UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de Rio Claro

Programa de Pós-Graduação em Ciência da Computação

Bionda Rozin

Machine Learning and Information Retrieval Techniques for Time Series Analysis

Rio Claro - SP

2024

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Bionda Rozin

Machine Learning and Information Retrieval Techniques for Time Series Analysis

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Daniel Carlos Guimarães Pedronette

Rio Claro - SP
2024

R893m

Rozin, Bionda
Machine learning and information retrieval techniques for time series analysis / Bionda Rozin. -- Rio Claro, 2024
134 p. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Instituto de Geociências e Ciências Exatas, Rio Claro
Orientador: Daniel Carlos Guimarães Pedronette

1. Aprendizado de Máquina. 2. Recuperação da Informação. 3. Extração de Características. 4. Ranqueamento. 5. Analise de Series Temporais. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA
"Júlio de Mesquita Filho"
Instituto de Geociências e Ciências Exatas
Câmpus de Rio Claro

Bionda Rozin

Machine Learning and Information Retrieval Techniques for Time Series Analysis

Dissertação de Mestrado apresentada ao Instituto de Geociências e Ciências Exatas do Câmpus de Rio Claro, da Universidade Estadual Paulista "Júlio de Mesquita Filho", como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação.

Comissão Examinadora

- Prof. Dr. Daniel Carlos Guimarães Pedronette (Orientador)
  Instituto de Geociências e Ciências Exatas (IGCE)
  Universidade Estadual Paulista - UNESP
- Prof. Dr. Diego Furtado Silva
  Instituto de Ciências Matemáticas e de Computação (ICMC)
  Universidade de São Paulo - USP
- Prof. Dr. Lucas C. Ribas
  Departamento de Ciências da Computação e Estatística (DCCE)
  Universidade Estadual Paulista - UNESP

Conceito: Aprovada.
Rio Claro (SP), 04 de setembro de 2024.

Acknowledgements

I am grateful for my life, health, and resilience.

I thank my parents and those who stand close to me, always encouraging me always to strive for more.

I am extremely grateful to my advisor, Professor Daniel Pedronette, for all the guidance and support.

I also extend my gratitude to Professor Ricardo da Silva Torres for agreeing to be my advisor during my exchange in a foreign country.

Thanks to Emilio Bergamin Junior, Professor Fabricio Aparecido Breve, and Professor Felipe Arruda Moura, for the collaborative work.

I appreciate the support from São Paulo State University (UNESP), professors and colleagues from the research group.

I am also thankful to the Wageningen University & Research, for welcoming me as an exchange student.

Grants #2022/01359-1 and #2023/08087-0, São Paulo Research Foundation (FAPESP)

Resumo

Considerando o vasto domínio de aplicações de dados temporais, como o setor médico, agrícola, financeiro e científico, por exemplo, exige-se cada vez mais a análise e processamento desse tipo de dado. Tarefas como recuperação da informação, classificação e agrupamento são cruciais para analisar séries temporais em diferentes contextos e com diferentes objetivos. Recuperação da informação aplicadas em conjuntos de séries temporais permitem a identificação de padrões e ranqueamento dos dados conforme a sua semelhança, enquanto a classificação rotula séries temporais com base em um conjunto de treinamento, e tarefas de *clustering* agrupam séries temporais com base em suas similaridades. Ainda, há a classificação semi-supervisionada, que considera ambos os dados rotulados e não-rotulados para classificar os dados. No geral, as tarefas de aprendizado de máquina e recuperação da informação são extremamente dependentes de uma boa representação computacional dos dados, gerando resultados mais eficazes e conclusões mais assertivas em relação à tarefa executada. Neste cenário, um dos desafios é obter uma boa representação computacional das séries temporais. Além disso, medidas de similaridade geralmente consideram apenas a similaridade par a par, desconsiderando informações importantes presentes na vizinhança dos itens analisados, no conjunto de dados. O objetivo dessa pesquisa é aplicar tecnicas de aprendizado de maquina e recuperação da informação para obter resultados efetivos em análises de séries temporais. Quatro diferentes métodos são empregados e diferentes extratores de características são avaliados em todas tarefas. Primeiro, um estudo comparativo de representação e ranqueamento de séries temporais univariadas por meio de aprendizado contextual de distância baseado em ranqueamento é conduzido em 10 conjuntos de dados diferentes, levando a ganhos de mAP de até 31,78%. Dando sequência a esta linha de pesquisa, propomos a análise de séries temporais multivariada processando cada dimensão da série individualmente e utilizando métodos de agregação contextual de ranques para mesclar resultados e obter uma representação de similaridade utilizada para recuperação e classificação, obtendo resultados competitivos a dois métodos do estado da arte. Também é proposto um arcabouço baseado em agrupamento para análise de dados baseada na codificação de gráficos temporais, onde os dados são divididos usando critérios de segmentação temporal, e resultados altamente interpretativos são alcançados neste arcabouço quando aplicado à análise de posse de bola em partidas de futebol. Por último, é proposta uma classificação semi-supervisionada de séries temporais univariadas utilizando métodos de representação por imagem e propagação de rótulos, alcançando resultados semelhantes à classificação supervisionada.

Palavras-chave: Classificação; Recuperação da Informação; Agrupamento; Aprendizado Semi-Supervisionado; Aprendizado de Maquina; Extração de Características; Ranqueamento; Analise de Series Temporais.

Abstract

Due to the great applicability of time series in diverse scenarios, such as medicine, agriculture, economics, and science, the analysis and processing of this kind of data is demanding. Tools such as information retrieval, classification, and clustering are crucial for analyzing time series in different contexts and with different objectives. Information retrieval tasks in time series data can identify patterns and rank data by similarity. At the same time, classification can label time series based on a training set, and clustering can group time series based on their similarities. Also, semi-supervised classification considers both labeled and unlabeled data to perform classification. In general, Machine Learning and Information Retrieval tasks are extremely dependent on a good computational representation of data, generating more effective results and assertive conclusions about the performed task. In this scenario, one of the main challenges is to obtain good features from Time Series. Also, similarity metrics usually consider only pairwise relations, not considering important information in the neighborhood of the analyzed items in the dataset. The objective of this research is to apply machine learning and information retrieval techniques for obtaining effective results in time series analysis. Four different methods are employed, and different feature extractors are evaluated in all tasks. First, a comparative study of univariate time series representation and ranking through contextual ranked-based distance learning is conducted in 10 different datasets, leading to mAP gains up to 31.78%. Giving sequence to this research line, we propose multivariate time series analysis by processing each dimension of the series individually and using contextual rank aggregation methods to merge results and obtain a similarity representation used for retrieval and classification, obtaining competitive results to two SOTA methods. A clustering-based framework for data analysis based on temporal graph encoding is also proposed, where data is split using time segmentation criteria, and highly interpretative results are reached in this framework when applied to ball possession analysis in football matches. Last, semi-supervised classification of univariate time series using imaging methods and label propagation is proposed, reaching similar results to supervised classification.

Keywords: Classification; Information Retrieval; Clustering; Semi-Supervised Learning; Machine Learning; Feature Extraction; Ranking; Time Series Analysis.

List of Figures

List of Tables

List of Abbreviations and Acronyms

ACC	Color Autocorrelogram
AF	AtrialFibrillation
AFFNet	Adaptive Feature Fusion Network
AI	Artificial Intelligence
AMI	Adjusted Mutual Information
AP	Affinity Propagation
APCA	Adaptive Piecewise Constant Approximation
ARIMA	Autoregressive integrated moving average
AWR	ArticularyWordRecognition
BAS	Beam Angle Statistics
BFS-Tree	Breadth-First Search Tree of Ranking References
BIC	Border/Interior Classification
BM	BasicMotions
BOSS	Bag-of-SFA-Symbols
BOVW	Bag of Visual Word
Bow	Bag-of-Words
CBR	Content-Based Information Retrieval
CBTSR	Content-based time series retrieval
CFD	Contour Features Descriptor
CHEB	Chebyshev Polynomials
CHI	Calisnki-Harabasz Index
CLS	Classification
CNN	Convolutional Neural Network
conDetSEC	constrained Deep embedding time SEries Clustering
CPU	Central Processing Unit
CVA	Change vector analysis
DB	DBSCAN
DBA	DTW Barycenter Averaging
DBI	Davies-Bouldin Index
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCT	Discrete Cosine Transform
Deep r-RSJBE	Deep r-th root of Rank Supervised Joint Binary Embedding
DFT	Discrete Fourier Transform
DHN-RAD	Deep Hashing Network for Retrieval-based Anomaly Detection
Dim. Red.	Dimensionality Reduction Method
DINO	DIrNification with NOisy labels
DISSIM	Dissimilarity metric
DNN	Deep neural network
DTW	Dynamic Time Warping
DUBCNs	Deep Unsupervised Binary Coding Networks
DV2	DINO V2
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram

EDR	Edit Distance on Real sequence
EEG	Electroencephalogram
EffNV2	EfficientNet V2
EO-BoW	Entropy-optimized BoW
ERP	Edit Distance with Real Penalty
ESN	Echo state network
FCN	Fully Convolutional Network
FF	Fusion Features
FRF	Functional Relation Field
FrOKShape	Fractional Order Shape-based k-cluster
GAN	Generative Adversarial Networks
GAF	Gramian Angular Field
GRU	Gated Recurrent Units
GRF	Gaussian Random Fields
GVR	Graph Visual Rhythm
HB	Heartbeat
HC	Hierarchical clustering
HDB	HDBSCAN
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
HP	Hodrick Prescott
IPLA	Indexable Piecewise Linear Approximation
IR	Information Retrieval
KL	Kullback-Leibler
kNN	k-Nearest Neighbors
LAS	Local Activity Spectrum
LBP	Local Binary Pattern
LCSS	Longest Common Subsequence
LHRR	Log-based Hypergraph of Ranking References
LSH	Locality-sensitive hashing
LSTM	Long Short-Term Memory
MALSTM-FCN	Multivariate Attention Long Short-Term Memory Fully Convolutional Network
mAP	Mean Average Precision
MCFGNN	Multichannel fusion graph neural network
MDCNet	Mode Decomposition and 2D Convolutional Network
MESAMN	Multiscale Echo Self-Attention Memory Network
MF-Net	multi-feature-based network
MHAP	Multivariate Highly Activated Period
MI	MotorImagery
MLP	Multilayer Perceptron
MTF	Markov Transition Field
MTRL	deep multi-task representation learning method
MTS	Multivariate Time Series
MTSC	Multivariate Time Series Classification
N	NATOPS
Net	Network
NILM	Non-Intrusive Load Monitoring

