



Graduação em Engenharia da Computação

Análise de parâmetros de algoritmos de seleção dinâmica na previsão de série temporais

Leandro de Andrade Moura

Recife
2022

Universidade Federal de Pernambuco
Centro de Informática

Análise de parâmetros de algoritmos de seleção dinâmica na previsão de série temporais

*Trabalho apresentado ao Centro de Informática da
Universidade Federal de Pernambuco como requisito
parcial para obtenção do grau de Bacharel em
Engenharia da Computação.*

Orientador: *Prof. Dr. Paulo Salgado Gomes de Mattos Neto*

Recife
2022

Resumo

Abordagens de seleção dinâmica têm sido aplicadas com sucesso no contexto de séries temporais. Um dos aspectos importantes para se obter uma alta acurácia é a coleção de preditores, que deve ser diversa o suficiente para os modelos gerarem diferentes previsões, mas não ao ponto de ser aleatória, pois eles precisam ter um bom desempenho. Outro aspecto crucial é o critério para realizar a seleção do preditor a ser utilizado na previsão do padrão de teste. Para isto, é preciso escolher qual métrica e quantidade de vizinhos tem maior afinidade com a coleção de preditores e o conjunto de dados que está sendo previsto. Este trabalho utiliza Algoritmo Genético para buscar uma coleção de preditores a ser utilizada em uma abordagem de seleção dinâmica e indicar quais métricas e quantidades de vizinhos têm maior afinidade com a coleção encontrada. Este trabalho mostra que a seleção dinâmica é altamente suscetível à escolha destes parâmetros.

Conteúdo

Resumo	2
Conteúdo	4
1. Introdução	5
1.1 Contexto	5
1.2 Objetivo	5
1.3 Estrutura	6
2. Conceitos básicos	7
2.1 Seleção Dinâmica	7
2.2 Regressão de Vetor de Suporte (SVR)	8
2.3 Algoritmo genético	8
2.4 Padrão Temporal	8
2.5 Erro quadrático médio	8
2.6 Métricas	9
3. Abordagem proposta	11
3.1 Configuração dos Genes	11
3.2 <i>Bagging</i>	12
3.3 Poda	12
3.4 Mutação	13
3.5 Cruzamento	13
3.6 Função de fitness	14
3.7 Finalizando geração	14
4. Experimento e Resultados	15
4.1 Protocolo Experimental	15
4.2 Resultados	15
5. Conclusão	20
5.1 Trabalhos futuros	20
Referências	21

Capítulo 1

Introdução

1.1 Contexto

Uma série temporal é composta por observações ordenadas, capturadas em um intervalo específico de tempo [1]. Séries temporais podem ter diferentes características. Exemplos de comportamento [2] que podem diferenciar as séries temporais são: Sazonalidade: quando um existe um padrão periódico e recorrente na série; Cíclico: que difere da sazonalidade por poder ocorrer em diferentes frequências; Tendência: que é a mudança sistemática e não periódica de crescimento ou queda dos valores da série temporal; Estacionaridade: que ocorre quando a série mantém constante suas propriedades estatísticas ao longo do tempo.

Diferentes abordagens têm sido utilizadas para prever séries temporais. O desafio, porém, reside no fato de que geralmente fenômenos temporais têm suas características alteradas através do tempo [3]. Assim, uma promissora abordagem tem sido a utilização de Ensemble, partindo da premissa de que diferentes preditores podem se especializar em diferentes padrões temporais da série temporal [4]. Em especial, destacam-se as abordagens dinâmicas, que tendem a apresentar melhores resultados quando utilizados à séries temporais. Espera-se, portanto, que os Ensembles comportem-se melhor que um preditor individualmente, conhecido, neste contexto, como monolítico.

A construção de Ensemble é composta por três etapas [5]: geração do conjunto de preditores, também conhecido como *Pool*, a seleção e a integração. Na primeira etapa, gera-se os preditores, sendo importante que eles sejam diversos entre si, mas sejam eficazes. Na segunda etapa, seleciona-se um ou mais preditores com base em algum critério. No terceiro, utiliza-se os preditores selecionados para compor o resultado final. Este estudo utiliza a Seleção Dinâmica como Ensemble, que verifica dinamicamente qual preditor utilizar para cada novo padrão temporal a ser previsto. Esta abordagem tem apresentado bons resultados pois ela tira proveito de que diferentes preditores podem aprender melhor sobre os padrões temporais de diferentes regiões da série temporal [6].

Para gerar um *pool* na Seleção Dinâmica, para além dos hiperparâmetros de cada preditor, há também os hiperparâmetros da Seleção Dinâmica, que são a métrica com a qual se compara os padrões de entrada com os já aprendidos pelos preditores e a quantidade de vizinhos com a qual um padrão será comparado para verificar a proximidade e, a partir disso, escolher o melhor preditor [7].

1.2 Objetivo

Diante da grande quantidade de parâmetros a serem escolhidos pelo Ensemble quando comparado ao uso de um único preditor, surge a proposta de utilizar uma busca baseada em algoritmo genético. Objetiva-se, assim, transformar estes hiperparâmetros em genes e a acurácia de previsão do sistema em *fitness* para que o algoritmo genético possa

buscar a melhor configuração para cada série temporal que for dada como entrada. O resultado da busca não só será os hiperparâmetros de todos os preditores como também a métrica e o número de vizinhos no qual o uso da Seleção Dinâmica obtenha mais afinidade com o conjunto de dados em questão. Este estudo, no entanto, foca em analisar o impacto de diferentes métricas e número de vizinhos para o resultado final do sistema de previsão.

1.3 Estrutura

Este trabalho está organizado da seguinte forma: capítulo 2 apresentará os conceitos básicos acerca de Seleção Dinâmica, Algoritmo Genético e equações utilizadas neste estudo. No capítulo 3 será utilizado estes conceitos para o alcance deste objetivo e será apresentado todo o conhecimento necessário para a reprodução deste estudo. No capítulo 4, será apresentado os valores escolhidos para rodar o experimento, assim como detalhes práticos que levam ao resultado deste estudo, que em seguida serão apresentados e discutidos. Por fim, o capítulo 5 conterà a conclusão deste estudo e também sugestões para futuras pesquisas.

Capítulo 2

Conceitos básicos

Este capítulo apresentará os conceitos de Seleção Dinâmica e Algoritmo Genético, centrais a este estudo, assim como o explicará qual preditor será usado para formular o conjunto de preditores, as métricas para comparação de padrões temporais e a fórmula utilizada para comparação de resultados, que é o *MSE*.

2.1 Seleção Dinâmica

A Seleção Dinâmica (já citado em [7], traduzido de *Dynamic Selection*, DS) trata-se de um Sistema de Múltiplos Preditores (*Multiple Predictor System*), também conhecido como Ensemble. Assim, para utilizar esta técnica, é necessário um conjunto de preditores bem treinados e diversos. Neste trabalho, isto é alcançado através da busca do algoritmo genético. O treinamento dos preditores consiste em apresentar diferentes padrões temporais como entrada e, como saída, a previsão deste padrão. Na seleção dinâmica, para cada padrão de teste, escolhe-se K padrões de treino ou validação mais próximos dele para, então, verificar qual preditor alcançou melhores resultados naquela previsão próxima. Esta proximidade é medida através de uma métrica de distância. O melhor preditor é escolhido e utilizado para prever o padrão de teste. O resultado então pode ser comparado com o *Oracle* (ou, em sua tradução livre, Oráculo), que é o resultado obtido hipoteticamente no caso do Ensemble escolher sempre o melhor preditor.

