



**INSTITUTO LATINO-AMERICANO DE  
CIÊNCIAS DA VIDA E DA NATUREZA  
(ILACVN)**

**ENGENHARIA FÍSICA**

**NOVO MÉTODO DE RECOMENDAÇÃO DE CLASSIFICADORES DE SÉRIES  
TEMPORAIS COM *META-LEARNING* E EXTRAÇÃO DE CARACTERÍSTICAS  
USANDO CATCH22**

**WALLACE PANNACE PALMA**

Foz do Iguaçu  
2025

**NOVO MÉTODO DE RECOMENDAÇÃO DE CLASSIFICADORES DE SÉRIES  
TEMPORAIS COM META-LEARNING E EXTRAÇÃO DE CARACTERÍSTICAS USANDO  
CATCH22**

**WALLACE PANNACE PALMA**

Trabalho de Conclusão de Curso apresentado ao Instituto latino- americano de ciência da vida e da natureza (ILACVN), como requisito parcial à obtenção do título de Bacharel em Engenharia física.

Orientador: Prof. Willian Zalewski (doutor)

Foz do Iguaçu  
2025

WALLACE PANNACE PALMA

EXTRAÇÃO DE CARACTERÍSTICAS USANDO CATCH22 E RECOMENDAÇÃO DE  
CLASSIFICADORES DE SÉRIES TEMPORAIS COM META-LEARNING



Trabalho de Conclusão de Curso apresentado ao Instituto latino- americano de ciência da vida e da natureza (ILACVN), como requisito parcial à obtenção do título de Bacharel em Engenharia física.

**BANCA EXAMINADORA**

---

Orientador: Prof. Doutor Willian Zalewski  
UNILA

---

Prof. Doutor Joylan Nunes Maciel  
UNILA

---

Prof. Doutor André Gustavo Maletzke  
UNIOESTE

Foz do Iguaçu, \_\_\_\_ de \_\_\_\_\_ de \_\_\_\_.

## **AGRADECIMENTOS**

Agradeço, em primeiro lugar, a Deus, pela saúde e pelas condições que me permitiram seguir nesta jornada. Em segundo lugar, à minha esposa, que esteve ao meu lado desde o início da graduação, cuidando de mim e do nosso filho, tornando possível a conclusão desta etapa tão importante. Também agradeço à minha mãe, que me motivou e tudo que faço é graças ao esforço e amor que ela dedicou em minha vida. Ao meu pai, sou grato pelo suporte financeiro nos primeiros anos da faculdade.

Aos professores que contribuíram para minha formação, deixo meu agradecimento pelo conhecimento compartilhado e pelo incentivo ao nosso crescimento. Um agradecimento especial ao meu orientador, Willian Zalewski, pelas valiosas orientações durante a iniciação científica, monitoria e no TCC, sempre demonstrando profissionalismo e servindo como uma inspiração. Aos colegas de curso, agradeço pela parceria e pelo apoio mútuo nos momentos desafiadores.

Por fim, agradeço à UNILA, instituição que me proporcionou a oportunidade de cursar Engenharia Física com acesso a uma infraestrutura de qualidade e diversas oportunidades.

## RESUMO

O grande volume de dados temporais e sua crescente complexidade exige o desenvolvimento de técnicas avançadas para a sua visualização, agrupamento, classificação e detecção de padrões. A classificação de séries temporais por meio de algoritmos de aprendizado de máquina tem sido amplamente explorada na literatura para atender a tais demandas. Como resultado, verifica-se o aumento da quantidade e da diversidade de algoritmos aplicados na solução de uma grande gama de problemas envolvendo dados temporais. Contudo, a adequada identificação do melhor algoritmo para cada domínio tornou-se um desafio em termos de desempenho e custo de tempo computacional. Nesse contexto, neste trabalho desenvolvemos e avaliamos um novo método de recomendação de classificadores de séries temporais ao extrair atributos das séries e empregar técnicas de *meta-learning*. É proposto a extração de atributos temporais por meio da ferramenta *Catch22* combinado com técnicas de extração de *meta-features* para criar uma base de *meta-knowledge*. A avaliação experimental conduzida neste estudo empregou 112 conjuntos de dados de séries temporais do repositório da *University of California, Riverside*, envolveu a avaliação do método proposto para a recomendação dentre 34 diferentes classificadores de séries temporais e 7 *meta-learners* para o mapeamento da *meta-knowledge*. Por meio dos resultados gerados foi possível constatar que o método proposto possibilitou redução significativa no tempo de construção de *meta-features*, mantendo um desempenho competitivo, em termos de acurácia, aos métodos da literatura. Nos experimentos, dentre os algoritmos testados, o algoritmo *Random Forest* obteve o melhor desempenho como *meta-learner* para o método proposto. Em comparação com a literatura destaca-se o baixo custo computacional do método proposto.

**Palavras-chave:** Séries Temporais; Aprendizado de Máquina; Meta Aprendizado; Extração de Atributos; Regressão.

## RESUMEN

El gran volumen de datos temporales y su creciente complejidad exigen el desarrollo de técnicas avanzadas para su visualización, agrupamiento, clasificación y detección de patrones. La clasificación de series temporales mediante algoritmos de aprendizaje automático ha sido ampliamente explorada en la literatura para atender tales demandas. Como resultado, se observa un aumento en la cantidad y diversidad de algoritmos aplicados en la solución de una amplia gama de problemas que involucran datos temporales. Sin embargo, la adecuada identificación del mejor algoritmo para cada dominio se ha convertido en un desafío en términos de rendimiento y costo de tiempo computacional. En este contexto, en este trabajo desarrollamos y evaluamos un nuevo método de recomendación de clasificadores de series temporales al extraer atributos de las series y emplear técnicas de meta-learning. Se propone la extracción de atributos temporales mediante la herramienta Catch22 combinada con técnicas de extracción de meta-características para crear una base de meta-conocimiento. La evaluación experimental realizada en este estudio empleó 112 conjuntos de datos de series temporales del repositorio de la University of California, Riverside, e involucró la evaluación del método propuesto para la recomendación entre 34 diferentes clasificadores de series temporales y 7 meta-learners para el mapeo del meta-conocimiento. A través de los resultados generados, fue posible constatar que el método propuesto permitió una reducción significativa en el tiempo de construcción de meta-características, manteniendo un rendimiento competitivo, en términos de precisión, con los métodos de la literatura. En los experimentos, entre los algoritmos probados, el algoritmo Random Forest obtuvo el mejor rendimiento como meta-learner para el método propuesto. En comparación con la literatura, se destaca el bajo costo computacional del método propuesto.

**Palabras clave:** Series de Tiempo; Aprendizaje Automático; Meta-aprendizaje; Extracción de Atributos; Regresión.

## ABSTRACT

The large volume of temporal data and its increasing complexity demand the development of advanced techniques for its visualization, clustering, classification, and pattern detection. The classification of time series through machine learning algorithms has been widely explored in the literature to address such demands. As a result, there has been an increase in the quantity and diversity of algorithms applied to solving a wide range of problems involving temporal data. However, the proper identification of the best algorithm for each domain has become a challenge in terms of performance and computational time cost. In this context, in this work, we developed and evaluated a new method for recommending time series classifiers by extracting attributes from the series and employing meta-learning techniques. The extraction of temporal attributes is proposed using the Catch22 tool combined with meta-feature extraction techniques to create a meta-knowledge base. The experimental evaluation conducted in this study used 112 time series datasets from the University of California, Riverside repository and involved assessing the proposed method for recommending among 34 different time series classifiers and 7 meta-learners for mapping the meta-knowledge. Through the generated results, it was possible to verify that the proposed method enabled a significant reduction in the time required to build meta-features while maintaining competitive performance, in terms of accuracy, compared to methods in the literature. In the experiments, among the tested algorithms, the Random Forest algorithm achieved the best performance as a meta-learner for the proposed method. Compared to the literature, the low computational cost of the proposed method stands out.

**Key words:** Temporal Series; Machine Learning; Meta-Learning; Attribute Extraction; Regression.

## LISTA DE TABELAS

<b>Tabela 1</b> – Abreviações utilizadas para cada regressor ( <i>meta-learner</i> ).....	31
<b>Tabela 2</b> – Classificadores utilizados de cada grupo.....	32
<b>Tabela 3</b> – Resultados da avaliação LOO de diferentes regressores nas três bases de <i>meta-knowledge</i> utilizadas.....	34
<b>Tabela 4</b> – Impacto da limitação ao intervalo de 0 a 1 nas predições efetuadas em cada construção de meta-features por meta-learner avaliado.....	36
<b>Tabela 5</b> - Avaliação LOO de cada regressor utilizado como <i>meta-learner</i> na <i>meta-knowledge</i> proposta.....	38
<b>Tabela 6</b> - Impacto da limitação ao intervalo de 0 a 1 nas predições efetuadas com cada regressor usado como <i>meta-learner</i> .....	39
<b>Tabela 7</b> - Resultados do RandomForest como <i>meta-learner</i> na LOO por grupo de classificadores de séries temporais.....	41



## LISTA DE FIGURAS

<b>Figura 1</b> - Representação ilustrativa das componentes de tendência e sazonalidade.....	17
<b>Figura 2</b> - Representação Rice do problema de seleção de algoritmos.....	21
<b>Figura 3</b> - Esquema de aplicação de meta-learning para a recomendação de classificadores de séries temporais.....	25
<b>Figura 4</b> - Fluxograma de trabalho para a recomendação de algoritmos.....	27
<b>Figura 5</b> - Resultados da avaliação LOO em cada base de <i>meta-knowledge</i> para o regressor <i>RandomForest</i> .....	35
<b>Figura 6</b> - Contagem de conjuntos de dados por faixa de tempo consumido para a conversão do conjunto em um vetor de <i>meta-features</i> na abordagem Catch22+mfe.....	37
<b>Figura 7</b> - Avaliação LOO de cada regressor utilizado como <i>meta-learner</i> na <i>meta-knowledge</i> proposta.....	38
<b>Figura 8</b> - Desempenho das predições do <i>RandomForest</i> como <i>meta-learner</i> por algoritmo candidato.....	40
<b>Figura 9</b> - Avaliação LOO do <i>RandomForest</i> como <i>meta-learner</i> por conjunto de dados.....	42
<b>Figura 10</b> - Resultados do <i>RandomForest</i> como <i>meta-learner</i> por tipo de conjunto de dados.....	43
<b>Figura 11</b> - Resultados do <i>RandomForest</i> como <i>meta-learner</i> por número de rótulos (classes) presentes nos conjuntos de dados.....	44
<b>Figura 12</b> - Resultados do <i>RandomForest</i> como <i>meta-learner</i> por comprimento ( <i>length</i> ) da ST dos conjuntos de dados.....	44
<b>Figura 13</b> - Resultados do <i>RandomForest</i> como <i>meta-learner</i> por número de amostras ( <i>size</i> ) nos conjuntos de dados.....	45

## DEFINIÇÕES E SIGLAS

ASP	<i>Algorithm selection problem</i>
BOSS	Algoritmo Bag-ofSFA-Symbols
Catch 22	Algoritmo <i>The Canonical Time Series Characteristics</i>
cBOSS	<i>Contractible BOSS</i>
CIF	<i>The Canonical Interval Forest</i>
CNN	<i>Convolutional Neural Networks</i>
DrCIF	<i>The Diverse Representation CIF</i>
EE	Algoritmo <i>Elastic Ensemble</i>
HC1	Algoritmo HIVE-COTE v1
HC2	Algoritmo HIVE-COTE v2.0
HCTSA	<i>Highly comparative time-series analysis</i>
Hydra	<i>Hybrid Dictionary ROCKET Architecture</i>
Hydra-MR	MultiROCKET-Hydra
InceptionT	Algoritmo <i>InceptionTime</i>
KNN	<i>K-Nearest Neighbors</i>
MAE	Erro médio absoluto ( mean absolute error)
MAPE	Erro médio absoluto percentual (mean absolute percentage error)
MFE	Extrator de <i>meta-features</i>
Mini-R	Mini-ROCKET
MLP	<i>Multi-layer Perceptron</i>
MSquareE	Erro médio quadrático ( <i>mean squared error</i> )
Multi-R	Multi-ROCKET
PF	Algoritmo <i>Proximity Forest</i>
RDST	Algoritmo <i>Random Dilated Shapelet Transform</i>
Resnet	<i>Residual Network</i>
RISE	<i>Random Interval Spectral Ensemble</i>
ROCKET	<i>Random Convolutional Kernel Transform</i>
RSF	<i>The Generalised Random Shapelet Forest</i>
RSTSF	<i>Randomised Surpevised Time Series Forest</i>
ShapeDTW	Algoritmo <i>Shape Dynamic Time Warping</i>
STC	<i>The Shapelet Transform Classifier</i>
SVR	<i>Support Vector Machine</i>

TDE	<i>Temporal Dictionary Ensemble</i>
Tree	<i>Decision Tree</i>
TS-CHIEF	<i>Time Series Combination of Heterogeneous and Integrated Embedding</i>
Forest	<i>Random Forest</i>
TSFresh	<i>Time Series Features Extraction based on Scalable Hypothesis Tests</i>
TSF	<i>Algoritmos Time Series Forest</i>
UCR	<i>University of California, Riverside</i>
WEASEL	<i>Algoritmo Word Extraction for Time Series Classification</i>
WEASEL-D	<i>Algoritmo Word Extraction for Time Series Classification com dilatação</i>
1NN-DTW	<i>Algoritmo 1 Nearest Neighbor Dynamic Time Warping</i>
LOO	<i>Leave-one-out cross validation</i>
UNILA	<i>Universidade Federal da Integração Latino-Americana</i>
ASP	<i>Problema de seleção de algoritmos</i>
<i>Meta-features</i>	No contexto de <i>meta-learning</i> aplicado a ASPs, características que descrevem conjuntos de dados.
<i>Meta-target</i>	No contexto de <i>meta-learning</i> aplicado a ASPs, caracterização dos algoritmos candidatos no espaço do desempenho.
<i>Meta-learner</i>	No contexto de <i>meta-learning</i> aplicado a ASPs, Algoritmo que modela a relação entre as características de um conjunto de dados ( <i>meta-features</i> ) e os algoritmos candidatos.

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 HIPÓTESE .....	13
1.2 OBJETIVOS.....	14
1.3 ORGANIZAÇÃO DO TRABALHO.....	14
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>15</b>
2.1 APRENDIZADO DE MÁQUINA.....	15
2.2 CLASSIFICAÇÃO DE SÉRIES TEMPORAIS.....	16
2.3 <i>META-LEARNING</i> .....	20
2.4 <i>META-FEATURES</i> .....	22
2.4.1 Caracterização Direta .....	22
2.4.3 Caracterização por Propriedades de Modelos .....	23
2.4.4 Marcadores.....	23
2.4.5 Medidas Baseadas Em Informações Estruturais.....	24
2.5 <i>META-TARGET</i> .....	24
2.6 <i>META-LEARNER</i> .....	24
2.7 TRABALHOS RELACIONADOS.....	26
<b>3 MÉTODO</b> .....	<b>27</b>
3.1 CONJUNTOS DE DADOS, TRANSFORMAÇÃO E EXTRAÇÃO DE <i>META-FEATURES</i> .....	28
3.1.1 Conjuntos De Dados.....	28
3.1.2 Transformação De Séries Temporais Para Um Vetor De Atributos Com <i>Catch22</i> ....	28
3.1.3 Extração De <i>Meta-Features</i> Com <i>Python Meta-Feature Extractor</i> .....	29
3.2 <i>DELINEAMENTO DOS EXPERIMENTOS</i> .....	29
3.2.1 Comparação Com A Literatura .....	30
3.2.2 Avaliação Da Abordagem Proposta Em 34 Classificadores De Séries Temporais...31	
<b>4 RESULTADOS E DISCUSSÃO</b> .....	<b>33</b>
4.1 RESULTADOS DA COMPARAÇÃO COM A LITERATURA.....	33
4.2 RESULTADOS DA ABORDAGEM PROPOSTA COM 34 CLASSIFICADORES DE SÉRIES TEMPORAIS.....	37
4.2.1 Resultados para o <i>Meta-Learner RandomForest</i> .....	39
4.2.1.1 <i>Perspectiva de classificadores candidatos</i> .....	39

4.2.1.2 <i>Perspectiva de conjuntos de dados</i> .....	41
4.3 CONTRIBUIÇÕES E LIMITAÇÕES.....	45
<b>5 CONCLUSÃO E TRABALHOS FUTUROS</b> .....	47
<b>REFERÊNCIAS</b> .....	48

**APÊNDICES**

<b>APÊNDICE A</b> – Tempo Consumido Para a Conversão Dos 112 Conjuntos De Dados Para Meta-Examples Utilizando Catch22 E MFE.....	55
<b>APÊNDICE B</b> - Resultados Da Seção 4.2 Para Os Regressores Utilizados.....	56
<b>APÊNDICE C</b> - Códigos e Resultados Disponibilizados .....	57

## 1 INTRODUÇÃO

O grande volume de dados gerados diariamente em diversos setores impulsiona a mineração de dados de séries temporais como um campo de pesquisa cada vez mais relevante [1]. Devido à grande variação em domínios de problemas que caracteriza a pesquisa em classificação de séries temporais [16], a compreensão de fenômenos e eventos temporais é vital para uma ampla gama de domínios como finanças, saúde, meteorologia, varejo, indústria e pesquisa científica [2,17]. Paralelamente a esses aspectos, a crescente complexidade de séries temporais, como no aumento do período de observações combinado a não linearidade dos dados, a correlação entre as observações e a relevância da ordem de observações, exige o desenvolvimento de técnicas específicas em tarefas como previsão, visualização, agrupamento, classificação e detecção de padrões anômalos [2]. Em especial, para a tarefa de classificação, diversos algoritmos de Aprendizado de Máquina (AM) têm sido propostos para atender à crescente demanda por soluções automatizadas em diferentes aplicações que envolvem dados de séries temporais [2].

Os algoritmos de AM apresentam uma otimização intrínseca que o torna especialmente adequado para determinadas tarefas. O desempenho de um algoritmo em um domínio ou atividade específica está diretamente relacionada à congruência entre o viés inerente ao algoritmo e as características do problema em questão [4, 5]. Cada algoritmo tem a possibilidade de apresentar um desempenho melhor em um limitado número de atividades, sendo este fenômeno conhecido como complementaridade de desempenho [4, 5]. Esse problema é também evidenciado pelo teorema “*no free lunch*” (sem almoço grátis) [3, 4], o qual baseia-se no fato de que nenhum algoritmo pode desempenhar melhor que todos os outros algoritmos em todos os problemas existentes.

No contexto de algoritmos de AM para a classificação de séries temporais, apesar da existência de uma taxonomia, na qual cada categoria deveria ser mais adequada a um tipo de problema ou base de dados, a eficácia dessa associação ainda não foi comprovada empiricamente de forma conclusiva [6]. Deste modo, diante da quantidade e variabilidade de opções disponíveis, a recomendação de um algoritmo adequado para um dado problema representa um desafio crescente na aplicação de algoritmos de AM [6]. Na literatura de AM, esse problema é formalmente conhecido como *algorithm selection problem (ASP)* [7,8]. Nesse contexto, métodos baseados em *meta-learning* têm se mostrado eficazes na resolução de problemas de ASPs em aprendizado

supervisionado [9].

O *meta-learning*, frequentemente descrito como 'aprender a aprender', consiste em uma área de estudo que tem por objetivo explorar conhecimentos prévios de aprendizagem para aprender novas tarefas. Em especial, abordagens baseadas em *meta-learning* podem ser aplicadas na tarefa de recomendação de classificadores, em contraponto com abordagens convencionais como tentativa e erro, análise teórica ou conhecimento de especialista [10]. Mais especificamente, estratégias que utilizam *meta-learning* possibilitam superar as desvantagens de abordagens convencionais, como custo computacional e subjetividade para a seleção de algoritmos [11,12].

No contexto de séries temporais, classificadores considerados como estado-da-arte, em geral, são computacionalmente caros, o que faz com que a avaliação de todos seja inacessível [13]. Por exemplo, diversos algoritmos de classificação de séries temporais utilizam cálculos de medida de similaridade entre todos os pares de séries e/ou subséries, conseqüentemente tornando o custo computacional quadrático em relação ao número e ao comprimento das séries [14]. A recomendação de algoritmos é um desafio que tem sido explorado recentemente no campo de classificação de séries temporais [13], tendo em vista a literatura de *meta-learning* e o estado-da-arte em descrever conjuntos através de atributos (*meta-features*) que foi projetado para dados sem ordem (sem dependência temporal) [13]. Na literatura existem trabalhos que utilizam recomendações por meio de *meta-learning* para classificadores de séries temporais. No entanto, uma abordagem que emprega transformações de séries para atributos para capturar padrões temporais em abordagens de *meta-learning*, ainda não foi explorada. Nesse contexto, neste trabalho, é realizada a proposição de um método para a obtenção de *meta-features* que absorvam a informação temporal das séries e incorpore-as ao processo de recomendação de classificadores de séries temporais.

## 1.1 HIPÓTESE

A extração de *meta-features* combinada a transformação do algoritmo *Catch22* [2] permite construir uma base de *meta-knowledge* para treinar regressores capazes de prever a acurácia de classificadores em novos conjuntos de dados, viabilizando um sistema de recomendação de modelos adequados para a classificação de séries temporais.

## 1.2 OBJETIVOS

O objetivo é desenvolver e avaliar um método computacional que, empregando *meta-learning*, recomenda os algoritmos de aprendizado de máquina mais adequados para a classificação de séries temporais, buscando transformadores para extrair características relevantes que considerem a informação temporal contida nas séries.

## 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado do seguinte modo. No Capítulo 2, é apresentada a fundamentação teórica, abordando conceitos como aprendizado de máquina, classificação de séries temporais e *meta-learning*. No Capítulo 3 é descrito o método adotado, incluindo os conjuntos de dados utilizados, a transformação das séries temporais, a extração de *meta-features* e o delineamento dos experimentos. No Capítulo 4, são apresentados e discutidos os resultados, tanto em relação à comparação com a literatura quanto à aplicação da abordagem em diversos classificadores de séries temporais. Por fim, o Capítulo 5 traz as considerações finais, destacando as contribuições e limitações do estudo.



## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos que possibilitam a compreensão da contribuição deste trabalho. São apresentados conceitos de aprendizado de máquina, classificação de séries temporais e definições relevantes para o entendimento de *meta-learning* e sua aplicabilidade em problemas de seleção de algoritmos.

### 2.1 APRENDIZADO DE MÁQUINA

O Aprendizado de Máquina (AM) é uma área da Inteligência Artificial que tem como objetivo o desenvolvimento de algoritmos e sistemas computacionais capazes de adquirir conhecimento de modo automático a partir de experiências prévias [62, 63, 19]. Em sistemas computacionais AM, a experiência assume a forma de dados, e a tarefa principal do AM consiste em induzir modelos a partir desses dados, de modo que o sistema seja capaz de fazer previsões em novas observações.

Desse modo, a aplicação de sistemas de AM baseia-se na existência de conjuntos de dados compostos por exemplos  $(x_i)$ , onde cada exemplo possibilita a representação de atributos ou características úteis para a descrição de um objeto ou evento. Os valores associados a estes atributos são chamados valores de atributos, podendo ser descrições (discretas), números (discretos ou contínuos) [19]. A tabela ou espaço composto por atributos é geralmente chamado de espaço de atributos, tabela atributo-valor ou espaço de entradas. Nessa estrutura, cada ponto no espaço corresponde a um vetor de posição. Desse modo, um exemplo também é chamado de vetor de características.

Formalmente, seja  $D = \{x_1, x_2, \dots, x_m\}$  um conjunto de dados contendo  $m$  exemplos, onde cada exemplo é descrita por  $d$  atributos, cada amostra  $x_i = (x_{i1}; x_{i2}; \dots; x_{id}) \in X$  é um vetor de  $d$  dimensões no espaço amostral  $X$ , onde  $d$  é a dimensionalidade da amostra  $x_i$ . e  $x_{ij}$  é o valor do  $j$ -ésimo atributo da amostra  $x_i$  [19].

