização geográfica que contém uma alta quantidade de crimes em comparação com a distribuição do crime na área de estudo (DRAWVE, 2016).

Como dito por Chainey *et al.* (2008), *hotspot mapping* é uma forma popular e bastante utilizada para prever áreas de risco para ocorrência de crimes, servindo de auxílio para as forças policiais decidirem onde alocar seus recursos de forma mais eficiente.

Nos estudos em criminologia, existem diversas técnicas que buscam encontrar padrões nas ocorrências de crimes baseados na sua localização geoespacial. De acordo com Chainey *et al.* (2008) a ocorrência de crimes não é um fator exclusivamente aleatório. Como dito por Rosser *et al.* (2016), crimes tendem a se concentrar em uma escala espacial, como bairros, ruas ou pontos específicos. Por isso, técnicas que levam em consideração a localização espacial são amplamente utilizadas para esse fim.

2.5 *Kernel Density Estimation*

O KDE é uma técnica para estimativa da função de densidade de probabilidade, que permite analisar melhor a distribuição de probabilidade do que quando usando um histograma tradicional. Dois conceitos desempenham papel fundamental no KDE: a função kernel e o *bandwidth*, que é o coeficiente de suavidade da função (WEĞLARCZYK, 2018).

Nos estudos de criminologia, o KDE é uma das mais populares técnicas para a geração de mapas de *hotspots* criminais. Essa técnica tem sido bastante trabalhada por diversos autores ao longo dos anos e mostrando cada vez mais bons resultados (ROSSER *et al.*, 2016).

O KDE com grade de células envolve a criação de células quadradas sobre a área de estudo, formando uma espécie de malha quadriculada. Para cada local de crime dentro da área de estudo, uma interpolação, usando uma função de densidade de kernel, é feita, ponderando cada célula da malha dentro de um *bandwidth* proporcional à distância do centro da célula para a localização do evento criminoso.

Na literatura existem diversas adaptações dessa técnica, como exemplificado em Rosser *et al.* (2016) e Nunes *et al.* (2019). Nos quais, foram propostas variações no KDE para abordar o problema de geração de *hotspots* tendo a rede de ruas como objeto de estudo, ao invés da tradicional grade de células.

2.6 Métricas de avaliação

Devido aos diferentes tipos de técnicas de geração de *hotspots* e aos diferentes formatos de saída de cada uma das técnicas, existe uma dificuldade em avaliar os resultados (DRAWVE, 2016). Ao longo do tempo diversas métricas foram criadas, desde as mais simples, como uma análise visual dos *hotspots* plotados em gráfico, *Hit Rate* (HR), até métricas mais elaboradas como *Predictive Accuracy Index* (PAI) e *Recapture Rate Index* (RRI). A seguir serão detalhadas as três métricas mais utilizadas na literatura para a avaliação de *hotspots*.

2.6.1 *Hit Rate - HR*

A métrica mais simples e popular é a HR (DRAWVE, 2016). Essa métrica é definida como a fração entre a quantidade de crimes registrados dentro dos *hotspots* e o total de crimes. Seja n a quantidade de crimes registrados dentro dos *hotspots* e N a quantidade total de crimes, a fórmula para o cálculo do HR é definida de acordo com a equação (2.1).

$$HR = \frac{n}{N} \quad (2.1)$$

Quanto maior for o HR, maiores as chances de que ao ocorrer novos eventos, esses estejam dentro dos *hotspots*. O principal problema em utilizar essa métrica é devido ao fato de que nesse cálculo a área dos *hotspots* em relação a área total do local de estudo não é considerada. Dessa forma, um conjunto de *hotspots* que ocupa uma maior área tem uma tendência de obter um melhor resultado, visto que há mais chances de que mais eventos ocorram dentro dessa área maior.

2.6.2 *Predictive Accuracy Index - PAI*

Em Chainey *et al.* (2008) é definida uma métrica que leva em consideração a área dos *hotspots* para obter um resultado mais consistente. O valor de PAI é obtido pela divisão entre a taxa de acerto e a fração da área coberta pelos *hotspots* em relação à área total. Seja n a quantidade de crimes registrados dentro dos *hotspots*, N a quantidade total de crimes, a sendo a área ocupada pelos *hotspots* e A a área total do local de estudo, a fórmula é definida de acordo com a equação (2.2).

$$PAI = \frac{n/N}{a/A} \quad (2.2)$$

Um valor de PAI mais alto apresenta uma proporção maior de eventos por unidade de área. Ou seja, uma área menor ocupada por *hotspots* que consegue conter uma grande quantidade de eventos.

2.6.3 *Recapture Rate Index - RRI*

Levine (2008) propôs um outro método para avaliar *hotspots* denominado RRI. Baseado no trabalho de Chainey *et al.* (2008), a métrica RRI foi definida como a proporção da densidade de crimes de um período de tempo em relação a outro. Por exemplo, se o número total de crimes aumenta em 5% do ano x para o ano $x + 1$, a densidade de crimes também deve aumentar em 5% em relação ao ano anterior. Assim, o índice seria 1.0. Um valor inferior a 1.0 indica uma perda de eventos de um ano para o próximo, enquanto um valor superior a 1.0 indica aumento da concentração de um ano para o próximo (LEVINE, 2008). Seja n a quantidade de crimes dentro de *hotspots*, $n_t - 1$ essa quantidade no ano anterior e T o total de crimes de um ano e $T_t - 1$ a quantidade total de crimes no ano anterior, o RRI é definido de acordo com a equação (2.3).

$$RRI = \frac{n_t}{n_t - 1} * \frac{T_t}{T_t - 1} \quad (2.3)$$

Geralmente, é esperado que o RRI seja inferior a 1.0, uma vez que alguns crimes que ocorrem dentro do *hotspot* no ano base não ocorreriam no próximo ano (LEVINE, 2008).

3 TRABALHOS RELACIONADOS

Os trabalhos descritos a seguir abordam temas que se enquadram ao contexto do trabalho proposto. No primeiro momento apresenta-se os trabalhos que tratam de técnicas para a otimização de hiperparâmetros e comparação de seus resultados. Em seguida são apresentados os trabalhos na área de geração de *hotspots*, mais especificamente sobre o KDE. Por fim, é feita uma comparação entre os trabalhos relacionados e o presente trabalho.

3.1 *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS*

Em Liashchynskyi e Liashchynskyi (2019) foi feito um comparativo entre três técnicas para a otimização de hiperparâmetros: *Grid Search*, *Random Search* e algoritmos genéticos, aplicando essas técnicas ao problema de *Neural Architecture Search* (NAS).