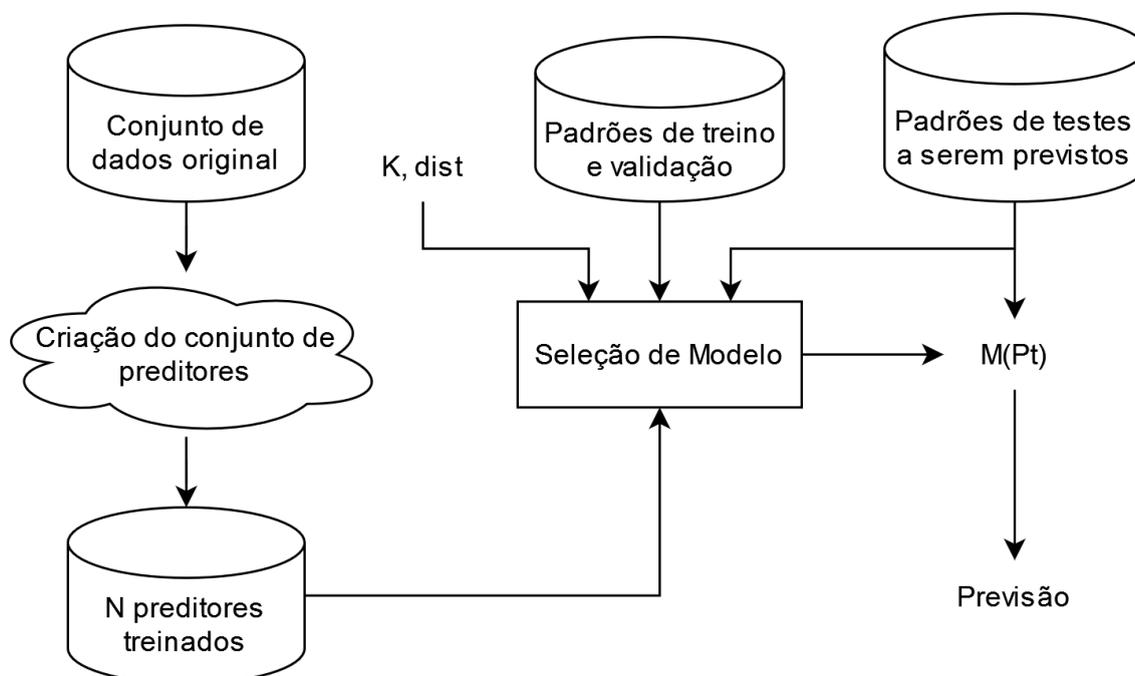


Figura 2.1 - Esquema da Seleção Dinâmica. Os hiperparâmetros K , o número de vizinhos, e $dist$, a métrica de comparação, são utilizados pela seleção de modelos, junto com os padrões de teste e validação e o padrão de teste a ser previsto. No fim deste processo, o modelo M é selecionado e aplicado ao padrão de teste (PT) para obter a previsão.

2.2 Regressão de Vetor de Suporte (SVR)

Para preditores da seleção dinâmica, foi escolhido a Regressão de Vetor de Suporte (tradução livre de *Support Vector Regression* - SVR). Este preditor, cujo treinamento é supervisionado, tem poucos parâmetros quando comparado às redes neurais, mas uma boa performance para tarefas de previsão [8]. Utilizou-se a implementação do *SKlearn* do SVR neste estudo, e como parâmetros para busca utilizou-se *kernel*, *C*, *Epsilon* e *Gamma*. Os demais parâmetros foram utilizados em seus valores padrão.

2.3 Algoritmo genético

O algoritmo genético baseia-se na estratégia de evolução encontrada empiricamente pela natureza. Ele consiste de uma população inicial aleatória que, a cada geração, sofre mutações e se reproduz, para então passar por uma pressão evolutiva, que seleciona os indivíduos mais aptos a resolver um problema. Este processo é repetido por várias gerações até que indivíduos satisfatórios sejam encontrados. Cada indivíduo é constituído por genes, que é onde há os efeitos da mutação e reprodução, e cada combinação genética implica em uma aptidão (tradução livre de *fitness*) diferente para resolver o problema [9-11].

Para utilizar esta abordagem para resolver problemas ou fazer buscas na computação, é necessário ser capaz de expressar este cenário em termos dos genes do indivíduo e pela função de *fitness*, que, para dado gene, expressa numericamente o quão apto um indivíduo é no cenário. Além disso, diversas modificações podem ser feitas na função de mutação e reprodução para controlar o processo evolucionário. Por exemplo, pode-se fazer este processo focar em uma busca ampla, através de mutações grandes, ou focar sua busca por soluções ótimas em um local específico, através de mutações pequenas. Na literatura isto é conhecido como *exploration* e *explotation*.

2.4 Padrão Temporal

Em previsões de séries temporais, um padrão temporal se refere à uma sequência ininterrupta de medições do conjunto de dados utilizada como entrada que contém como saída a medição imediatamente após a última medição da sequência de entrada. O tamanho ideal desta janela varia de acordo com a série temporal.

2.5 Erro quadrático médio

O erro quadrático médio (tradução livre do inglês *Mean Square Error* - MSE) foi utilizado para avaliação dos resultados (Equação 2.1). No texto de previsão, recebe como entrada as N previsões do preditor e utiliza N resultados esperados e retorna um número real. Esta métrica foi escolhida por estar associada à avaliação de predição de séries temporais em vários artigos, inclusive ao artigo de referência deste estudo (já citado em [6]).

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.1)$$

em que y_i e \hat{y}_i são o valor real da série e a previsão no tempo i , respectivamente.

2.6 Métricas

A métrica, ou medição de similaridade (tradução livre do inglês *Similarity Measures* utilizado em [6]), é utilizada pela Seleção Dinâmica para medir o quão próximos são dois padrões temporais. As métricas são um parâmetro que influenciam no resultado final da Seleção Dinâmica. Das sete métricas escolhidas, cinco (*Euclidian*, *Chebyshev*, *Manhattan*, *Cosine*, *DTW*) foram utilizadas para comparação com [6], duas (*Canberra*, *Braycurtis*) foram utilizadas para avaliar novas métricas.

Euclidean

$$dis(y, \hat{y}) = \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2.2)$$

Canberra

$$dist(y, \hat{y}) = \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|} \quad (2.3)$$

Braycurtis

$$dist(y, \hat{y}) = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i + \hat{y}_i|} \quad (2.4)$$

Chebyshev

$$dist(y, \hat{y}) = \max |y_i - \hat{y}_i| \quad (2.5)$$

Cosine

$$dist(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n y_i \times \hat{y}_i}{\sqrt{\sum_{i=1}^n y_i^2} \times \sqrt{\sum_{i=1}^n \hat{y}_i^2}} \quad (2.6)$$

Manhattan

$$dis(y, \hat{y}) = \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.7)$$

Dynamic Time Warping (DTW)

$$dist(i, j) = d(y_i, \hat{y}_j) + \min\{dist(i-1, j-1), dist(i-1, j), dist(i, j-1)\} \quad [12] (2.8)$$

obs: DTW foi implementada em python pelo autor seguindo o seguinte link:
<https://towardsdatascience.com/dynamic-time-warping-3933f25fcdd>

Capítulo 3

Abordagem proposta

Usar o Algoritmo Genético para gerar um conjunto diverso e eficaz para a Seleção Dinâmica. Para isto, será gerada uma população em que cada indivíduo consiste em um conjunto de preditores de SVR. A função de *fitness* a ser usada para avaliar cada indivíduo é o resultado do MSE dos melhores parâmetros da Seleção Dinâmica da combinação entre as diversas métricas e valores de K. Assim, busca-se garantir que a população esteja bem equilibrada entre sua diversidade e capacidade de predição e encontrar os parâmetros da Seleção Dinâmica com maior afinidade à série temporal em questão.