NN	Neural Network
NSNP	nonlinear spiking neural P
NSST	on-subsampled shearlet transform
NTM	Neural Turing Machines
P@k	Precision@k
PAA	Piecewise Aggregate Approximation
PCA	Principal Component Analysis
PCF	Polynomial curve fitting
PD	PenDigits
PFD_DTW	Polynomial function derivative features-based dynamic time warping
PSEUDo	Interactive Pattern Search in Multivariate Time Series with Locality-Sensitive Hashing and Relevance Feedback
R@k	Recall@k
RDPAC	Rank Diffusion Process with Assured Convergence
RegEx	Regular Expression
ResNet	Residual Network
RFE	Rank Flow Embedding
RMSE	Root mean squared error
RN152	CNN ResNet-152
RobDecomp	Robust series decomposition block
RobTF	Robust trend forecasting block
ROCKET	RandOm Convolutional KErnel Transform
RP	Recurrence Plot
SAB	Seasonal component adjustment block
SampEn	Sample entropy
SAX	Symbolic Aggregate approXimation
SCARNet	Stacked Convolution sequence AutoRegressive encoding Network
SCV	Spectral change vector
Sil	Silhouette Score
SOTA	State-of-the-art
SpADe	Spatial Assembling Distance
SSL	Semi-supervised learning
Swale	Sequence Weighted Alignment model
SWJ	StandWalkJump
TF-IDF	Term Frequency — Inverse Document Frequency
TQuEST	Threshold Queries
T-encoder	Transformer-based
T-decoder	Transformer-based decoder
TS	Time Series
TCCT	Tightly-coupled convolutional Transformer
t-SNE	t-distributed Stochastic Neighbor Embedding
UDLF	Unsupervised Distance Learning Framework
UMAP	Uniform Manifold Approximation and Projection
UTBCN	Unsupervised Transformer-based Binary Coding Networks
VGG	Visual Geometry Group
ViT	Vision Transformers

List of Symbols

$D(\mathbf{x}, \mathbf{y})$	Pairwise distance measure between two vectors $\mathbf{x}$ and $\mathbf{y}$
i	Generic counter
j	Generic counter
k	Generic neighborhood parameter
k_c	Number of clusters
c_c	Cluster center
T	Univariate time series
M	Multivariate time series
$\mathbf{R}^{m \times n}$	Distance matrix of two time series of sizes m and n, respectively
d	Number of dimensions
k_r	Number of convolutional kernels in ROCKET algorithm
I	Unit line vector
K	Neighborhood parameter
P	Set of points in a contour
α	Angle
k_b	Beam angle statistics distance parameter
n	Number of points or dimensions
x_s	Line segment
y_s	Line segment
$\tilde{T}$	Polar coordinate representation of time series
W	Markov Transition Matrix
q_{b_j}	Quantile bin
$w_{i,j}$	Represents the frequency at which a point in quantile q_{b_j} is followed by a point in quantile q_{b_i}
$\vec{T}$	Trajectory of time series
d_{rp}	Temporal delay
m_{rp}	Trajectory dimension
$corr\_preds$	Correct predictions in a labeled dataset
$preds$	Total number of predictions
U	Cluster
V	Cluster
$H(U)$	Entropy of cluster U
$MI(U, V)$	Mutual Information between clusters U and V
$E(MI(U, V))$	Expected mutual information
TP	True positives, i.e., number of relevant retrieved items in a ranked list
q_l	Query l
r_i	Represents the relevance (irrelevant = 0 or relevant = 1) of the i-th item in a ranked list
N_r	Number of relevant items in a collection for a query q
z	Mean standard deviation of a dataset
n_coefs	Number of Fourier coefficients to keep in DFT [58]
$norm\_mean$	If True, center each time series before scaling in DFT [58]
$norm\_std$	If True, scale each time series to unit variance in DFT [58]

$wavelet$	Wavelet family, containing important information such as the wavelet filters coefficients, which are used in DWT
$mode$	Signal extrapolation mode
L	Ranked List size
L_{MULT}	Multiplier of ranked list size in RDPAC algorithm
t_i	Timestamp
G	Graph
V	Set of nodes in a graph
s	Node in a graph
t	Node in a graph
v	Node in a graph
$d(s,t)$	Shortest path from node s to node t
a	Average path length
$c_b(v)$	Betweenness centrality of node v
$c_{b_n}(v)$	Normalized betweenness centrality
$\sigma(s,t)$	Count of shortest paths from s to t
$\sigma(s,t\|v)$	Count of shortest paths from s to t passing through v
$e(v)$	Eccentricity of node v
$E(v)$	Global Efficiency of node v
$deg(v)$	Degree of of node v
$E_{loc}(v)$	Local Efficiency of node v
$V(v)$	Vulnerability of node v
$\mathbb{G}$	Temporal Graph
$\mathbb{F}$	Graph Visual Rhythm
m_{ic}	Mean intra-cluster distance
m_{nc}	Mean nearest-cluster distance for each sample
S_i	Average dissimilarity of cluster i to all other clusters
S_j	Average dissimilarity of the neighboring cluster of cluster i to all other clusters
$d(C_i, C_j)$	Similarity between centroids of clusters i and j
D_W	Dispersion within clusters
D_B	Dispersion between clusters
L_t	Lower triangular matrix of symmetric distance matrix
$\|\|L_t(t_k)\|\|_F$	Frobenius norm of L_t for a time stamp t_k
l_{ij}	Point in the L_t matrix
m	Embedding dimension
r	Tolerance
$T_m(i)$	Time series segments
B	Number of cases where $D[T_m(i), T_m(j)](i \neq j) < r$
C	Number of cases where $D[T_{m+1}(i), T_{m+1}(j)](i \neq j) < r$
p	Probability of a result being equal or superior to an observed result
$\mathcal{C}$	Collection of time series
D_t	Time series descriptor
ϵ	Function that extracts a feature vector
v_{T_i}	Feature vector of a time series T_i
$\rho(i,j)$	Function that computes the distance between two time series T_i and T_j based on their corresponding feature vectors

A	Matrix of dimension $n \times n$, where $A_{ij} = \rho(i, j)$
τ_q	Ranked list of time series T_q
$\mathcal{R}$	Set of ranked lists for each time series in $\mathcal{C}$
$\rho_r(i, j, \mathcal{R})$	More effective distance function used in unsupervised distance learning algorithms
$\mathcal{A}_a$	Set of distance matrices
$\mathcal{R}_a$	Set of ranked lists sets
$f_a(\mathcal{A}_a, \mathcal{R}_a)$	Contextual rank aggregation function
$\hat{\mathcal{A}}_c$	Resulting distance matrix of $f_a(\mathcal{A}_a, \mathcal{R}_a)$
$\mathcal{R}_c$	Resulting ranked list based on $\hat{\mathcal{A}}_c$
$f_a(\mathcal{R}_a)$	Contextual rank aggregation function that does not use the set $\mathcal{A}_a$
σ	Parameter of Gaussian Similarity
$w_{g_{i,j}}$	Gaussian Similarity
H_f	Cost function
L_e	Number of labeled elements
$\psi_{i,s}$	Represents the probability of the $i-$th node of a graph belonging to the $s-$th class
t	Iteration marker
e	upper bound for the difference between consecutive iterations
$\hat{y}_i$	Unknown label of $i-$th node of a graph
$\mathcal{D}$	Dataset
r_l	Percentage of labeled elements

Contents

1 Introduction

Analysing time series is not only of paramount importance, nowadays, but also a challenging task. It depends on available labels, time series representation, similarity measures, and results' interpretability. This work proposes four different approaches that contribute to these directions.

This work discusses and presents contributions to the field of time series analysis by using machine learning and information retrieval techniques. This introductory chapter gives an overview of this work and is organized as follows: Section 1.1 discusses the motivations of the research. Section 1.2 presents the challenges and research questions approached during the development of the work. Section 1.3 discusses the goals and contributions of the study. Section 1.4 outlines the structure of this work, including a brief description of each chapter's content.

1.1 Motivation

Time series data finds extensive applications in our daily lives, spanning diverse domains such as medicine, economics, and science. ECGs [200] and EEGs [169], yearly sales [24], sensors [180], weather forecasts [73], bitcoin charts [130], and general finances [223] are just a few examples of how time series data is employed. Formally, a time series refers to a set of numerical values arranged chronologically, with each point directly correlated to preceding points [149].

Considering the broad application in different domains, the development of tools for studying and analyzing time series is of central relevance. Additionally, various facets can be taken into account based on the desired objectives. From a computational perspective, supervised and unsupervised learning algorithms process time series data, utilizing either labeled or unlabeled information, while semi-supervised learning tasks combine both approaches [197]. Semi-supervised learning algorithms stand out when labeled data is scarce, considering that obtaining labels can be expensive and a time-consuming task. Classification and clustering algorithms, for instance, enable the labeling or grouping, respectively, of time series samples, where series with the same label (or group) exhibit similarity, while those with different labels (groups) are distantly related [83, 13]. Such techniques have proven to be effective in detecting cancer in early stages [228], identifying mobility patterns in populations [225], and enhancing poultry welfare through the analysis of chicken behaviors captured by 3-axis sensors [4]. Forecasting tasks involve predicting future values of time series based on historical data, encompassing applications like stock market variations [223] and even ECG predictions [157].

Despite the large number of successful applications, the effectiveness of machine learning tasks is directly dependent on the features extracted from the data. The features are intrinsically associated with the data representation in the space, impacting similarity measures. Time series features consider various aspects for data representation, including shape [10], temporal correlation [204], as well as bag-of-words based models [111] and kernel-based models [45].

Data representation and similarity measures also profoundly impact the effectiveness of another critical area: Information Retrieval (IR). IR applications generally involve managing, indexing, and retrieving data based on a given query, which represents an information need [120]. The retrieval process revolves around ranking the most similar elements relative to a given query. Therefore, data representation and similarity measures directly influence the effectiveness of information retrieval tasks, as different features can generate varying similarities for the same query, and different similarity measures can result in distinct ranked lists. Moreover, traditional measures such as Euclidean distance solely consider pair-to-pair comparisons, disregarding neighboring interactions, which could be leveraged to compute improved similarity measures [193].

Information Retrieval has diverse applications in the realm of time series analysis. For instance, it can be employed in plant phenology studies to identify the same species in different locations based on time series obtained from satellite images [7]. Similar techniques can be utilized to study the dispersion of volcanic plume height, which poses risks to airplanes [135]. Additionally, selecting a slice of a time series as a query enables the identification of similar slices, as exemplified in the case of distinguishing open and closed eyes based on EEG signals [227]. Building information retrieval methods capable of reaching high-quality results is a challenging task. An option to consider the existing relationships within a dataset beyond traditional pairwise similarities is the utilization of unsupervised distance learning algorithms after the information retrieval process. These algorithms can consider the relationships within a neighborhood and compute more efficient distances based on such information, enabling the generation of improved ranked lists [193].

Machine Learning and Information Retrieval algorithms are crucial tools for time series analysis, especially when strategies from both areas are combined. Our objectives are to explore the applicability of these methodologies in diverse contexts of time series analysis. Additionally, we aim to assess the impact of different time series representations and contextual similarity measures on these tasks.