Como o problema tratado envolve a arquitetura de redes neurais, a abordagem para a produção de um modelo foi feita definindo uma arquitetura base, e em seguida, as técnicas de otimização geraram outros modelos adicionando novas camadas na arquitetura base. O hiperparâmetro a ser otimizado, neste caso, é a quantidade de camadas convolucionais na rede.

Ao final, a comparação é feita baseada no tempo em que a técnica de otimização levou até encontrar o melhor modelo e também na acurácia do modelo encontrado. Nesse estudo, com o método *grid search* foi obtida uma acurácia de 83%, já com o *random search* e o algoritmo genético esse valor chegou a 86%. Em relação ao tempo de execução, o *random search* se mostrou o mais rápido, levando aproximadamente 2.7 horas para executar, contra 4.13 horas do algoritmo genético e 4.3 horas do *grid search*.

3.2 *Random Search for Hyper-Parameter Optimization*

Bergstra e Bengio (2012) mostraram de forma empírica e teórica que escolher os hiperparâmetros de forma randômica é mais eficiente do que fazer uma busca em grade, ou seja, que a técnica *Random Search* supera o *Grid Search*.

De acordo com o trabalho, a evidência empírica dessa afirmação se dá baseada em uma comparação feita em um estudo anterior que usou *Grid Search* e *Random Search* para encontrar os hiperparâmetros de uma rede neural. Os resultados obtidos apontaram que a busca aleatória encontrou na maioria das vezes modelos tão bons ou melhores em menos tempo. Tendo um espaço de busca de 32 dimensões, ou seja, 32 hiperparâmetros para otimizar, foi

obtido um desempenho estatisticamente igual em quatro dos sete conjuntos de dados testados e desempenho superior em um dos sete.

Foi feita uma análise e observado que existe pouca influência de alguns hiperparâmetros no resultado final do modelo. Isso poderia explicar a baixa performance do *Grid Search*, já que essa técnica faz uma busca em todos os hiperparâmetros definidos, e por isso se torna uma escolha ruim, principalmente para grandes espaços de busca.

3.3 *Hyperparameter Tuning and Performance Assessment of Statistical and Machine-Learning Algorithms Using Spatial Data*

O trabalho de Schratz *et al.* (2019) tem como objetivo mostrar os efeitos que a autocorrelação espacial tem na otimização de hiperparâmetros. Para fazer isso, vários algoritmos de aprendizado de máquina foram usados na comparação, como BRT, KNN, *Random Forest* (RF), SVM e regressão logística.

Para o ajuste dos hiperparâmetros, a técnica utilizada foi a *Sequential Model-Based Optimization* (SMBO). Nessa técnica o usuário escolhe um espaço de busca e os hiperparâmetros são selecionados inicialmente de forma aleatória, a cada iteração os resultados obtidos na iteração anterior são levados em consideração para definir um próximo conjunto de hiperparâmetros que será usado no modelo a ser avaliado. Neste caso, foram escolhidos inicialmente um conjunto de 30 hiperparâmetros e um critério de parada de 70 iterações.

Ao final dos experimentos, constatou-se que a diferença de performance usando a validação cruzada espacial e a não-espacial foi pouca. Entretanto é recomendado que seja feita a validação cruzada espacial, pois isso tende a diminuir o bias.

Quadro 1 – Comparação com os trabalhos relacionados.

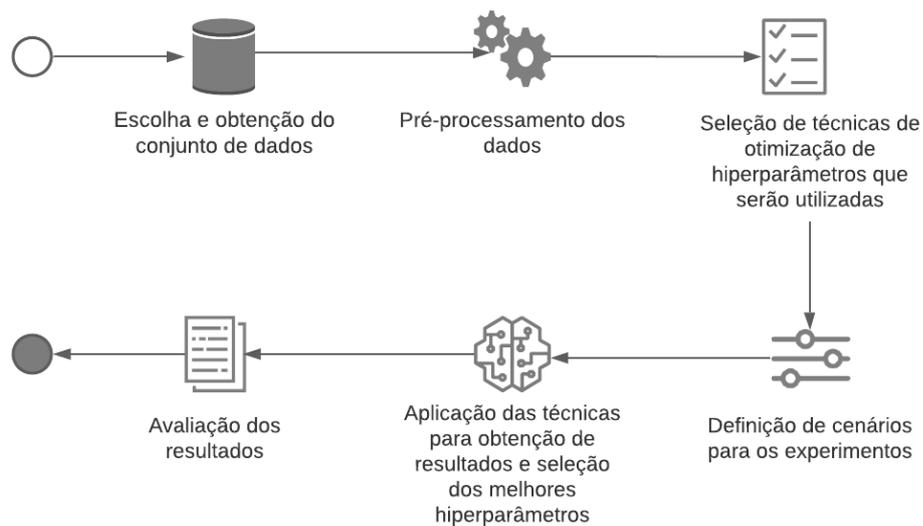
	Liashchynskiy e Liashchynskiy (2019)	Bergstra e Bengio (2012)	Schratz <i>et al.</i> (2019)	Este trabalho
Domínio	Arquitetura de redes neurais	Classificação de imagens	Predição de doenças florestais (<i>Diplodia sapinea</i>)	Geração de mapas de hotspots de crimes
Técnicas de aprendizado de máquina	Rede neural	Rede neural	BRT, KNN, RF, SVM e regressão logística	KDE
Técnicas de otimização de hiperparâmetros	Algoritmos genéticos, <i>Grid Search</i> e <i>Random Search</i>	<i>Grid Search</i> e <i>Random Search</i>	<i>SMBO</i>	<i>Grid Search</i> e <i>Random Search</i>
Métricas de avaliação	Acurácia	Acurácia	<i>Brier score</i>	HR e PAI
Conjuntos de dados	CIFAR-10	Variações do <i>mnist dataset</i> , <i>rectangles images</i> e <i>convex dataset</i>	Subconjunto dos dados ambientais utilizados em Iturritxa <i>et al.</i> (2015)	<i>Crimes in San Francisco</i> ,

Fonte: elaborado pelo autor

4 METODOLOGIA

Para analisar os hiperparâmetros foi proposta a seguinte metodologia representada pela Figura 3. Considerando os passos que foram propostos, espera-se poder observar como diferentes valores de hiperparâmetros impactam nos resultados do KDE, além de observar também o efeito que diferentes cenários de experimentos provocam na avaliação do modelo.