3.1 Configuração dos Genes

Cada indivíduo da população possui os seguintes parâmetros:

- *Mutation_gen*, a intensidade que todos os demais genes, excetuando-se o ele próprio, irão mutacionar. Espera-se, através deste parâmetro, que o próprio algoritmo genético seja capaz de escolher entre *exploitation* e *exploration*.
- *Step_lower_window* e *step_higher_window*: são, respectivamente, os valores mínimos e máximos que cada parâmetro *step* pode alcançar.

Além desses parâmetros, cada indivíduo também possui seu conjunto de preditores. Cada parâmetro de cada um desses preditores se configura como gene do indivíduo. São eles[13]:

- *Step*: determina o tamanho da janela do padrão de previsão do preditor.
- *Kernel*: o núcleo utilizado no SVR.
- *Gamma*: coeficiente para o kernel.
- *C*: o parâmetro de regularização do SVR.
- *Epsilon*: a distância para qual não haverá penalidade caso o SVR erre.

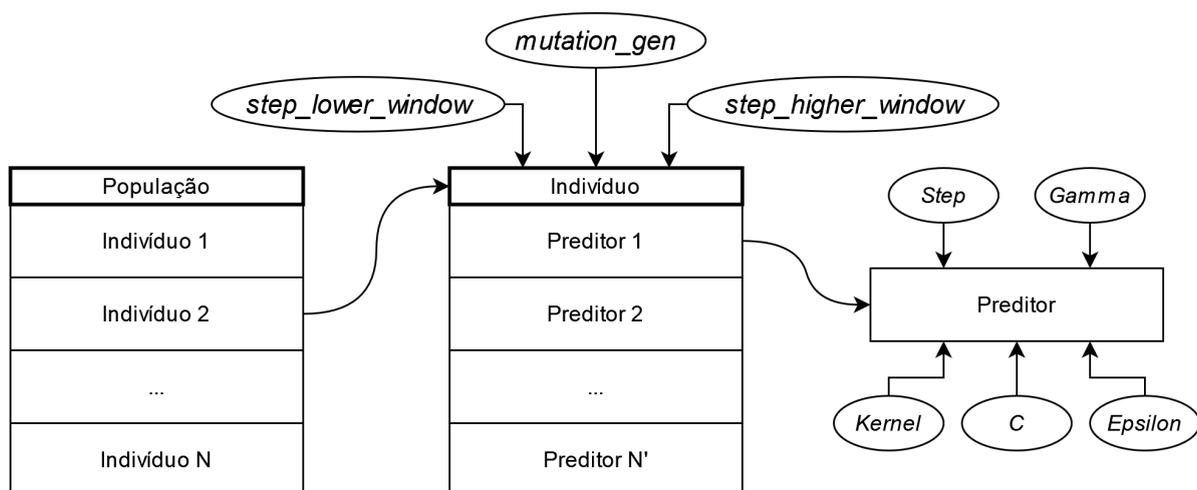


Figura 3.1: esquema descrevendo a modelagem da população que será utilizada no Algoritmo genético. N é o número de indivíduos e N' é o número de preditores de cada indivíduo.

Assim, pode-se afirmar que o Gene de cada indivíduo da população será expresso da seguinte forma:

$Gene(mutation_gen, step_lower_window, step_higher_window, step_1, kernel_1, gamma_1, C_1, epsilon_1, step_2, kernel_2, gamma_2, C_2, epsilon_2, \dots, step_n, kernel_n, gamma_n, C_n, epsilon_n)$
em que n é número de preditores por indivíduo.

3.2 Bagging

Para buscar criar preditores especializados em diferentes regiões de competência, cada preditor recebeu uma diferente parte do conjunto de dados ao ser treinado. Mais especificamente, cada conjunto de treino foi dividido em dez partes e os preditores foram divididos em dez grupos. O grupo 1 recebeu a parte 1 e 2, o grupo 2 recebeu a parte 2 e 3, o grupo 3 recebeu a parte 3 e 4, assim por diante. O grupo 10 recebeu todo o conjunto de treinamento. Os conjuntos de validação e teste são usados por inteiro em todos os preditores na durante todo o processo de busca e Seleção Dinâmica.

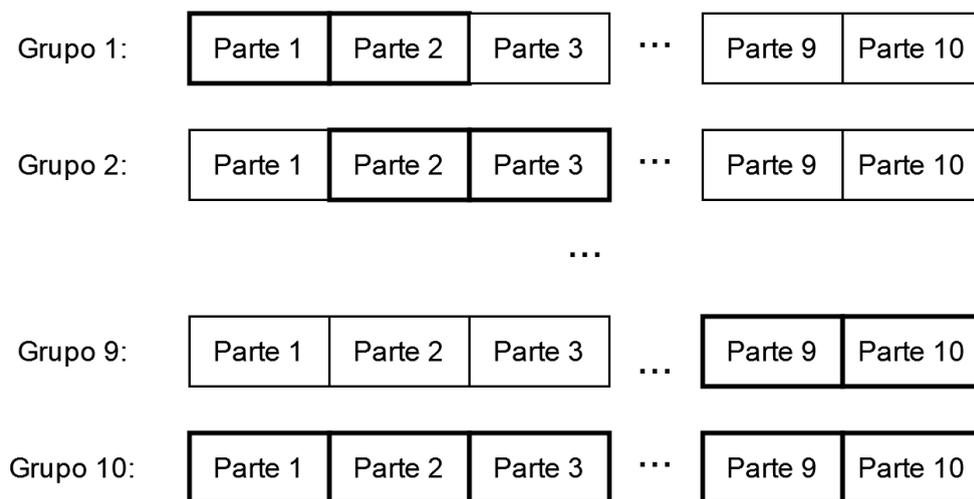


Figura 3.2: *bagging* utilizado neste estudo. As partes em negrito foram as utilizadas no grupo especificado. O último grupo recebe o conjunto de dados completo.

3.3 Poda

Cada novo indivíduo é testado todas as combinações de seleção dinâmica, combinando entre as métricas e o número de vizinhos, para verificar qual é a melhor. Os preditores de um indivíduo que não foram utilizados por nenhuma seleção dinâmica entram na lista de poda. Deste modo, antes da mutação, substitui-se os preditores que não foram utilizados pela seleção dinâmica pelos que participaram, preferencialmente do mesmo grupo. Se não houver nenhum outro preditor do mesmo grupo para substituí-lo, escolhe-se outro aleatoriamente outro preditor que tenha sido usado na Seleção Dinâmica. É preciso atentar para, ao escolher um preditor de outro grupo, manter o mesmo conjunto de treino associado a ele.