1.2 Research Challenges

While time series have a wide range of applicability, analyzing it is very challenging. There are diverse research challenges related to exploiting the encoded information in time

series and how to analyze such data properly. Following, some topics related to this work are discussed:

- **Information encoded in the time series:** There are diverse characteristics intrinsic to time series, such as shapes, frequency, trend, seasonality, among others. How can we extract meaningful features for a required task? Which features are more suitable for a machine learning or information retrieval task?
- **Impact of distance measurements:** Diverse tasks depend on a similarity measure which allows to compare different time series. How do we find distance measurements that translate effectively the relationship between the data points?
- **Explore 2D representations of time series:** There are diverse methods to represent time series as images [204, 53]. How the information contained in the images can be explored?
- **Explore contextual information for effective time series information retrieval:** How can we explore more global relationships between time series to perform information retrieval?
- **Exploit information of each dimension of a multivariate time series:** Multivariate time series are composed of $n-$dimensional data points and each time series dimension carries its own information. How to effectively explore information encoded in the dimensions of a multivariate series?
- **Temporal data analysis with lack of ground truth:** Some time series datasets do not have labels, and clustering analysis emerges in these scenarios. How do we properly interpret the clustering results?
- **Effective use of labeled and non-labeled time series data:** How to properly explore supervised and unsupervised information in time series data?

1.3 Goals and Contributions

Our hypothesis is that information retrieval and machine learning techniques can be explored for effective time series analysis, especially under conditions of scarcity or absence of labeled data. Given the direct time series applications in real-world scenarios, our main goal is to obtain effective results on time series analysis by applying unsupervised, supervised, and semi-supervised learning approaches for time series clustering, classification, and similarity search. Also, our secondary goal is to study the impact of different time series feature extractors in the performed tasks.

Efficiency is also important in machine learning and information retrieval tasks, but the experimental evaluation performed in this work only focused on effectiveness. Works such as [146, 69, 145, 195] discuss algorithmic complexity and efficiency in re-ranking tasks.

Figure 1.1 presents the goals and contributions and how they relate to proposed approaches, considering different supervision strategies.

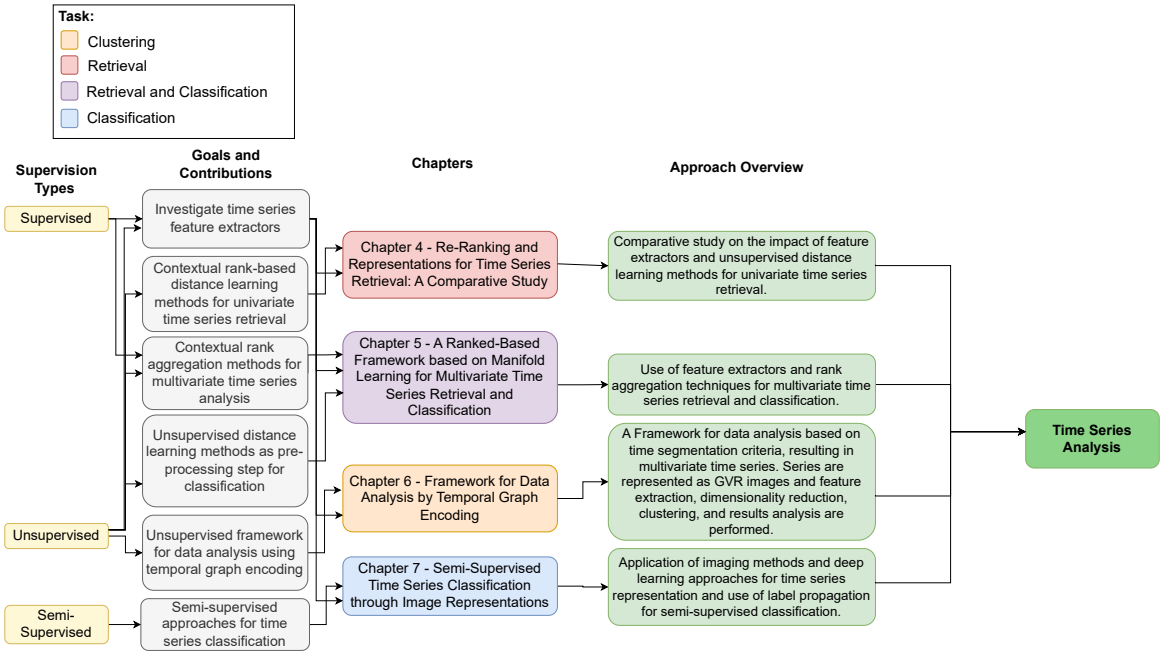


Figure 1.1 – Overview of goals and contributions of this work.

The proposed approaches directly relate to the research challenges discussed before. Following, an overview of each goal and contribution is given:

- **Investigate time series feature extractors:** Diverse feature extractors were applied in time series for different tasks, i.e. classification, clustering, and information retrieval. Both 1D and imaging methods were evaluated. Features were extracted from the images using deep learning approaches.

- **Employ contextual rank-based distance learning methods for univariate time series retrieval:** Pairwise distance measurements often ignore contextual information available in the dataset. Four unsupervised distance learning methods were applied in time series datasets for computing more global distance measurements between points in a dataset, by considering neighborhood relationships, and performing re-ranking.

- **Employ contextual rank aggregation methods for multivariate time series analysis:** To fully explore information present in the dimensions of multivariate time

series, we process each dimension individually and use rank aggregation methods to obtain a final result for multivariate time series retrieval.

- **Use unsupervised distance learning methods as a pre-processing step for classification:** This topic is directly correlated to the previous one. The resulting similarity measure defined by the rank aggregation step is considered for performing a kNN classification in a multivariate time series dataset.
- **Proposal of an unsupervised framework for data analysis using temporal graph encoding:** We propose a novel clustering-based framework based on time segmentation criteria, encoding information as multivariate time series, using domain-specific variables to evaluate and enhance the interpretability of results. This framework was validated for ball possession analysis in football matches.
- **Investigate the application of semi-supervised approaches for time series classification:** To explore the information available in both labeled and unlabeled time series datasets, we employ the label propagation algorithm for time series semi-supervised classification.

1.4 Organization

This work is organized according to the main contributions obtained in our research, which were published or submitted to conferences and journals. Below, the contents of each chapter are briefly described, indicating the associated publication or submission:

- **Chapter 2 - Foundation:** defines the theoretical foundation of the work, presenting main concepts, and formal definitions;
- **Chapter 3 - Related Work:** presents the related work, relevant to the topics approached in this work;
- **Chapter 4 - Re-Ranking and Representations for Time Series Retrieval: A Comparative Study:** presents the comparative study of univariate time series retrieval approaches, considering different data representation and contextual re-ranking methods. The content of this chapter was submitted to a journal;
- **Chapter 5 - A Ranked-Based Framework based on Manifold Learning for Multivariate Time Series Retrieval and Classification:** discusses a framework for multivariate time series retrieval and classification. The content of this chapter was submitted to a journal;
- **Chapter 6 - Framework for Data Analysis by Temporal Graph Encoding:** presents a framework for temporal data analysis by considering time segmentation

criteria. Experiments were conducted considering ball possessions from football matches. The content of this chapter is in the final process of writing and it is going to be submitted to a journal;

- **Chapter 7 - Semi-Supervised Time Series Classification through Image Representations:** approaches the developed work about semi-supervised time series classification using image representations. The content of this chapter is available in a paper published in the proceedings of the *International Conference on Computational Science and Its Applications (ICCSA 2023)* [163];
- **Chapter 8 - Conclusions:** presents the relationships between the research questions and contributions and discusses future works.

Figure 1.2 presents the organization of the work.

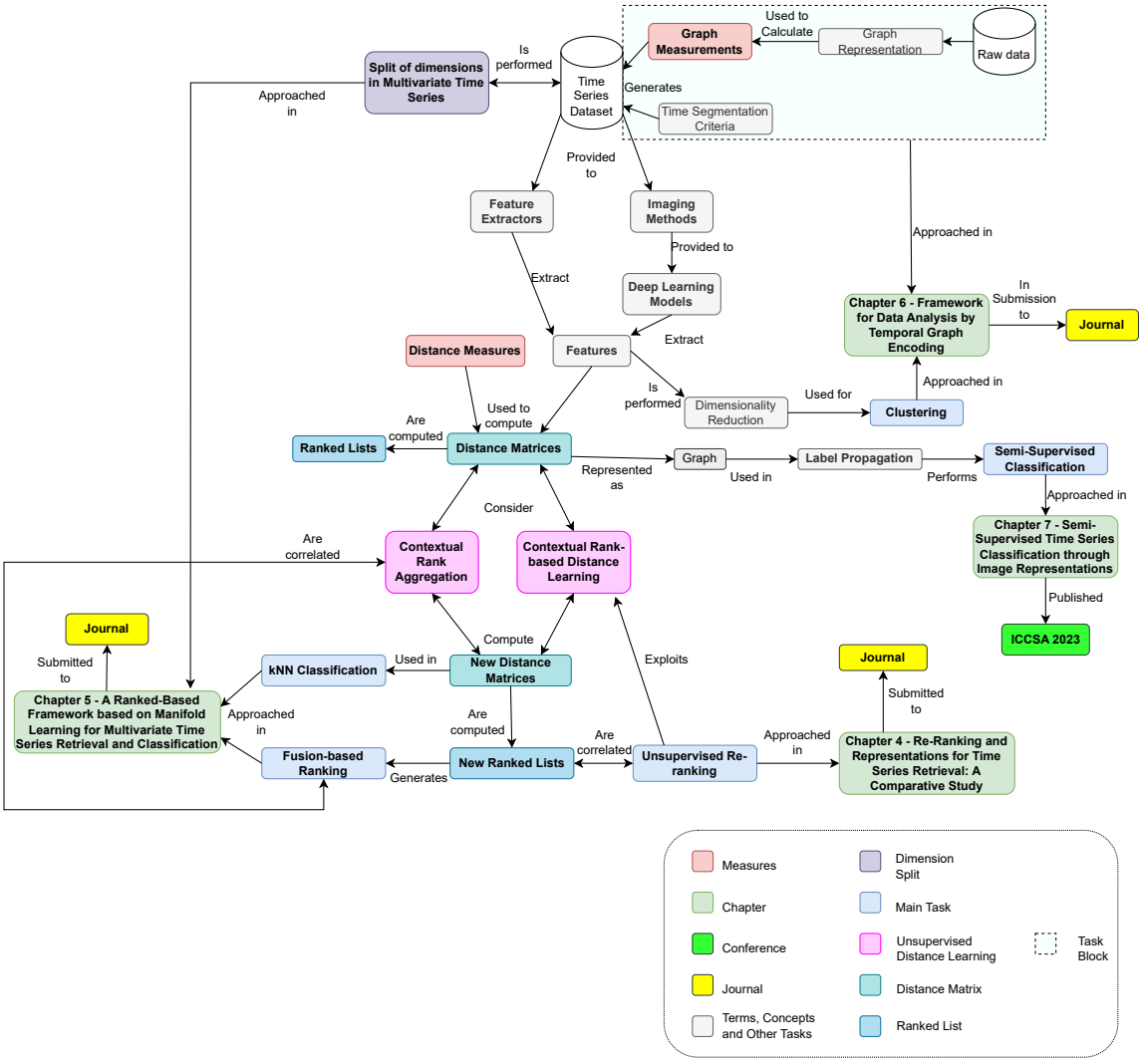


Figure 1.2 – Organization of the work

2 Foundation

In this chapter, the main theoretical aspects of this work are defined. In Section 2.1, the overview of the time series is presented. Section 2.2 presents the principal concepts related to information retrieval. Section 2.3 describes the information representation concepts, while Section 2.4 presents the main concepts about machine learning. Section 2.5 presents examples of time series descriptors. In Section 2.6, the core concepts of Unsupervised Distance Learning are introduced. In Section 2.7 are described some univariate time series datasets, the primary evaluation metrics for information retrieval and classification tasks are defined in Section 2.8, and the leading software and frameworks utilized in this work are described in Section 2.9.