Figura 3 – Fluxo de atividades



Fonte: elaborado pelo autor

4.1 Escolha e obtenção do conjunto de dados

Para o treino e avaliação do KDE é necessário obter um conjunto de dados com uma quantidade significativa de crimes, visto que, geralmente, um número maior de dados resulta na melhora da performance do modelo. Como dito em Koch *et al.* (2018), os hiperparâmetros ideais podem variar de acordo com os dados.

É preciso que os dados apresentem a localização com as coordenadas de latitude e longitude, para que seja possível aplicar o KDE. Além disso, é importante que tenha a data do ocorrido, possibilitando verificar se é um evento recente ou antigo, e também o tipo do delito, para que seja possível especificar exatamente quais dados serão usados, pois alguns tipos de crime não tem uma correlação direta com o local em que foram registrados.

4.2 Pré-processamento dos dados

Tendo definido o conjunto de dados, nessa etapa os dados serão processados antes de seguir para o próximo passo. Esse processamento inclui a remoção de dados que estão fora da área de estudo, a filtragem para utilizar um tipo de crime específico, e se necessário, realizar a conversão das coordenadas dos eventos para o sistema métrico, para facilitar os cálculos de distância que serão feitos posteriormente.

4.3 Seleção de técnicas de otimização de hiperparâmetros

Nesta etapa são definidas quais técnicas serão utilizadas para o problema de otimização de hiperparâmetros. Atualmente existem diversos frameworks de AutoML, que são ferramentas que automatizam o processo de criação de um modelo de aprendizado de máquina. Alguns deles são feitos especificamente para a otimização de hiperparâmetros, tais como AutoTune (KOCH *et al.*, 2018) e Optuna (AKIBA *et al.*, 2019), que são exemplos de frameworks que disponibilizam soluções do estado da arte para esse problema. Essas ferramentas utilizam de diversas técnicas, desde as mais avançadas até as mais simples, combinando-as a fim de obter um melhor resultado. Neste trabalho, o foco dos experimentos será utilizar as duas técnicas clássicas mais utilizadas e que servem como base para outras mais complexas: *Grid Search* e *Random Search*.

4.4 Definição de cenários para realização dos experimentos

Os cenários dos experimentos são propostos de modo a explorar uma maior possibilidade de hiperparâmetros. Usualmente, para geração de mapas *hotspots* criminais, os dados utilizados para treino e testes são divididos com base em um período de tempo, de modo que os dados de treino precedem os dados de teste. Os cenários serão feitos variando esses períodos de tempo, além disso os experimentos também serão conduzidos com os dados estratificados e não estratificados. O objetivo principal de criar esses cenários é observar como as técnicas de otimização de hiperparâmetros variam seus resultados em diferentes contextos.

4.5 Aplicação das técnicas para obtenção de resultados e seleção dos melhores hiperparâmetros

Para aplicar as técnicas, primeiramente são definidos os espaços de busca, de acordo com os hiperparâmetros do algoritmo KDE, que são: *bandwidth* e a função de kernel. Para cada novo modelo gerado é feito o processo de treino utilizando um subconjunto dos dados disponíveis. Depois de treinados, os modelos podem ser avaliados de acordo com o processo descrito na Seção 4.6.

4.6 Avaliação dos resultados

Após aplicar as técnicas de otimização de hiperparâmetros e executar os modelos gerados por cada técnica, os resultados são comparados através das métricas de avaliação HR e PAI, previamente definidas na Seção 2.6 do Capítulo 2. A métrica RRI não foi utilizada, pois de acordo como a métrica funciona, seria necessário realizar a avaliação usando dados de dois anos distintos, o que seria muito diferente de como é feito nas outras duas métricas.

O objetivo dessa avaliação é verificar qual modelo obteve o melhor resultado, e por consequência, qual das técnicas de otimização teve uma melhor performance. No capítulo seguinte são apresentados os resultados obtidos com a execução dos procedimentos metodológicos.

5 EXPERIMENTOS E RESULTADOS

Este capítulo apresenta os resultados obtidos na execução das etapas descritas nos procedimentos metodológicos deste trabalho.

5.1 Conjunto de dados

Nos experimentos foi utilizado o conjunto de dados *Police Department Incident Reports: Historical 2003 to May 2018 - San Francisco*, que contém registros de crimes disponibilizado de forma pública pelo departamento de polícia de São Francisco¹, os dados são de ocorrências registradas durante o período de 2003 até 2018, somando um total de 2,13 milhões de registros distribuídos em 37 categorias diferentes de crimes. Entre os atributos do conjunto de dados estão as coordenadas de longitude e latitude onde o delito foi registrado, a data e horário do ocorrido, e também a categoria, que armazena a informação do tipo de crime cometido.

5.1.1 Pré-processamento

O conjunto de dados contém uma série histórica de eventos no período de janeiro de 2003 até maio de 2018. Devido a grande quantidade de registros nesse período, por questões de limitação de processamento computacional, foram escolhidos dados de um ano para seguir com os experimentos. Considerando que o ano de 2018 não apresenta os dados em todos os meses, o ano de 2017 foi escolhido por ser o mais recente que apresenta dados nos 12 meses.

Dentre as vários tipos de crimes presentes nos dados, apenas os delitos do tipo *Robbery* foram considerados, sendo esse um dos tipos de crime usado também em Chainey *et al.* (2008). Além disso, foi necessário converter as coordenadas de latitude e longitude que estavam em graus para metros, para facilitar o cálculo da distância entre a localização dos eventos e as células do *grid*. Essa conversão e as demais operações geoespaciais, como visualização, filtragem espacial e a própria criação da grade de células foram feitas usando a biblioteca GeoPandas².