3.4 Mutação

Cada progenitor é escolhido pelo método roleta, utilizando como pontuação a fórmula 3.1, a cada rodada do algoritmo, onde o *fitness* é o MSE do conjunto de validação. Seleciona-se dois progenitores e aplica-se neles a mutação. Cada mutação varia o *mutation_gen* de acordo com a fórmula 3.2. Os *step_lower_window* e *step_higher_window* variam de acordo com a fórmula 3.3. Depois seleciona-se os preditores de acordo com a chance de mutação. Se um preditor for escolhido, aplica-se a fórmula 3.4 em *C*, *gamma* e *epsilon* e a fórmula 3.5 no *step*. Uma vez que um preditor é mutacionado, ele é treinado pelo conjunto de treino de acordo com o *bagging* e o seu grupo.

$$peso_rolleta = \log\left(\frac{1}{fitness}\right) \quad (3.1)$$

$$mutation_gen = mutation_gen + random.gauss(0, tau_l) \quad (3.2)$$

$$step_window = step_window + 10 * random.gauss(0, mutation_gen) \quad (3.3)$$

$$gene = gene * math.exp(random.gauss(0, mutation_gen)) \quad (3.4)$$

$$step = \max(step_lower_window, \min(step_heigher_window, step + random.gauss(0, mutation_gen) * 3)) \quad (3.5)$$

3.5 Cruzamento

Obtém-se, através disto, 4 indivíduos a serem cruzados: 2 progenitores e 2 progenitores mutacionados. O cruzamento é feito 2 a 2 e é multiplicado pelo *children_number*, a quantidade de cruzamentos. Cada cruzamento é feito da seguinte forma: dois filhos são gerados fazendo corte de 1 ponto e cruzando estas partes, dois filhos são gerados com cortes de 2 pontos. Outros quatro filhos são escolhidos através da escolha aleatória entre o progenitor 1 e o progenitor 2. Gera-se, portanto, um total de 32 filhos vezes *children_number* por passo. Este desenho foi concebido para tirar proveito de que a parte mais custosa de todo o processo é treinar um preditor.

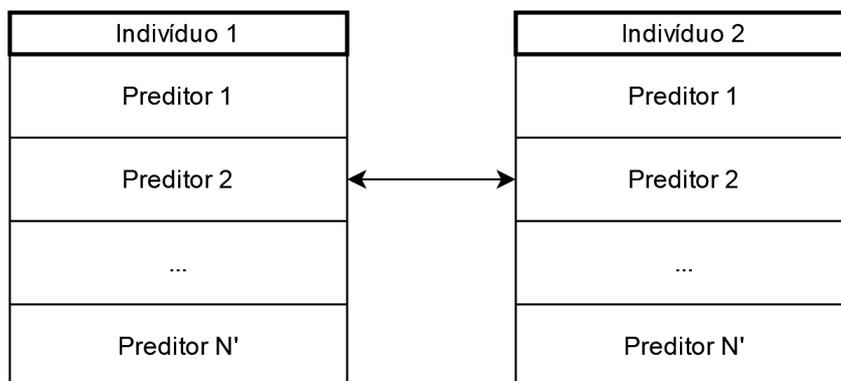


Figura 3.3: esquema de reprodução utilizado neste estudo.

3.6 Função de *fitness*

Após a criação dos diversos filhos, aplica-se neles a função de *fitness*. Como já dito, esta função consistirá em utilizar a combinação de todas as métricas e todos os valores de K na Seleção Dinâmica no conjunto de dados em questão e, depois de feita a previsão, selecionar aquela que obtiver menor erro associado através do MSE. A Seleção Dinâmica se torna barata quando comparada à treinar um preditor uma vez que os valores do conjunto de treino e validação, seus respectivos erros, sejam pré-computados cada vez que se treina um novo preditor. Além disso, também foram pré-computados os $\max(K)$ vizinhos de todos os padrões temporais para todos os passos, o que ajudou ainda mais a acelerar a Seleção Dinâmica e, por consequência, a função de *fitness*.

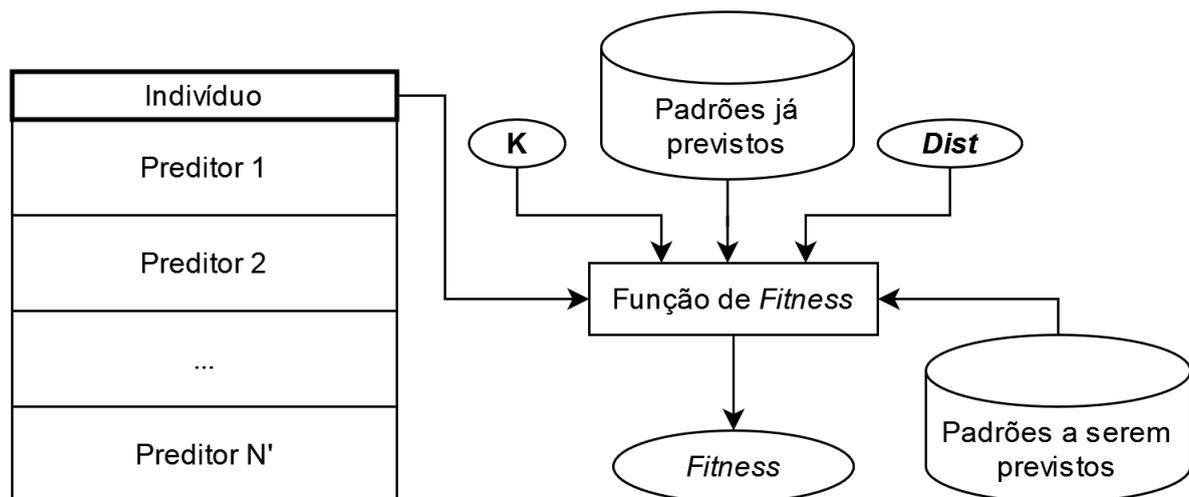


Figura 3.4: esquema da função fitness. Atentar para K e $dist$ em negrito, que representam um vetor de valores. Padrões já previstos são os padrões nos quais já se conhece o erro ao utilizá-los em cada preditor do indivíduo. Durante a busca estes padrões são apenas o conjunto de treino. No teste, são o conjunto de treino e o de validação. Já os padrões a serem previstos são o conjunto de validação na busca e conjunto de teste no teste. A função de *fitness* utiliza os padrões já conhecidos para, a cada novo padrão a ser previsto que chega, decidir qual modelo utilizar. Um conjunto de predições é construído para cada combinação de K e $dist$. Aplica-se MSE em todos eles e escolhe-se o melhor. Retorna-se este valor como *fitness* associado ao indivíduo.

3.7 Finalizando geração

Ao aplicar a função de *fitness* em todos os filhos, trocar o pior indivíduo da população pelo melhor filho desta nova geração. Assim, finaliza-se uma geração e os passos 3.4, 3.5 e e.6 se repetem até que o critério de parada seja alcançado. Neste estudo, utilizou-se o critério de tempo decorrido desde o início por oferecer previsibilidade de tempo e por não haver prejuízo associado à continuar a busca em uma população já convergida uma vez que não se observou *overfitting* utilizando a função de *fitness* descrita na seção anterior.