Treinamento ou aprendizado é o nome dado ao uso de algoritmos de aprendizado de máquina para a construção de um modelo baseado em dados. Os dados utilizados na indução do modelo são chamados de conjunto de treinamento. Um modelo induzido corresponde às regras subjacentes sobre os dados, também chamado de

hipótese ( $h(X)$ ), e as regras subjacentes reais são chamadas de fatos ( $f(X)$ ), de modo que o aprendizado de máquina tem por objetivo encontrar ou aproximar  $f(X)$  através de  $h(X)$ .

As tarefas de aprendizado podem ser divididas em preditivas e descritivas. As preditivas fazem parte do paradigma de aprendizado supervisionado, onde a saída (rótulo) é conhecida. Tarefas descritivas fazem parte do paradigma de aprendizado não supervisionado, onde o objetivo é explorar ou descrever um conjunto de dados sem fazer uso do rótulo [63].

O resultado de cada amostra é comumente chamado de rótulo. Denota-se o  $i$ -ésimo exemplo como  $(x_i, y_i)$ , onde  $y_i \in Y$  é o rótulo da amostra  $x_i$ , e  $Y$  é o conjunto de todos os rótulos, também chamado de espaço de saída ou espaço de rótulos.

Se o modelo busca prever um resultado discreto, é chamado de problema de classificação. Quando o resultado da predição é uma resposta contínua, é chamado de problema de regressão. Em um problema de predição busca-se estabelecer um mapeamento  $f: X \rightarrow Y$  do espaço amostral  $X$  para o espaço de rótulos  $Y$  através do aprendizado obtido de um conjunto de treinamento  $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ .

Para avaliar o desempenho de um modelo induzido a partir do treinamento, é necessário utilizar um conjunto de exemplos não fornecidos no conjunto de treinamento. O ato de fazer previsões com um modelo de aprendizado é chamado teste. Os exemplos a serem utilizados para o teste são chamados de exemplos de teste, que através do modelo induzido podemos prever  $y = f(x)$ .

A habilidade de um modelo prever corretamente novos exemplos é chamada de habilidade de generalização. Nesse sentido, é necessário que o conjunto de treinamento seja representativo do espaço amostral, possibilitando a utilização de modelos treinados para a inferência de conhecimento em novas amostras [19].

## 2.2 CLASSIFICAÇÃO DE SÉRIES TEMPORAIS

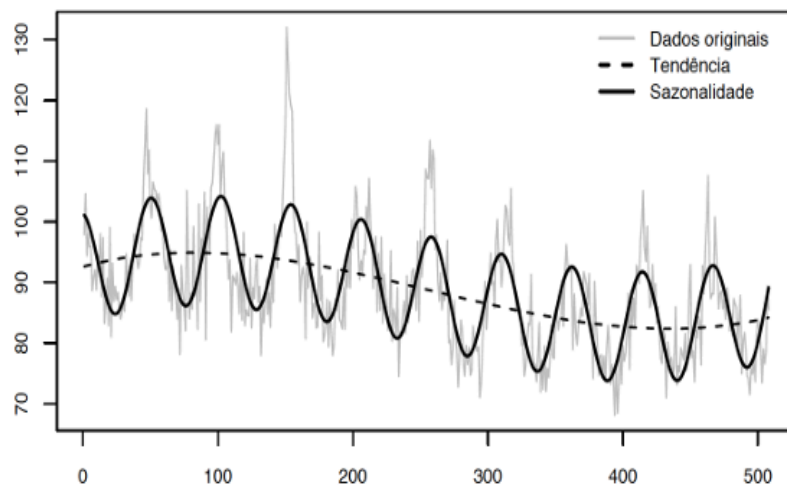
Uma Série Temporal (ST) é uma sequência ordenada de valores reais, cuja estrutura sequencial se diferencia de outros tipos de dados. Essa característica exige métodos específicos para a mineração de ST, as técnicas tradicionais de AM nem sempre são adequadas para considerar a dependência temporal presente nesses dados.

Uma série temporal  $T = \{t_1, \dots, t_i, t_j, \dots, t_m\}$  é composta de  $m$  valores ordenados,

$m \geq 2$ , tal que se  $i < j$ ,  $t_i$  ocorre cronologicamente antes que  $t_j$ .

Para a compreensão e análise dos eventos representados por uma ST, se faz útil a decomposição da série em um conjunto finito de componentes independentes. Sendo as principais componentes tendência, sazonalidade e resíduo. Portanto, um evento observado em  $t_j$  de uma  $T$  pode estar influenciado por uma ou mais dessas componentes.

**Figura 1** - Representação ilustrativa das componentes de tendência e sazonalidade.



Fonte: Figura presente em [64].

Tendência corresponde ao movimento dominante em uma série temporal, exercendo influência sobre as observações por longos períodos de tempo, representada por uma linha tracejada na Figura 1. Sazonalidade corresponde ao comportamento que tende a se repetir em uma ST em diferentes períodos de tempo, representando oscilações ao longo do componente tendência. Resíduo corresponde aos eventos aleatórios característicos do fenômeno observado ou gerados pelo observador [65].

Séries temporais podem ser univariadas, apenas uma variável é observada a cada ponto no tempo, ou multivariada, múltiplas variáveis são observadas a cada ponto no tempo [16]. A tarefa de classificação de séries temporais envolve a implementação de um modelo usando as séries temporais para uma resposta variável discreta [14]. Formalmente, seja  $T$  o conjunto de todas as séries possíveis de um domínio e  $C = \{c_1, \dots, c_w\}$  um conjunto de  $w$  classes, tal que:

$$\forall T_i \in T: ((T_i \in c_1) \vee \dots \vee (T_i \in c_w)) \wedge (T_i \in c_j \rightarrow T_i \notin c_k, j \neq k)$$

Um classificador de séries temporais (TSC) consiste em uma função  $f$  que

permite mapear uma série  $T_i \in T$  para uma classe  $c \in C$ :

$$f: T \rightarrow \{c_1, \dots, c_w\}$$

Diversos algoritmos para a classificação de STs têm sido propostos nos últimos anos. Em MIDDLEHURST [16] é proposta uma categorização contendo 8 grupos que organiza os classificadores de séries temporais pelo tipo de características que cada classificador utiliza e o modo que as aplica para discriminar entre diferentes rótulos. Em geral, espera-se que métodos em cada categoria sejam especialmente adequados para um tipo de problema ou *dataset*, porém esta correlação direta não é confirmada [13]. A seguir, é apresentada uma breve descrição de cada grupo:

- Baseados em distância

A classificação é baseada no uso de uma medida de distância entre séries integralmente [16]. Exemplos: *Elastic Ensemble* (EE), *Proximity Forest* (PF), *Shape Dynamic Time Warping* (ShapeDTW) e *1-Nearest Neighbor Dynamic Time Warping* (1NN-DTW).

- Baseados em características

Estes classificadores extraem características globais das séries temporais e as utilizam como atributos em classificadores tradicionais. Em geral, essas características resumem cada série transformando-as para vetores de atributos [16]. Exemplos: *The Canonical Time Series Characteristics* (Catch22), *Time Series Features Extraction based on Scalable Hypothesis Tests* (TSFresh) e sua implementação *FreshPRINCE*, assinaturas generalizadas (*Signature*).

- Baseados em intervalos

Os algoritmos utilizam características derivadas de intervalos dependentes de fase extraídos de um *offset* fixado e descritores estatísticos são computados destes intervalos em um conjunto de *pipelines*. Geralmente é utilizada alguma forma de seleção randômica na escolha de intervalos combinando características de múltiplos intervalos aleatórios [16]. Exemplos: *Time Series Forest* (TSF), *Random Interval Spectral Ensemble*

(RISE), *Randomised Surpevised Time Series Forest* (RSTSF), *The Canonical Interval Forest* (CIF) e *The Diverse Representation CIF* (DrCIF).

- Baseados em *shapelets*

*Shapelets* são subséries, independentes de fase que possibilitam a discriminação entre classes baseando-se em sua presença ou ausência [16]. Exemplos: *The Shapelet Transform Classifier* (STC), *The Generalised Random Shapelet Forest* (RSF), *Multiple Representaion Squence Learner* em sua implementação MrSQM, *Random Dilated Shapelet Transform* (RDST).

- Baseados em dicionários

Nessa abordagem as séries são divididas em intervalos, denominados janelas, as quais são convertidas em uma pequena sequência de símbolos discretos. As características para os classificadores são compostas por histogramas de contagens de padrões repetidos através da extração de subséries, independentes de fase [16]. Exemplos: *Bag-ofSFA-Symbols* (BOSS) e *Contractable BOSS* (cBOSS), *Temporal Dictionary Ensemble* (TDE), *Word Extraction for Time Series Classification* (WEASEL) e sua implementação WEASEL com dilatação (WEASEL-D).

- Baseados em convoluções

Estes classificadores usam convoluções com subséries chamadas *kernels*. Cada *kernel* é operado com uma série temporal pelo produto escalar em uma janela deslizante, criando um mapa de ativação seguido de operações de *pool* que buscam extrair atributos relevantes de cada operação. Estes atributos são concatenados para formar um único vetor de recursos para cada série temporal [16]. Exemplos: *Random Convolutional Kernel Transform* (ROCKET), *Mini-ROCKET* (Mini-R) e *Multi-ROCKET* (Multi-R), *Hybrid Dictionary ROCKET Architecture* (Hydra) e *MultiROCKET-Hydra* (Hydra-MR).

- Baseados em *deep learning*

Classificação com uso de redes neurais profundas. Exemplos: *InceptionTime*

(InceptionT), *Residual Network* (ResNet) e *Convolution Neural Networks* (CNN).

- Abordagens híbridas

São métodos que constroem modelos a partir da combinação de diferentes descritores de características e algoritmos de classificação de ST. Em geral, são os algoritmos com maior acurácia [16]. Exemplos: HIVE-COTE v1 (HC1) e v2.0 (HC2) e o *Time Series Combination of Heterogeneous and Integrated Embedding Forest* (TS-CHIEF).

## 2.3 META-LEARNING

*Meta-learning* é uma área do AM, é o estudo de métodos que exploram metachecimento para construir ou melhorar modelos e soluções por meio da adaptação de processos de aprendizado de máquina e mineração de dados [21, 44, 63], comumente descrito como a tarefa de aprender a aprender. Os principais objetivos são fornecer uma recomendação automática na seleção de modelos para um usuário não especialista em AM [9,22], e economizar tempo ao reduzir o número de algoritmos a serem testados. Busca-se atingir ambos objetivos com perdas mínimas na qualidade dos resultados obtidos quando comparados a real aplicação dos modelos [6].

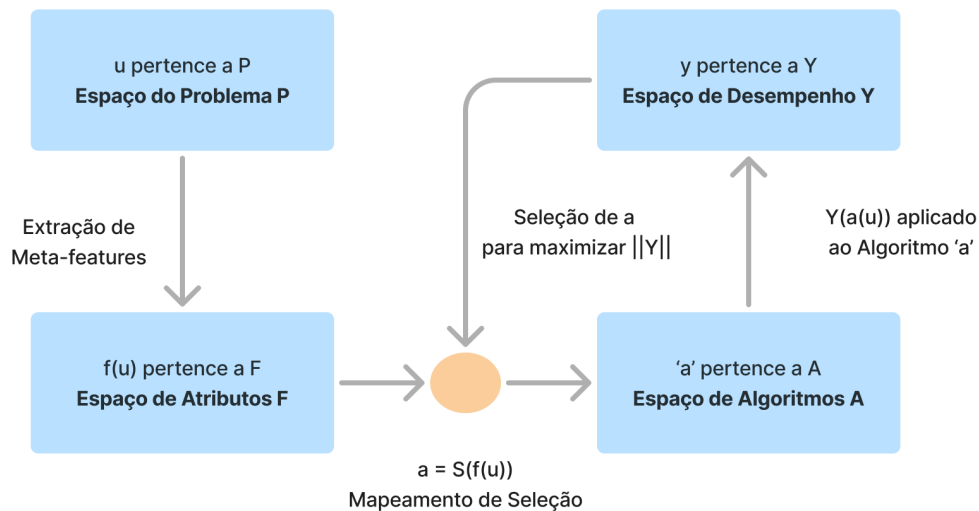
A recomendação fornecida por sistemas de meta-learning são construídas ao explorar conhecimento que relaciona as características de um conjunto de dados com o desempenho de algoritmos de aprendizado de máquina, este conhecimento é chamado de *meta-knowledge* [6]. Através do conhecimento acumulado da aplicação de algoritmos em tarefas similares presente no *meta-knowledge*, os algoritmos de *meta-learning* utilizam os resultados encontrados em tarefas já conhecidas para inferir e recomendar um conjunto de algoritmos potencialmente adequados para uma tarefa específica [23].

O *meta-knowledge* pode ser extraído de diversos modos. No trabalho de FACELI (2021) [63] é apresentada uma separação em três principais maneiras de extração de informações a cerca de conjuntos de dados (*meta-features*) discutidas na seção 2.4. No trabalho de ABANDA (2022) [13] são discutidos 5 tipos de representações do desempenho de classificadores nos conjuntos de dados (*meta-target*) e a hierarquia de inferência entre estas.

O problema de seleção de algoritmo, *algorithm selection problem* (ASP), é

definido como “para um dada uma instância de problema  $u \in P$  com atributos  $f(u) \in F$ , encontre o mapa de seleção  $S(f(u))$  no espaço de algoritmos  $A$ , tal que o algoritmo selecionado  $a \in A$  maximiza o mapeamento do desempenho  $y(a(u)) \in Y$ ” [9], como demonstrado na Figura 2. O espaço do problema  $P$  é um componente altamente dimensional com cardinalidade infinita dado a impossibilidade de incluir todos os problemas em qualquer sistema de recomendação de algoritmos, bem como no espaço de algoritmos  $A$  possui normalmente alta cardinalidade devido a impossibilidade de avaliar todos os algoritmos candidatos possíveis [28].

**Figura 2-** Representação Rice do problema de seleção de algoritmos.



Fonte: Adaptado de [29].

Uma das principais vantagens de um modelo de recomendação de algoritmos baseado em *meta-learning* é que uma vez construído, ele pode ser adaptado e aprimorado com novas experiências através da adição de mais algoritmos e problemas [9,30].

Comparando *meta-learning* a outras formas de solução de ASPs, a abordagem de tentativa e erro não é possível em diversos casos práticos devido ao alto custo computacional, especialmente para grandes conjuntos de dados [9]. A ênfase de métodos teóricos visa obter conhecimento sobre a utilidade de algoritmos através da avaliação de seu viés representativo, porém, a aplicabilidade de todos os classificadores não pode ser analisada teoricamente [9]. Profissionais com experiência em lidar com problemas similares podem fornecer assistência na seleção de algoritmos adequados ao problema, contudo, adquirir o conhecimento profissional é geralmente oneroso e/ou não rapidamente

disponível na maioria das situações, podendo ser propenso a preconceitos ou preferências pessoais [8].

A principal diferença entre os métodos para a resolução de ASPs está na construção do *meta-knowledge* e do *meta-learner* escolhido [9]. No *meta-knowledge*, na variação presente no grupo de *meta-features* (atributos extraídos do espaço de problemas) utilizado e na variação de procedimentos escolhidos para o mapeamento do espaço de atributos para o espaço de desempenho. Por fim, no *meta-learner*, varia o tipo de método utilizado para estimar o desempenho de algoritmos candidatos em conjuntos de dados.

## 2.4 META-FEATURES

A coleção de características medidas e extraídas de um conjunto de dados que possibilitem a estimativa do desempenho de algoritmos de aprendizagem em determinados problemas é denominado *meta-features* ou caracterização dos dados [9]. O propósito das *meta-features* consiste em buscar a caracterização uniforme de diferentes conjuntos de dados para que seja possível a comparação entre estes e o desempenho de algoritmos por meio dos mesmos atributos.

Tendo em vista os objetivos em soluções de ASP, é necessário que as *meta-features* tenham três principais propriedades [9]:

1- As medidas devem ser relevantes e uniformemente computáveis para uma grande gama de problemas em um domínio particular.

2- As medidas devem reduzir o custo computacional de modo que haja economia de recursos ao usar o recomendador se comparado ao teste direto dos algoritmos.

3- Deve haver uma relação intrínseca entre as *meta-features* e o desempenho de algoritmos.

Segundo FACELI (2021) [63], as *meta-features* usadas para a seleção de classificadores podem ser agrupadas em três subcategorias. Cada grupo representa um número de conjuntos de medidas que compartilham alguma similaridade.

### 2.4.1 Caracterização Direta

Este grupo consiste em medidas de estatísticas descritivas ou medidas através da teoria da informação extraídas dos conjuntos de dados para descrevê-los. Representa



o maior e mais diversificado grupo de *meta-features*, conhecido como medidas gerais e possuem um baixo custo computacional e é o grupo mais utilizado em seleção de classificadores [11, 31, 32]. As medidas simples estão associadas a informações básicas como o número de amostras, atributos ou classes. As medidas estatísticas possibilitam a extração de informações das distribuições de atributos numéricos como dispersão e tendência. Medidas de informações teóricas são baseadas em entropia que descreve a variabilidade e redundância de atributos discretos [33].

Nessa categoria agrega-se as medidas que possibilitam a descrição da distribuição espacial dos dados e a dificuldade em um conjunto de dados por meio de sua complexidade geométrica. Essas medidas caracterizam o tamanho e forma aproximados do limite de decisão que separa as classes [34, 35]. Neste grupo estão inclusos as medidas baseadas em características, medidas de linearidade, medidas de vizinhança, medidas de dimensionalidade e medidas de balanceamento de classes.

#### 2.4.2 Caracterização por Propriedades de Modelos

As medidas desse grupo fornecem uma caracterização indireta do conjunto de dados e são calculados ao induzir um modelo para extrair informações sobre as estruturas subjacentes dos dados e as propriedades do modelo são usados como *meta-features* [36]. Por exemplo, ao utilizar uma árvore de decisão, a profundidade e a quantidade de nós podem ser utilizadas como medidas para descrever o conjunto de dados.

#### 2.4.3 Marcadores

Nessa abordagem, as medidas são baseadas no desempenho de algoritmos simples e rápidos para prever o desempenho de algoritmos computacionalmente mais complexos e custosos. Chama-se tais medidas de marcadores (*landmarks*) e há duas formas de obtê-las: (1) ao executar uma versão simplificada dos classificadores candidatos (redução de algoritmo) [35] ou (2) ao executar os algoritmos dos classificadores em um subconjunto dos dados (redução do conjunto de dados) [37]. Assim, utiliza-se a informação do desempenho obtido na redução (marcador) para a recomendação dos algoritmos apropriados [39]. Como desvantagem, quanto maior for a redução utilizada, maior pode ser a perda da relação entre o marcador e o algoritmo

original [38]

Abordagens baseadas em marcadores levam em consideração que cada problema possui características que o associam a uma área de domínio de algoritmos específicos. Desse modo, os algoritmos candidatos que possuem o viés indutivo mais próximo aos marcadores e que apresentem os melhores desempenho em um dado problema, tendem a ser os mais adequados para recomendação. Na construção de recomendadores baseados em marcadores busca-se a utilização de marcadores com baixo custo computacional e com distintos tipos de viés de indução [40].

## 2.5 META-TARGET

Em *meta-learning* o espaço de resultado  $Y$  é representado por *meta-target*, ou seja, este corresponde ao tipo de saída que o sistema produz na forma de desempenho estimado de algoritmos candidatos para qualquer dado problema. Alguns exemplos de formas de *meta-targets* são: acurácia de classificadores, *ranking* completo, *top-k ranking* (sendo  $k$  um inteiro positivo de interesse), melhor conjunto e melhor classificador.

A escolha da forma de *meta-target* a ser considerada para a recomendação afeta também a escolha do *meta-learner*. Na maioria dos estudos são utilizados *rankings* como *meta-target* [45, 46, 31], porém a inferência hierárquica entre as formas de *meta-target* é possível [13]. A *meta-target* escolhida será também a forma da saída do sistema de recomendação.

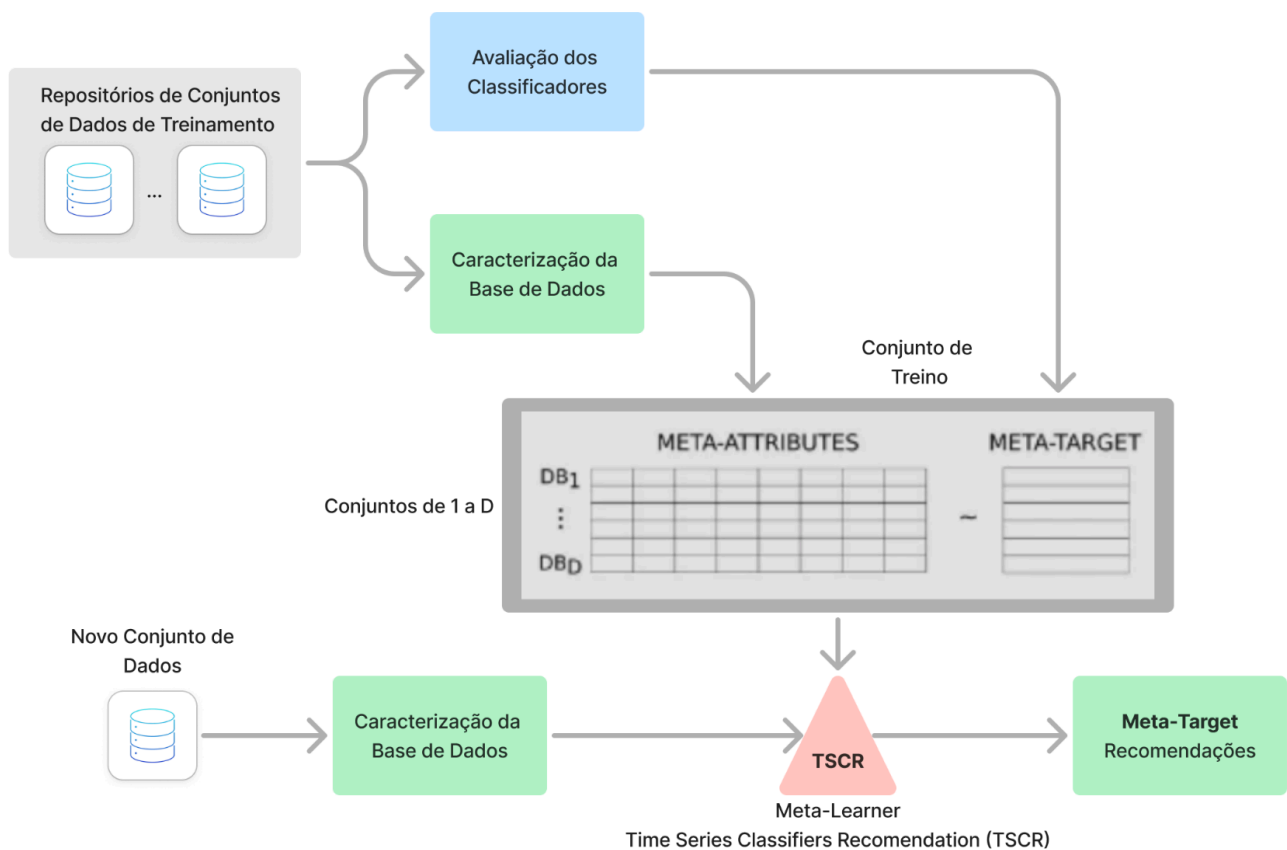
## 2.6 META-LEARNER

O algoritmo que modela a relação entre as características de um conjunto de dados (*meta-features*) e os algoritmos candidatos é denominado *meta-learner*. Inicialmente os *meta-learners* utilizados foram os indutores de regras, o primeiro utilizado foi o algoritmo de árvore de decisão [47] como recomendador (*meta-learner*). Posteriormente foram aplicados algoritmos baseados em instâncias como o *k-Nearest Neighbors* (kNN), vantajosos quando o conjunto de treino disponível é pequeno. Essa característica faz com que haja dificuldade de induzir modelos que sejam generalistas e produzam com constância limites que separam os casos em que cada recomendação é adequada [48]. Os métodos baseados em instâncias fornecem flexibilidade e facilidade de expansão para o sistema de *meta-learning*, esta abordagem é usada na maioria dos

estudos para a recomendação de classificadores [9].