2.1 Time Series

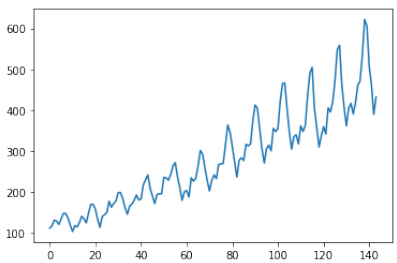
Time series are data that describe the evolution of one or more variables over time, considering equidistant periods [55]. Formally, a time series of size n can be defined as:

$$T_s = (t_1, t_2, ..., t_n), t_i \in \mathbb{R}^d, d \in \mathbb{N}^* \tag{2.1}$$

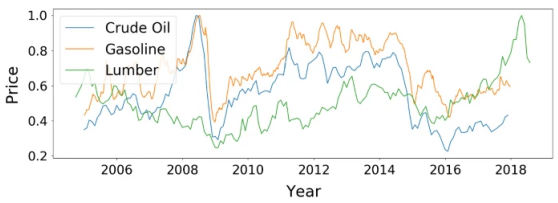
Time series composed of a single observed variable over time, i.e., $d = 1$, are called univariate [55], and an example to consider is a patient's heartbeat. Figure 2.1(a) illustrates a univariate time series.

Similarly, multivariate time series are composed of variables that exhibit temporal dependence [26], i.e. $d > 1$, such as data on gas concentrations in the atmosphere measured every hour. Figure 2.1(b) illustrates a multivariate time series.

For instance, we will define a univariate time series as T and a multivariate time series as M.



(a) Univariate Time Series



(b) Multivariate Time series. Extracted from [178]

Figure 2.1 – Univariate and Multivariate Time Series

Considering the definitions, time series are complex structures capable of encoding a wealth of information. As a result, time series have a wide range of applicability, including the medical field [200], industrial sector [57], financial domain [130], among others. Therefore, analyzing and representing these structures are tasks of paramount importance and quite challenging.

Also, aligning two time series is a difficult task. The alignment provided by traditional distance measures, such as the Euclidean Distance, usually does not provide the optimum alignment. In this context, the Dynamic Time Warping (DTW) metric calculates the optimum non-linear alignment between two time series of any size [175]. The value returned by the DTW function represents the distance between two-time series with better alignment. Formally, let the time series $T_x = (t_{x_1}, t_{x_2}, ..., t_{x_m})$ and $T_y = (t_{y_1}, t_{y_2}, ..., t_{y_n})$ and $\mathbf{R}^{m \times n}$ a matrix containing the distances $D(T_x, T_y)_{i,j}$ of the sequences T_x and T_y, where D is, usually, the Euclidean Distance. The objective is to find two sequences a and b, both of size $l \geq max(m, n)$, in which the index of combination $a(i)$ corresponds to the time series T_x and $b(i)$ corresponds to the time series T_y, where the total cost of combinations $\sum_{i=1}^{l} D(T_x, T_y)_{a(i),b(i)}$ are minimized. The alignment (a, b) matches the following conditions:

1. $a(1) = b(1) = 1$;
2. $a(l) = m, b(l) = n$;
3. $(a(i+1), b(i+1)) - (a(i), b(i)) \in \{(1,0), (1,1), (0,1)\}$.

In Figure 2.2, it is possible to observe the comparison between the Euclidean Distance's linear alignment and the DTW algorithm's non-linear alignment.

2.2 Information Retrieval

Information retrieval is an area of computer science that deals with the representation, storage, organization, and access of items from diverse datasets, such as documents and web pages, for example [158]. An information retrieval problem involves a search task defined according to pre-selected criteria, which should return relevant results, where the organization and representation of the selected items facilitate user access.

The significant growth in data generation, such as audio, images, 3D models, and other objects, makes searching a challenging problem [201]. Unlike text-based search, which has mature results [158, 119], conducting textual searches for multimedia data can be unsatisfactory, as textual descriptions may not capture all the characteristics of the given data [201] and may potentially introduce ambiguities.

In this context, Content-Based Retrieval (CBR) systems enable comparisons between similar data based solely on their content [201]. The data is computationally

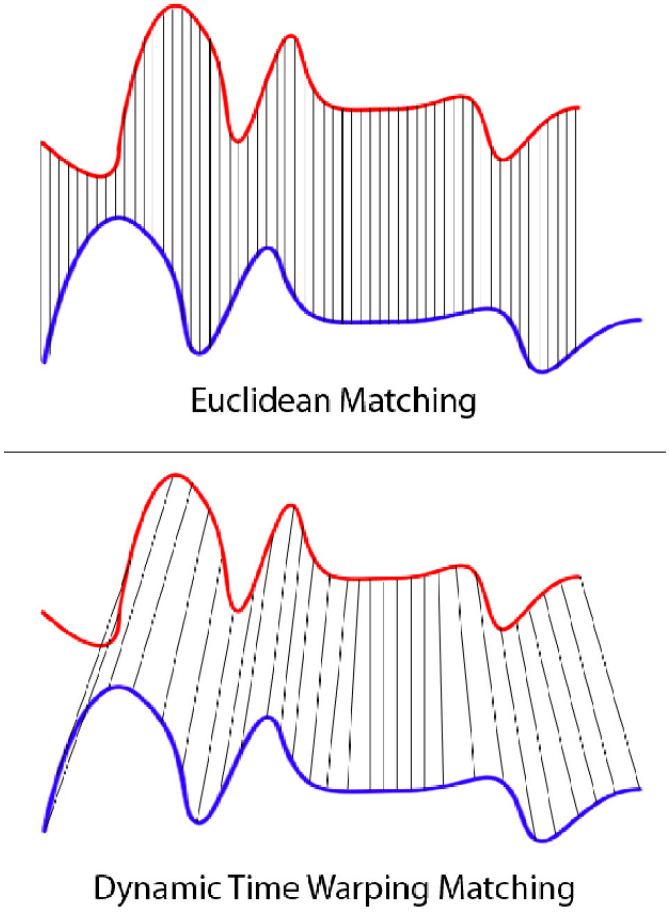


Figure 2.2 – Comparison between the Euclidean Distance alignment and DTW alignment. Extracted from [36]

represented as feature vectors, considering approaches such as texture, color, and shape for images [107], Zero Crossing Rate, and Short-time energy, for example, for audio [124], among other features for different types of media. In this way, extracting feature vectors for an entire dataset and enabling comparisons with any given query is possible. For each query, a result list is calculated containing the distances, in ascending order, between the elements of the dataset and the query [107].

Figure 2.3 illustrates a typical architecture of a CBR general model for time series application adapted from [191]. This architecture consists of *(i)* an interface, *(ii)* a query processing module, and *(iii)* a dataset. In the interface, user interaction occurs, where data is inputted, and query results are returned. Data characteristics are extracted in the query processing module, and ranking tasks are performed based on the obtained features. Lastly, the data and their corresponding feature vectors are stored in the dataset to avoid recalculations.

The distances between the obtained feature vectors can be calculated in different ways. Let $\mathbf{x} = (x_1, x_2, ..., x_d), x_i \in \mathbb{R}$ and $\mathbf{y} = (y_1, y_2, ..., y_d), y_i \in \mathbb{R}$, where d represents the number of dimensions, be two feature vectors that represent times series. The following

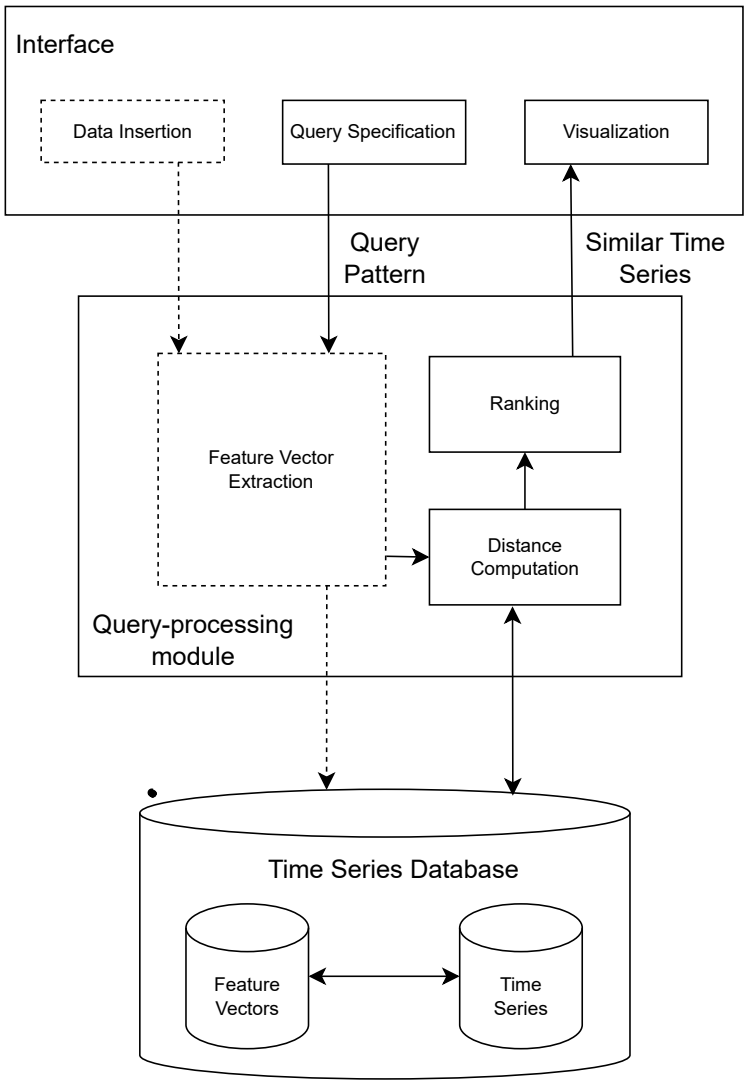


Figure 2.3 – Typical architecture of a content-based time series retrieval system. Inspired in the content-based image retrieval system presented in [191]

distance metrics can be used:

- **Euclidean**: $D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$;
- **Chebysev**: $D(\mathbf{x}, \mathbf{y}) = max_{i=1}^{d}|x_i - y_i|$;
- **Manhattan**: $D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d}|x_i - y_i|$;
- **Minkowsky**: $D(\mathbf{x}, \mathbf{y}) = [\sum_{i=1}^{d}|x_i - y_i|^p]^{1/p}$;
- **Canberra**: $D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d}\frac{|x_i - y_i|}{|x_i| + |y_i|}$;
- **Cosine**: $D(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}||||\mathbf{y}||}$.

In general, the Euclidean distance is widely used to calculate distances between elements [139]. However, despite the importance of distance metrics, the quality of the retrieved information is primarily influenced by the choice of the used representations (features).