Capítulo 4

Experimento e Resultados

4.1 Protocolo Experimental

Este estudo utilizou os seguintes conjuntos de dados: *Goldman Sachs (Goldman)*, *Microsoft (MSFT)*, *Pollution*, *Red Wine*, *Star Brightness (Star)* e *Vehicle*. *Goldman* e *Microsoft* tratam-se de séries temporais de ações financeiras no período de 04/01/2010 à 31/12/2012, totalizando 754 pontos cada. *Star* trata-se do brilho de uma estrela à meia-noite diariamente, tendo 600 pontos. *Vehicle* são os preços de veículos mensalmente nos EUA de 1971 à 1991, totalizando 187 pontos. *Pollution* refere-se a quantidade de poluentes na França mensalmente, totalizando 130 pontos. *Red Wine* são preços de vinho australiano entre 1980 à 1994, totalizando 187 pontos.

Cada conjunto de dados se divide em 60% para treino, 20% para validação e 20% para teste sequencialmente. Estes conjuntos foram escolhidos por possuir diferentes comportamentos e também por serem famosos na literatura. Os dados foram normalizados para valores proporcionais entre 0 e 1. Foram utilizados 100 SVRs por indivíduo na população. Foram feitas 3 rodadas de 2h30min por conjunto de dados.

Foram testadas as seguintes métricas: *euclidian*, *canberra*, *braycurtis*, *chebyshev*, *cosine*, *manhattan* e *dtw*. Os valores para K, o número de vizinhos, avaliados foram: 1, 3, 5, 10 e 15. Por questões de performance, a quantidade de cruzamento, *children_number*, variou de acordo com o tamanho do conjunto de dados. Para os maiores optou por 1 filho, para os conjuntos menores, 8 filhos. Os parâmetros do SVR *C*, *Gamma* e *Epsilon* variaram de 10^{-8} até 10^6 para os conjuntos menores e para os maiores, de 10^{-8} até 10^4 . Já o *kernel* do SVR variou entre *sigmoid* e *rbf*. O valor de *step_lower_window* é limitado inferiormente por 3, já o valor de *step_highier_window* é limitado superiormente em 25, e *step_lower_window* não pode ser igual ou maior que *step_highier_window*. A chance de mutação ficou entre 0.75 para os conjuntos menores a 0.25 para os conjuntos maiores. *Mutation_gen* foi limitado entre 0 e 0.5 e *tau_l* foi fixado em 0.2.

4.2 Resultados

Para analisar a relação entre as métricas e os K número de vizinhos na seleção dinâmica, optou-se por gráficos de barras, apresentados abaixo através de uma figura para cada série, contendo todas as métricas, havendo no eixo X o valor de K e no eixo Y o erro em *MSE* para cada conjunto de métrica/valor de K. Assim, observa-se que as escolhas de K e da métrica causam um grande impacto na seleção dinâmica e que cada conjunto de dados possui um diferente padrão de comportamento em relação a essas variações. Cada uma dessas figuras trata-se de uma série diferente.

Nas Figuras 4.2 e 4.6, é perceptível que há uma relação entre K e o MSE. Porém, esta relação é invertida em ambas, pois em uma, quanto maior o K, melhor, em outra, quanto menor o K, melhor. Nas Figuras 4.2, 4.3, 4.4 e 4.6 verifica-se um aumento no MSE

para a métrica *cosine*. Na Figura 5 não se verifica nenhum padrão visual significativo nos resultados além de uma leve piora para K igual a um em todas as métricas exceto na *dtw*. Na Figura 7, as métricas *chebyshev* e *cosine* apresentaram um padrão diferente das demais em relação ao K.

Este resultado indica uma alta variação de comportamentos a depender dos parâmetros da seleção dinâmica de tal modo que, em casos como a Figura 2, a acurácia pode ficar muito aquém do possível caso escolhida a métrica com pouca afinidade à série temporal em questão.

Na Tabela 1, apresentada após as figuras, estão os melhores resultados da seleção dinâmica para cada série temporal, o monolítico que obteve melhor desempenho em validação na *pool* e o *Oracle*. Observa-se que a métrica *dtw* possuiu um bom desempenho na maioria das séries temporais e que, neste estudo, o monolítico ficou melhor que a Seleção Dinâmica na maioria das séries.

Com a Tabela 1, observa-se a possibilidade de uma melhora ainda maior na Seleção Dinâmica, uma vez que os resultados da Seleção Dinâmica estão, em todos os casos excetuando-se o *redwine*, pelo menos à uma ordem de grandeza maior que o *Oracle*.

Tabela 1 - Resultados comparados

	goldman	msft	pollution	redwine	star	vehicle
<i>Oracle</i>	2,46E-05	2,16E-04	2,37E-03	1,34E-02	7,82E-06	4,53E-03
Monolítico	3,80E-04	1,32E-03	1,44E-02	7,53E-02	1,23E-04	1,40E-02
Melhor DS	DTW, k=15	Canberra, k=10	DTW, k=5	Cosine, k=10	DTW, k=1	Euclidian, k=10
	4,92E-04	1,83E-03	1,20E-02	3,97E-02	5,71E-04	1,38E-02