Algoritmos de regressão como *meta-learner* podem ser aplicados para aprender a relação entre as *meta-features* e o desempenho dos classificadores candidatos [24, 45, 49, 50]. Nesta abordagem, é possível a predição de métricas específicas (acurácia, f1, precision, *recall*) de cada classificador candidato independentemente. A desvantagem é que o regressor utilizado como *meta-learner* deve ser treinado separadamente para cada algoritmo candidato e requer o treinamento cada vez que novas combinação de *meta-features* e *meta-target* de um conjunto de dados (*meta-examples*) forem adicionados ao *meta-knowledge* [9]. Na Figura 2 é apresentado um diagrama de como seria a criação e aplicação de *meta-learning* de acordo com o discutido até aqui.

**Figura 3** - Esquema de aplicação de *meta-learning* para a recomendação de classificadores de séries temporais.



Fonte: Adaptado de [13].

## 2.7 TRABALHOS RELACIONADOS

A aplicação mais comum do *meta-learning* é na recomendação de algoritmos de classificação [24, 25]. Contudo, especificamente para a classificação de séries temporais, ainda que com resultados promissores, poucos estudos têm abordado a tarefa de recomendação [26, 27]. Nesta seção são comentados três trabalhos, sendo estes os trabalhos mais recentes presentes na literatura que envolvem um sistema de recomendação utilizando *meta-learning* no contexto de classificação de séries temporais.

Em ABANDA (2023) [13], os autores introduzem um sistema de recomendação utilizando marcadores (*landmarkers*) como *meta-features*. O método proposto supera abordagens padrão de *meta-learning* para dados não temporais, além de explorar a inferência hierárquica de *meta-targets*, demonstrando que um único modelo pode inferir múltiplos tipos de *meta-targets* com resultados competitivos.

Já em MU (2023) [67], é proposta uma abordagem chamada TSC-AutoML, que utiliza *meta-learning* para resolver o problema de seleção combinada de algoritmos e otimização de hiperparâmetros em classificação de séries temporais (TSC). Empregando aprendizado por reforço para extrair conhecimento de tarefas históricas e um *meta-learner* com estratégia de "*warm start*" para seleção e ajuste automáticos de algoritmos e hiperparâmetros.

Em NARWARIYA (2020) [66], os autores aplicam *meta-learning* baseado em gradientes para classificação de séries temporais com poucos exemplos (*few-shot* TSC). O método utiliza uma rede neural residual treinada em tarefas de diversos domínios, com uma função de perda baseada em *triplet loss*.

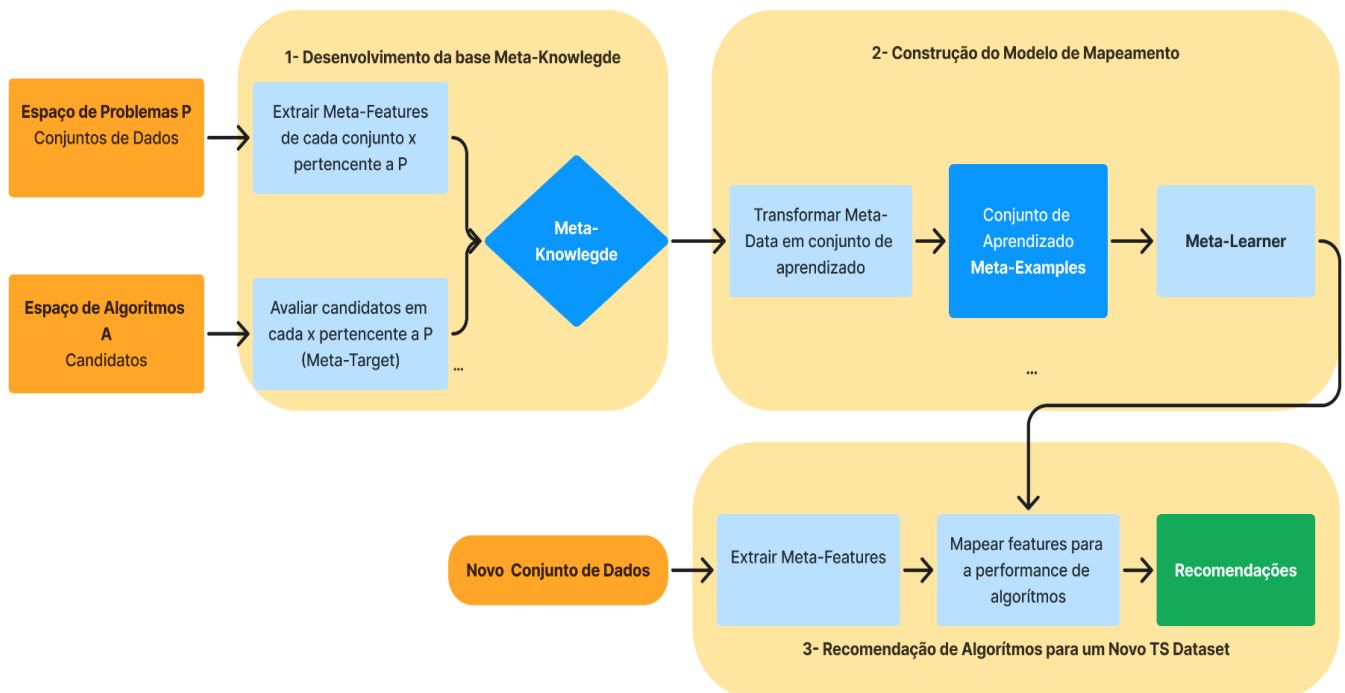
Nesse contexto, o método apresentado neste trabalho se assemelha a [13]. Contudo, não há registros na literatura de abordagens que explorem a inserção de informações temporais em técnicas de *meta-learning* através de transformações para atributos voltadas à recomendação de classificadores de STs. Assim, buscamos investigar essa abordagem, combinando transformações de séries temporais, *meta-learning* e a avaliação de diferentes sistemas de recomendação.

### 3 MÉTODO

Neste capítulo serão destacadas as escolhas feitas para a construção do método proposto. A avaliação do método proposto é dada em dois experimentos, a comparação com métodos presentes na literatura e a avaliação do método para 34 classificadores de séries temporais. Na Figura 3 está demonstrado um fluxo de trabalho para a criação de um algoritmo de recomendação, onde a adaptação proposta neste trabalho está na construção de *meta-features*, discutido em mais detalhes no subcapítulo 3.1.

Para o uso de uma recomendação com *meta-learning* três principais etapas devem ser executadas: (1) o desenvolvimento da base de *meta-knowledge*, onde é extraído as *meta-features* dos conjuntos de dados e separada a *meta-target* de interesse; (2) a construção do modelo de mapeamento, onde a *meta-knowledge* é transformada em um conjunto de aprendizado e um *meta-learner* é treinado; (3) a recomendação em um novo conjunto de dados, onde é feita a extração das *meta-features* e o *meta-learner* é aplicado para se obter uma recomendação de quais candidatos são adequados ao problema.

**Figura 4** - Fluxograma de trabalho para a recomendação de algoritmos.



Fonte: Adaptado de [9].

### 3.1 CONJUNTOS DE DADOS, TRANSFORMAÇÃO E EXTRAÇÃO DE *META-FEATURES*.

Nesta seção são descritos os procedimentos e dados utilizados até a construção da tabela de *meta-features* do método proposto, que fará parte da *meta-knowledge*. Com base na Figura 3, os seguintes passos fazem parte do desenvolvimento da base *meta-knowledge*: Espaço do problema P, contendo os conjuntos de dados e as transformações nas séries temporais de cada conjunto; e a extração de *meta-features* de cada cada conjunto.

#### 3.1.1 Conjuntos De Dados

Os experimentos foram executados nos conjuntos de dados da University of California, Riverside (UCR *time series repository*) [58]. Esse repositório é frequentemente utilizado como *benchmark* para a avaliação de métodos de classificação em séries temporais. Além disso, o repositório contém mais de 100 conjuntos univariados de séries temporais de diferentes domínios, variando em tamanho de 24 a mais de 270.000 pontos por série. O número de séries em cada conjunto varia até mais de 10.000, com uma divisão padronizada em conjuntos de treino e teste. Neste trabalho, utilizou-se 112 conjuntos de dados de diferentes domínios, obtidos através da biblioteca para *python* *aeon* [56].

#### 3.1.2 Transformação De Séries Temporais Para Um Vetor De Atributos Com Catch22

O algoritmo Catch22 [2] é utilizado para a extração de atributos nas séries temporais. Este algoritmo tem como antecessor o HCTSA [15, 51, 52] e se trata do resultado de uma seleção de 22 atributos dentre 4791 gerados pelo HCTSA buscando reduzir o custo computacional e manter a capacidade descritiva [2]. Os atributos refletem a interdisciplinaridade e a diversidade presente na literatura de métodos de análises de séries temporais [51], estando presentes propriedades da distribuição de valores, auto correlação linear e não linear, previsibilidade, escala de flutuações e outros.

Utilizando os 112 conjuntos de dados selecionados do repositório UCR, aplicamos o algoritmo Catch22 para extrair atributos de cada conjunto de série temporal. Como

resultado, cada conjunto foi transformado em um novo conjunto de dados, no qual cada série temporal passou a ser representada por meio das 22 atributos canônicos do Catch22.

### 3.1.3 Extração De *Meta-Features* Com *Python Meta-Feature Extractor*

Com as séries temporais de cada conjunto de dados convertidas para os 22 atributos através da transformação executada pelo algoritmo Catch22, utilizou-se a biblioteca *pymfe* [57] para extrair *meta-features* do tipo gerais, info-teóricas e estatísticas. Foram aplicados descritores supervisionados e não supervisionados, totalizando 73 *meta-features* para cada conjunto de dados. A tabela de *meta-features* extraídas pela biblioteca *pymfe*, com suas descrições e referências, pode ser consultada no link em [59].

O resultado desta etapa é uma tabela atributo-valor de *meta-features* de 112 linhas x 73 colunas, na qual cada linha representa um dos conjuntos de dados de séries temporais selecionados do repositório UCR e cada coluna representa uma *meta-feature*. Para que não haja nenhum valor que impossibilite o treinamento de algum dos regressores utilizados, valores de *meta-features* superiores a 100000 são substituídos por 100000.

## 3.2 DELINEAMENTO DOS EXPERIMENTOS

Os experimentos foram executados em um notebook Acer Aspire 5, com processador intel I5-1135G7, com memória RAM de 8GB. Foram utilizadas principalmente as ferramentas *python* com as bibliotecas *scikit learn* [55], *aeon* [56] e *pymfe* [18, 57]. Para a análise dos dados utilizou-se a combinação de *python* e *Microsoft Power BI*. Os resultados e códigos estão disponíveis no APÊNDICE C.

O procedimento de avaliação experimental adotado neste trabalho foi o método *leave-one-out cross validation (Loo)*. Na avaliação *Loo*, um exemplo é retirado do conjunto de dados, o modelo é treinado utilizando os exemplos restantes e testado para a predição do exemplo retirado. Esse processo é repetido para todos os exemplos do conjunto de dados. Uma característica da *Loo* é seu maior custo computacional se comparados aos demais casos de *k-fold* possíveis, no entanto, nos casos em que o volume de exemplos é pequeno, *leave-one-out* é recomendado para obter uma estimativa mais confiável sobre o desempenho de um algoritmo [53].

É importante destacar que, sendo o objetivo a predição da acurácia de classificadores, os valores possíveis estão no intervalo [0,1]. Percebendo a existência de casos distantes do intervalo esperado, o processo de *clipping* se fez adequado, desse modo, foi imposta uma limitação onde predições de acurácia maiores que 1 são substituídas por 1 e menores que 0 são substituídas por 0. O impacto desta limitação é brevemente avaliado nos resultados.

Para avaliar a qualidade dos resultados foram adotadas as métricas de desempenho Erro médio absoluto (MAE) e Erro percentual médio absoluto (MAPE), conforme descritas no Quadro 1. Para a análise da distribuição dos erros nas predições dos *meta-learners* é adequado a visualização em boxplot e o destaque dos valores do 1° quartil (valor que separa os 25% inferiores dos dados dos 75% superiores), 2° quartil (mediana) e 3° quartil (valor que separa os 75% inferiores dos dados dos 25% superiores).

**Quadro 1-** Métricas avaliativas utilizadas na acurácia de *predições* ( $y_{ri}$ ) referentes ao seus respectivos valores reais ( $y_i$ ).

MÉTRICAS	FÓRMULA
Erro médio absoluto, <i>mean absolute error</i> (MAE)	$MAE(y, y_r) = \frac{1}{n_{amostras}} \sum_{i=0}^{n_{amostras}-1}  y_i - y_{ri} $
Erro percentual médio absoluto, <i>mean absolute percentage error</i> (MAPE)	$MAPE(y, y_r) = \frac{1}{n_{amostras}} \sum_{i=0}^{n_{amostras}-1} \frac{ y_i - y_{ri} }{\max(\epsilon,  y_i )}$

Fonte: Autor.

### 3.2.1 Comparação Com A Literatura

Nesta etapa do experimento buscou-se validar a hipótese ao comparar o método proposto com o executado em [13], considerando exatamente os mesmos conjuntos de dados. Em [13], os autores aplicaram duas abordagens diferentes: a aplicação da ferramenta *pymfe* direta nas séries temporais e a utilização de 24 marcadores (*landmarkers*). Os códigos e arquivos disponibilizados do estudo estão disponíveis em [60].

Para este trabalho, utilizamos como *meta-target* a acurácia de classificadores de



séries temporais. Neste primeiro experimento, a *meta-target* consiste da acurácia de 24 classificadores de séries temporais [13]. Os resultados utilizados são os publicados em [61] no qual cada classificador é executado em 100 re-amostragens estratificadas de treino/teste e a média da acurácia é reportada. Desse modo, neste estudo, avaliou-se três *meta-knowledges*, possuindo a mesma *meta-target*, diferenciando na construção das *meta-features*: (1) construída a partir da transformação com Catch22 e a extração MFE (Catch22 + mfe), (2) extração MFE (mfe) e (3) marcadores (*landmarkers*).

Em cada meta-knowledge foi feita a avaliação *leave-one-out cross validation* utilizando modelos induzidos de 7 regressores (Tabela 1) como *meta-learner* em suas configurações *default* e registrando as métricas de desempenho.

**Tabela 1-** Abreviações utilizadas para cada regressor (*meta-learner*).

REGRESSOR ( <i>META-LEARNER</i> )	ABREVIÇÃO UTILIZADA
<i>K Nearest Neighbors</i>	K-NN
<i>Regressor Linear</i>	linear
<i>Linear Bayesian Ridge</i>	linearBayesianRidge
<i>Multi-layer Perceptron</i>	MLP
<i>Random Forest</i>	RandomForest
<i>Support Vector Machine</i>	SVR
<i>Decision Tree</i>	Tree

Fonte: Autor.

### 3.2.2 Avaliação Da Abordagem Proposta Em 34 Classificadores De Séries Temporais

Nesta abordagem, a análise experimental foi expandida para avaliação de 34 classificadores de séries temporais como *meta-target* nos 112 conjuntos de dados da UCR. Foi disponibilizado um diagrama junto aos códigos e resultados no repositório do APÊNDICE C que demonstra o fluxo completo empregado neste experimento.

Utilizou-se os dados publicados no trabalho de MIDDLEHURST (2024) [16] como *meta-target*. Nesse estudo, os resultados de 34 classificadores de séries temporais

foram executados em 30 amostragens estratificadas de treino/teste e a média da acurácia é reportada. Estes resultados foram obtidos através da biblioteca *aeon* [56].

**Tabela 2-** Classificadores utilizados de cada grupo.

<b>GRUPO</b>	<b>CLASSIFICADORES CANDIDATOS</b>
Baseados em distância	EE, PF, ShapeDTW, 1NN-DTW
Baseados em características	FreshPRINCE, TSFresh, Signatures, Catch22
Baseados em intervalos	DrCIF, RSTSF, CIF, TSF, RISE
Baseados em shapelets	RDST, MrSQM, STC, RSF
Baseados em dicionários	WEASEL, WEASEL-D, TDE, cBOSS, BOSS
Baseados em convoluções	Hydra-MR, Hydra, Multi-R, Mini-R, ROCKET, Arsenal
Baseados em <i>deep learning</i>	InceptionT, ResNet, CNN
Abordagens híbridas	HC1, HC2, TS-CHIEF

Fonte: Autor

Os regressores utilizados como *meta-learners* são os mesmos da Tabela 2. Foi utilizado a *leave-one-out cross validation* (LOO) para cada regressor registrando a acurácia obtida nas predições através das métricas MAE e MAPE.

No capítulo presente discutiu-se o método proposto para a criação de *meta-features* em conjuntos de dados de séries temporais após a transformação de séries para vetores de atributos, bem como o delineamento experimental para a avaliação do método. No capítulo seguinte serão apresentados os resultados obtidos destes experimentos e discutido a escolha de *meta-learner*, o desempenho do método em comparação com outras alternativas e o desempenho das recomendações nos diferentes contextos aplicados.

## 4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentadas a interpretação e a discussão dos resultados gerados a partir do delineamento experimental definido no método proposto deste trabalho. Separados em duas etapas: comparação com a literatura e a avaliação da abordagem expandida para 34 classificadores candidatos. Também são discutidos aspectos que fundamentam o método proposto, as principais contribuições e limitações deste estudo.

### 4.1 RESULTADOS DA COMPARAÇÃO COM A LITERATURA

Nesta seção são apresentados os resultados obtidos na avaliação da construção de *meta-features* por meio do método proposto, bem como os resultados obtidos ao utilizar duas das construções de *meta-features* propostas em [13].

Na Tabela 3 são apresentadas as métricas MAE e MAPE da avaliação Loo organizada por cada *meta-learner* nas três bases de *meta-knowledge* distintas. Pela análise dos resultados, nota-se que em 6 dos 7 regressores utilizados como *meta-learner*, a mediana obtida na base de *meta-knowledge* Catch22+mfe é inferior ao da base mfe. Isso significa que ao considerar metade das predições de menor erro, aquelas realizadas por meio do Catch22+mfe ocupam uma região de menor erro, logo foram mais próximas à realidade. Há também uma clara superioridade na qualidade das predições executadas na base de *meta-knowledge landmarks*, demonstrando predições com menor erro e menor distribuição dos erros. Constata-se que o *meta-learner* com melhor desempenho foi o *RandomForest*, apresentando os menores conjuntos de erros e menor distribuição de erros.

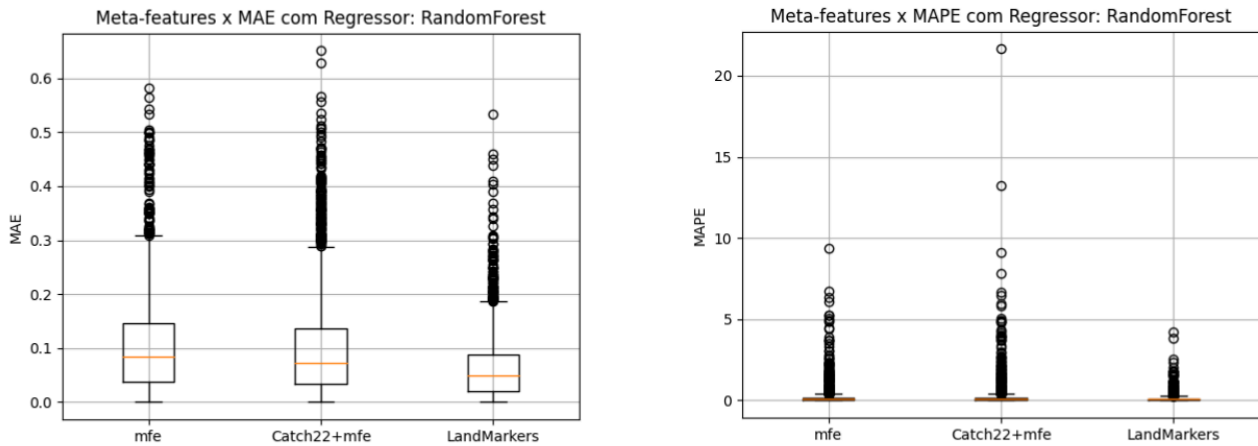
**Tabela 3** - Resultados da avaliação LOO de diferentes regressores nas três bases de *meta-knowledge* utilizadas.

Métrica	MAE			MAPE		
	1° Quartil	Mediana	3° Quartil	1° Quartil	Mediana	3° Quartil
<b>MLP</b>						
Catch22+mfe	0,11	0,24	0,45	0,13	0,33	0,86
mfe	0,10	0,22	0,38	0,11	0,28	0,61
LandMarkers	0,06	0,15	0,23	0,07	0,19	0,35
<b>SVR</b>						
mfe	0,06	0,12	0,19	0,07	0,14	0,23
Catch22+mfe	0,06	0,11	0,18	0,08	0,14	0,22
LandMarkers	0,03	0,06	0,10	0,04	0,08	0,13
<b>linearBayesianRidge</b>						
Catch22+mfe	0,05	0,12	0,20	0,07	0,15	0,26
mfe	0,06	0,12	0,19	0,07	0,15	0,26
LandMarkers	0,02	0,05	0,09	0,03	0,06	0,12
<b>Linear</b>						
mfe	0,05	0,11	0,22	0,06	0,14	0,31
Catch22+mfe	0,04	0,10	0,19	0,05	0,14	0,27
LandMarkers	0,02	0,05	0,09	0,03	0,06	0,13
<b>Tree</b>						
mfe	0,04	0,09	0,19	0,05	0,12	0,27
Catch22+mfe	0,04	0,09	0,17	0,05	0,12	0,24
LandMarkers	0,03	0,06	0,12	0,03	0,09	0,17
<b>KNN</b>						
Catch22+mfe	0,05	0,11	0,20	0,07	0,15	0,27
mfe	0,04	0,10	0,17	0,05	0,12	0,24
LandMarkers	0,03	0,05	0,10	0,03	0,07	0,13
<b>RandomForest</b>						
mfe	0,04	0,08	0,15	0,05	0,11	0,18
Catch22+mfe	0,03	0,07	0,14	0,04	0,09	0,18
LandMarkers	0,02	0,05	0,09	0,03	0,06	0,12

Fonte: Autor.

Na Figura 5 são detalhados os resultados do algoritmo *RandomForest* como *meta-learner*, organizado na forma de *boxplot*, ordenados pela mediana, com a distribuição de erros das métricas MAE e MAPE. Por meio da análise dos resultados reportados na Figura 5 é possível observar uma melhora nos resultados com a aplicação da transformação pelo Catch22, como também fica evidenciado em mais detalhes as informações contidas nas linhas da Tabela 3 referentes a estes dados.

**Figura 5** - Resultados da avaliação LOO em cada base de *meta-knowledge* para o regressor *RandomForest*.



Fonte: Autor.

Nesta avaliação experimental, ao total, foram realizadas 2688 predições (112 conjuntos de dados x 24 classificadores candidatos) em cada *meta-knowledge* por *meta-learner*. Na Tabela 4, são apresentadas informações adicionais sobre o processo de limitação das predições (*clipping*) realizadas. Adicionalmente, pela análise dos dados apresentados na Tabela 4, é possível notar que o MLP como *meta-learner*, com suas configurações *default*, não deve ser utilizado, pois uma grande parte de suas predições foram realizadas fora do intervalo esperado. Nota-se também que as limitações ocorreram apenas no uso dos algoritmos MLP, regressão linear e regressão *Linear Bayesian Ridge*.