2.3 Information Representation

Describing a specific type of data is a crucial step in machine learning tasks, as the representation of the dataset directly influences the effectiveness of the performed tasks [132]. D-dimensional feature vectors represent each element in a dataset.

Methods used for data description depend on the type of information encoded in the data. Various algorithms can be considered for feature extraction in the context of images. Shape extractors, such as Beam Angle Statistics [9], Contour Features Descriptor (CFD) [141], and Aspect Shape Context [113], allow for the study of different contours present in the image. Color-based methods, such as Color Autocorrelogram (ACC) [76] and Border/Interior Classification (BIC) [182], take into account color histograms of the images. Texture-based methods consider elements such as bricks or velvet, which produce variations in surface appearance. Examples include Local Binary Patterns (LBP) [133] and Local Activity Spectrum (LAS) [189]. Other methods, such as Bag of Visual Words (BOVW) [47] and deep learning-based models [89], can be mentioned in the scope of image feature extraction. In the case of text, representations such as Bag-of-Words, TF-IDF [158, 119], and again, deep learning-based approaches [110] are some examples found in the literature. Time series representation will be approached in Section 2.5.

Given the presented overview, it is evident that various characteristics can be explored in multimedia datasets. On the other hand, different feature extractors can be used, leading to diverse results in machine learning tasks. Figure 2.4 illustrates extracting features from multimedia content and calculating the distance measure between the obtained features, resulting in a ranking score. The chosen representation for a given data directly influences the calculation of distance measures and, consequently, the ranking of similarities among items in the dataset. Therefore, it is crucial to select features carefully, considering the dataset's characteristics and the executed tasks.

2.4 Machine Learning

Machine learning is an area of artificial intelligence (AI) that involves learning algorithms capable of performing tasks or predictions without being explicitly set, only considering the characteristics of a provided dataset, finding patterns, and relevant information for a specific task [98], e.g., pattern recognition [224], sales prediction [211],

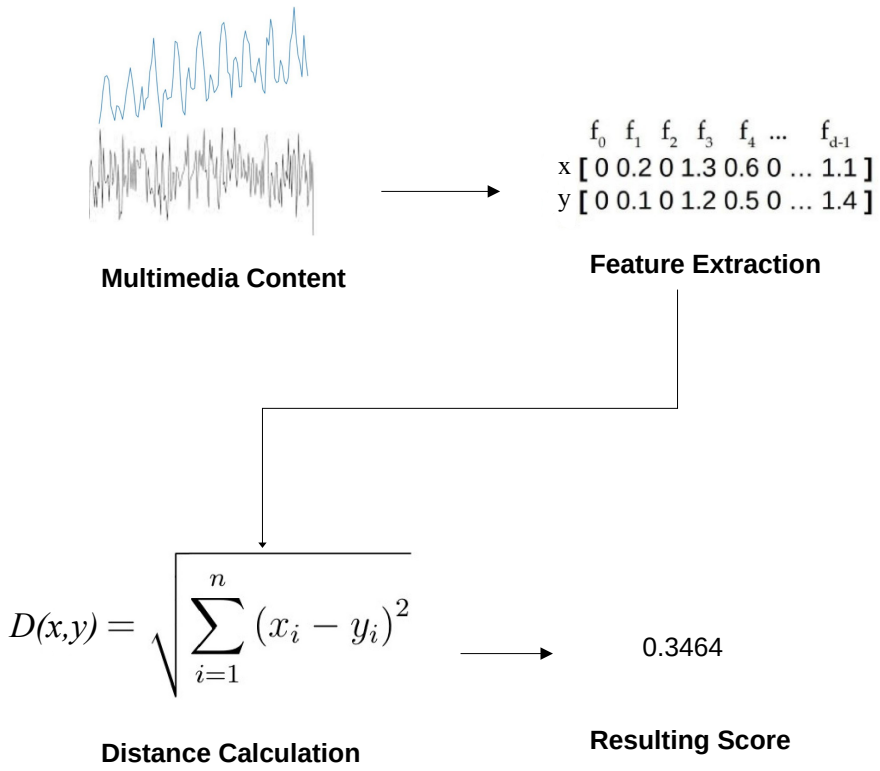


Figure 2.4 – Feature extraction and distance calculation between two time series

simultaneous machine translation [77], among others.

There are diverse categories of machine learning algorithms, and their choice depends on the problem to be solved and data availability. This work explored Supervised, Unsupervised, and Semi-Supervised Learning, which will be explained below.

- **Supervised Learning**

Supervised Learning algorithms are trained on labeled data, i.e., data with class information, and make decisions based on unlabeled data, i.e., unknown data [98]. The patterns in labeled data are considered to build a model capable of making predictions about the unknown data. The main supervised learning tasks are classification, in which the main goal is predicting the class of the elements, such as in image recognition [72], and regression, where the objective is to predict continuous values, such as forecasting about sales [54]. Examples of supervised algorithms are: Linear Regression [62], Support Vector Machines [20], Convolutional Neural Networks [72], Decision Trees [185], among others. Despite their popularity and highly accurate results, supervised tasks require large amounts of quality labeled data and elevated training time, making their execution impracticable in many situations.

- **Unsupervised Learning**

Unlike supervised algorithms, unsupervised learning tasks do not require labels, information about data, or user intervention, using only intrinsic data characteristics, as similarities between the objects [97], to perform the task. We can cite as examples of unsupervised learning tasks the dimensionality reduction, simplifying data while keeping its characteristics, and clustering, where data is grouped based on the similarities between points [97], among others. Examples of unsupervised learning algorithms are Principal Component Analysis (PCA) [90], for dimensionality reduction, and K-Means [116], for clustering. There are unsupervised distance learning algorithms in information retrieval tasks, exploring intrinsic characteristics and contextual information on datasets to find more global similarity measures, in contrast to the traditional pairwise measurements [214]. Unsupervised learning is an excellent tool for unlabeled data but lacks a ground truth for performance evaluation.

- **Semi-supervised Learning**

Semi-supervised learning combines concepts of both Supervised and Unsupervised Learning [29, 229], utilizing information present in the labeled and unlabeled data. An example is the semi-supervised classification, where the classifier is trained in a small sample of labeled data, and the information contained in the unlabeled data is also considered for performing the task. The main advantage of semi-supervised methods is the capability to perform with a scarcity of labeled data. Semi-supervised methods are divided into transductive and inductive methods. Transductive algorithms do not produce a model, considering only samples found during the trainment [198] for labeling data. When new data is added, it is necessary to perform the trainment again to consider the new samples. Otherwise, as traditional supervised methods, inductive algorithms build a model during the training, utilized for classifying new elements [198]. As examples of semi-supervised algorithms, we can cite Label Propagation [230] and Semi-Supervised Support Vector Machine (S3VM) [88]. Diverse assumptions are made about data distribution [29], being considered individually or in groups, depending on the semi-supervised method [198]. The assumptions are:

- **Clustering Assumption:** Data points belonging to the same cluster must belong to the same class;
- **Smoothness Assumption:** Samples close to each other in the feature space tend to belong to the same class, and the decision boundary must be placed in regions with low data density;
- **Manifold Assumption:** Data must belong to the same class if placed close to each other in a lower dimensional representation.

2.5 Time Series Representation

A high-quality data representation is essential for achieving effective results in machine learning tasks. Obtaining suitable feature vectors from time series is also a crucial step in content-based information retrieval tasks, as the arrangement of ranked lists depends directly on this information. Data representation can consider shapes, frequencies, pattern repetitions, temporal correlation between points, and other characteristics.

Several methods for feature extraction in time series are briefly described below. These methods were selected based on the variability of feature extraction approaches in time series.

2.5.1 1D-Signal-based Feature Extractors

Effective machine learning relies on robust data representation, particularly for time series. Feature extraction from time series is pivotal for accurate information retrieval, encompassing various properties, such as shapes, frequencies, pattern repetitions, and temporal correlations.

- **Beam Angle Statistic**

The Beam Angle Statistics (BAS) algorithm [10, 164] is a shape descriptor that identifies concavities and convexities in contours and extracts related features. The method, including in prior research [215, 192], has demonstrated promise [192]. By considering a parameter k_b, a contour described by points $P = (p_1, p_2, \ldots, p_n)$ undergoes traversal. Features are generated based on the angle α_i, formed by the intersection of line segments x_s (connecting points $(i - k_b, p_{i-k_b})$ and (i, p_i)) and y_s (connecting points (i, p_i) and $(i + k_b, p_{i+k_b})$), where $i = k_b, k_b + 1, k_b + 2, \ldots, n - k_b$. This yields the representation $BAS = (\alpha_1, \alpha_2, \ldots, \alpha_{n-2k_b})$ for any contour. Figure 2.5 presents an example of the BAS execution.

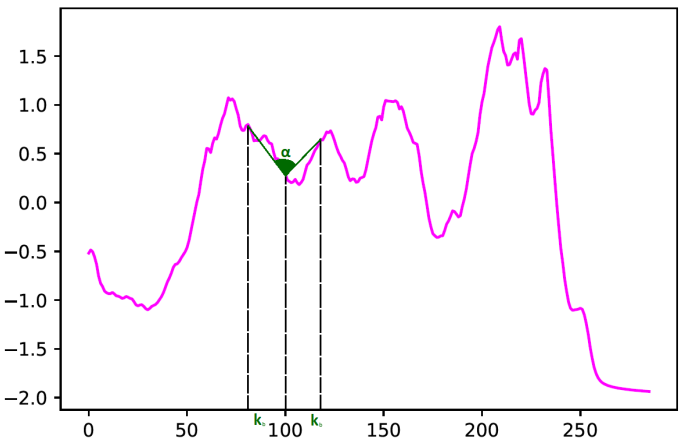


Figure 2.5 – Lines x_s, y_s and angle α for the point p_i ($i = 100$ and $0 < k_b < 50$).

- **Discrete Fourier Transform (DFT)**

The Discrete Fourier Transform (DFT) preserves point count while transforming a signal. It dissects time series into basis functions, where early functions capture trends and later ones account for noise. Each function possesses a complex Fourier coefficient. The initial coefficients approximate the time series [171]. Figure 2.6 shows an example of DFT representation.

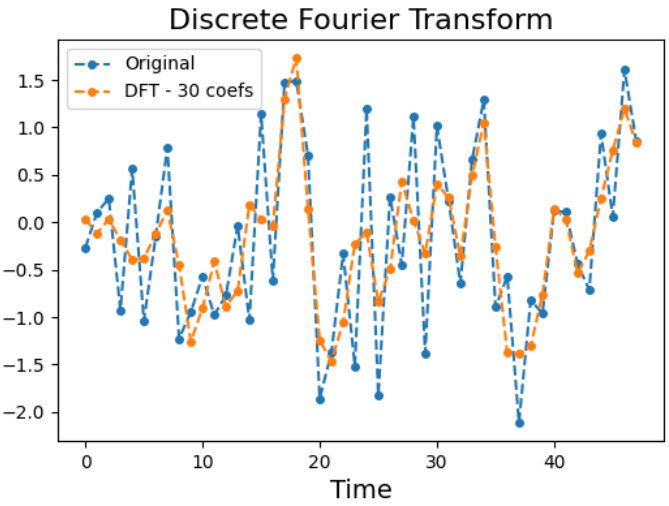


Figure 2.6 – Time series represented by DFT. Extracted from [151]

- **Discrete Wavelet Transform (DWT)**

The Discrete Wavelet Transform (DWT) breaks time series into diverse frequencies across scales, reducing dimensions and noise [28]. It yields coarse-grained approximation and fine-grained detail coefficients through specific wavelet functions involving convolution and downsampling. Figure 2.7 shows a DWT representation.