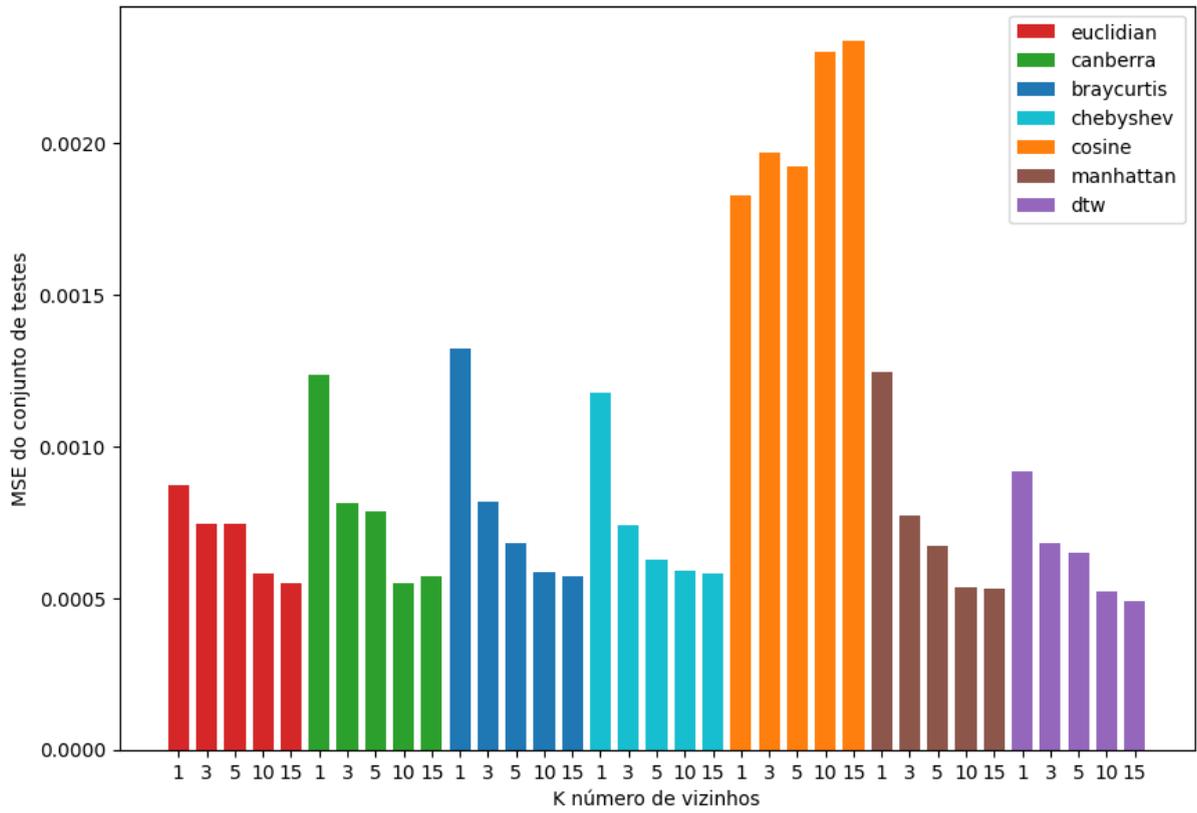


Figura 4.1 - Goldman

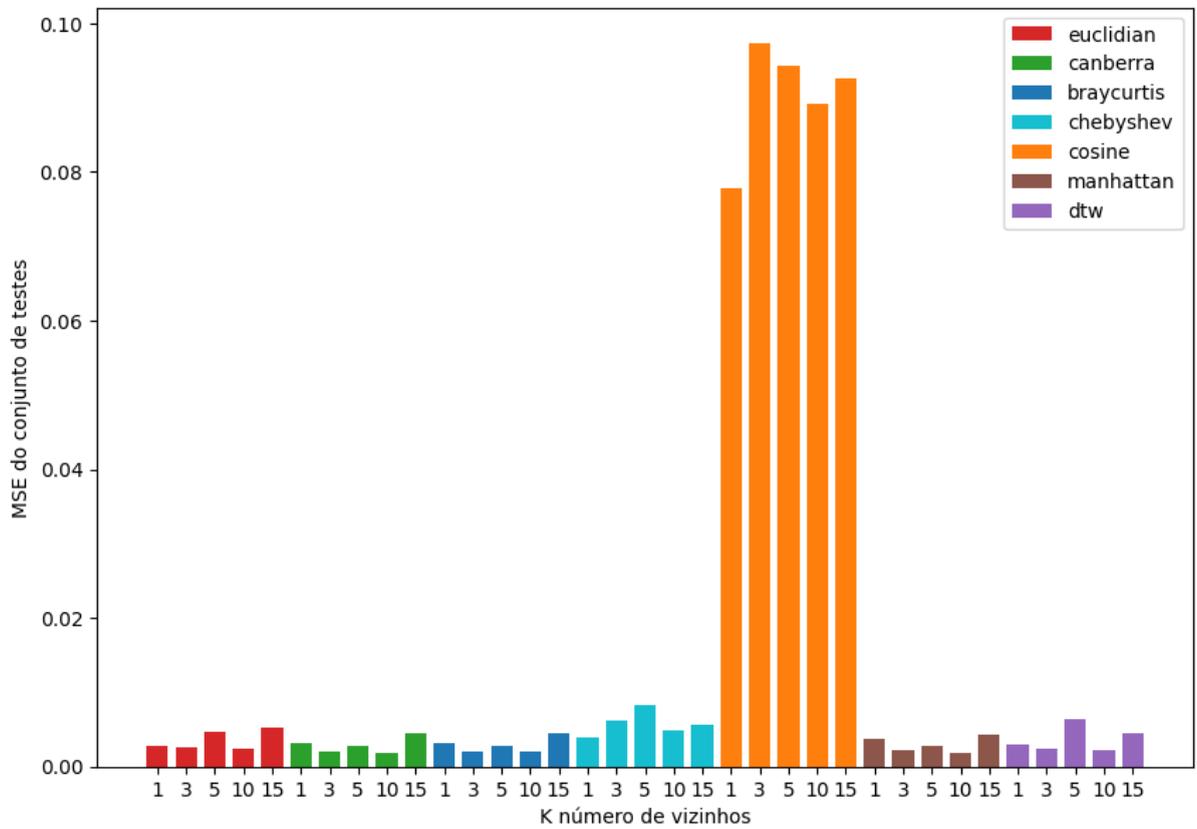


Figura 4.2 - MSFT

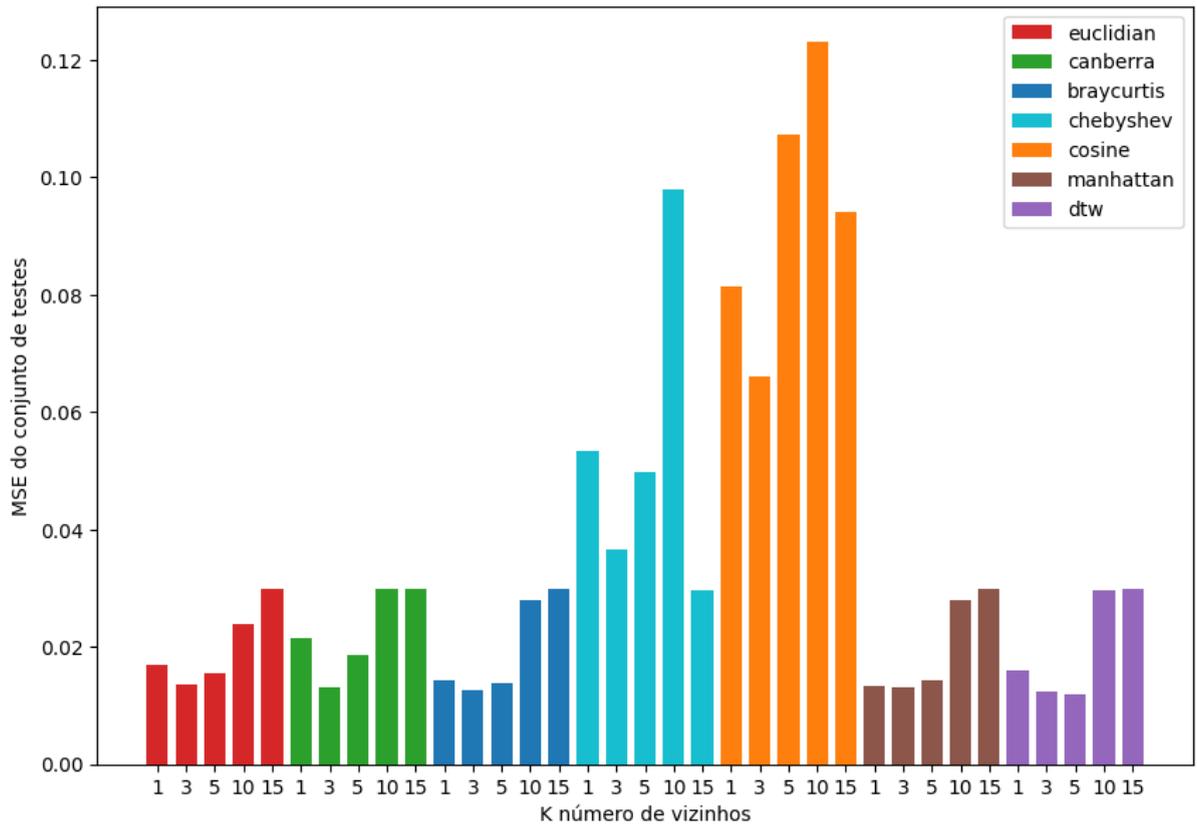


Figura 4.3 - Pollution

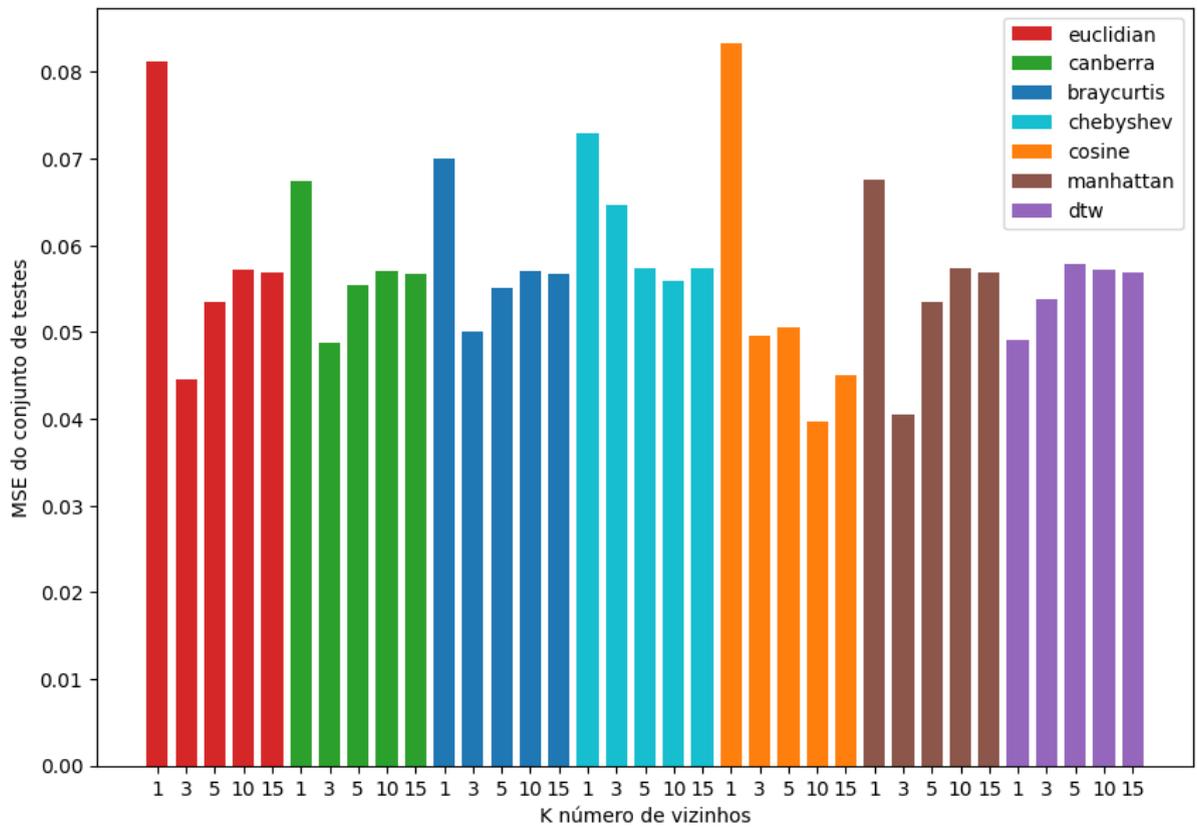


Figura 4.4 - Redwine

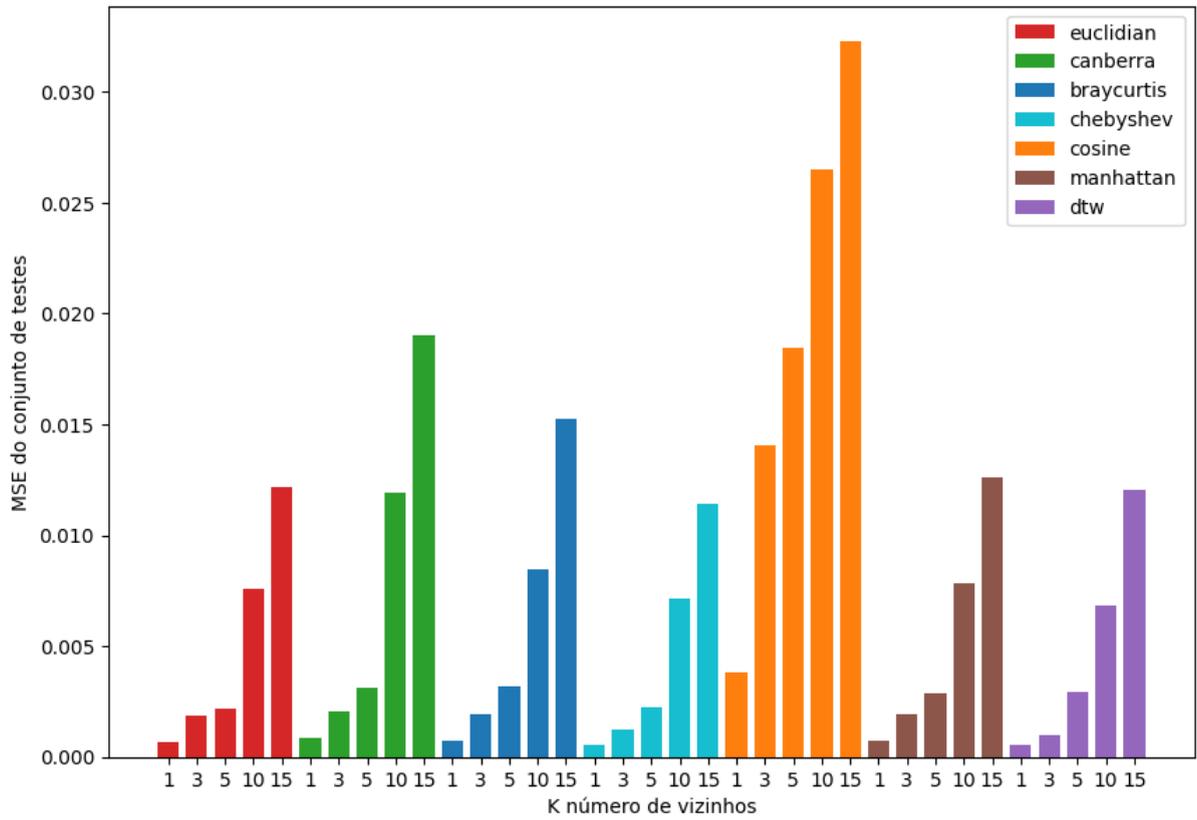


Figura 4.5 - Star

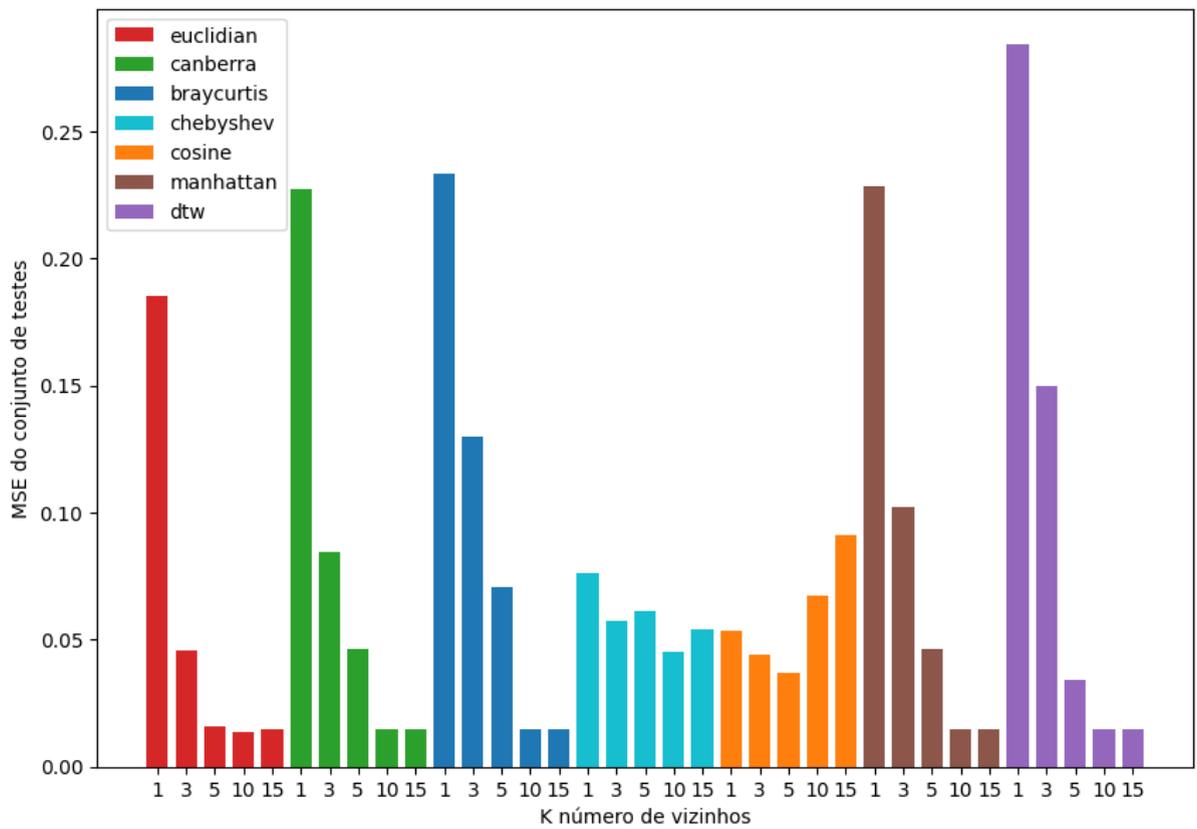


Figura 4.6 - Vehicle

Capítulo 5

Conclusão

Este estudo analisa o comportamento da seleção dinâmica para diferentes métricas e valores para K . Para isto, o algoritmo genético foi utilizado para buscar os hiperparâmetros, mas também buscando alcançar uma população com bom desempenho e diversa. Observa-se, através dos resultados, que cada conjunto de dados se comporta de maneira diferente.

Em comparação com o artigo de referência, observa-se que estes resultados ficaram um pouco piores. Isto pode ser explicado por diferenças no método de *bagging* e na divisão entre conjunto de treino, validação e teste ou, ainda, porque o algoritmo genético ficou preso em algum mínimo local e, por isso, gerou uma população uniforme.

Já quanto aos resultados dos monolíticos, isto provavelmente acontece devido à eficiência de busca do algoritmo genético e também ao SVR ser uma máquina de aprendizagem poderosa, de modo que um dos monolíticos tenham conseguido aprender de forma mais eficiente todos os padrões temporais das séries quando comparado aos demais. Isto, no entanto, implica num maior tempo de treinamento, sendo este um possível *trade-off* a ser considerado.

5.1 Trabalhos futuros

A seleção dinâmica baseia-se na premissa de diversidade de preditores. Existe, no entanto, a possibilidade do *pool* de preditores de um indivíduo do algoritmo genético ter ficado preso em um ótimo local. Assim, a ideia é adicionar ao algoritmo genético alguma métrica para selecionar o indivíduo cujos preditores sejam mais diversos.