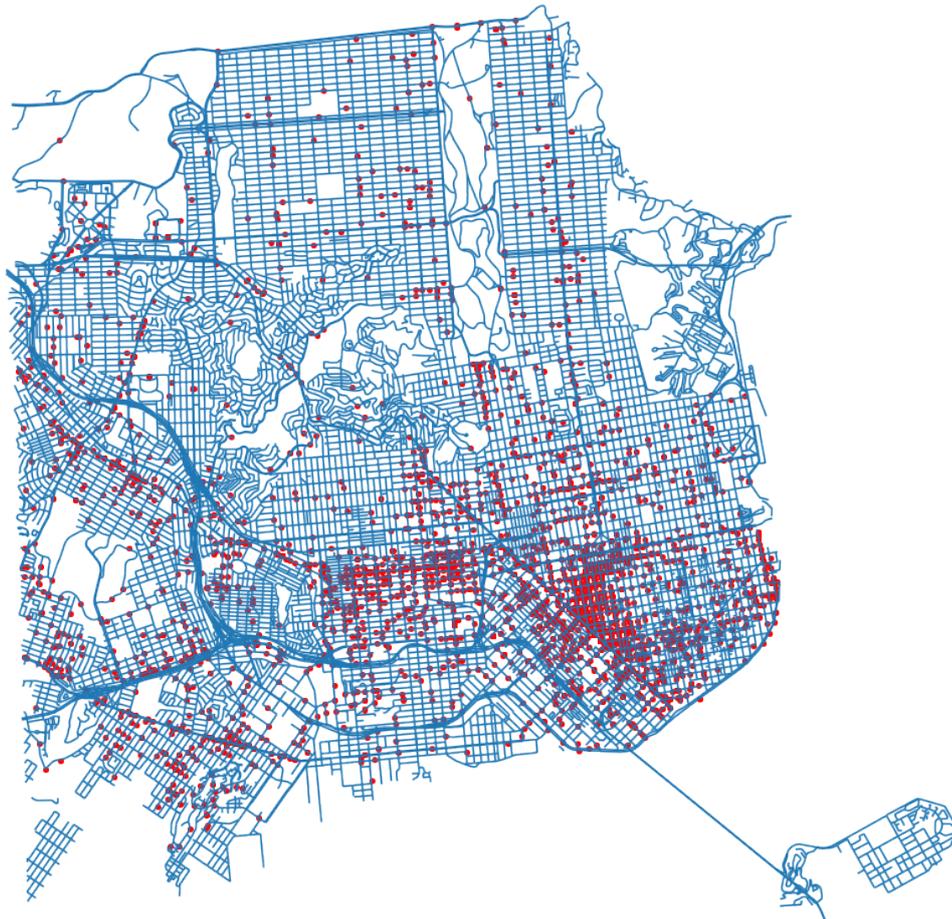
Após o processamento inicial, para validar a integridade dos dados foi realizada uma filtragem espacial com o objetivo de remover dados com ruído, ou seja, remover eventos cuja localização não estava dentro da área de estudo, o que poderia prejudicar o resultado final. Após as filtrações, a quantidade final de eventos foi de 3256. A Figura 4 mostra a localização dos

¹ <https://data.sfgov.org/>

² <https://geopandas.org/>

eventos na rede de ruas da cidade. Os dados da rede de ruas foram obtidos a partir da plataforma aberta de dados geospaciais Koodirnates³.

Figura 4 – Visualização dos eventos na cidade de São Francisco.



Fonte: elaborado pelo autor

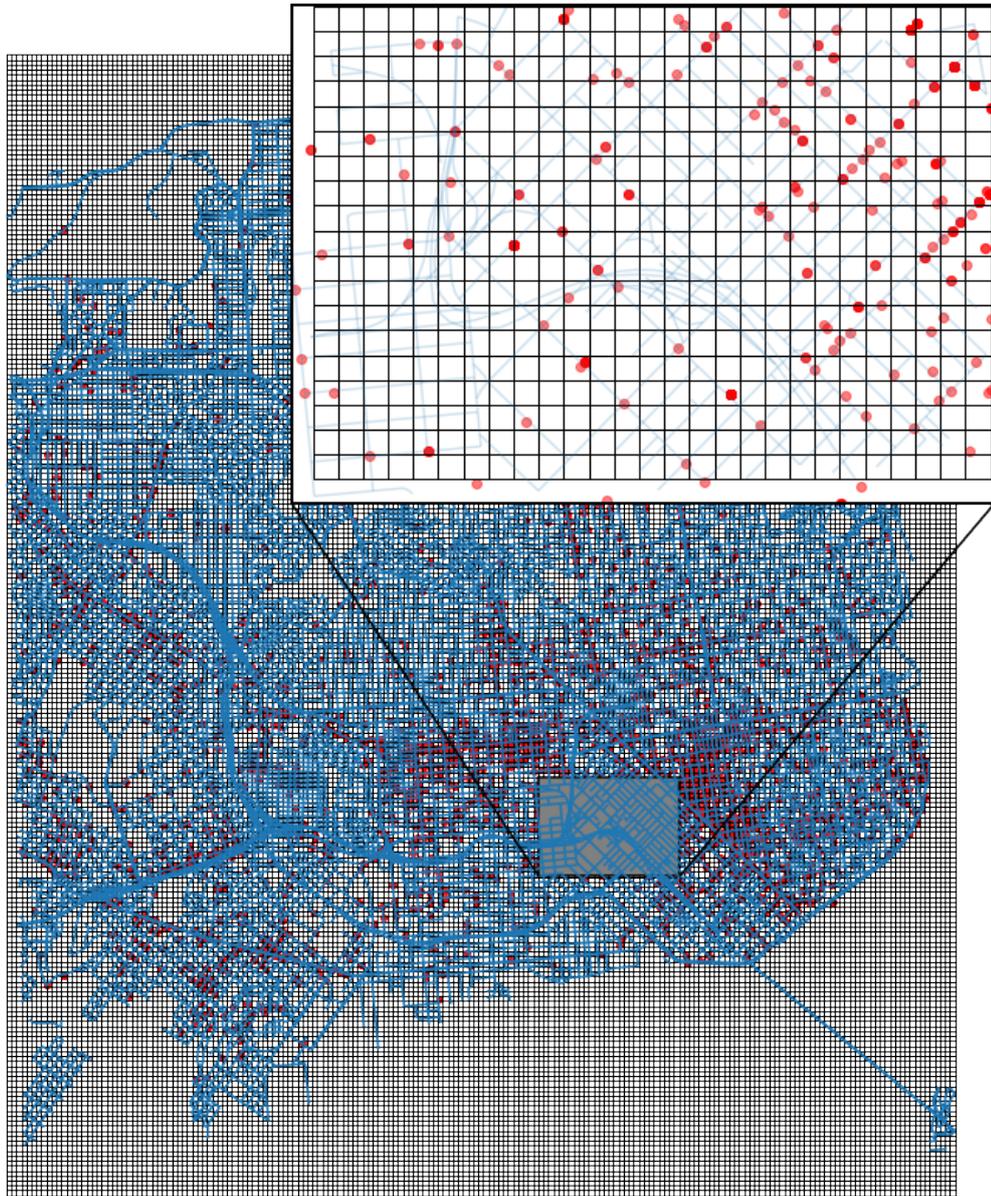
5.2 Geração dos mapas de *hotspots*

Para a geração dos mapas de *hotspots* foi criada uma grade de células sobre a área de estudo. Chainey *et al.* (2008) cita uma forma de estimar o tamanho da célula, que consiste em dividir o comprimento do menor lado da fronteira da área de estudo por 150. Neste trabalho, foi usado 100m como tamanho da célula, próximo aos 125m que seria o valor encontrado pelo método citado anteriormente. Ao aplicar o KDE, cada uma dessas células receberá um score que representa a densidade de eventos na célula. Para definir quais dessas células serão *hotspots*, é

³ <https://koordinates.com/>

escolhido a quantidade correspondente a aproximadamente 3% da área total de estudo com as maiores densidades. A Figura 5 mostra uma visualização da grade de células sobre a cidade.