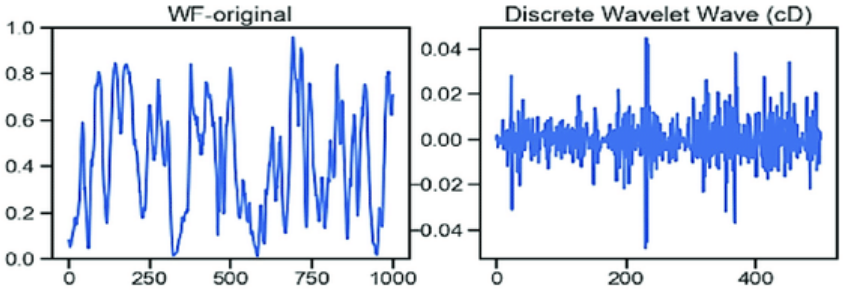


Figure 2.7 – Time series and its DTW representation. Extracted from [123]

- **RandOm Convolutional KErnel Transform (ROCKET)**

ROCKET [45] is a time series classifier that can be used for feature extraction. This method reached impressive results for many time series datasets and is computationally inexpensive. The method is based on convolutional kernels generated randomly. It generates k_r convolutional kernels, extracting two features per kernel: global max pooling and the

proportion of positive values. Then, a linear classifier is trained on the features obtained for the classification task. ROCKET excels in efficiency, boasting lower computational complexity than alternatives, needing just one hyperparameter evaluation, the number of kernels k_r. Figure 2.8 presents the weights of the most representative kernels, based on mutual information, for executing the algorithm.

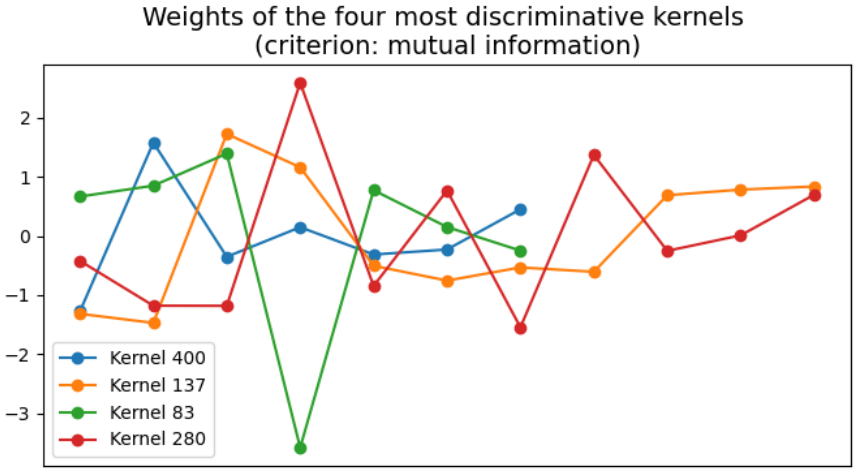


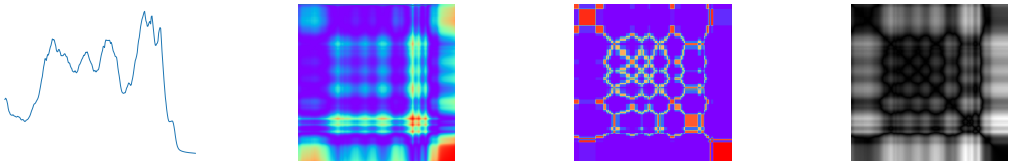
Figure 2.8 – Weights of the most representative kernels. Extracted from [152]

The descriptors were selected based on many criteria. DWT and DFT were chosen as representatives of traditional approaches, i.e., well-established methods in the field [27]. ROCKET is a recent algorithm that can be used for time series classification and feature extraction. Still, to the best of our knowledge, this work was the first to apply it in information retrieval tasks. The use of the BAS algorithm is based on [192]. BAS represents the family of methods that rely on shape-based characterization.

2.5.2 Image-Based Time Series Description

The representation of times series through images which are subsequently exploited for feature extraction is a promising approach [204, 53], as demonstrated in several applications, ranging from phenology [60] to land cover analysis studies [122, 49, 48]. In those methods, time series patterns are encoded in bi-dimensional representations, and pre-trained models (e.g., based on transfer learning procedures relying on Convolutional Neural Networks and Transformer-based models) are used for feature extraction. In the following, we discuss the approaches used for obtaining the bi-dimensional representations and the deep learning approaches employed for feature extraction. Figure 2.9 presents a time series and its respective 2-dimensional representations.

- **Gramian Angular Fields (GAF) [204]**



(a) Time series (b) GAF (c) MTF (d) RP

Figure 2.9 – Time series and its respective image representations

This method generates an image based on polar coordinates of a time series T. It transforms normalized series T into $\tilde{T}$ using polar coordinates in $[-1, 1]$. Using $\tilde{T}$, a Gramian Angular Field Matrix ($N \times N$) forms, depicted in Equation 2.2:

$$GAF = \tilde{T}'\tilde{T} - \sqrt{I - \tilde{T}^2}'\sqrt{I - \tilde{T}^2} \tag{2.2}$$

Here, I is a unit line vector. Each point $GAF_{i,j}$ in the Gramian Angular Field Matrix calculates the trigonometric sum of angles between corresponding points in $\tilde{T}$, representing the temporal correlation between different time intervals in the series.

- **Markov Transition Fields (MTF) [204]**

This method creates time series images based on Markov transition probabilities. Given a time series T, it is identified Q quantile bins and assigned each t_i to the corresponding bin q_{b_j} ($j \in [1, Q]$), that divide the sorted series into equal subsets. A $Q \times Q$ weighted adjacency matrix W is constructed by counting transitions among quantile bins, employing a first-order Markov chain methodology along the temporal axis. $w_{i,j}$ represents the frequency at which a point in quantile q_{b_j} is followed by a point in quantile q_{b_i}. After normalization by $\sum_j w_{i,j} = 1$, we have the Markov Transition Matrix, defined as in Equation 2.3:

$$W = \begin{bmatrix} w_{11}|P(t_x \epsilon q_{b_1}|t_{x-1} \epsilon q_{b_1}) & \dots & w_{1Q}|P(t_x \epsilon q_{b_1}|t_{x-1} \epsilon q_{b_Q}) \\ w_{21}|P(t_x \epsilon q_{b_2}|t_{x-1} \epsilon q_{b_1}) & \dots & w_{2Q}|P(t_x \epsilon q_{b_2}|t_{x-1} \epsilon q_{b_Q}) \\ \dots & \dots & \dots \\ w_{Q1}|P(t_x \epsilon q_{b_Q}|t_{x-1} \epsilon q_{b_1}) & \dots & w_{QQ}|P(t_x \epsilon q_{b_Q}|t_{x-1} \epsilon q_{b_Q}) \end{bmatrix} \tag{2.3}$$

W is insensitive to the distribution of the time series T and temporal dependency on time steps x_i. This results in information loss. Next, a Markov Transition Field (MTF) of dimensions $N \times N$ is built per Equation 2.4:

$$MTF = \begin{bmatrix} w_{ij}|t_1 \epsilon q_{b_i}, t_1 \epsilon q_{b_j} & \dots & w_{ij}|t_1 \epsilon q_{b_i}, t_n \epsilon q_{b_j} \\ w_{ij}|t_2 \epsilon q_{b_i}, t_1 \epsilon q_{b_j} & \dots & w_{ij}|t_2 \epsilon q_{b_i}, t_n \epsilon q_{b_j} \\ \dots & \dots & \dots \\ w_{ij}|t_n \epsilon q_{b_i}, t_1 \epsilon q_{b_j} & \dots & w_{ij}|t_n \epsilon q_{b_i}, t_n \epsilon q_{b_j} \end{bmatrix} \tag{2.4}$$

Here, $MTF_{i,j}$ is the probability of transitioning from quantile bin q_{b_i} to q_{b_j}. Specifically, $MTF_{i,j||i-j|=k}$ denotes the probability of transitioning between points with a time interval of k.

- **Recurrence Plots (RP) [53]**

A Recurrence Plot (RP) is a binary depiction of a time series, revealing temporal correlations. Given time series T, its trajectory $\vec{T} = (\vec{t_i}, \vec{t_{i+d_{rp}}}, \ldots, \vec{t_{i+(m_{rp}-1)d_{rp}}})$ is obtained for $i \in 1, 2, \ldots, n-(m_{rp}-1)d_{rp}$, where d_{rp} is temporal delay and m_{rp} is trajectory dimension. $RP_{i,j}$ becomes 1 if Euclidean distance between $\vec{t}(i)$ and $\vec{t}(j)$ is $\leq k$. For $X = n-(m_{rp}-1)d_{rp}$, the Recurrence Plot matrix, $X \times X$, is computed according to Equation 2.5:

$$R_{i,j} = \begin{cases} 1, & \text{if } ||\vec{x}(i) - \vec{x}(j)|| \leq k \\ 0, & \text{else,} \end{cases} \tag{2.5}$$

$$\forall i, j \in 1, 2, \ldots, X$$

2.5.3 Deep Learning Methods as Image Feature Extractors

Images hold diverse information like patterns, colors, textures, and shapes. Leveraging pre-trained neural networks via transfer learning has exhibited substantial potential in diverse domains due to robust generalization and feature learning [115]. In this research, we utilized five neural networks for feature extraction, pre-trained on ImageNet [46].

- **CNN ResNet-152**

ResNet [72] represents a Convolutional Neural Network series explicitly built for image recognition tasks. The ResNet-152 variant, characterized by its depth comprising up to 152 layers, effectively addresses the challenge of gradient vanishing existing in deep architectures by leveraging skip connections. These skip connections play a fundamental role in augmenting the flow of gradients during the backpropagation phase, promoting the efficient training of deep networks. The feature extraction output is sourced from the terminal's fully connected layer of the network architecture, which has 2048 dimensions.

- **Vision Transformers**

Vision Transformers (ViT) [96] demonstrate particular efficacy in object detection and image classification tasks. Thanks to the self-attention mechanism, ViTs can simultaneously focus on essential parts of the image. The input image transforms patches, which are subsequently processed by an encoder to produce hidden states. A decoder component generates output tokens through layers of self-attention and feedforward

networks. Masked self-attention within the decoder selectively focuses on preceding tokens. In the context of feature extraction, we utilize the *CLS* token, as delineated in the work by Kim et al. [95]. The obtained features contain 1280 dimensions.

- **VGG-19**

VGG-19 [179] is a Convolutional Neural Network developed by the Visual Geometry Group (VGG). Composed of 19 layers, it features 16 convolutional layers stacked sequentially, followed by max-pooling layers for dimensionality reduction. The network closes with three fully connected layers. For feature extraction, the last fully connected layer before classification was utilized. The resulting features are 4096-dimensional.