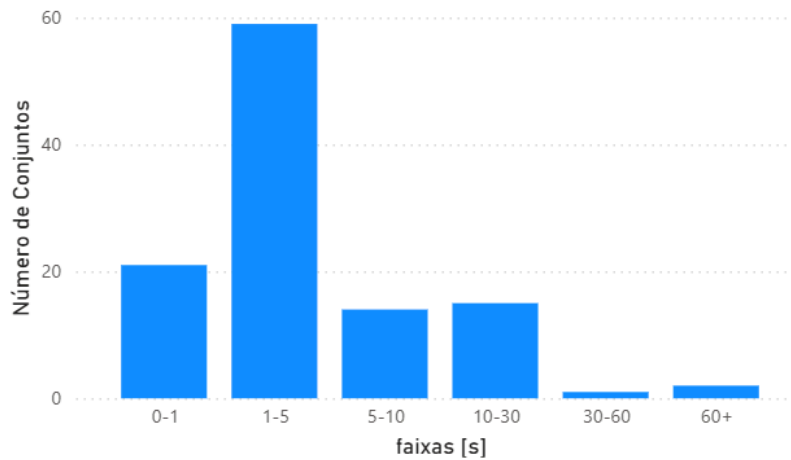
**Tabela 4** - Impacto da limitação ao intervalo de 0 a 1 nas predições efetuadas em cada construção de *meta-features* por *meta-learner* avaliado.

Meta-learner	Predições	% de Predições Limitadas	Predições Limitadas	Predições Limitadas em 1	Predições Limitadas em 0
<b>KNN</b>					
Catch22+mfe	2688	0,00%	0	0	0
LandMarkers	2688	0,00%	0	0	0
mfe	2688	0,00%	0	0	0
<b>Linear</b>					
Catch22+mfe	2688	14,81%	398	322	76
LandMarkers	2688	4,50%	121	121	0
mfe	2688	19,23%	517	383	134
<b>linearBayesianRidge</b>					
Catch22+mfe	2688	6,10%	164	84	80
LandMarkers	2688	2,86%	77	77	0
mfe	2688	4,76%	128	82	46
<b>MLP</b>					
Catch22+mfe	2688	99,11%	2664	2352	312
LandMarkers	2688	58,15%	1563	1563	0
mfe	2688	100,00%	2688	2592	96
<b>RandomForest</b>					
Catch22+mfe	2688	0,00%	0	0	0
LandMarkers	2688	0,00%	0	0	0
mfe	2688	0,00%	0	0	0
<b>SVR</b>					
Catch22+mfe	2688	0,00%	0	0	0
LandMarkers	2688	0,00%	0	0	0
mfe	2688	0,00%	0	0	0
<b>Tree</b>					
Catch22+mfe	2688	0,00%	0	0	0
LandMarkers	2688	0,00%	0	0	0
mfe	2688	0,00%	0	0	0

Fonte: Autor.

O tempo consumido na conversão de todos os conjuntos de dados para a tabela de *meta-features* foi um total de 13 minutos e 04 segundos. O tempo consumido individualmente em cada conjunto pode ser verificado no Apêndice A. Na Figura 6 pode-se observar a contagem por faixa de minutos. É possível verificar que a conversão da maioria dos conjuntos de dados consumiu menos de 1 minuto. Nos resultados disponibilizados por [13], para a criação da tabela de *meta-features* utilizando 24 marcadores para descrever os mesmos 112 conjuntos atingiu um total de 3081 minutos e 12 segundos.

**Figura 6** - Contagem de conjuntos de dados por faixa de tempo consumido para a conversão do conjunto em um vetor de *meta-features* na abordagem *Catch22+mfe*.



Fonte: Autor.

Com base nos resultados reportados, a hipótese testada demonstrou resultados promissores, em termos de eficácia, para sistemas de recomendação de classificadores em séries temporais. A partir dessa constatação é possível conjecturar que a utilização de um extrator de atributos temporais com maior poder descritivo poderia melhorar o desempenho obtido nas recomendações. O algoritmo utilizado para a extração de atributos de séries temporais, *Catch22*, é um dos algoritmos de menor custo computacional, e demonstrou, visualmente na Figura 5, que ao ser utilizado na construção das *meta-features* uma melhora na recomendação de classificadores. Nesse contexto, é possível constatar que o método proposto cumpre com as 2 primeiras propriedades necessárias para as *meta-features* discutidas na seção 2.4 e o desempenho obtido na avaliação dos *meta-learners* demonstra que a abordagem também cumpre com a terceira propriedade, validando a hipótese deste trabalho.

#### 4.2 RESULTADOS DA ABORDAGEM PROPOSTA COM 34 CLASSIFICADORES DE SÉRIES TEMPORAIS

Esta abordagem de avaliação experimental baseou-se no mesmo delineamento executado anteriormente, contudo, utilizando apenas a base de *meta-knowledge* *Catch22+mfe* e expandindo a meta-target para 34 classificadores de séries temporais. Primeiramente, são apresentados os resultados gerais dos *meta-learners*. Em seguida, os resultados do algoritmo *RandomForest* são analisados mais detalhadamente sob

perspectivas de classificadores candidatos e conjuntos de dados. Resultados análogos aos do *RandomForest* para os outros *meta-learners* estão disponíveis no apêndice B.

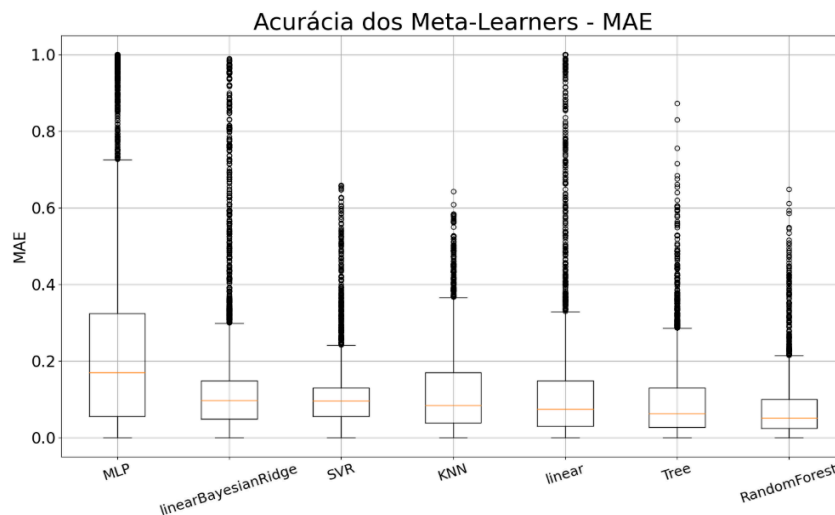
Na Tabela 5, são apresentados os resultados em termos das métricas MAE e a MAPE da avaliação Loo de cada *meta-learner* aplicado na base de *meta-knowledge* proposta. Na Figura 7, a dispersão dos resultados de cada *meta-learner* são detalhados visualmente na forma de gráfico *boxplot*. os resultados de cada *meta-learner* temos o *boxplot* da mesma fonte de informação da MAE da Tabela 5, combinando estas duas visualizações, assim como no experimento anterior, Com base nesses resultados é possível constatar que o melhor desempenho foi do algoritmo *RandomForest* como *meta-learner*, apresentando a menor mediana e a menor distribuição de erros.

**Tabela 5** - Avaliação LOO de cada regressor utilizado como *meta-learner* na *meta-knowledge* proposta.

Métrica	MAE			MAPE		
	1° Quartil	Mediana	3° Quartil	1° Quartil	Mediana	3° Quartil
MLP	0,06	0,17	0,32	0,06	0,21	0,48
linearBayesianRidge	0,05	0,10	0,15	0,06	0,11	0,17
SVR	0,06	0,10	0,13	0,06	0,10	0,15
KNN	0,04	0,08	0,17	0,04	0,10	0,20
linear	0,03	0,07	0,15	0,03	0,09	0,19
Tree	0,03	0,06	0,13	0,03	0,08	0,16
RandomForest	0,02	0,05	0,10	0,03	0,06	0,12

Fonte: Autor.

**Figura 7** - Avaliação LOO de cada regressor utilizado como *meta-learner* na *meta-knowledge* proposta.



Fonte: Autor.



De acordo com a Tabela 6, de estrutura semelhante à Tabela 4, é possível avaliar o impacto da limitação imposta a predições fora do intervalo fechado de 0 a 1. De modo similar à avaliação realizada na seção anterior, os mesmos algoritmos sofreram limitações, corroborando o fato de que, de modo geral, são menos confiáveis para a tarefa no contexto aplicado.

**Tabela 6-** Impacto da limitação ao intervalo de 0 a 1 nas predições efetuadas com cada regressor usado como *meta-learner*.

Meta-learner	Predições	% de Predições Limitadas	Predições Limitadas	Predições Limitadas em 1	Predições Limitadas em 0
KNN	3808	0,00%	0	0	0
linear	3808	18,28%	696	579	117
linearBayesianRidge	3808	6,85%	261	153	108
MLP	3808	99,11%	3774	3332	442
RandomForest	3808	0,00%	0	0	0
SVR	3808	0,00%	0	0	0
Tree	3808	0,00%	0	0	0

Fonte: Autor.

Com base nos resultados apresentados na Tabela 5 e Tabela 6, verifica-se uma grande variação na qualidade das predições feitas por diferentes regressores utilizados como *meta-learner*. Pela análise, é possível verificar que os melhores resultados estão associados ao algoritmo *RandomForest*.

#### 4.2.1 Resultados para o *Meta-Learner RandomForest*

Nesta subseção, será aprofundada a avaliação da aplicação do *RandomForest* como *meta-learner*. Esta escolha de discussão se deve ao fato deste algoritmo de regressão ter obtido o melhor desempenho como *meta-learner* entre os algoritmos considerados neste trabalho. Primeiro discute-se seu desempenho sob a perspectiva do erro obtido nas predições por classificadores candidatos e, posteriormente, a análise é realizada sob a perspectiva dos conjuntos de dados utilizados.

##### 4.2.1.1 *Perspectiva de classificadores candidatos*

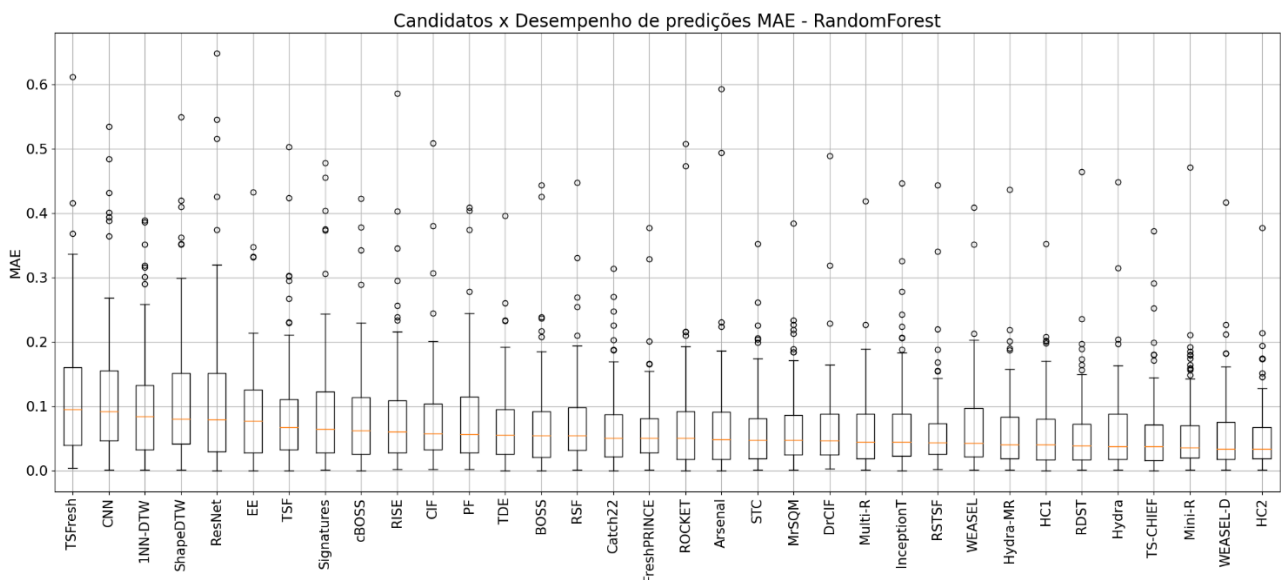
O desempenho do *RandomForest* é avaliado e discutido a partir da perspectiva

dos classificadores candidatos, apresentando sua distribuição de erros em previsões para cada candidato individualmente e por tipo de classificador de séries temporais (de acordo com a seção 2.2).

Na Figura 8, os resultados da avaliação Loo com o *meta-learner RandomForest* são apresentados na forma de gráfico *boxplot*, em termos da métrica MAE. Nota-se que há classificadores em que o *RandomForest* executou suas previsões com menores erros e menor dispersão. O conjunto de previsões com menor erro foram do HC2, e com o maior erro foram as previsões no conjunto referente ao TSFresh.

Percebe-se que há previsões feitas pelo *RandomForest* de candidatos que são mais consistentes que as de outros. Considerando essa característica, é possível, para alguma determinada aplicação, considerar apenas um subconjunto de candidatos que tenham sido preditos com maior acurácia, com isso, reduzindo o número de candidatos porém aumentando a confiabilidade da recomendação a ser utilizada. Neste mesmo contexto, há também uma leve correlação com as previsões por tipos de classificadores (Tabela 7), fornecendo assim qual o tipo de classificador o sistema proposto faz sua previsão com maior acurácia.

**Figura 8** - Desempenho das previsões do *RandomForest* como *meta-learner* por algoritmo candidato.



Fonte: Autor.

Na Tabela 7, temos os resultados da Loo nas métricas MAE e MAPE organizadas por tipo de classificador. Os valores fornecem indícios da capacidade do *meta-learner* de fazer previsões melhores para certos tipos de algoritmos como no caso

de classificadores de abordagens híbridas.

**Tabela 7** - Resultados do RandomForest como *meta-learner* na LOO por grupo de classificadores de séries temporais.

Métrica	MAE			MAPE		
	1° Quartil	Mediana	3° Quartil	1° Quartil	Mediana	3° Quartil
Distance based	0,03	0,07	0,13	0,04	0,09	0,17
Deep learning based	0,03	0,07	0,13	0,03	0,08	0,16
Feature based	0,03	0,06	0,11	0,03	0,07	0,14
Interval based	0,03	0,05	0,10	0,03	0,06	0,12
Dictionary based	0,02	0,05	0,10	0,03	0,06	0,11
Shapelet based	0,02	0,05	0,08	0,03	0,05	0,10
Convolution based	0,02	0,04	0,09	0,02	0,05	0,10
Hybrid approaches	0,02	0,04	0,08	0,02	0,04	0,09

Fonte: Autor.

#### 4.2.1.2 Perspectiva de conjuntos de dados

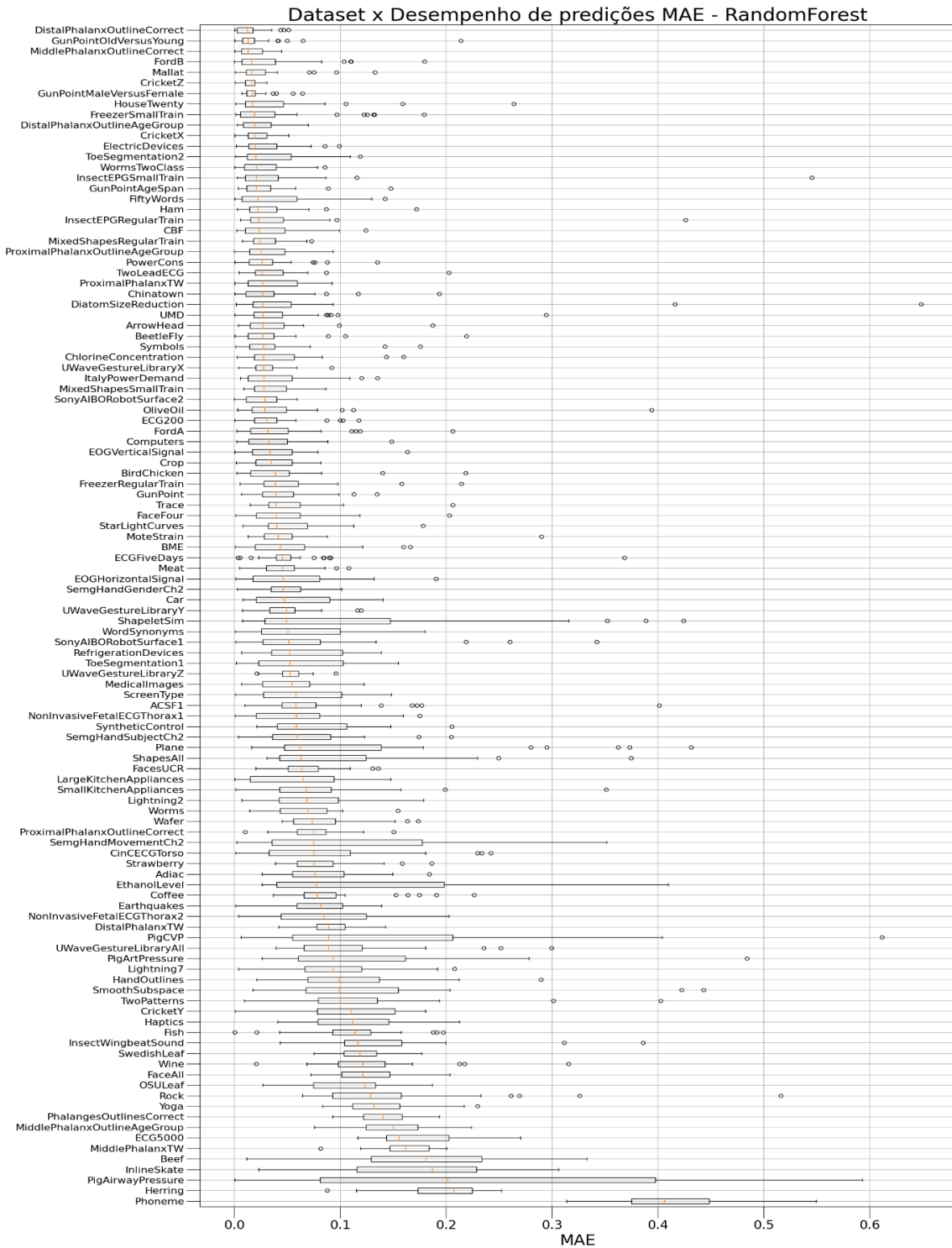
Nesta seção são avaliados os resultados do desempenho do *RandomForest* como *meta-learner* em diferentes perspectivas dos conjuntos de dados, considerando: por conjunto individualmente; por tamanho de séries temporais; por número de amostras; e por quantidade de rótulos.

Os resultados reportados na Figura 9 fornecem a distribuição dos erros das predições feitas pelo *RandomForest* em cada um dos 112 conjuntos de dados utilizados na métrica MAE e ordenados pela mediana. É possível verificar na parte superior da figura que há diversos conjuntos de dados com baixa mediana e baixa dispersão. Nesses casos, é possível constatar que as predições realizadas foram mais próximas da realidade, logo, casos em que a recomendação pelo modelo induzido seria mais confiável. Há também os casos com menores medianas na parte inferior da figura, onde percebe-se um grande aumento na dispersão dos erros. Nestes casos, as predições se distanciam da realidade, logo, as recomendações nestes casos seriam menos confiáveis.

Do desempenho das previsões por conjunto de dados (Figura 9) é possível constatar a possibilidade de inferência para problemas semelhantes. Por exemplo, no conjunto *Phonema* as predições foram distantes da realidade, apresentando maior erro, como foi o caso de outros conjuntos de dados do tipo áudio. Contudo no conjunto *Cricket Z* as predições foram próximas às reais, apresentando menor erro, como foi o caso de outros conjuntos de dados do tipo HAR. Assim, podemos delimitar quais tipos de

problemas o sistema tem potencial para fazer as melhores previsões e com isso fornecer uma melhor recomendação com base dos conhecimentos acerca do problema que se deseja aplicar a recomendação.

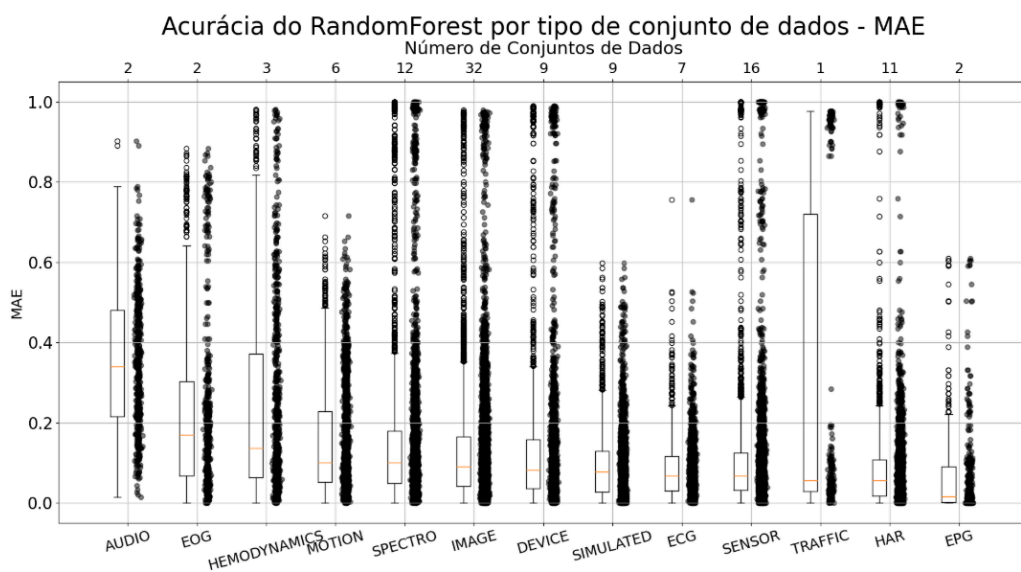
**Figura 9 - Avaliação LOO do *RandomForest* como *meta-learner* por conjunto de dados.**



Fonte: Autor.

Na Figura 10, são apresentados os resultados da avaliação Loo, em termos das métricas MAE para o algoritmo *RandomForest* organizados por tipo de conjunto de dados. Pela análise dos dados reportados é possível constatar que o *meta-learner* obteve desempenho distinto em predições acerca de certos tipos de dados. Como exemplo, o pior desempenho foi observado nos conjuntos de dados do tipo áudio, destacando, pela Figura 9, o conjunto chamado *Phoneme*.