Figura 5 – Visualização da grade de células de $100m^2$ sobre a cidade de São Francisco.



Fonte: elaborado pelo autor

5.2.1 Conjuntos de treino e teste

Para realizar os experimentos, de acordo com os testes elaborados no trabalho Chainey *et al.* (2008) foram escolhidos eventos por 12 meses a partir de 01/01/2017. Neste trabalho, como há também a intenção de observar o impacto em considerar eventos mais antigos

no processo de treino, os conjuntos de treino foram divididos usando três faixas de valores, sendo elas com 90 dias, 180 dias e 365 dias. Os conjuntos de testes foram fixos em 60 dias, iniciando a partir do dia subsequente ao último dia do período de treino. A Figura 6 mostra essa divisão dos conjuntos.

Figura 6 – Divisões dos conjuntos de dados para treino e teste.



Fonte: elaborado pelo autor

5.2.2 Execução e Avaliação

5.2.2.1 Grid Search

Com relação aos hiperparâmetros, para definir os valores de *bandwidth* foi usado como base o tamanho da célula, de forma que se fossem escolhidos valores muito pequenos, menores do que os 100m da própria célula, isso faria com que a influência de um evento não se propagasse entre as células vizinhas, o que seria análogo a realizar uma contagem dos eventos em cada célula para selecionar os *hotspots*. Por outro lado, se fossem valores muito altos, como 10 ou 15 vezes o tamanho da célula, a influência de um evento estaria chegando muito longe da sua real localização, o que também poderia ser um problema. O quadro 2 apresenta os valores usados no KDE com a técnica de otimização *grid search*

Para montar o conjunto de combinações possíveis é feito o produto cartesiano entre os valores de *bandwidth* e as funções de kernel. Dessa forma, o espaço de busca tem 9 possibilidades de hiperparâmetros para cada conjunto de dados definidos na Seção 5.2.1.

Quadro 2 – Hiperparâmetros utilizados no KDE com grade de células.

<i>Bandwidth</i>	<i>Função de Kernel</i>
400m	Quártico
600m	TopHat
800m	Gaussiano

Fonte: elaborado pelo autor

No *grid search*, é selecionada uma combinação de hiperparâmetros e posteriormente é feito o treinamento do KDE. O primeiro experimento é realizado considerando os eventos que ocorreram no intervalo de 90 dias, depois de 160 dias e por último de 365 dias. Após o treino, é feita a avaliação com a métrica HR usando os eventos dos 60 dias subsequentes. As tabelas 1 e 2 apresentam os resultados obtidos.

Tabela 1 – Resultados do *grid search* aplicado ao KDE avaliado com a métrica HR.

<i>bandwidth</i>	90 dias de treino			180 dias de treino			365 dias de treino		
	quartico	tophat	gaussiano	quartico	tophat	gaussiano	quartico	tophat	gaussiano
400m	38,38%	35,86%	36,40%	38,92%	36,31%	36,50%	38,06%	35,53%	35,15%
600m	37,12%	33,33%	34,05%	36,87%	32,77%	34,64%	35,92%	35,15%	35,73%
800m	35,86%	32,61%	32,79%	35,57%	32,22%	31,66%	36,50%	33,59%	33,59%

Fonte: elaborado pelo autor

Tabela 2 – Resultados do *grid search* aplicado ao KDE avaliado com a métrica PAI.

<i>bandwidth</i>	90 dias de treino			180 dias de treino			365 dias de treino		
	quartico	tophat	gaussiano	quartico	tophat	gaussiano	quartico	tophat	gaussiano
400m	17,41	16,27	16,51	17,66	16,47	16,56	17,27	16,12	15,94
600m	16,84	15,12	15,45	16,73	14,87	15,71	16,30	15,94	16,21
800m	16,27	14,80	14,88	16,14	14,62	14,36	16,56	15,24	15,24

Fonte: elaborado pelo autor

O valor destacado em negrito representa o melhor valor daquele conjunto de dados. O kernel quártico apresentou melhores resultados em relação às demais funções de kernel. Juntamente com o *bandwidth* de 400m e 180 dias de treino, foi obtido o melhor resultado no geral de 38,92% na métrica HR e 17,41 na métrica PAI. É importante destacar que os hiperparâmetros com os melhores resultados foram os mesmos para as duas métricas, isso acontece porque a área dos *hotspots* em ambos os casos foi a mesma, aproximadamente 3% da área total da cidade.

5.2.2.2 *Random Search*

Para o *random search*, a divisão do conjunto de dados segue o mesmo padrão definido na Figura 6. Nesse caso, os hiperparâmetros não são explicitamente definidos, ao invés disso foi

definido um intervalo de valores.

Para o bandwidth foi definido um espaço de busca entre 300m e 1000m, apenas com valores inteiros, e um número de 10 iterações para procurar entre os valores nesse espaço. Essa busca acontece selecionando um valor de forma aleatória, logo em seguida seguindo o fluxo de treino e avaliação, conforme ocorre no *grid search*. As funções de kernel não foram variadas, seguindo com o kernel quártico, tophat e gaussiano.

A quantidade de iterações para encontrar os melhores hiperparâmetros é igual ao produto das possibilidades no espaço de busca. Como foram definidas 10 iterações para selecionar o bandwidth, três funções de kernel e três conjuntos de treino e teste, isso resulta um total de 90 iterações. As tabelas 3 e 4 apresentam apenas os melhores resultados obtidos na avaliação. Apenas o kernel quartico é mostrado, pois foi o que obteve os melhores resultados.

Tabela 3 – Melhores parâmetros obtidos no *random search* aplicado ao KDE avaliado com a métrica HR.