- **EfficientNet V2**

EfficientNetV2 [187] is a family of convolutional neural networks based on the EfficientNet architecture. Like its predecessor, EfficientNetV2 employs the compound scaling method to balance the network's depth, width, and resolution. By adjusting these aspects, EfficientNetV2 can be resized efficiently to work well on different devices, even those with limited resources. While retaining core components such as depthwise separable convolutions, inverted residual blocks, and squeeze-and-excitation modules, EfficientNetV2 introduces optimizations and improved normalization layers to stabilize and accelerate the training process. Advanced training techniques are also employed, including learning rate schedules, data augmentation strategies, and regularization methods. The Large model variant was selected for feature extraction, utilizing the last average pooling layer as output, and the obtained features contain 1280 dimensions.

- **DIrNification with NOisy labels (DINO) V2**

DINO V2 [134] is a self-supervised vision transformer-based model developed by Meta AI. The model is pre-trained based on the "Instance Discrimination with No Annotations"(INDA), meaning no labeled data is required, only a large dataset of unlabeled images. Richer representations are obtained as the model is trained directly on image data. DINO V2 has two main components: an encoder and discriminative self-supervised learning. The first is responsible for extracting visual features from images. It consists of multiple layers of Transformer blocks, each composed of a self-attention layer and a feed-forward network. This step generates a representation that captures crucial visual features of the image. The second is a Transformer-based discriminator consisting of multiple layers of Transformer blocks. The discriminator receives global representations of different augmented versions of the same image, and its task is to differentiate the different versions. The discriminator performance directly impacts the encoder training. Confusing original and augmented versions of an image means that the encoder needs to improve its feature extraction, as the features are not fully capturing the image's identity. Also, if the discriminator struggles to differentiate unrelated images, there may be issues with the

training, or the encoder needs to refine its visual understanding. Through this feedback loop, the encoder is helped to learn to produce robust features that capture the essence of the image and differentiate between different images and their variations. The raw model was used for feature extraction [74], generating features with 384 dimensions.

2.6 Unsupervised Distance Learning

The feature vectors associated with dataset samples can vary in spatial distribution within the feature space, potentially leading to severe impacts on machine learning tasks. On the other hand, unsupervised distance learning algorithms can improve retrieval and machine learning effectiveness [146, 193] by enhancing sample representations based on their rearrangement in the feature or distance space. We follow the retrieval models proposed for image search [37] and re-ranking [142, 150]. Formally, let $\mathcal{C} = \{T_1, T_2, \ldots, T_n\}$ be a collection composed of n time series. Let D_t be a time series descriptor, which can be defined by the tuple (ϵ, ρ), where $\epsilon : T_i \longrightarrow \mathbb{R}^d$ is a function that extracts a feature vector v_{T_i} from a time series T_i, and $\rho : \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}^+$ is a function that computes the distance between two time series T_i and T_j based on their corresponding feature vectors, formally defined as $\rho(\epsilon(T_i), \epsilon(T_j))$ or simply $\rho(i, j)$. The distance $\rho(i, j)$ for each pair (i, j) can be computed to form a square matrix A of dimension $n \times n$, where $A_{ij} = \rho(i, j)$. Based on the distance ρ, a ranked list τ_q is computed for each time series T_q present in the collection. The ranked list $\tau_q = (T_1, T_2, \ldots, T_n)$ is a permutation of the collection $\mathcal{C}$, where $\tau_q(i)$ is the rank of the time series T_i in τ_q. The rank is defined according to the distance measure, such that if $\tau_q(i) < \tau_q(j)$, then $\rho(q, i) < \rho(q, j)$. The set of ranked lists for each time series in $\mathcal{C}$ is defined as $\mathcal{R} = \{\tau_1, \tau_2, \ldots, \tau_n\}$. The re-ranking task can be seen as recalculating the distance ρ using a more effective distance function ρ_r. The function $\rho_r(i, j, \mathcal{R})$ aims to explore the contextual information present in the set $\mathcal{R}$, analyzing the relationships existing in a k-neighborhood of the ranked lists and improving the effectiveness of distances between time series. Formally, $\rho_r : \mathcal{C} \times \mathcal{C} \times \mathcal{R} \longrightarrow \mathbb{R}^+$ is a distance function between time series T_i and T_j that takes into account the contextual similarity information contained in the set $\mathcal{R}$. [195].

Rank aggregation tasks can also be performed. Let $\mathcal{A}_a = \{A_1, A_2, ..., A_n\}$ a set of distance matrices from a dataset and $\mathcal{R}_a = \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_n\}$ a set of the corresponding ranked lists. The objective is utilizing the sets $\mathcal{A}_a$ and $\mathcal{R}_a$ to obtain a new distance matrix $\hat{\mathcal{A}}_c$, in which $\hat{\mathcal{A}}_c = f_a(\mathcal{A}_a, \mathcal{R}_a)$. With the new distance matrix $\hat{\mathcal{A}}_c$, it is possible to calculate a new ranked list $\mathcal{R}_c$. Some rank aggregation methods do not utilize the set $\mathcal{A}_a$, calculating directly the new ranking $\mathcal{R}_c$, being characterized by $\mathcal{R}_c = f_a(\mathcal{R}_a)$.

Figure 2.10 illustrates unsupervised learning methods' ability to consider the dataset's geometry. Query points, highlighted with triangles, are defined in each moon of

the dataset, and the remaining points are assigned colors based on their closest query point. In (a), Euclidean Distance is used for distance calculation. It can be observed that the metric could not classify the data correctly as it does not consider the dataset's structure. In (b), the Reciprocal kNN Graph+CCs method [143] is used to recalculate the distances, which can consider the dataset's geometry and correctly classify the elements of the two moons.

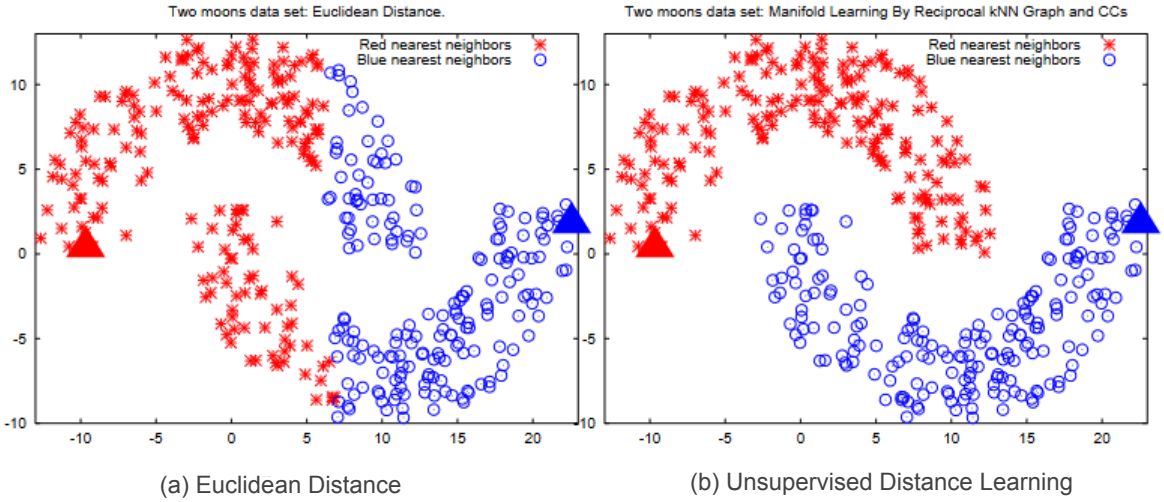


Figure 2.10 – Dataset TwoMoons: the points are colored considering the lower distance between the query points (triangles). In (a), it utilizes the Euclidean Distance, and in (b), it utilizes an Unsupervised distance learning method. Figure taken from [143]

The ability to incorporate the complete dataset structure significantly influences machine learning tasks, particularly in retrieval scenarios. Numerous methods, initially designed for image retrieval, have extended their application to various domains, including audio [41], video [42], and time series [7, 170].

Below, four recent unsupervised distance learning methods used in this work are described.

2.6.1 Log-based Hypergraph of Ranking References (LHRR)

Hypergraphs generalize graphs with hyperedges, which connect multiple vertices and can represent higher-order similarity relationships. The LHRR method uses a hypergraph model from ranking data, assigning hyperedge weights via a log-based function. Hyperedges provide contextual dataset representation, enhancing similarity measurement efficiency through the product of hyperedge similarities [146].

2.6.2 Rank Diffusion Process with Assured Convergence (RDPAC)

RDPAC [69] is an unsupervised distance learning algorithm with low computational complexity that exploits a diffusion process considering only the top-k positions from ranked lists. The algorithm starts with a similarity measure based on ranking information, and then normalization is performed to improve the ranking symmetry. The third step is a rank diffusion process with assured convergence that requires a small number of iterations. A post-diffusion step is performed to explore the reciprocal rank information.

2.6.3 Breadth-First Search (BFS) Tree of Ranking References

This algorithm constructs a tree using the breadth-first search on a graph created from dataset ranking references [148]. Starting from a query object root, the first level includes top-k similar objects according to rankings. Subsequent levels stem from top-k objects in the first level's ranking references The BFS-Tree can represent similarity information between the query and its k neighbors in a dataset. Objects occurring at different levels of the structure possibly exhibit higher similarity, while images occurring infrequently may indicate non-relevant elements or noise [148].

2.6.4 Rank Flow Embedding (RFE)

RFE is an algorithm proposed for both retrieval and classification tasks, considering contextual information to obtain better results [193]. RFE comprises five steps. The first is a ranked list normalization considering a sigmoid score based on reciprocal ranked list positions. Then, a re-ranking step is performed utilizing a hypergraph structure. The obtained hypergraph is also utilized for computing embeddings. Then, another two steps are performed for re-ranking, utilizing Cartesian Product and Connected Components, which are defined based on the hypergraph structures. More effective embeddings are computed based on the element similarity and its identified Connected Components for semi-supervised classification tasks. Enhanced unsupervised retrieval and semi-supervised classification are achieved through stepwise improvement of rank-based similarity information.

2.7 Univariate Time Series Datasets

This section introduces seven univariate time series datasets sourced from the UCR Archive [39], with several sizes, number of classes, time series lengths, and domains. The UCR Time Series Archive is the primary source of univariate and multivariate time series datasets with a wide range of domains and sizes. Not all datasets from UCR Time Series Archive were contemplated due to restricted computational resources and time.

• **Cylinder-Bell-Funnel (CBF)** [167]: A simulated dataset with 930 elements of size 128, divided into 3 classes. Each class consists of standard normal noise plus an offset term that varies for each class.

• **ECG5000** [31]: Derived from a 20-hour-long ECG record called BIDMC Congestive Heart Failure Database (chfdb). It contains 5000 pre-processed heartbeats of length 140, randomly selected from the dataset "chf07", with five classes.

• **Yoga** [207]: Composed of 3300 time series of size 426, generated from videos of actors transitioning between yoga poses. Each image was converted to a time series considering the distances between the actor's contour and the image center. The task is to distinguish between male and female actors.