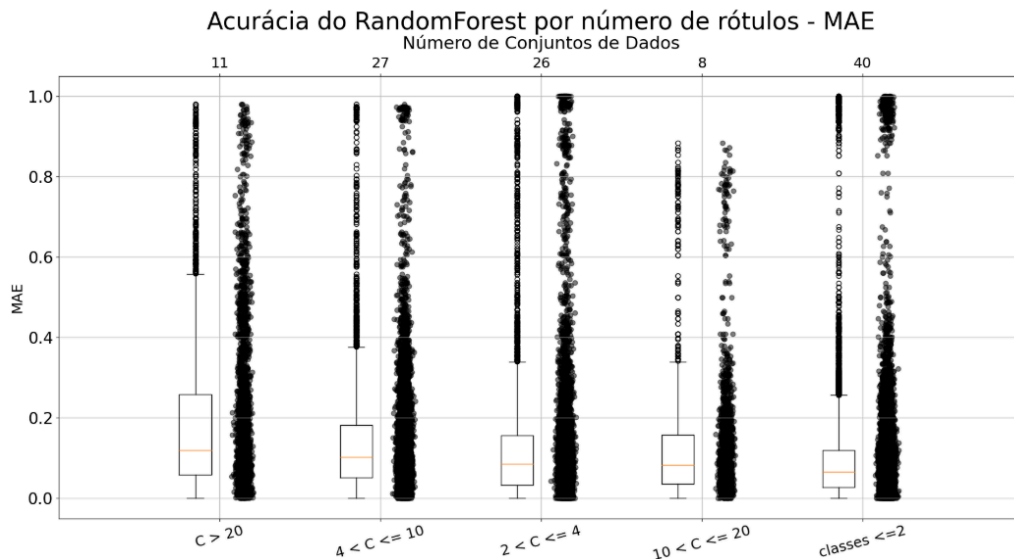
**Figura 10** - Resultados do *RandomForest* como *meta-learner* por tipo de conjunto de dados.



Fonte: Autor.

Na Figura 11, os resultados são reportados em função do número de classes em cada conjunto de dados. Pela análise dos resultados fornecidos, é possível verificar que o *meta-learner* não fará, necessariamente, predições melhores em conjuntos de dados com menor número de classes. Como exemplo, o desempenho em problemas com 10 a 20 classes foi superior aos conjuntos com 4 a 10 classes.

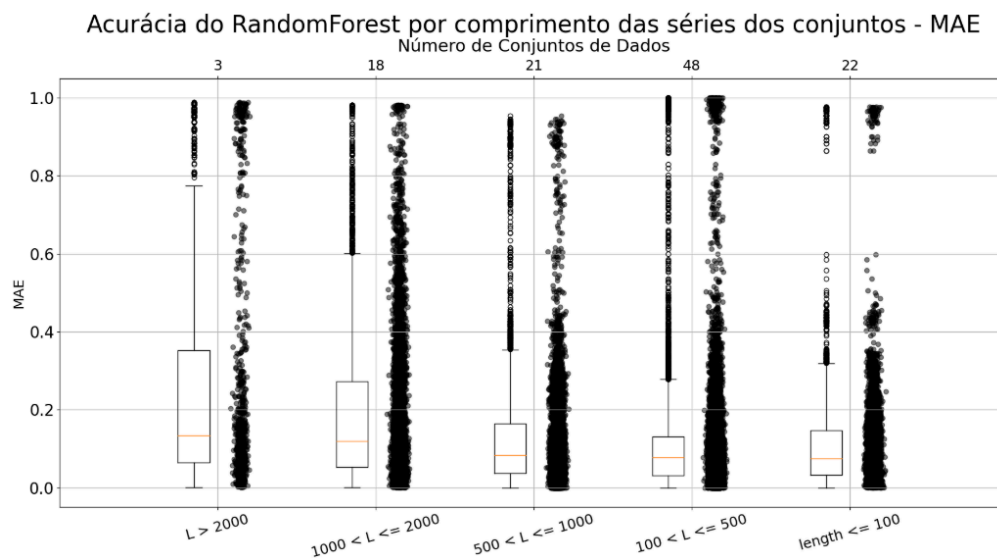
**Figura 11** - Resultados do *RandomForest* como *meta-learner* por número de rótulos (classes) presentes nos conjuntos de dados.



Fonte: Autor.

Na Figura 12, os resultados são organizados em função do comprimento das séries temporais dos conjuntos de dados. Por meio da análise dos dados reportados, nota-se um indício de um padrão em que o *meta-learner* faz previsões melhores em conjuntos nos quais as séries possuem menor comprimento.

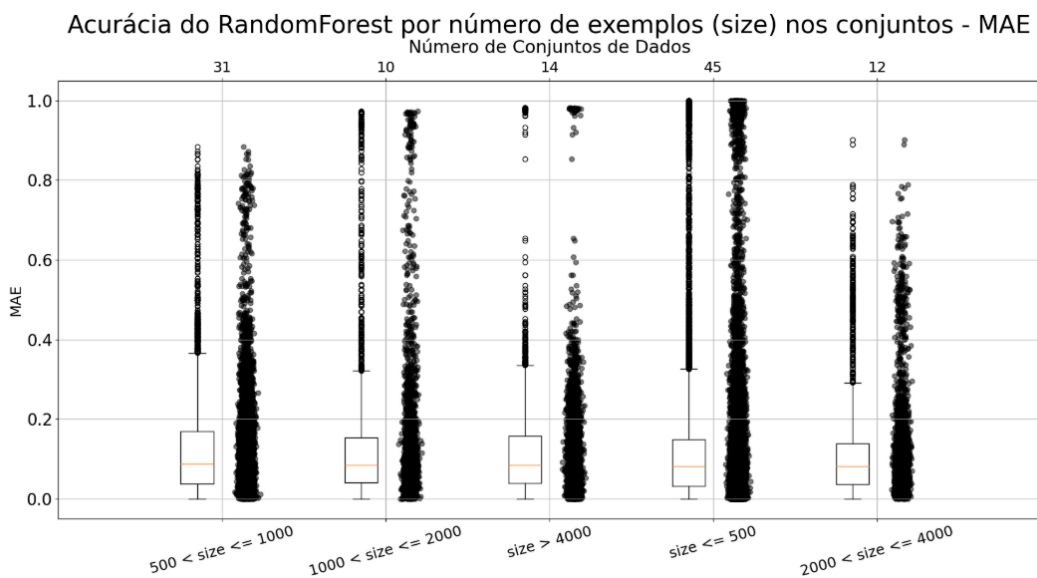
**Figura 12** - Resultados do *RandomForest* como *meta-learner* por comprimento (*length*) da ST dos conjuntos de dados.



Fonte: Autor.

Na Figura 13, os resultados são estruturados em função do tamanho dos conjuntos de dados, ou seja, o número de séries temporais que os compõem. Novamente, com base na análise desses resultados, é possível verificar que as recomendações não serão, necessariamente, melhores em conjuntos menores. Como exemplo, foi identificado um melhor desempenho em conjuntos com um número de séries superior a 4000 se comparados aos com números de séries entre 2000 e 4000 exemplos.

**Figura 13** - Resultados do *RandomForest* como *meta-learner* por número de amostras (*size*) nos conjuntos de dados.



Fonte: Autor.

#### 4.3 CONTRIBUIÇÕES E LIMITAÇÕES

No contexto de séries temporais, aplicações de representações baseadas em atributos de séries temporais demonstraram que, apesar de uma grande redução de dimensionalidade, possibilitaram a construção de classificadores competitivos [2,15]. Desse modo, pela primeira vez na literatura, neste trabalho, foi proposto um método para incorporar a informação temporal ao processo de *meta-learning* para recomendação de classificadores de séries temporais. A intuição que motivou a construção deste método baseia-se na possibilidade de retenção de parte da informação temporal presente nas séries, e paralelamente, para permitir a aplicação de estratégias tradicionais de *meta-learning*. Assim, a proposta apresentada baseou-se na aplicação do algoritmo *Catch22*, o qual possibilitou a extração de atributos que refletem a interdisciplinaridade e a

diversidade presente na literatura de métodos de análises de séries temporais [51].

Adicionalmente, foi apresentada uma ampla análise experimental para avaliar o método proposto. Nos experimentos conduzidos avaliou-se o desempenho de diferentes *meta-learners* executando a *leave-one-out cross validation* nos 112 conjuntos de dados de séries temporais univariados e de iguais comprimentos do repositório da *UCR*. A avaliação foi feita em duas diferentes *abordagens*, a primeira para a comparação direta com trabalhos usados como referência e a segunda para expandir e explorar a análise para 34 classificadores de séries temporais. Os resultados nos permitem avaliar o desempenho dos *meta-learners* em diferentes métricas, no desempenho geral e por conjunto de dados, por candidatos, e suas subdivisões. Em ambos experimentos dentre os *meta-learners* utilizados o *Random Forest* obteve o melhor desempenho. A escolha da utilização de acurácias de classificadores como *meta-target* tem como motivação a possibilidade de utilizá-la para inferir outros tipos de *meta-target*, possibilitando diferentes tipos de recomendação a partir da saída gerada pelo *meta-learner* [13].

De modo geral, ao considerar o método proposto em combinação com o *RandomForest* como *meta-learner*, é possível constatar (Tabela 3 e Figura 6) que o método representa uma ferramenta de recomendação de baixo custo computacional e com resultados que se aproximam da abordagem, utilizando marcadores, apresentada em [13].

Apesar das vantagens mencionadas sobre o algoritmo *Catch22*, os atributos utilizados podem não ser suficientes para capturar padrões com maior discriminabilidade entre os possíveis rótulos de um conjunto de dados com maior complexidade. Outra limitação do método proposto é o processo de extração de *meta-features* aplicado à representação do *Catch22*. Considerando os descritores *mfe* utilizados, alguns podem capturar padrões mais discriminantes enquanto outros podem prejudicar o desempenho de determinados *meta-learners*. Não foi avaliada a possibilidade de que ao utilizar outras técnicas para a extração de *meta-features* nos conjuntos transformados exista uma melhora significativa no desempenho dos *meta-learners*.

## 5 CONCLUSÃO E TRABALHOS FUTUROS



Pela primeira vez na literatura, neste trabalho, foi desenvolvido e avaliado um método para incorporar a informação temporal ao processo de *meta-learning*, utilizando Catch22 como um extrator de atributos de séries temporais, para recomendação de classificadores de séries temporais. Foi conduzida uma ampla avaliação experimental, envolvendo um total de 112 conjuntos de dados, 34 classificadores temporais e 7 algoritmos de regressão, analisados sob diferentes perspectivas de análise de desempenho. Como resultado, constatou-se o algoritmo RandomForest com melhor desempenho como *meta-learner* para o método proposto. Desse modo, em comparação com a literatura destaca-se o baixo custo computacional do método proposto, bem como a validação da hipótese de que a utilização *meta-features* construídas a partir da transformação de séries temporais em atributos discriminatórios é possível a criação de um sistema de recomendação competitivo com outras abordagens conhecidas.

Para a melhoria do sistema de recomendação e possíveis trabalhos futuros, é considerável o potencial de aprimoramento do recomendador ao explorar outros extratores de atributos específicos para séries temporais no método proposto, a combinação de diferentes tipos de *meta-features* para a discriminação entre conjuntos de dados, os parâmetros dos modelos utilizados como *meta-learners*, a construção de *meta-knowledge* e uma diferente aplicação partir de problemas não supervisionados para a aplicação de recomendação sobre os mesmos.

## REFERÊNCIAS

- [1] KEOGH, E.; KASETTY, S.. On the need for time series data mining benchmarks: a survey and empirical demonstration. **Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining**, p. 102-111, 2002.
- [2] LUBBA, C.H. *et al.* *Catch22: Canonical Time-series Characteristics*. **Data Min Knowl Disc** **33**, p.1821–1852, 2019.
- [3] WOLPERT, D. H. *et al.* No free lunch theorems for search. **Technical Report SFI-TR-95-02-010**, Santa Fe Institute, v.10, n. 12, p. 2756-2760, 1995.
- [4] WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE transactions on evolutionary computation**, v. 1, n. 1, p. 67-82, 1997.
- [5] KERSCHKE, P. *et al.* Automated algorithm selection: Survey and perspectives. **Evolutionary computation**, v. 27, n. 1, p. 3-45, 2019.
- [6] BRAZDIL, P. *et al.* *Metalearning: Applications to data mining*. **Springer Science & Business Media**, 2008.
- [7] BISCHL, B., *et al.* *Aslib: A benchmark library for algorithm selection*. **Artificial Intelligence**, v. 237, p. 41-58, 2016.
- [8] ALI, R.; LEE, S.; CHUNG, T. C. Accurate multi-criteria decision making methodology for recommending machine learning algorithm. **Expert Systems with Applications**, v. 71, p. 257-278, 2017.
- [9] KHAN, I. *et al.* A literature survey and empirical study of meta-learning for classifier selection. **IEEE Access**, v. 8, p. 10262-10281, 2020.
- [10] WANG, G.; SONG, Q.; ZHU, X. An improved data characterization method and its application in classification algorithm recommendation. **Applied Intelligence**, v. 43, p. 892-912, 2015.
- [11] SERBAN, F., *et al.* A survey of intelligent assistants for data analysis. **ACM Computing Surveys (CSUR)**, v. 45, n. 3, p. 1-35, 2013.
- [12] DEGROOTE, H. Online Algorithm Selection. **IJCAI**, p. 5173-5174, 2017.
- [13] ABANDA, A.; MORI, U.; LOZANO, J. A. Time series classifier recommendation by a meta-learning approach. **Pattern Recognition**, v. 128, p. 108671, 2022.
- [14] BAGNALL, A., *et al.* The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. **Data mining and knowledge discovery**, v. 31, p. 606-660, 2017.
- [15] FULCHER, B.D.; JONES, N.S. Highly comparative feature-based time-series classification. **IEEE Trans Knowl Data Eng**, v. 26 n.12, p.3026–3037, 2014.

- [16] MIDDLEHURST, M.; SCHÄFER, P.; BAGNALL, A. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. **Data Mining and Knowledge Discovery**, p. 1-74, 2024.
- [17] ABANDA, A.; MORI, U.; LOZANO, J.A. Review on distance based time series classification. **Data Mining and Knowledge Discovery**, v. 33, n.2, p.378–412, 2019.
- [18] ALCOBAÇA, E. *et al.* MFE: Towards reproducible meta-feature extraction. **Journal of Machine Learning Research**, v. 21, n. 111, p. 1-5, 2020.
- [19] ZHOU, ZH. Machine learning. **Springer nature**, 2021.
- [20] ESLING, P.; AGON, C. Time-series data mining. **ACM Computing Surveys (CSUR)**, v. 45, n. 1, p. 1-34, 2012.
- [21] LEMKE, C.; BUDKA, M.; GABRYS, B. Metalearning: a survey of trends and technologies. **Artificial intelligence review**, v. 44, p. 117-130, 2015.
- [22] SOHN, S. Y. Meta analysis of classification algorithms for pattern recognition. **IEEE transactions on pattern analysis and machine intelligence**, v. 21, n. 11, p. 1137-1144, 1999.
- [23] MUSTAFA, M.; SEBAG, M. ALORS: An algorithm recommender system for Classifiers. **Artif. Intell**, v. 244, p. 291-314, 2017.
- [24] ZHU, X. *et al.* A new classification algorithm recommendation method based on link prediction. **Knowledge-Based Systems**, v. 159, p. 171-185, 2018.
- [25] WANG, G. *et al.* A generic multilabel learning-based classification algorithm recommendation method. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, v. 9, n. 1, p. 1-30, 2014.
- [26] LEMKE, C.; GABRYS, B. Meta-learning for time series forecasting and forecast combination. **Neurocomputing**, v. 73, n. 10-12, p. 2006-2016, 2010.
- [27] ROSSI, A. L. D. *et al.* MetaStream: A meta-learning based method for periodic algorithm selection in time-changing data. **Neurocomputing**, v. 127, p. 52-64, 2014.
- [28] MUÑOZ, M. A. *et al.* Algorithm selection for black-box continuous optimization problems: A survey on methods and challenges. **Information Sciences**, v. 317, p. 224-245, 2015.
- [29] SMITH-MILES, K. A. Cross-disciplinary perspectives on meta-learning for algorithm selection. **ACM Computing Surveys (CSUR)**, v. 41, n. 1, p. 1-25, 2009.
- [30] CUI, C. *et al.* A recommendation system for meta-modeling: A meta-learning based approach. **Expert Systems with Applications**, v. 46, p. 33-44, 2016.
- [31] ABDULRAHMAN, S. M. *et al.* Speeding up algorithm selection using average ranking and active testing by introducing runtime. **Machine learning**, v. 107, p. 79-108, 2018.
- [32] VAN RIJN, J. N. *et al.* The online performance estimation framework: heterogeneous ensemble learning for data streams. **Machine Learning**, v. 107, p. 149-176, 2018.

- [33] RIVOLLI, A. et al. Meta-features for meta-learning. **Knowledge-Based Systems**, v. 240, p. 108101, 2022.
- [34] LORENA, A. C. et al. How complex is your classification problem? a survey on measuring classification complexity. **ACM Computing Surveys (CSUR)**, v. 52, n. 5, p. 1-34, 2019.
- [35] LUENGO, J.; HERRERA, F. An automatic extraction method of the domains of competence for learning classifiers using data complexity measures. **Knowledge and Information Systems**, v. 42, p. 147-180, 2015.
- [36] BENSUSAN, H. God doesn't always shave with Occam's razor—Learning when and how to prune. In: **Machine Learning: ECML-98: 10th European Conference on Machine Learning Chemnitz, Germany, April 21–23, 1998 Proceedings 10**. Springer Berlin Heidelberg, 1998. p. 119-124.
- [37] HO, T. K.; BASU, M. Complexity measures of supervised classification problems. **IEEE transactions on pattern analysis and machine intelligence**, v. 24, n. 3, p. 289-300, 2002.
- [38] SÁEZ, J. A.; LUENGO, J.; HERRERA, F. Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification. **Pattern Recognition**, v. 46, n. 1, p. 355-364, 2013.
- [39] BENSUSAN, H.; GIRAUD-CARRIER, C. Discovering task neighbourhoods through landmark learning performances. In: **European Conference on Principles of Data Mining and Knowledge Discovery**. Berlin, Heidelberg: Springer Berlin Heidelberg, p. 325-330, 2000.
- [40] PFAHRINGER, B.; BENSUSAN, H.; GIRAUD-CARRIER, C. G. Meta-Learning by Landmarking Various Learning Algorithms. In: **ICML**. 2000. p. 743-750.
- [41] TATTI, N. Distances between Data Sets Based on Summary Statistics. **Journal of Machine Learning Research**, v. 8, n. 1, 2007.
- [42] SONG, Q.; WANG, G.; WANG, C. Automatic recommendation of classification algorithms based on data set characteristics. **Pattern recognition**, v. 45, n. 7, p. 2672-2689, 2012.
- [43] ZHANG, X. et al. An instance-based learning recommendation algorithm of imbalance handling methods. **Applied Mathematics and Computation**, v. 351, p. 204-218, 2019.
- [44] HUTTER, F.; KOTTHOFF, L; VANSCHOREN, J. Automated machine learning: methods, systems, challenges. **Springer Nature**, 2019.
- [45] GARCIA, Luís PF et al. Ensembles of label noise filters: a ranking approach. **Data Mining and Knowledge Discovery**, v. 30, p. 1192-1216, 2016.
- [46] WANG, Guangtao et al. A feature subset selection algorithm automatic recommendation method. **Journal of Artificial Intelligence Research**, v. 47, p. 1-34, 2013.

- [47] BRAZDIL, PI; GAMA, J; HENERY, B. Characterizing the applicability of classification algorithms using meta-level learning. In: **Machine Learning: ECML-94: European Conference on Machine Learning Catania**, Italy, April 6–8, 1994 Proceedings 7. Springer Berlin Heidelberg, 1994. p. 83-102.
- [48] SOUSA, A. et al. Active learning and data manipulation techniques for generating training examples in meta-learning. **Neurocomputing**, v. 194, p. 45-55, 2016.
- [49] LEYVA, Enrique et al. On the use of meta-learning for instance selection: An architecture and an experimental study. **Information Sciences**, v. 266, p. 16-30, 2014.
- [50] REIF, Matthias et al. Automatic classifier selection for non-experts. **Pattern Analysis and Applications**, v. 17, p. 83-96, 2014.
- [51] FULCHER, B; LITTLE, M.; JONES, N. Highly comparative time-series analysis: the empirical structure of time series and their methods. **Journal of the Royal Society Interface**, v. 10, n. 83, p. 20130048, 2013.
- [52] FULCHER, B.; JONES, N. hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. **Cell systems**, v. 5, n. 5, p. 527-531. e3, 2017.
- [53] WONG, T. Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. **Pattern recognition**, v. 48, n. 9, p. 2839-2846, 2015.
- [54] WITTEN, Ian H.; FRANK, Eibe. Data mining: practical machine learning tools and techniques with Java implementations. **Acm Sigmod Record**, v. 31, n. 1, p. 76-77, 2002.
- [55] SCIKIT-LEARN. scikit-learn: Machine Learning in Python. Disponível em: <<https://scikit-learn.org/stable/>>. Acesso em: 24 jan. 2025.
- [56] **Welcome to aeon**. Disponível em: <<https://www.aeon-toolkit.org/en/stable/>>. Acesso em: 24 jan. 2025.
- [57] **The PyMFE example gallery — pymfe 0.4.2 documentation**. Disponível em: <[https://pymfe.readthedocs.io/en/latest/auto\\_examples/index.html](https://pymfe.readthedocs.io/en/latest/auto_examples/index.html)>. Acesso em: 24 jan. 2025.
- [58] CHEN, Yanping et al. The UCR time series classification archive. 2015.
- [59] **Meta-feature Description Table — pymfe 0.4.2 documentation**. Disponível em: <[https://pymfe.readthedocs.io/en/latest/auto\\_pages/meta\\_features\\_description.html](https://pymfe.readthedocs.io/en/latest/auto_pages/meta_features_description.html)>. Acesso em: 24 jan. 2025.
- [60] **Amaia Abanda Elustondo / TSCR**. Disponível em: <<https://gitlab.bcamath.org/aabanda/tscr>>. Acesso em: 24 jan. 2025.
- [61] BAGNALL, Anthony et al. The UEA & UCR time series classification repository, 2017. **URL: <http://www.timeseriesclassification.com>**, 2021.
- [62] RUSSELL, Stuart J.; NORVIG, Peter. **Artificial intelligence: a modern approach**. pearson, 2016.

[63] FACELI, Katti et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. 2021.

[64] FERRERO, Carlos Andres. **Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia**. 2009. Tese de Doutorado. Universidade de São Paulo.

[65] ZALEWSKI, Willian. **Modelagem simbólica de padrões morfológicos para classificação de séries temporais**. 2015.

[66] NARWARIYA, Jyoti et al. **Meta-learning for few-shot time series classification**. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD. 2020. p. 28-36.

[67] MU, Tianyu et al. **Tsc-automl: Meta-learning for automatic time series classification algorithm selection**. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE). IEEE, 2023. p. 1032-1044.