É possível avaliar este estudo em uma série temporal ainda, de tal modo que fique a associação de um preditor com uma região de competência e torne-se inviável utilizar o Algoritmo Genético se os preditores forem expostos a toda série temporal. Objetiva-se, através disto, extrair o melhor da Seleção Dinâmica quando comparada aos monolíticos.

Por fim, alterar o método de treinamento/validação para, ao invés de passar os primeiros 60% da série temporal e os 20% seguintes para validação, passar 60% e 20% escolhidos de modo aleatório nos primeiros 80% da série. Deste modo é possível que os preditores aprendam sobre os padrões mais recentes e obtenham melhores resultados de maneira individual.

Referências

- [1] BROCKWELL, P. J.; DAVIS, R. A. Introduction to time series and forecasting. [S.l.]: Springer, 2016.
- [2] BROCKWELL, P. J.; DAVIS, R. A.; FIENBERG, S. E. Time series: theory and methods: theory and methods. [S.l.]: Springer Science & Business Media, 1991.
- [3] Xingquan Zhu, Xindong Wu and Ying Yang, "Dynamic classifier selection for effective mining from noisy data streams," *Fourth IEEE International Conference on Data Mining (ICDM'04)*, 2004, pp. 305-312, doi: 10.1109/ICDM.2004.10091.
- [4] Cao, Lijuan and Gu, Qingming. 'Dynamic Support Vector Machines for Non-stationary Time Series Forecasting'. 1 Jan. 2002 : 67 – 83.
- [5] Yao, C., Dai, Q. & Song, G. Several Novel Dynamic Ensemble Selection Algorithms for Time Series Prediction. *Neural Process Lett* 50, 1789–1829 (2019).
<https://doi.org/10.1007/s11063-018-9957-7>
- [6] SILVA, Eraylson Galdino da. Uma abordagem de seleção dinâmica de preditores baseada nas janelas temporais mais recentes. 2021. Tese (Doutorado em Ciência da Computação) - Universidade Federal de Pernambuco, Recife, 2021.