	90 dias de treino	180 dias de treino	365 dias de treino
bandwidth	quartico	quartico	quartico
318m	39,63%	41,89%	39,02%

Fonte: elaborado pelo autor

Tabela 4 – Melhores parâmetros obtidos no *random search* aplicado ao KDE avaliado com a métrica PAI.

	90 dias de treino	180 dias de treino	365 dias de treino
bandwidth	quartico	quartico	quartico
318m	17,98	19,01	17,70

Fonte: elaborado pelo autor

5.2.2.3 KDE e Kernel no tempo

O KDE com grade de células usualmente considera apenas a dimensão espacial dos eventos, ou seja, a localização em relação às células do *grid*. Porém, como dito em Hu *et al.* (2018), o crime tende a se agrupar também temporalmente. É comum que alvos próximos a locais onde houveram ocorrência de crimes recentes venham a ser vítimas de novas ocorrências em um curto espaço de tempo.

Por esse motivo, surgiram técnicas que estendem o KDE para ser capaz de tratar além do local, a data em que o evento ocorreu, passando a utilizar essa informação de tempo também como um hiperparâmetro para a geração dos *hotspots*. Hu *et al.* (2018) propôs uma redefinição da

técnica denominada *Space Time Kernel Density* (STKDE), que combina o componente espacial com o temporal no KDE. A seguir, é apresentado um outro método para realizar essa combinação.

A Figura 7 apresenta a forma de aplicação de um kernel temporal ao KDE. A matriz A é o componente espacial de densidade, que é obtida calculando a distância de cada célula do *grid* para cada um dos eventos criminosos e aplicando uma função de kernel. A matriz B é o componente temporal, que é obtido calculando a densidade no tempo dos eventos.

A densidade no tempo é calculada obtendo a diferença em dias da data de ocorrência do evento para uma data base, que pode ser o dia atual ou uma outra data que esteja próxima ao fim do período de treino. Após calcular essa distância em dias, é aplicada uma função de kernel para obter uma densidade desses valores, de forma que eventos que ocorreram há mais tempo recebam um valor menor, e eventos mais recentes tenham um peso maior.

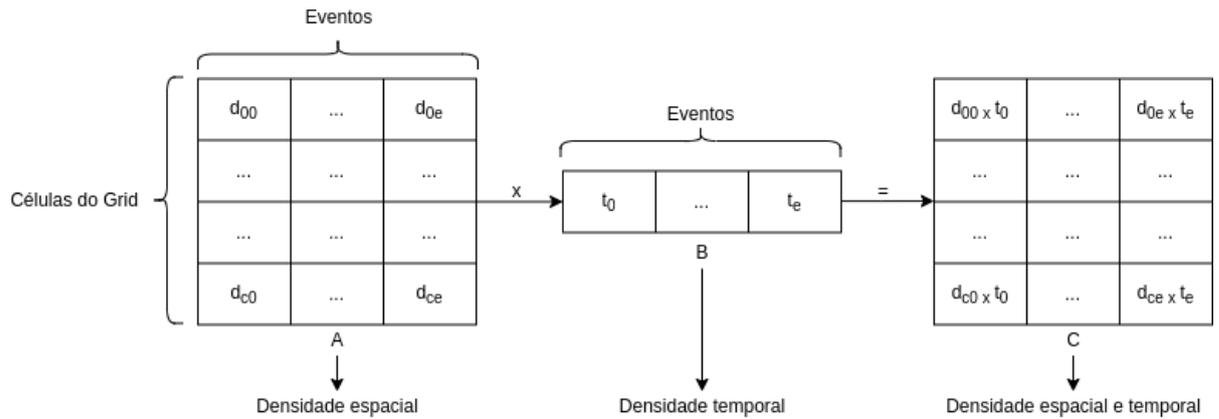
Tendo calculado os componentes de espaço e de tempo, cada uma das densidades no espaço é multiplicada por sua densidade no tempo, conforme a matriz C da Figura 7. E por fim, o somatório das densidades de cada evento é feito, resultando na densidade total da célula, assim como acontece usualmente no KDE, entretanto agora essas densidades receberam uma penalidade em relação ao tempo.

É importante observar que para aplicar essa técnica é necessário guardar a informação de densidade de cada um dos eventos para cada uma das células (matriz A). Considerando que a quantidade de células e a quantidade de eventos pode ser muito grande, manter essas informações em memória tem um alto custo computacional.

A intenção desse processo é fazer com que as células do *grid* contendo eventos mais antigos, mesmo que tenham uma densidade espacial mais alta, sejam penalizadas, enquanto as que contém eventos mais recentes tenham uma penalidade consideravelmente menor. Isso tende a fazer com que a localização dos ocorridos registrados há menos tempo tenham preferência no momento da escolha das células que serão os *hotspots*, e por consequência, espera-se obter resultado melhor na avaliação final do modelo.

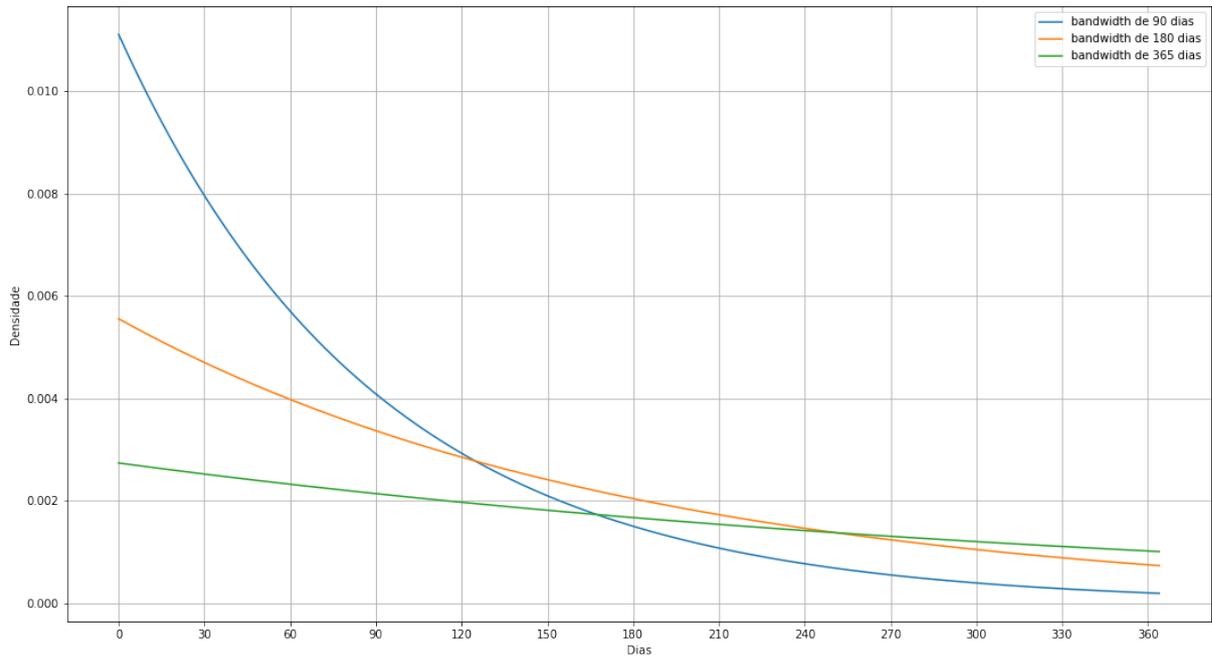
Nos experimentos foi utilizada uma função de decaimento exponencial de acordo com a fórmula (5.1), onde B é o *bandwidth* em dias, e T representa quantos dias se passaram da data que o evento ocorreu, até a data base. Para o kernel no tempo, também foram definidos três valores de *bandwidth*, sendo eles 90, 180 e 365 dias. A Figura 8 mostra o formato das curvas ao aplicar os *bandwidths* na função (5.1).