**APÊNDICES**

**APÊNDICE A – TEMPO CONSUMIDO PARA A CONVERSÃO DOS 112 CONJUNTOS DE DADOS PARA META-EXAMPLES UTILIZANDO CATCH22 E MFE.**

Conjunto de Dados	Tempo [s]	Conjunto de Dados	Tempo [s]
ACSF1	2.43	Mallat	9.63
Adiac	2.53	Meat	1.11
ArrowHead	1.14	MedicalImages	3.12
Beef	0.83	MiddlePhalanxOutlineAgeGroup	1.41
BeetleFly	0.87	MiddlePhalanxOutlineCorrect	1.54
BirdChicken	0.89	MiddlePhalanxTW	1.55
BME	0.90	MixedShapesRegularTrain	10.09
Car	1.04	MixedShapesSmallTrain	6.52
CBF	1.78	MoteStrain	1.61
Chinatown	20.88	NonInvasiveFetalECGThorax1	11.19
ChlorineConcentration	5.75	NonInvasiveFetalECGThorax2	14.63
CinCECGTorso	8.20	OliveOil	0.88
Coffee	0.89	OSULeaf	1.62
Computers	1.82	PhalangesOutlinesCorrect	2.31
CricketX	2.55	Phoneme	7.35
CricketY	2.67	PigAirwayPressure	3.84
CricketZ	2.59	PigArtPressure	19.54
Crop	13.60	PigCVP	3.93
DiatomSizeReduction	1.39	Plane	0.83
DistalPhalanxOutlineAgeGroup	1.45	PowerCons	1.15
DistalPhalanxOutlineCorrect	1.53	ProximalPhalanxOutlineAgeGroup	1.22
DistalPhalanxTW	1.44	ProximalPhalanxOutlineCorrect	1.35
Earthquakes	1.60	ProximalPhalanxTW	1.65
ECG200	1.06	RefrigerationDevices	2.22
ECG5000	5.25	Rock	3.58
ECGFiveDays	1.76	ScreenType	2.56
ElectricDevices	12.08	SemgHandGenderCh2	25.83
EOGHorizontalSignal	72.32	SemgHandMovementCh2	24.18
EOGVerticalSignal	46.65	SemgHandSubjectCh2	21.34
EthanolLevel	3.63	ShapeletSim	0.93
FaceAll	2.08	ShapesAll	2.62
FaceFour	0.67	SmallKitchenAppliances	2.26
FacesUCR	2.40	SmoothSubspace	0.72
FiftyWords	1.82	SonyAIBORobotSurface1	0.94
Fish	0.97	SonyAIBORobotSurface2	1.11
FordA	6.26	StarLightCurves	20.68
FordB	5.64	Strawberry	1.44
FreezerRegularTrain	3.58	SwedishLeaf	1.49
FreezerSmallTrain	3.45	Symbols	1.84
GunPoint	0.66	SyntheticControl	0.99
GunPointAgeSpan	24.82	ToeSegmentation1	0.93
GunPointMaleVersusFemale	24.35	ToeSegmentation2	0.81
GunPointOldVersusYoung	24.25	Trace	0.86
Ham	0.80	TwoLeadECG	1.27
HandOutlines	7.52	TwoPatterns	3.30
Haptics	1.87	UMD	0.71
Herring	0.72	UWaveGestureLibraryAll	10.34
HouseTwenty	157.76	UWaveGestureLibraryX	5.55
InlineSkate	3.77	UWaveGestureLibraryY	5.37
InsectEPGRegularTrain	1.20	UWaveGestureLibraryZ	5.68
InsectEPGSmallTrain	1.26	Wafer	6.19
InsectWingbeatSound	3.46	Wine	0.85
ItalyPowerDemand	1.22	WordSynonyms	3.35
LargeKitchenAppliances	2.95	Worms	1.52
Lightning2	1.36	WormsTwoClass	1.43
Lightning7	1.14	Yoga	6.17

Fonte: Autor.

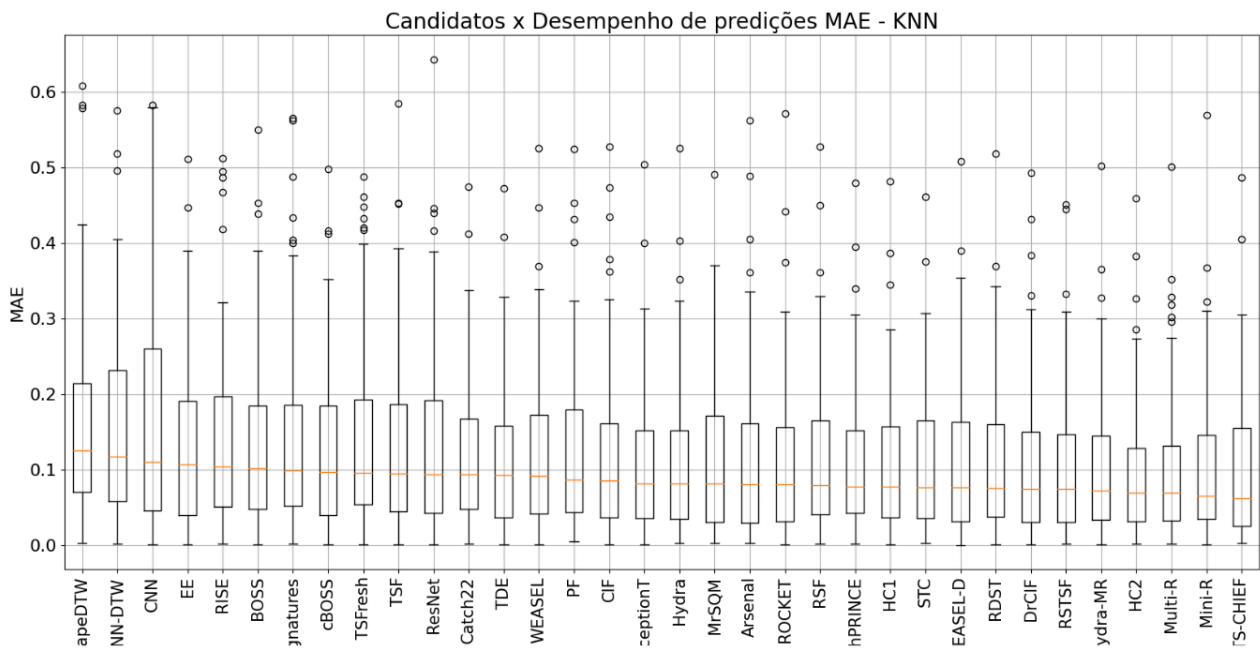


## APÊNDICE B – RESULTADOS DA SEÇÃO 4.2 PARA OS REGRESSORES UTILIZADOS

### B.1. Resultados Para O Knn

#### B.1.1 Perspectiva de classificadores candidatos

**Gráfico B.1** - Desempenho das previsões do KNN como meta-learner por candidato.



Fonte: Autor.

**Tabela B.1** - Resultados do KNN como meta-learner na LOO por grupo de classificadores de séries temporais.

Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Distance based	0,04	0,09	0,16	0,05	0,11	0,21	0,00	0,01	0,03
Deep learning based	0,04	0,08	0,16	0,04	0,10	0,20	0,00	0,01	0,03
Feature based	0,04	0,08	0,15	0,05	0,10	0,18	0,00	0,01	0,02
Interval based	0,03	0,07	0,13	0,04	0,09	0,16	0,00	0,01	0,02
Dictionary based	0,03	0,07	0,13	0,04	0,09	0,15	0,00	0,01	0,02
Shapelet based	0,03	0,07	0,12	0,04	0,08	0,14	0,00	0,00	0,01
Convolution based	0,03	0,06	0,11	0,03	0,07	0,13	0,00	0,00	0,01
Hybrid approaches	0,03	0,06	0,11	0,03	0,07	0,12	0,00	0,00	0,01

Fonte: Autor.

### B.1.2 Perspectiva de conjuntos de dados

**Gráfico B.2 - Avaliação do KNN como meta-learner em LOO por conjunto de dados.**



Fonte: Autor.

**Tabela B.2** - Resultados do KNN como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
AUDIO	0,20	0,32	0,47	0,33	0,80	1,40	0,04	0,10	0,22
HEMODYNAMICS	0,06	0,11	0,28	0,06	0,14	0,70	0,00	0,01	0,08
MOTION	0,04	0,09	0,18	0,06	0,12	0,29	0,00	0,01	0,03
SPECTRO	0,05	0,09	0,14	0,05	0,10	0,17	0,00	0,01	0,02
EOG	0,04	0,08	0,15	0,05	0,11	0,20	0,00	0,01	0,02
IMAGE	0,04	0,08	0,15	0,04	0,09	0,18	0,00	0,01	0,02
SIMULATED	0,03	0,08	0,13	0,04	0,09	0,14	0,00	0,01	0,02
DEVICE	0,03	0,07	0,13	0,04	0,08	0,16	0,00	0,00	0,02
SENSOR	0,03	0,06	0,11	0,03	0,07	0,13	0,00	0,00	0,01
ECG	0,03	0,07	0,11	0,03	0,07	0,11	0,00	0,00	0,01
TRAFFIC	0,02	0,04	0,09	0,02	0,04	0,09	0,00	0,00	0,01
HAR	0,02	0,05	0,09	0,02	0,06	0,10	0,00	0,00	0,01
EPG	0,01	0,05	0,10	0,01	0,05	0,10	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.3** - Resultados do KNN como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,05	0,10	0,20	0,06	0,12	0,28	0,00	0,01	0,04
4 < C <= 10	0,05	0,09	0,16	0,06	0,11	0,20	0,00	0,01	0,02
2 < C <= 4	0,03	0,07	0,13	0,04	0,08	0,15	0,00	0,01	0,02
10 < C <= 20	0,03	0,07	0,13	0,04	0,08	0,15	0,00	0,00	0,02
classes <=2	0,03	0,06	0,11	0,03	0,07	0,12	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.4** - Resultados do KNN como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
1000 < L <= 2000	0,05	0,10	0,19	0,05	0,11	0,29	0,00	0,01	0,04
L > 2000	0,04	0,10	0,16	0,05	0,11	0,19	0,00	0,01	0,03
500 < L <= 1000	0,03	0,07	0,13	0,04	0,09	0,18	0,00	0,01	0,02
100 < L <= 500	0,03	0,07	0,12	0,03	0,08	0,13	0,00	0,01	0,01
length <= 100	0,03	0,06	0,12	0,03	0,07	0,15	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.5** - Resultados do KNN como meta-learner em LOO por número de amostras (size) nos conjuntos de dados.

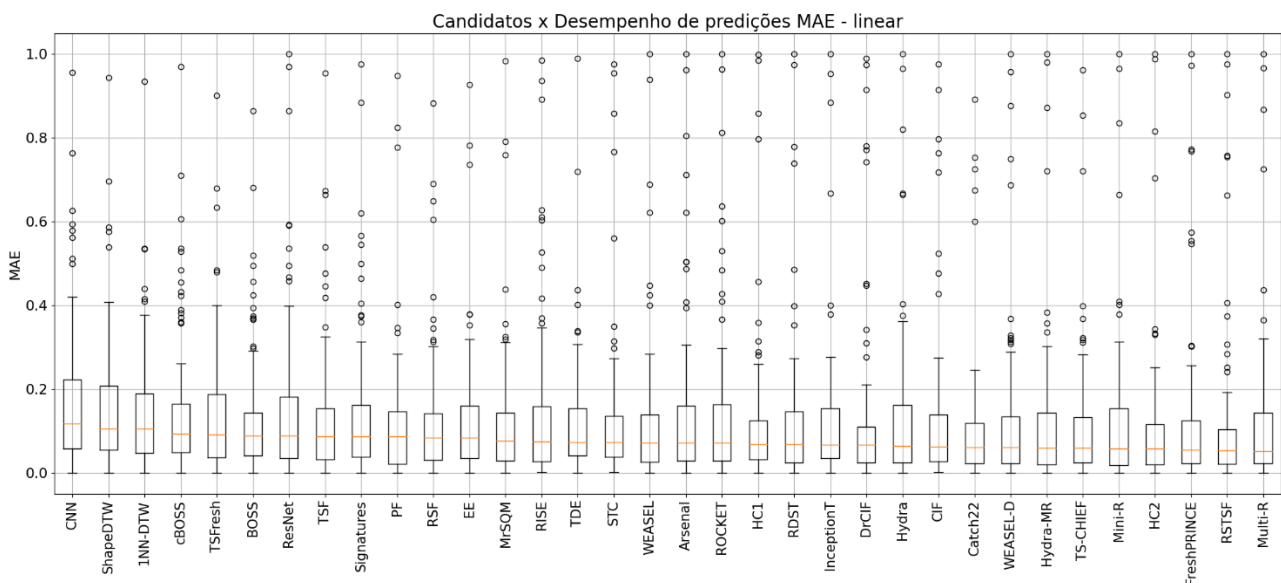
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
2000 < size <= 4000	0,04	0,08	0,15	0,04	0,09	0,17	0,00	0,01	0,02
1000 < size <= 2000	0,04	0,08	0,14	0,04	0,09	0,16	0,00	0,01	0,02
500 < size <= 1000	0,03	0,07	0,14	0,04	0,09	0,18	0,00	0,01	0,02
size <= 500	0,03	0,07	0,12	0,03	0,08	0,14	0,00	0,01	0,01
size > 4000	0,03	0,07	0,13	0,04	0,08	0,15	0,00	0,00	0,02

Fonte: Autor.

## B.2. Resultados Para A Regressão Linear

### B.2.1 Perspectiva de classificadores candidatos

**Gráfico B.3** - Desempenho das predições da regressão linear como meta-learner por candidato.



Fonte: Autor.

**Tabela B.6** - Resultados da regressão linear como meta-learner na LOO por grupo de classificadores de séries temporais.

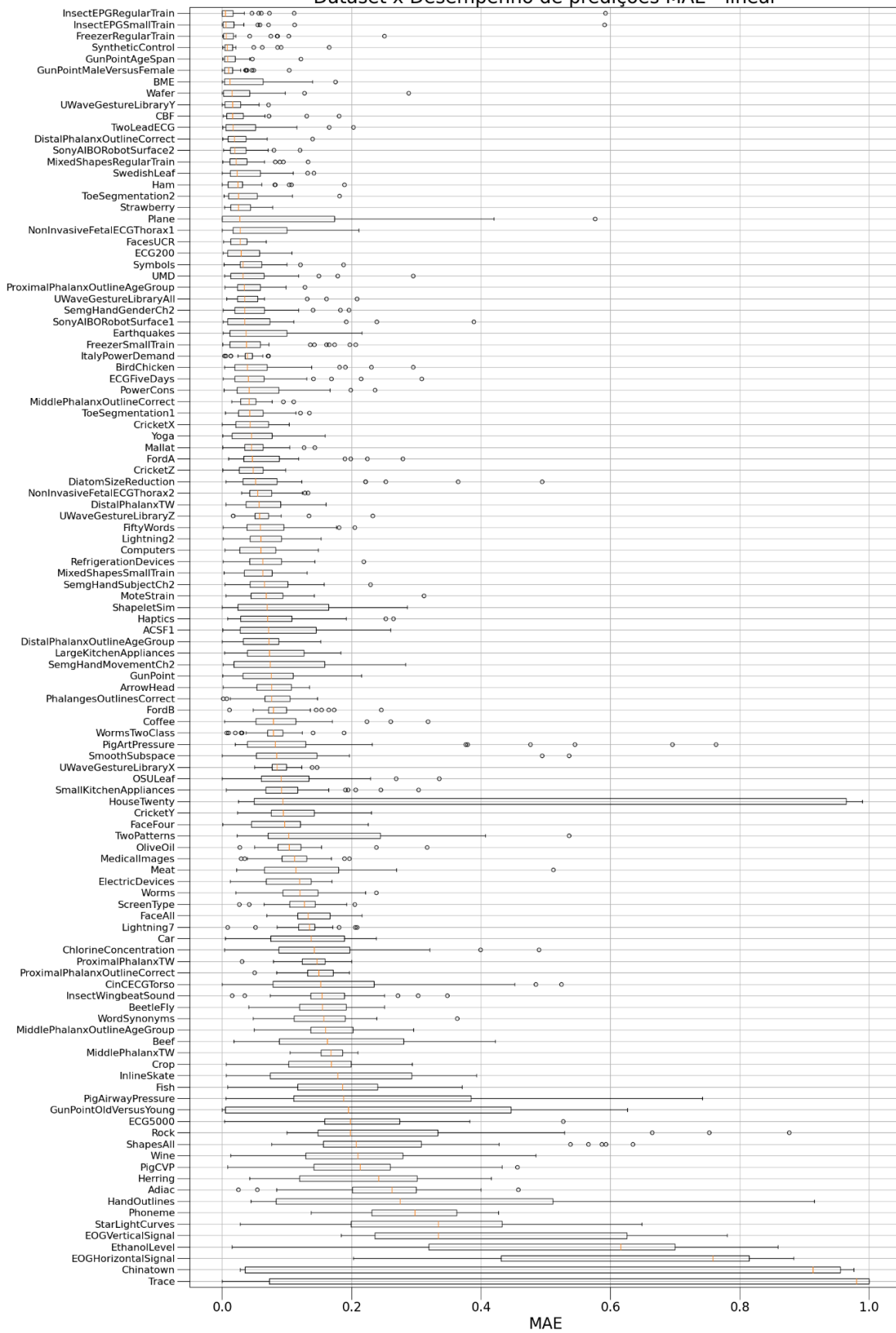
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Distance based	0,04	0,10	0,18	0,04	0,13	0,25	0,00	0,01	0,03
Deep learning based	0,04	0,09	0,18	0,04	0,11	0,23	0,00	0,01	0,03
Dictionary based	0,04	0,08	0,15	0,04	0,09	0,19	0,00	0,01	0,02
Shapelet based	0,03	0,08	0,14	0,04	0,09	0,18	0,00	0,01	0,02
Feature based	0,03	0,08	0,14	0,03	0,09	0,19	0,00	0,01	0,02
Convolution based	0,02	0,06	0,16	0,02	0,07	0,19	0,00	0,00	0,02
Interval based	0,03	0,07	0,13	0,03	0,08	0,17	0,00	0,00	0,02
Hybrid approaches	0,02	0,06	0,13	0,03	0,07	0,15	0,00	0,00	0,02

Fonte: Autor.

### *B.2.2 Perspectiva de conjuntos de dados*

**Gráfico B.4** - Avaliação da regressão linear como meta-learner em LOO por conjunto de dados.

Dataset x Desempenho de predições MAE - linear



Fonte: Autor.

**Tabela B.7** - Resultados da regressão linear como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
TRAFFIC	0,03	0,91	0,96	0,04	1,00	1,00	0,00	0,83	0,92
EOG	0,27	0,61	0,78	0,38	1,00	1,00	0,08	0,37	0,60
AUDIO	0,15	0,22	0,31	0,25	0,51	0,98	0,02	0,05	0,09
HEMODYNAMICS	0,07	0,15	0,28	0,09	0,23	0,59	0,01	0,02	0,08
SPECTRO	0,04	0,10	0,19	0,04	0,11	0,24	0,00	0,01	0,04
IMAGE	0,04	0,09	0,16	0,04	0,11	0,20	0,00	0,01	0,02
DEVICE	0,04	0,08	0,13	0,05	0,10	0,17	0,00	0,01	0,02
MOTION	0,03	0,07	0,12	0,04	0,10	0,19	0,00	0,01	0,01
ECG	0,02	0,05	0,13	0,02	0,06	0,14	0,00	0,00	0,02
SENSOR	0,02	0,06	0,13	0,02	0,07	0,16	0,00	0,00	0,02
SIMULATED	0,01	0,05	0,11	0,01	0,05	0,13	0,00	0,00	0,01
HAR	0,01	0,05	0,08	0,02	0,05	0,10	0,00	0,00	0,01
EPG	0,00	0,00	0,02	0,00	0,00	0,02	0,00	0,00	0,00

Fonte: Autor.

**Tabela B.8** - Resultados da regressão linear como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,07	0,15	0,25	0,08	0,20	0,36	0,00	0,02	0,06
2 < C <= 4	0,04	0,09	0,16	0,04	0,11	0,21	0,00	0,01	0,03
10 < C <= 20	0,03	0,08	0,20	0,04	0,10	0,24	0,00	0,01	0,04
4 < C <= 10	0,03	0,08	0,15	0,04	0,10	0,20	0,00	0,01	0,02
classes <=2	0,02	0,05	0,10	0,02	0,05	0,12	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.9** - Resultados da regressão linear como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
L > 2000	0,08	0,19	0,53	0,09	0,24	0,64	0,01	0,04	0,28
1000 < L <= 2000	0,04	0,11	0,29	0,05	0,17	0,51	0,00	0,01	0,09
500 < L <= 1000	0,04	0,08	0,14	0,04	0,10	0,18	0,00	0,01	0,02
length <= 100	0,03	0,07	0,14	0,03	0,08	0,17	0,00	0,00	0,02
100 < L <= 500	0,02	0,06	0,13	0,02	0,07	0,15	0,00	0,00	0,02

Fonte: Autor.

**Tabela B.10** - Resultados da regressão linear como meta-learner em LOO por número de

amostras (size) nos conjuntos de dados.

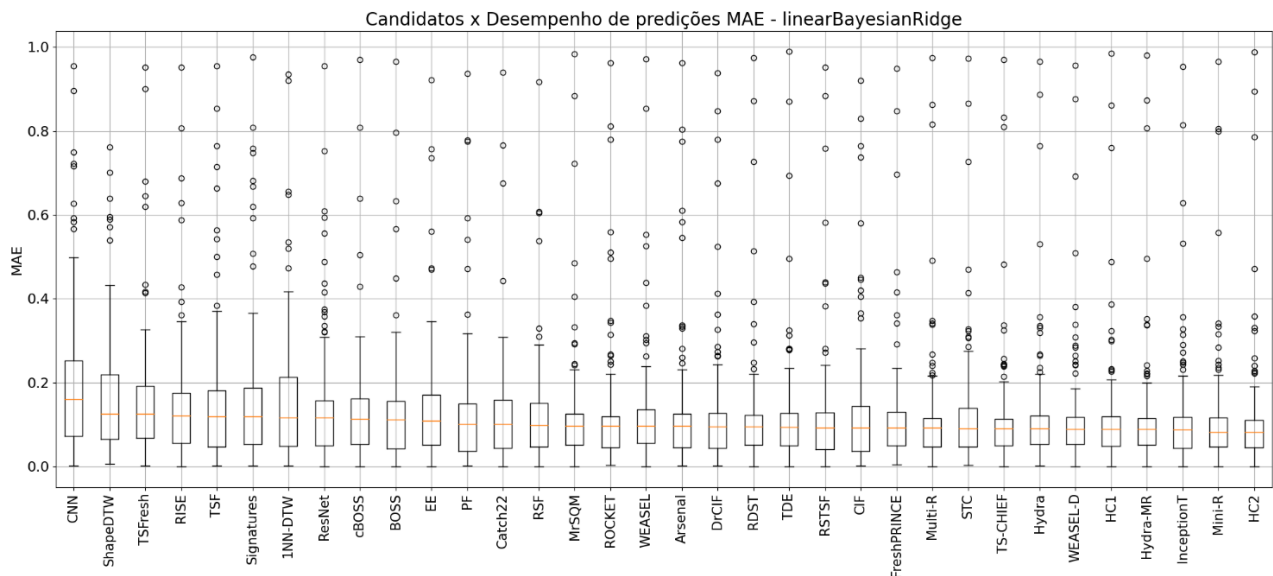
Métrica	MAE			MAPE			MSquareE			
	Num. de ST	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
size > 4000		0,04	0,09	0,16	0,05	0,10	0,21	0,00	0,01	0,03
1000 < size <= 2000		0,04	0,08	0,20	0,04	0,09	0,22	0,00	0,01	0,04
size <= 500		0,03	0,08	0,15	0,03	0,09	0,19	0,00	0,01	0,02
500 < size <= 1000		0,03	0,07	0,15	0,03	0,09	0,20	0,00	0,01	0,02
2000 < size <= 4000		0,02	0,05	0,11	0,02	0,06	0,12	0,00	0,00	0,01

Fonte: Autor.

### B.3. Resultados Para A Regressão Linear Bayesian Ridge

#### B.3.1 Perspectiva de classificadores candidatos

**Gráfico B.5** - Desempenho das predições da regressão linear Bayesian Ridge como meta-learner por candidato.



Fonte: Autor.

**Tabela B.11** - Resultados da regressão linear Bayesian Ridge como meta-learner na LOO por grupo de classificadores de séries temporais.