- [7] Silva, E. G., Cavalcanti, G. D., de Oliveira, J. F. L., & de Mattos Neto, P. S. (2020, July). On the evaluation of dynamic selection parameters for time series forecasting. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.
- [8] E. Silva, S. d. O. Domingos, G. D. Cavalcanti, and P. S. de Mattos Neto, "Improving the accuracy of intelligent forecasting models using the perturbation theory," in 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018, pp. 1–7.
- [9] G. H. T. Ribeiro, P. S. G. d. M. Neto, G. D. C. Cavalcanti and I. R. Tsang, "Lag selection for time series forecasting using Particle Swarm Optimization," *The 2011 International Joint Conference on Neural Networks*, 2011, pp. 2437-2444, doi: 10.1109/IJCNN.2011.6033535.
- [10] Aranildo Rodrigues Lima Junior, David Augusto Silva, Paulo Salgado Mattos Neto, and Tiago A.E. Ferreira. 2010. An experimental study of fitness function and time series forecasting using artificial neural networks. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation (GECCO '10)*. Association for Computing Machinery, New York, NY, USA, 2015–2018.
<https://doi.org/10.1145/1830761.1830848>
- [11] A. Rodrigues L. J., P. S. G. de Mattos Neto and T. A. E. Ferreira, "A prime step in the time series forecasting with hybrid methods: The fitness function choice," *2009 International Joint Conference on Neural Networks*, 2009, pp. 2703-2710, doi: 10.1109/IJCNN.2009.5178928.

[12] Keogh, Eamonn, and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping." *Knowledge and information systems* 7.3 (2005): 358-386.

[13] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2, 3, Article 27 (April 2011), 27 pages. <https://doi.org/10.1145/1961189.1961199>