Figura 7 – Kernel no tempo aplicado ao KDE.



Fonte: elaborado pelo autor

$$D = \frac{1}{B} * e^{-\frac{T}{B}} \quad (5.1)$$

Figura 8 – Visualização da função de decaimento exponencial com diferentes *bandwidths*.

Fonte: elaborado pelo autor

É possível observar que para um valor de *bandwidth* menor (90 dias) a curva é mais acentuada. Isso indica que a influência do evento diminui mais rapidamente à medida em que sua distância no tempo aumenta em relação ao ponto zero. Enquanto para *bandwidths* maiores (180 e 365 dias), apesar de já iniciarem com valores mais baixos, a diminuição desse valor com o passar dos dias ocorre de forma mais lenta.

A Tabela 5 mostra os resultados ao aplicar o método *grid search* no KDE juntamente com o kernel no tempo na cidade de São Francisco. Os valores de *bandwidth* mostrados são os hiperparâmetros de espaço e tempo, respectivamente. Em todas as execuções a função de kernel no tempo se manteve a mesma (5.1), por isso não é mostrada na tabela. As funções de kernel para o espaço são as mesmas usadas no experimento anterior.

Tabela 5 – Resultados do *grid search* aplicado ao KDE e kernel no tempo avaliado com a métrica HR.

bandwidths	90 dias de treino			180 dias de treino			365 dias de treino		
	quartico	tophat	gaussiano	quartico	tophat	gaussiano	quartico	tophat	gaussiano
400m e 90d	40,36%	37,12%	33,51%	39,85%	35,38%	34,82%	39,22%	34,76%	35,34%
400m e 180d	40,00%	36,22%	33,69%	40,60%	35,94%	34,45%	38,64%	35,15%	36,12%
400m e 365d	39,46%	36,22%	33,51%	39,85%	35,75%	34,26%	38,06%	35,73%	36,12%
600m e 90d	37,48%	33,33%	32,79%	37,24%	33,15%	33,33%	37,48%	35,34%	33,20%
600m e 180d	37,84%	33,51%	31,89%	36,87%	32,59%	33,33%	35,92%	35,15%	32,62%
600m e 365d	38,20%	33,33%	31,53%	37,24%	32,77%	33,71%	36,31%	34,95%	32,43%
800m e 90d	36,04%	32,25%	29,37%	36,87%	32,03%	33,71%	36,31%	33,79%	31,26%
800m e 180d	36,04%	32,43%	29,37%	36,13%	32,22%	33,89%	36,12%	33,59%	31,26%
800m e 365d	35,08%	32,43%	29,37%	35,75%	32,03%	33,52%	36,31%	33,40%	31,07%

Fonte: elaborado pelo autor

Tabela 6 – Resultados do *grid search* aplicado ao KDE e kernel no tempo avaliado com a métrica PAI.

bandwidths	90 dias de treino			180 dias de treino			365 dias de treino		
	quartico	tophat	gaussiano	quartico	tophat	gaussiano	quartico	tophat	gaussiano
400m e 90d	18,31	16,84	15,20	18,08	16,05	15,80	17,79	15,77	16,03
400m e 180d	18,15	16,43	15,29	18,42	16,31	15,63	17,53	15,94	16,39
400m e 365d	17,90	16,43	15,20	18,08	16,22	15,55	17,27	16,21	16,39
600m e 90d	17,00	15,12	14,88	16,90	15,04	15,12	17,00	16,03	15,06
600m e 180d	17,17	15,20	14,47	16,73	14,78	15,12	16,30	15,94	14,80
600m e 365d	17,33	15,12	14,31	16,90	14,87	15,29	16,47	15,86	14,71
800m e 90d	16,35	14,63	13,32	16,73	14,53	15,29	16,47	15,33	14,18
800m e 180d	16,35	14,71	13,32	16,39	14,62	15,38	16,39	15,24	14,18
800m e 365d	16,19	14,71	13,32	16,22	14,53	15,21	16,47	15,15	14,09

Fonte: elaborado pelo autor

É possível notar uma melhora nos resultados obtidos. Agora, o melhor caso na métrica HR obteve um resultado de 40,6%, aproximadamente 4,31% superior ao experimento anterior que usava os mesmos hiperparâmetros no espaço, mas sem considerar o componente de tempo.

Como mencionado anteriormente, aplicar o kernel no tempo é muito custoso em termos de memória. Devido a limitações de poder computacional, esse método não foi utilizado junto com o *random search*.

6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho analisou diferentes valores de hiperparâmetros aplicados ao problema de geração de mapas de *hotspots* criminais usando KDE com grade de células. Para atingir esse objetivo foi definido que as técnicas de seleção de hiperparâmetros utilizadas seriam o *grid search* e *random search*, para explorar o espaço de busca de formas diferentes, abrangendo mais valores. Foram utilizadas as métricas HR e PAI, que permitiram avaliar os *hotspots* tanto pela quantidade de eventos contidos, quanto a proporção entre quantidade de eventos dentro *hotspot* e a área geográfica ocupada. Para realizar os experimentos foram obtidos dados de crimes a partir do portal online de dados abertos da cidade de São Francisco, o que possibilitou realizar os experimentos com dados reais.