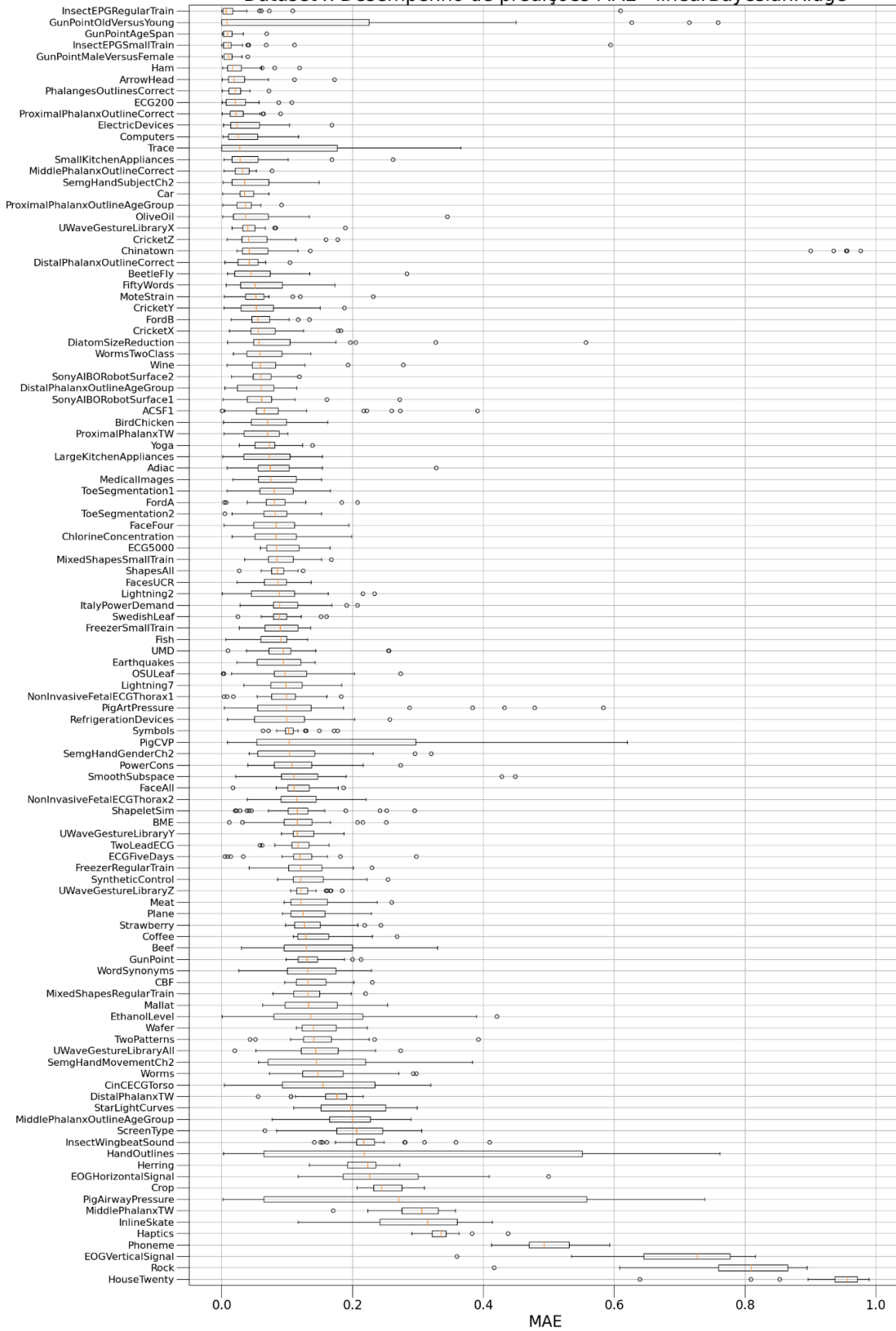
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Deep learning based	0,05	0,11	0,19	0,06	0,12	0,25	0,00	0,01	0,03
Distance based	0,05	0,11	0,19	0,06	0,14	0,23	0,00	0,01	0,04
Feature based	0,05	0,11	0,17	0,06	0,12	0,20	0,00	0,01	0,03
Dictionary based	0,05	0,10	0,14	0,06	0,11	0,16	0,00	0,01	0,02
Shapelet based	0,05	0,09	0,13	0,06	0,10	0,15	0,00	0,01	0,02
Convolution based	0,05	0,09	0,12	0,05	0,10	0,13	0,00	0,01	0,01
Interval based	0,04	0,10	0,16	0,05	0,11	0,19	0,00	0,01	0,02
Hybrid approaches	0,05	0,09	0,11	0,05	0,09	0,12	0,00	0,01	0,01

Fonte: Autor.

### B.3.2 Perspectiva de conjuntos de dados

**Gráfico B.6** - Avaliação da regressão linear Bayesian Ridge como meta-learner em LOO por conjunto de dados.

Dataset x Desempenho de predições MAE - linearBayesianRidge



Fonte: Autor.

**Tabela B.12** - Resultados da regressão linear Bayesian Ridge como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
EOG	0,22	0,45	0,73	0,29	1,00	1,00	0,05	0,21	0,53
AUDIO	0,22	0,41	0,49	0,34	1,06	1,50	0,05	0,17	0,24
MOTION	0,08	0,12	0,30	0,08	0,16	0,60	0,01	0,01	0,09
SIMULATED	0,09	0,12	0,15	0,10	0,12	0,16	0,01	0,01	0,02
HEMODYNAMICS	0,05	0,12	0,41	0,06	0,14	0,97	0,00	0,02	0,17
SPECTRO	0,05	0,11	0,17	0,06	0,12	0,21	0,00	0,01	0,03
ECG	0,07	0,11	0,13	0,07	0,11	0,14	0,00	0,01	0,02
SENSOR	0,05	0,09	0,13	0,06	0,10	0,14	0,00	0,01	0,02
IMAGE	0,04	0,08	0,13	0,05	0,09	0,16	0,00	0,01	0,02
DEVICE	0,03	0,08	0,16	0,04	0,10	0,23	0,00	0,01	0,03
TRAFFIC	0,03	0,04	0,08	0,03	0,04	0,09	0,00	0,00	0,01
HAR	0,02	0,07	0,12	0,03	0,09	0,15	0,00	0,00	0,01
EPG	0,00	0,01	0,02	0,00	0,01	0,02	0,00	0,00	0,00

Fonte: Autor.

**Tabela B.13** - Resultados da regressão linear Bayesian Ridge como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,07	0,11	0,23	0,08	0,13	0,32	0,01	0,01	0,05
4 < C ≤ 10	0,08	0,11	0,17	0,09	0,13	0,24	0,01	0,01	0,03
10 < C ≤ 20	0,06	0,09	0,18	0,07	0,11	0,21	0,00	0,01	0,03
2 < C ≤ 4	0,03	0,10	0,15	0,04	0,11	0,18	0,00	0,01	0,02
classes ≤ 2	0,03	0,07	0,12	0,04	0,08	0,13	0,00	0,01	0,01

Fonte: Autor.

**Tabela B.14** - Resultados da regressão linear Bayesian Ridge como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
L > 2000	0,52	0,81	0,94	0,61	1,00	1,00	0,27	0,65	0,88
1000 < L ≤ 2000	0,09	0,15	0,32	0,10	0,19	0,61	0,01	0,02	0,10
100 < L ≤ 500	0,05	0,10	0,12	0,06	0,11	0,14	0,00	0,01	0,02
500 < L ≤ 1000	0,03	0,08	0,13	0,04	0,09	0,17	0,00	0,01	0,02
length ≤ 100	0,03	0,07	0,12	0,04	0,08	0,15	0,00	0,00	0,02

Fonte: Autor.

**Tabela B.15** - Resultados da regressão linear Bayesian Ridge como meta-learner em LOO por número de amostras (size) nos conjuntos de dados.

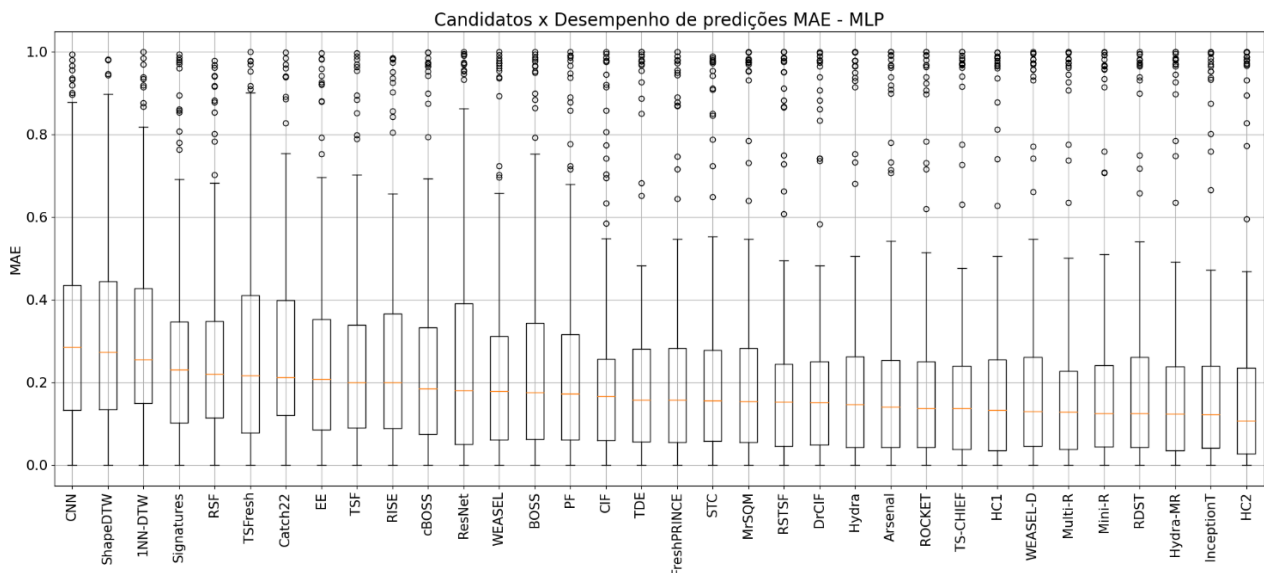
Métrica	MAE			MAPE			MSquareE		
Num. de ST	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
2000 < size <= 4000	0,08	0,10	0,14	0,08	0,11	0,15	0,01	0,01	0,02
1000 < size <= 2000	0,07	0,10	0,13	0,08	0,11	0,16	0,01	0,01	0,02
size > 4000	0,07	0,12	0,18	0,08	0,14	0,21	0,00	0,01	0,03
500 < size <= 1000	0,04	0,09	0,17	0,05	0,11	0,23	0,00	0,01	0,03
size <= 500	0,03	0,09	0,14	0,04	0,10	0,16	0,00	0,01	0,02

Fonte: Autor.

## B.4. Resultados Para O MLP

### B.4.1 Perspectiva de classificadores candidatos

**Gráfico B.7** - Desempenho das predições do MLP como meta-learner por candidato.



Fonte: Autor.

**Tabela B.16** - Resultados do MLP como meta-learner na LOO por grupo de classificadores de séries temporais.

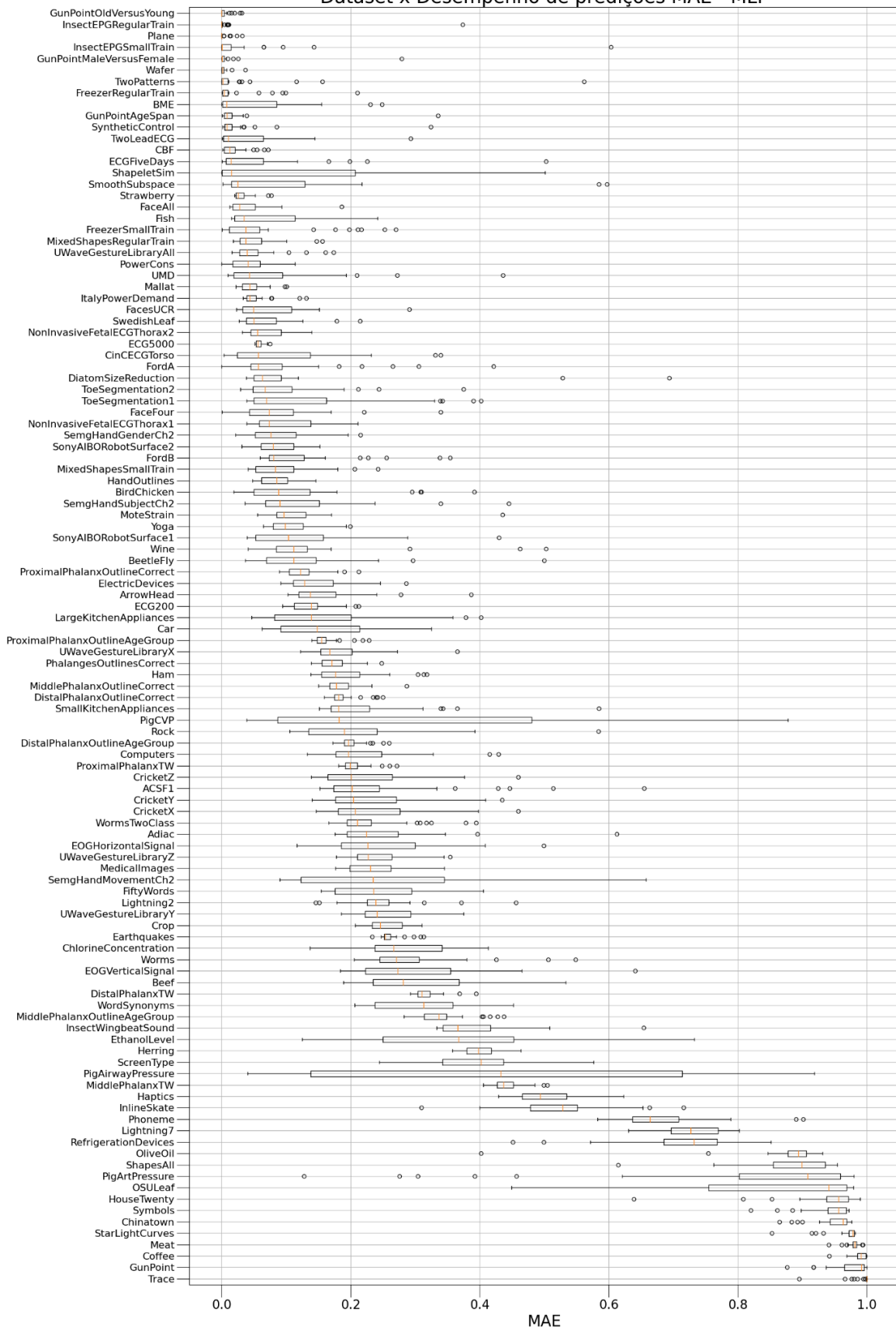
Métrica	MAE			MAPE			MSquareE		
Grupo	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Distance based	0,10	0,22	0,41	0,11	0,28	0,71	0,01	0,05	0,17
Feature based	0,09	0,20	0,36	0,10	0,26	0,58	0,01	0,04	0,13
Deep learning based	0,06	0,18	0,40	0,06	0,23	0,68	0,00	0,03	0,16
Interval based	0,06	0,18	0,31	0,07	0,21	0,45	0,00	0,03	0,10
Dictionary based	0,06	0,17	0,31	0,06	0,20	0,45	0,00	0,03	0,10
Shapelet based	0,06	0,17	0,30	0,06	0,20	0,43	0,00	0,03	0,09
Convolution based	0,04	0,13	0,25	0,04	0,15	0,33	0,00	0,02	0,06
Hybrid approaches	0,03	0,13	0,24	0,04	0,14	0,32	0,00	0,02	0,06

Fonte: Autor.

#### *B.4.2 Perspectiva de conjuntos de dados*

**Gráfico B.8** - Avaliação do MLP como meta-learner em LOO por conjunto de dados.

Dataset x Desempenho de predições MAE - MLP



Fonte: Autor.

**Tabela B.17** - Resultados do MLP como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
TRAFFIC	0,94	0,96	0,97	1,00	1,00	1,00	0,89	0,93	0,94
HEMODYNAMICS	0,14	0,55	0,87	0,17	1,00	1,00	0,02	0,31	0,75
AUDIO	0,36	0,59	0,66	0,57	1,43	1,98	0,13	0,35	0,44
MOTION	0,15	0,26	0,47	0,17	0,36	0,90	0,02	0,07	0,22
EOG	0,20	0,25	0,32	0,25	0,32	0,47	0,04	0,06	0,10
DEVICE	0,14	0,21	0,46	0,16	0,26	0,94	0,02	0,04	0,21
SPECTRO	0,11	0,22	0,73	0,13	0,28	1,00	0,01	0,05	0,53
IMAGE	0,09	0,18	0,30	0,10	0,22	0,44	0,01	0,03	0,09
HAR	0,02	0,18	0,26	0,02	0,23	0,36	0,00	0,03	0,07
SENSOR	0,04	0,10	0,26	0,04	0,11	0,35	0,00	0,01	0,07
ECG	0,04	0,06	0,12	0,04	0,06	0,13	0,00	0,00	0,01
SIMULATED	0,01	0,02	0,10	0,01	0,03	0,11	0,00	0,00	0,01
EPG	0,00	0,00	0,01	0,00	0,00	0,01	0,00	0,00	0,00

Fonte: Autor.

**Tabela B.18** - Resultados do MLP como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,16	0,28	0,68	0,20	0,38	1,00	0,03	0,08	0,46
4 < C ≤ 10	0,07	0,22	0,41	0,08	0,28	0,68	0,01	0,05	0,16
2 < C ≤ 4	0,05	0,17	0,39	0,05	0,21	0,65	0,00	0,03	0,16
10 < C ≤ 20	0,06	0,18	0,25	0,06	0,22	0,33	0,00	0,03	0,06
classes ≤ 2	0,04	0,11	0,21	0,04	0,13	0,27	0,00	0,01	0,05

Fonte: Autor.

**Tabela B.19** - Resultados do MLP como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
1000 < L ≤ 2000	0,09	0,23	0,54	0,10	0,31	1,00	0,01	0,06	0,29
500 < L ≤ 1000	0,08	0,22	0,37	0,09	0,28	0,60	0,01	0,05	0,14
L > 2000	0,10	0,19	0,94	0,11	0,24	1,00	0,01	0,04	0,88
length ≤ 100	0,10	0,17	0,24	0,11	0,21	0,32	0,01	0,03	0,06
100 < L ≤ 500	0,03	0,12	0,27	0,03	0,13	0,36	0,00	0,01	0,07

Fonte: Autor.

**Tabela B.20** - Resultados do MLP como meta-learner em LOO por número de amostras

(size) nos conjuntos de dados.

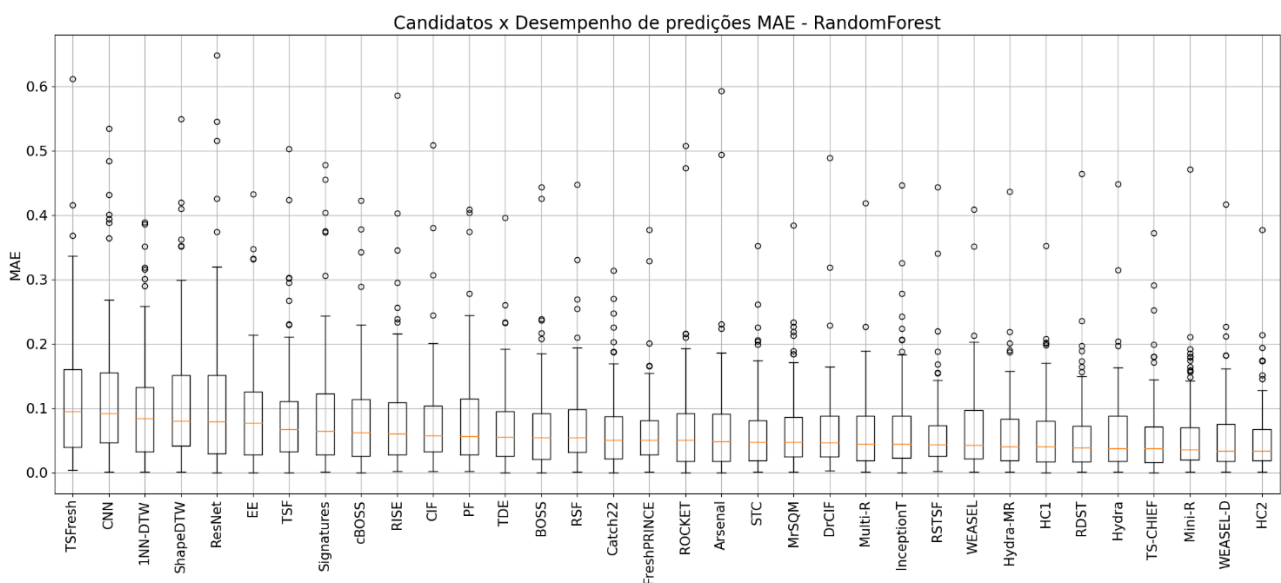
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
500 < size <= 1000	0,12	0,19	0,31	0,13	0,24	0,45	0,01	0,04	0,10
size <= 500	0,05	0,19	0,50	0,06	0,23	0,98	0,00	0,03	0,25
1000 < size <= 2000	0,05	0,12	0,38	0,06	0,14	0,61	0,00	0,01	0,14
size > 4000	0,05	0,17	0,28	0,06	0,21	0,38	0,00	0,03	0,08
2000 < size <= 4000	0,03	0,06	0,14	0,04	0,07	0,16	0,00	0,00	0,02

Fonte: Autor.

## B.5. Resultados Para O Randomforest

### B.5.1 Perspectiva de classificadores candidatos

**Gráfico B.9** - Desempenho das previsões do RandomForest como meta-learner por candidato.



Fonte: Autor.

**Tabela B.21** - Resultados do RandomForest como meta-learner na LOO por grupo de classificadores de séries temporais.



Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Distance based	0,03	0,07	0,13	0,04	0,09	0,17	0,00	0,01	0,02
Deep learning based	0,03	0,07	0,13	0,03	0,08	0,16	0,00	0,01	0,02
Feature based	0,03	0,06	0,11	0,03	0,07	0,14	0,00	0,00	0,01
Interval based	0,03	0,05	0,10	0,03	0,06	0,12	0,00	0,00	0,01
Shapelet based	0,02	0,05	0,08	0,03	0,05	0,10	0,00	0,00	0,01
Dictionary based	0,02	0,05	0,10	0,03	0,06	0,11	0,00	0,00	0,01
Convolution based	0,02	0,04	0,09	0,02	0,05	0,10	0,00	0,00	0,01
Hybrid approaches	0,02	0,04	0,08	0,02	0,04	0,09	0,00	0,00	0,01

Fonte: Autor.

## 5.2 Perspectiva de conjuntos de dados

**Gráfico B.10** - Avaliação do RandomForest como meta-learner em LOO por conjunto de dados.



Fonte: Autor.

**Tabela B.22** - Resultados do RandomForest como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
AUDIO	0,12	0,32	0,41	0,18	0,82	1,23	0,01	0,10	0,17
HEMODYNAMICS	0,06	0,10	0,24	0,07	0,13	0,69	0,00	0,01	0,06
SPECTRO	0,04	0,07	0,12	0,04	0,08	0,15	0,00	0,00	0,01
MOTION	0,02	0,06	0,11	0,03	0,09	0,22	0,00	0,00	0,01
ECG	0,03	0,05	0,12	0,03	0,05	0,12	0,00	0,00	0,01
IMAGE	0,02	0,06	0,12	0,03	0,07	0,13	0,00	0,00	0,01
SENSOR	0,02	0,05	0,08	0,03	0,05	0,10	0,00	0,00	0,01
SIMULATED	0,02	0,05	0,09	0,02	0,05	0,10	0,00	0,00	0,01
EOG	0,02	0,04	0,07	0,02	0,05	0,09	0,00	0,00	0,00
DEVICE	0,02	0,04	0,08	0,02	0,05	0,10	0,00	0,00	0,01
HAR	0,02	0,03	0,06	0,02	0,04	0,07	0,00	0,00	0,00
EPG	0,01	0,02	0,05	0,01	0,02	0,05	0,00	0,00	0,00
TRAFFIC	0,01	0,03	0,04	0,01	0,03	0,04	0,00	0,00	0,00

Fonte: Autor.

**Tabela B.23** - Resultados do RandomForest como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,04	0,07	0,14	0,05	0,09	0,18	0,00	0,01	0,02
4 < C <= 10	0,03	0,07	0,12	0,04	0,09	0,16	0,00	0,00	0,02
10 < C <= 20	0,02	0,06	0,11	0,03	0,07	0,12	0,00	0,00	0,01
2 < C <= 4	0,02	0,05	0,09	0,02	0,05	0,10	0,00	0,00	0,01
classes <=2	0,02	0,04	0,08	0,02	0,05	0,09	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.24** - Resultados do RandomForest como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
L > 2000	0,03	0,08	0,14	0,04	0,09	0,16	0,00	0,01	0,02
1000 < L <= 2000	0,03	0,06	0,13	0,04	0,07	0,22	0,00	0,00	0,02
500 < L <= 1000	0,03	0,06	0,10	0,03	0,07	0,13	0,00	0,00	0,01
100 < L <= 500	0,02	0,05	0,10	0,03	0,06	0,11	0,00	0,00	0,01
length <= 100	0,02	0,04	0,09	0,02	0,05	0,11	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.25** - Resultados do RandomForest como meta-learner em LOO por número de amostras (size) nos conjuntos de dados.

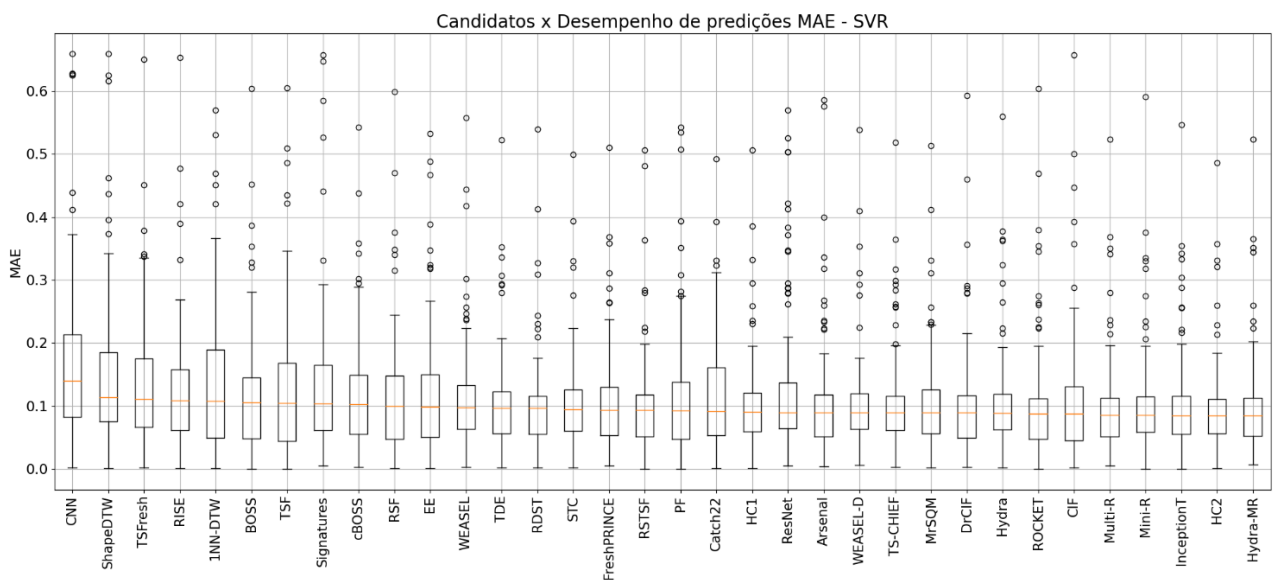
Métrica	MAE			MAPE			MSquareE		
Num. de ST	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
2000 < size <= 4000	0,03	0,07	0,13	0,03	0,07	0,15	0,00	0,00	0,02
1000 < size <= 2000	0,03	0,06	0,10	0,03	0,06	0,12	0,00	0,00	0,01
size > 4000	0,03	0,05	0,10	0,03	0,07	0,11	0,00	0,00	0,01
size <= 500	0,02	0,05	0,10	0,02	0,06	0,12	0,00	0,00	0,01
500 < size <= 1000	0,02	0,05	0,09	0,03	0,06	0,12	0,00	0,00	0,01

Fonte: Autor.

## B.6. Resultados Para O SVR

### B.6.1 Perspectiva de classificadores candidatos

**Gráfico B.11** - Desempenho das predições do SVR como meta-learner por candidato.



Fonte: Autor.

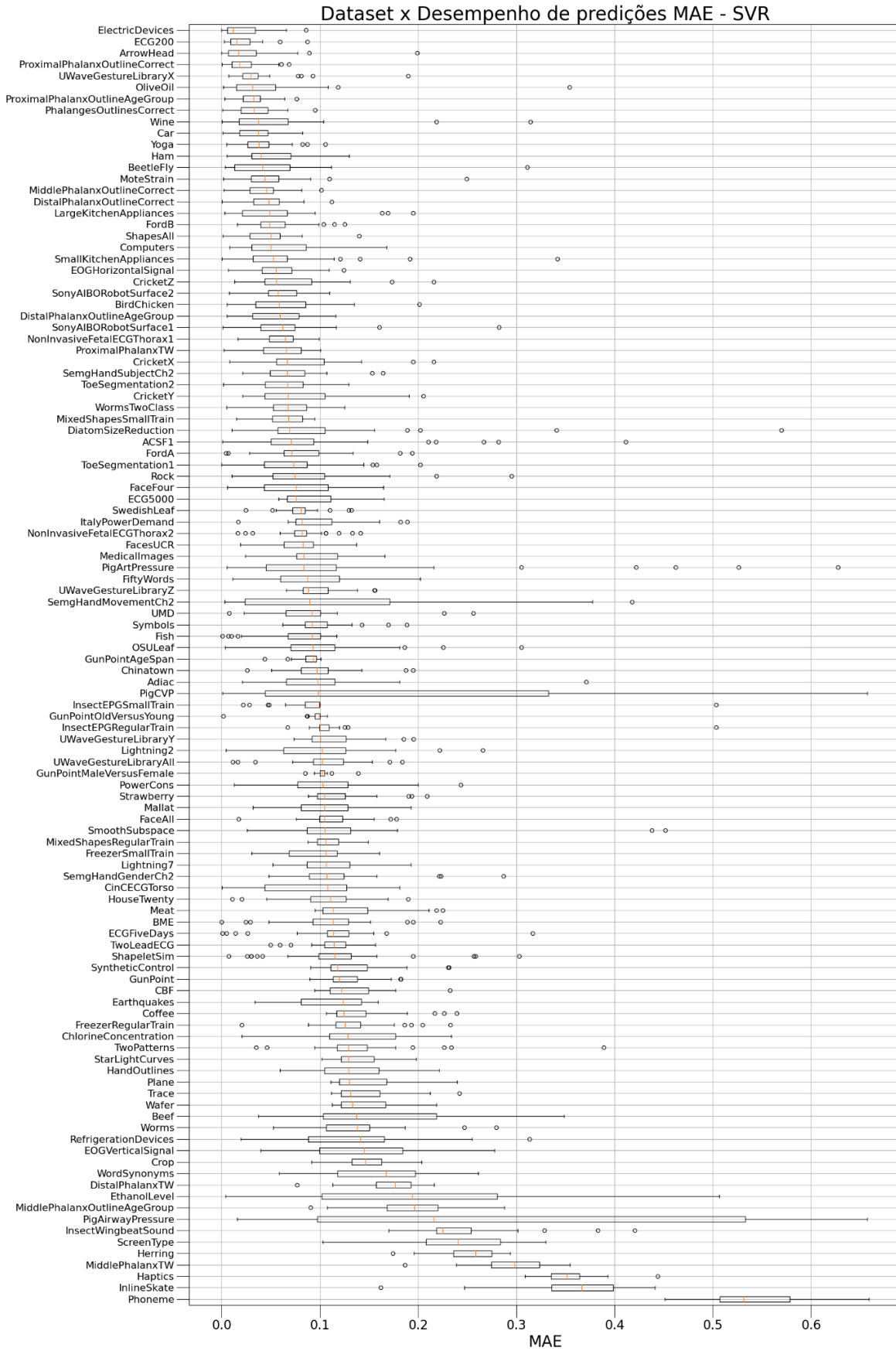
**Tabela B.26** - Resultados do SVR como meta-learner na LOO por grupo de classificadores de séries temporais.

Métrica	MAE			MAPE			MSquareE		
Grupo	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Deep learning based	0,06	0,10	0,16	0,07	0,11	0,20	0,00	0,01	0,03
Distance based	0,05	0,10	0,16	0,06	0,12	0,20	0,00	0,01	0,03
Dictionary based	0,06	0,10	0,13	0,07	0,11	0,14	0,00	0,01	0,02
Feature based	0,06	0,10	0,16	0,07	0,11	0,18	0,00	0,01	0,02
Hybrid approaches	0,06	0,09	0,11	0,07	0,09	0,12	0,00	0,01	0,01
Shapelet based	0,06	0,09	0,13	0,06	0,10	0,14	0,00	0,01	0,02
Interval based	0,05	0,10	0,13	0,06	0,10	0,16	0,00	0,01	0,02
Convolution based	0,05	0,09	0,11	0,06	0,10	0,12	0,00	0,01	0,01

Fonte: Autor.

### *B.6.2 Perspectiva de conjuntos de dados*

**Gráfico B.12** - Avaliação do SVR como meta-learner em LOO por conjunto de dados.



Fonte: Autor.

**Tabela B.27** - Resultados do SVR como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
AUDIO	0,22	0,44	0,53	0,35	1,15	1,61	0,05	0,19	0,28
SIMULATED	0,10	0,12	0,14	0,10	0,12	0,15	0,01	0,01	0,02
EPG	0,09	0,10	0,10	0,10	0,10	0,10	0,01	0,01	0,01
MOTION	0,07	0,11	0,34	0,08	0,14	0,65	0,00	0,01	0,11
HEMODYNAMICS	0,05	0,10	0,39	0,06	0,12	0,91	0,00	0,01	0,15
TRAFFIC	0,08	0,10	0,11	0,08	0,10	0,11	0,01	0,01	0,01
HAR	0,07	0,10	0,11	0,08	0,10	0,13	0,00	0,01	0,01
EOG	0,05	0,08	0,15	0,07	0,11	0,20	0,00	0,01	0,02
SENSOR	0,06	0,10	0,13	0,07	0,11	0,14	0,00	0,01	0,02
SPECTRO	0,05	0,10	0,13	0,05	0,10	0,16	0,00	0,01	0,02
ECG	0,05	0,08	0,11	0,06	0,09	0,12	0,00	0,01	0,01
IMAGE	0,04	0,08	0,12	0,05	0,09	0,15	0,00	0,01	0,01
DEVICE	0,04	0,07	0,14	0,05	0,09	0,17	0,00	0,01	0,02

Fonte: Autor.

**Tabela B.28** - Resultados do SVR como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
4 < C ≤ 10	0,07	0,10	0,16	0,08	0,12	0,22	0,01	0,01	0,03
C > 20	0,06	0,10	0,18	0,07	0,12	0,27	0,00	0,01	0,03
10 < C ≤ 20	0,06	0,08	0,11	0,07	0,09	0,13	0,00	0,01	0,01
2 < C ≤ 4	0,05	0,10	0,14	0,06	0,11	0,15	0,00	0,01	0,02
classes ≤ 2	0,05	0,09	0,12	0,05	0,09	0,12	0,00	0,01	0,01

Fonte: Autor.

**Tabela B.29** - Resultados do SVR como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
1000 < L ≤ 2000	0,07	0,11	0,26	0,08	0,12	0,46	0,01	0,01	0,07
L > 2000	0,07	0,11	0,14	0,09	0,12	0,15	0,01	0,01	0,02
100 < L ≤ 500	0,07	0,10	0,12	0,07	0,11	0,14	0,00	0,01	0,02
500 < L ≤ 1000	0,05	0,08	0,13	0,06	0,09	0,18	0,00	0,01	0,02
length ≤ 100	0,03	0,07	0,12	0,04	0,08	0,14	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.30** - Resultados do SVR como meta-learner em LOO por número de amostras

(size) nos conjuntos de dados.

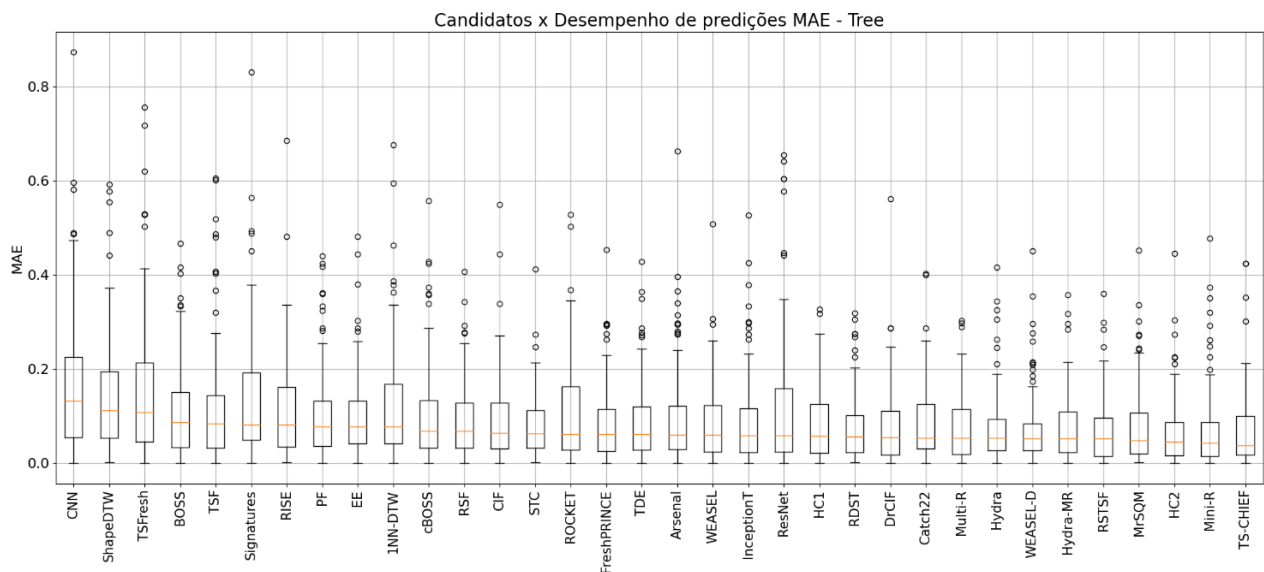
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
size > 4000	0,07	0,11	0,14	0,07	0,12	0,17	0,00	0,01	0,02
1000 < size <= 2000	0,06	0,09	0,12	0,07	0,10	0,14	0,00	0,01	0,02
2000 < size <= 4000	0,06	0,09	0,12	0,06	0,09	0,12	0,00	0,01	0,01
size <= 500	0,06	0,10	0,13	0,07	0,10	0,14	0,00	0,01	0,02
500 < size <= 1000	0,05	0,09	0,15	0,06	0,11	0,20	0,00	0,01	0,02

Fonte: Autor.

## B.7. Resultados Para O Tree

### B.7.1 Perspectiva de classificadores candidatos

**Gráfico B.12** - Desempenho das predições do Tree como meta-learner por candidato.



Fonte: Autor.

**Tabela B.31** - Resultados do Tree como meta-learner na LOO por grupo de classificadores de séries temporais.



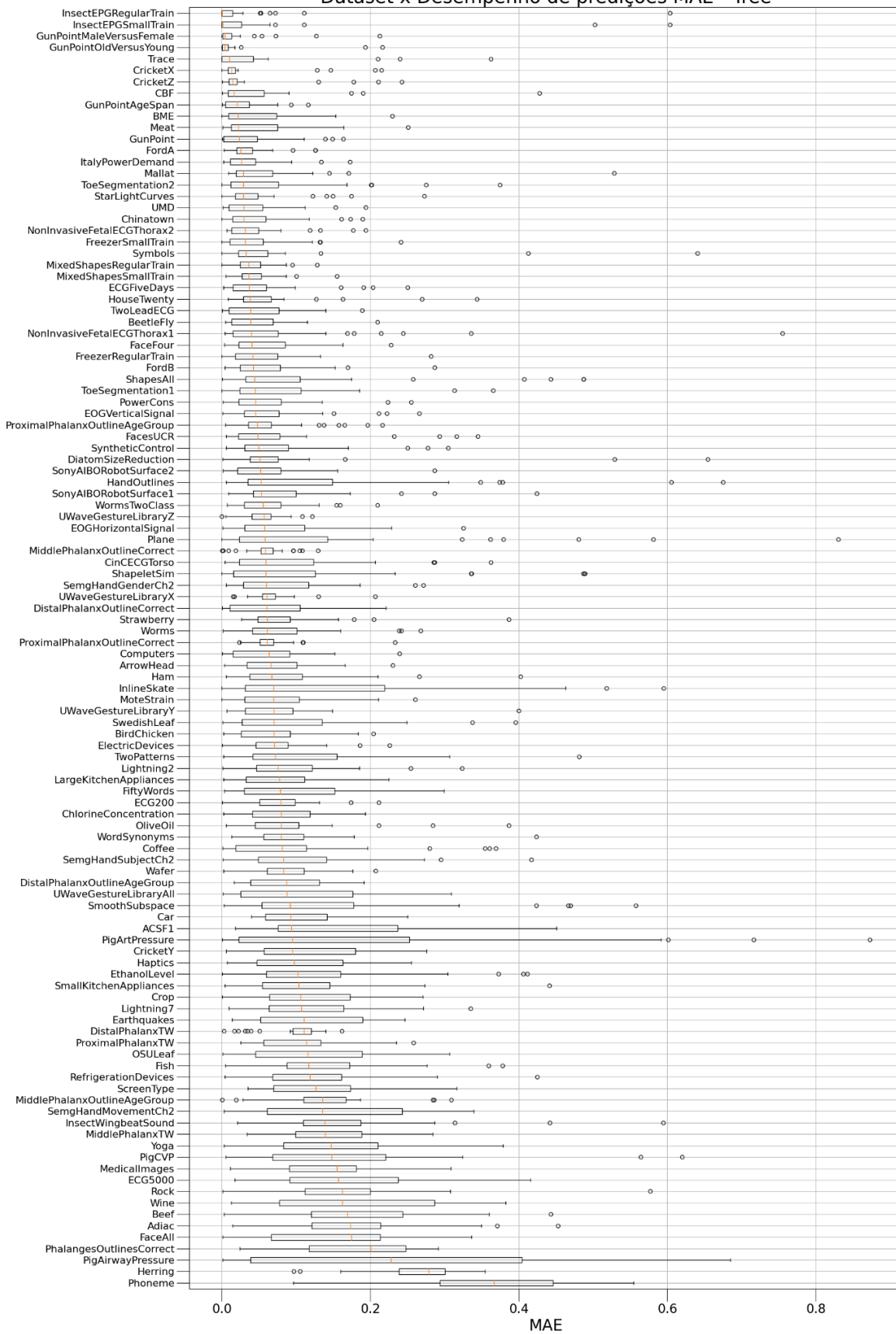
Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
Distance based	0,04	0,08	0,15	0,05	0,11	0,21	0,00	0,01	0,02
Feature based	0,04	0,07	0,15	0,04	0,09	0,21	0,00	0,01	0,02
Deep learning based	0,03	0,07	0,18	0,03	0,09	0,24	0,00	0,01	0,03
Dictionary based	0,03	0,06	0,12	0,03	0,07	0,16	0,00	0,00	0,02
Interval based	0,02	0,07	0,13	0,02	0,08	0,16	0,00	0,00	0,02
Shapelet based	0,02	0,06	0,11	0,03	0,07	0,14	0,00	0,00	0,01
Convolution based	0,02	0,05	0,11	0,02	0,06	0,14	0,00	0,00	0,01
Hybrid approaches	0,02	0,05	0,10	0,02	0,05	0,12	0,00	0,00	0,01

Fonte: Autor.

### B.7.2 Perspectiva de conjuntos de dados

**Gráfico B.13** - Avaliação do Tree como meta-learner em LOO por conjunto de dados.

Dataset x Desempenho de predições MAE - Tree



Fonte: Autor.

**Tabela B.32** - Resultados do Tree como meta-learner em LOO por tipo de conjunto de dados.

Métrica Tipo	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
AUDIO	0,14	0,26	0,40	0,22	0,55	1,24	0,02	0,07	0,16
HEMODYNAMICS	0,04	0,13	0,28	0,04	0,15	0,62	0,00	0,02	0,08
SPECTRO	0,04	0,09	0,16	0,05	0,10	0,21	0,00	0,01	0,03
DEVICE	0,04	0,07	0,13	0,05	0,10	0,17	0,00	0,01	0,02
IMAGE	0,04	0,08	0,15	0,04	0,10	0,19	0,00	0,01	0,02
MOTION	0,03	0,06	0,13	0,04	0,08	0,19	0,00	0,00	0,02
EOG	0,03	0,05	0,10	0,04	0,07	0,13	0,00	0,00	0,01
ECG	0,02	0,05	0,11	0,02	0,06	0,12	0,00	0,00	0,01
SENSOR	0,02	0,05	0,10	0,03	0,06	0,12	0,00	0,00	0,01
SIMULATED	0,02	0,05	0,10	0,02	0,05	0,11	0,00	0,00	0,01
TRAFFIC	0,01	0,03	0,06	0,01	0,03	0,06	0,00	0,00	0,00
HAR	0,01	0,03	0,07	0,01	0,03	0,09	0,00	0,00	0,01
EPG	0,00	0,00	0,02	0,00	0,00	0,02	0,00	0,00	0,00

Fonte: Autor.

**Tabela B.33** - Resultados do Tree como meta-learner em LOO por número de rótulos (classes) presentes no conjunto de dados.

Métrica Num. Classes	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
C > 20	0,04	0,09	0,22	0,04	0,12	0,30	0,00	0,01	0,05
4 < C ≤ 10	0,04	0,08	0,14	0,05	0,10	0,20	0,00	0,01	0,02
2 < C ≤ 4	0,02	0,06	0,13	0,02	0,07	0,16	0,00	0,00	0,02
classes ≤ 2	0,02	0,05	0,10	0,02	0,06	0,12	0,00	0,00	0,01
10 < C ≤ 20	0,02	0,05	0,13	0,02	0,06	0,16	0,00	0,00	0,02

Fonte: Autor.

**Tabela B.34** - Resultados do Tree como meta-learner em LOO por comprimento (length) da ST do conjunto de dados.

Métrica Comprimento da ST	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
length ≤ 100	0,04	0,07	0,13	0,05	0,08	0,17	0,00	0,00	0,02
1000 < L ≤ 2000	0,03	0,07	0,17	0,04	0,10	0,27	0,00	0,00	0,03
500 < L ≤ 1000	0,03	0,07	0,14	0,03	0,09	0,18	0,00	0,00	0,02
L > 2000	0,04	0,06	0,17	0,04	0,07	0,20	0,00	0,00	0,03
100 < L ≤ 500	0,02	0,05	0,11	0,02	0,06	0,13	0,00	0,00	0,01

Fonte: Autor.

**Tabela B.35** - Resultados do Tree como meta-learner em LOO por número de amostras

(size) nos conjuntos de dados.

Métrica	MAE			MAPE			MSquareE		
	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°	Quarti 1°	Quarti 2°	Quarti 3°
500 < size <= 1000	0,03	0,07	0,13	0,04	0,09	0,18	0,00	0,00	0,02
size > 4000	0,04	0,07	0,12	0,04	0,08	0,15	0,00	0,00	0,01
2000 < size <= 4000	0,02	0,05	0,16	0,03	0,06	0,18	0,00	0,00	0,03
1000 < size <= 2000	0,03	0,06	0,13	0,03	0,06	0,15	0,00	0,00	0,02
size <= 500	0,02	0,06	0,13	0,02	0,07	0,15	0,00	0,00	0,02

Fonte: Autor.

## APÊNDICE C – CÓDIGOS E RESULTADOS DISPONIBILIZADOS

Os resultados e códigos estão disponíveis na plataforma github no link <https://shorturl.at/9WGuW>.