A divisão do conjunto de dados se deu em três partes, com 90, 180, e 365 dias de treino, e em todos os casos, os dados usados como teste foram os 60 dias subsequentes ao último dia do treino, isso possibilitou uma redução do viés de teste, pois considerou eventos em quantidade e datas diferentes em cada conjunto. Foram utilizados para treino apenas dados do ano de 2017, que dentro do conjunto de dados é o ano mais recente em que há dados em todos os meses. Dentre os tipos de delito, foram selecionados os registros do tipo *Robbery*, um dos tipos de crime também usado em Chainey *et al.* (2008).

Para o *grid search* foi definido de forma fixa quais os valores de hiperparâmetros seriam utilizados baseado em valores apresentados em outros trabalhos, como Chainey *et al.* (2008). No *random search* o espaço de busca foi definido por uma faixa de valores, sendo que a escolha do valor se deu de forma aleatória, a cada vez escolhendo um valor dentro do intervalo e limitado a um número máximo de 10 iterações. Os *hotspots* gerados em ambos os casos foram avaliados com as métricas HR e PAI.

Também houve a introdução de um componente temporal combinado ao KDE. Com o objetivo de investigar os impactos de um kernel temporal junto com o kernel espacial e buscar otimizar também os hiperparâmetros desse novo modelo utilizando as técnicas já citadas. A combinação ocorre basicamente multiplicando a densidade de cada evento no espaço por sua densidade no tempo, utilizando uma função que prioriza eventos mais recentes.

Ao final dos experimentos pôde-se observar que os resultados obtidos pelo *random search* superaram os do *grid search* em 7,63% utilizando um valor de *bandwidth* menor (318m). Também foi observado que combinar um kernel de tempo que prioriza eventos mais recentes com KDE resultou em uma melhora de 4,31% no resultado do *grid search*. Isso reforça o que

é dito em Hu *et al.* (2018), indicando que novas ocorrências de crimes tendem a acontecer em locais que registraram delitos recentemente. Outra informação que reforça esse ponto, é a de que o conjunto de dados de 180 dias de treino obteve melhores resultados, isso se dá provavelmente o fato de que em 90 dias de treino há poucos eventos, o que prejudica os resultados e em 365 dias, mesmo com muitos eventos, a maioria são eventos antigos o que também pode ser o motivo de resultados não tão bons. Nesse cenário, 180 dias está como um meio termo e leva a melhor avaliação.

Como proposta de melhorias e trabalhos futuros, poderia ser desenvolvido um novo método para a otimização de hiperparâmetros baseado no *random search*, mas de forma que a escolha do valor não fosse puramente aleatória, explorar melhor o espaço de busca poderia trazer excelentes resultados. Outro ponto seria melhorar o método de combinação entre o kernel no espaço e o kernel no tempo, pois gerar uma matriz com todos as densidades de todos os eventos em cada uma das células da grade é muito custoso em termos de memória, o que dificulta a aplicação da técnica para áreas muito grandes ou com muitos eventos. Outra possibilidade é combinar a densidade espacial com outras funções de kernel, como por exemplo: os turnos durante o dia, ou ainda eventos sazonais, que acontecem em momentos específicos como feriados ou datas comemorativas.

REFERÊNCIAS

- AKIBA, T.; SANO, S.; YANASE, T.; OHTA, T.; KOYAMA, M. Optuna: A next-generation hyperparameter optimization framework. In: **Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining**. [S. l.: s. n.], 2019. p. 2623–2631.
- BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of machine learning research**, [S. l.], v. 13, n. 2, 2012.
- CHAINEDY, S.; TOMPSON, L.; UHLIG, S. The utility of hotspot mapping for predicting spatial patterns of crime. **Security Journal**, [S. l.], v. 21, p. 4–28, 02 2008.
- DRAWVE, G. A metric comparison of predictive hot spot techniques and rtm. **Justice Quarterly**, Taylor & Francis, [S. l.], v. 33, n. 3, p. 369–397, 2016.
- HU, Y.; WANG, F.; GUIN, C.; ZHU, H. A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation. **Applied geography**, Elsevier, [S. l.], v. 99, p. 89–97, 2018.
- ITURRITXA, E.; MESANZA, N.; BRENNING, A. Spatial analysis of the risk of major forest diseases in monterey pine plantations. **Plant Pathology**, Wiley Online Library, [S. l.], v. 64, n. 4, p. 880–889, 2015.
- KOCH, P.; GOLOVIDOV, O.; GARDNER, S.; WUJEK, B.; GRIFFIN, J.; XU, Y. Autotune: A derivative-free optimization framework for hyperparameter tuning. In: **Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining**. [S. l.]: [S. n.], 2018. p. 443–452.
- LEVINE, N. The “hottest” part of a hotspot: comments on “the utility of hotspot mapping for predicting spatial patterns of crime”. **Security journal**, Springer, [S. l.], v. 21, n. 4, p. 295–302, 2008.
- LIASHCHYNSKYI, P.; LIASHCHYNSKYI, P. Grid search, random search, genetic algorithm: a big comparison for nas. **arXiv preprint arXiv:1912.06059**, [S. l.], 2019.
- NAQA, I. E.; MURPHY, M. J. What is machine learning? In: **machine learning in radiation oncology**. [S. l.]: Springer, 2015. p. 3–11.
- NUNES, F. C.; SILVA, T. L. C. d.; QUEIROZ, J. F. d.; MACÊDO, J. A. F. d.; PORCINO, W. C. A novel approach to approximate crime hotspots to the road network. In: **Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility**. [S. l.: s. n.], 2019. p. 53–61.
- PROBST, P.; BOULESTEIX, A.-L.; BISCHL, B. Tunability: importance of hyperparameters of machine learning algorithms. **The Journal of Machine Learning Research**, JMLR. org, [S. l.], v. 20, n. 1, p. 1934–1965, 2019.
- ROSSER, G.; DAVIES, T.; BOWERS, K.; JOHNSON, S.; CHENG, T. Predictive crime mapping: Arbitrary grids or street networks? **Journal of Quantitative Criminology**, [S. l.], v. 33, p. 1–26, 09 2016.

- SCHRATZ, P.; MUENCHOW, J.; ITURRITXA, E.; RICHTER, J.; BRENNING, A. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. **Ecological Modelling**, Elsevier, [S.l.], v. 406, p. 109–120, 2019.
- WEGLARCZYK, S. Kernel density estimation and its application. In: EDP SCIENCES. **ITM Web of Conferences**. [S. l.], 2018. v. 23, p. 00037.