• **Electric Devices** [12]: Contains 16637 time series of length 96, divided into 7 classes. The data represents daily power consumption measurements of devices from 187 households, including washing machines, ovens, dishwashers, kettles, immersion heaters, cold groups (fridge, freezer), and screen groups (computer, television). The goal is to classify each device based on its daily measurements.

• **GunPoint** [156]: This dataset is based on the movement of drawing a gun from a holster, aiming at a target, and then returning the gun to the holster. The dataset contains two classes: "Gun-Draw,"where actors use a gun replica to perform the movement, and "Point,"where they use their index finger to point at the target. The time series is captured by mapping the X-axis coordinates of the centroid of the actor's right hand. It consists of 200 time series, each with a length of 150.

• **Beef** [5]: 60 samples of silverside beef were equally divided into five classes: one of pure beef, and the other four classes are contaminated samples with 20% w/w of tripe, liver, kidney, and heart, respectively. Meat samples were submitted to cooking in a microwave oven for a limited time in different power settings. From both cooked and uncooked samples, spectrograms were obtained in the same conditions, and a chemometric analysis was conducted considering 470 data points. The problem consists of distinguishing each class.

• **Rock** [14]: Contains 70 rock samples from the *ASTER spectral library*. The series corresponds to the spectral reflectance as a function of wavelength in microns, not time. There are 4 types of rocks, containing time series of lengths of 2844, and the problem consists in distinguishing them.

This Section described diverse univariate time series datasets, used in two different experiments. Chapter 5 presents multivariate time series datasets used in the experiments.

2.8 Evaluation Measures

The evaluation of the effectiveness of retrieval and machine learning tasks relies on evaluation measures [158, 163], which will be discussed in this Section.

- **Accuracy**

This metric, widely used to evaluate classification tasks, calculates the percentage of elements correctly labeled within a dataset that has been labeled [168]. The multi-class accuracy is defined by Equation 2.6:

$$accuracy = \frac{corr\_preds}{preds}, \tag{2.6}$$

where *corr_preds* stands for the correct predictions in a labeled dataset, and *preds* is the total number of predictions.

- **Adjusted Mutual Information (AMI)**

AMI is a statistical measure, utilized to evaluate clustering tasks, that quantifies the similarities between two sets of clusterings, considering the Mutual Information (MI) score and correcting the chance agreement [199]. The AMI between two clusters is given as Equation 2.7:

$$AMI(U,V) = \frac{[MI(U,V) - E(MI(U,V))]}{[avg(H(U),H(V)) - E(MI(U,V))]}, \tag{2.7}$$

where $E(MI(U,V))$ is the expected mutual information and $H(U)$ is the entropy of a cluster U.

- **Precision@k**

The Precision is widely used to evaluate retrieval tasks and aims to evaluate the proportion of relevant items among all the retrieved items in a depth k. It is defined as follows in Equation 2.8:

$$P = \frac{TP@k}{k}, \tag{2.8}$$

where $TP@k$ represents the number of relevant retrieved items in a depth k of the ranked list.

- **Recall@k**

Recall is also utilized to evaluate retrieval tasks and measure the proportion of relevant instances retrieved. Recall is defined as follows in Equation 2.9:

$$R = \frac{TP@k}{TP}, \tag{2.9}$$

where $TP@k$ stands for the number of relevant retrieved items in depth k of the ranked list, and TP is the total number of retrieved relevant items in a ranked list.

- **Mean Average Precision (mAP)**

Mean Average Precision (mAP) is a common metric for evaluating the effectiveness of ranked lists obtained in information retrieval tasks. This metric calculates the average of the Average Precision (AP) for each query [231]. The mAP is defined in Equation 2.10:

$$MAP = \frac{\sum_{l=1}^{Q} AP(q_l)}{Q}, \tag{2.10}$$

where Q is the number of queries and q_l represents a query l. The Average Precision (AP), as defined in Equation 2.11, is the average precision for a recall varying from 0 to 100% [231]:

$$AP = \frac{1}{N_r} \sum_{i=1}^{d} (\frac{r_i}{1} \sum_{j=1}^{i} r_j), \tag{2.11}$$

where N_r is the number of relevant items in a collection for a query q and r_i represents the relevance (irrelevant = 0 or relevant = 1) of the i-th item in a ranked list of size d.

2.9 Software and Frameworks

This Section briefly presents the main frameworks and software utilized in this work. Their main uses are also discussed.

- **UDLF [194]:** This is a framework for unsupervised distance learning, implemented in C++, and containing diverse algorithms, including the ones explained in Section 2.6. It performs re-ranking and contextual rank aggregation starting from distance, similarity, or ranking matrices;
- **pyts [59]:** This is a Python programming language library designed with tools for time series processing. It contains machine learning algorithms for time series, datasets, preprocessing, and utility tools;
- **Matplotlib [79]:** This Python library is utilized for image and graphics generation and manipulation. Besides the graphic generation, we also utilize Matplotlib to generate the images described in Section 2.5.2;
- **Scikit Learn [140]:** A complete machine learning library for Python. It contains machine learning algorithms, evaluation measurements, and pairwise distance measurements, among other tools;
- **NumPy [70]:** A fundamental tool in Python for manipulating arrays and matrices;
- **Tensorflow [3] and PyTorch [138]:** These are two Python libraries for deep learning algorithms. Mainly utilized to execute the neural networks described in Section 2.5.3.

3 Related Work

Time series data contains diverse encoded information, and effectively analyzing them poses significant challenges. Additionally, the increasing production of time series datasets demands efficient management strategies. One way to work with time series datasets is through information retrieval tasks.

In information retrieval problems applied to time series, two evaluation approaches depend on the available data. In problems involving a dataset with multiple time series, a single series is selected as a query, and other series are ranked in ascending order based on a distance measure, representing their similarity to the query [7]. For instance, in a database containing time series derived from satellite images for studying plant phenology, one might want to retrieve similar data to a given query series to identify the same plant species in different regions [7]. On the other hand, problems with a dataset containing a single time series involve selecting a segment of defined size and ranking segments closest to the query based on a distance measure [227]. For example, in a time series containing an electroencephalogram associated with eye states (open or closed), one might select a segment where the eyes are in a specific state and then retrieve more segments with the same eye state [227].

Over the years, information retrieval in time series has been approached in various ways and contexts. Numerous models have been proposed, ranging from general approaches for diverse data scopes to specific methods for applications such as analyzing time series obtained from satellite images captured over time [135], or techniques tailored for searching multivariate time series [227], for instance.

Other tasks provide alternative perspectives on time series analysis. Tasks like clustering and classification allow grouping sets of time series considering data characteristics, revealing patterns within a dataset. This can be used to analyze changes in population mobility patterns caused by external factors [225], detect early-stage cancer [228], or predict disease outbreaks [71]. On the other hand, forecasting tasks aim to predict future values of a time series. Relevant applications include financial market predictions [223] and echocardiogram forecasting [157], which can identify abnormal heartbeat patterns and enable early medical intervention.

The literature presents a variety of methods for time series analysis. Although information retrieval is a crucial task in data processing, it is a relatively unexplored area in the context of time series compared to labeling and forecasting tasks. Section 3.1 discusses models applied to univariate time series, while Section 3.2 presents methods for multivariate time series. Section 3.3 discusses techniques for processing time series

generated from images and methods that encode time series as images. Finally, Section 3.4 briefly discusses the literature gaps filled by this work.

3.1 Univariate Time Series

3.1.1 Information Retrieval

Analyzing information in time series efficiently and effectively is a challenging task. Over the years, various methods have been proposed to address and optimize this issue. Univariate time series are more common, and much of the existing research focuses on handling this data type.

Comparative studies, such as the one proposed by Ding *et al.* [50], allow for evaluating and comparing various available time series retrieval methods. In this work, comparisons are made among 8 different time series representation methods, including DFT, DCT, DWT, PAA, APCA, SAX, CHEB, and IPLA, as well as 9 similarity metrics, including SpADe, TQuEST, ERP, EDR, LCSS, DTW, DISSIM, Swale, and Euclidean, across 38 univariate time series datasets. This study provides a wide range of results for various datasets.

Furthermore, another challenging aspect in the field is dealing with extremely large datasets. Retrieving information from large datasets, not only in the context of time series but also for any multimedia data, is a computationally demanding task since comparing a query with all elements in the dataset is highly costly. In this regard, various algorithm optimizations are made to reduce the search time. In [155], several optimizations for distance metrics (such as Euclidean distance and DTW) are presented, enabling retrieval tasks to be performed much faster than other works developed during the same period. The algorithm DTW, which accomplishes retrieval quickly and delivers comparable results to the state of the art in the period, demonstrated to be consistent. Recent approaches propose tackling this issue differently, as seen in [172]. In this work, an adaptive model for time series information retrieval is proposed, based on relevance feedback, to retrieve a small number of series per query, considering large datasets. As searches are performed, the algorithm builds a collection of patterns existing in the time series, aiding in adapting the algorithm to perform more efficient searches.

In addition to relevance feedback, other machine learning methods can be used to improve the efficiency and effectiveness of information retrieval results. For instance, in [137], a supervised dictionary learning method, called EO-BoW, is introduced for optimizing the Bag-of-Words (BoW) representation in Information Retrieval. The method utilizes an entropy-based optimization criterion, suitable for retrieval tasks rather than classification, following the cluster hypothesis. Experiments conducted in time series

databases, and data from other domains, demonstrate that EO-BoW improves retrieval performance, and reduces encoding time and database storage requirements. Moreover, EO-BoW maintains its representation ability when used for retrieving objects from unseen classes during training. Chen et al. [30] propose the deep multi-task representation learning method (MTRL) for time series classification and retrieval tasks, leveraging supervised and unsupervised information. Supervised information for classification tasks aims to reduce intra-class differences and increase inter-class differences. Unsupervised information for retrieval tasks aims to preserve DTW distances between pairs. These tasks share networks, so the information obtained from one task can benefit the other. The results presented in the experiments outperform the other state-of-the-art methods at the time.

Often, labeled information is not available, making supervised learning tasks infeasible. In [21], the problem of content-based time series retrieval (CBTSR) is addressed, which bears similarities to the proposal of this work. Using satellite sensor images, the goal is to retrieve data related to abrupt changes. For this purpose, features of the type "change vector analysis"(CVA) are extracted, and a clustering and spectral change vectors (SCV) based approach is used for retrieval: the elements to be retrieved have SVCs in the cluster to which the query belongs. This membership is verified using Euclidean distance, and other distance measures can also be utilized. The results, considering the top 20 retrieved elements, achieve precision rates between 89% and 99%.

3.1.2 Clustering

Clustering tasks in time series data are also fundamental, and several approaches have been proposed. Many current clustering methods consider traditional distance measures, which do not take shape characteristics into account. In this context, algorithms that consider this property have become popular for time series data. The "Fractional Order Shape-based k-cluster (FrOKShape)"algorithm [109] utilizes a distance measure based on multivariate shape by normalized fractional order cross-correlation and the "DTW Barycenter Averaging (DBA)"measure for centroid calculation. The obtained results are comparable to the "KShape"algorithm, which is widely used for time series data. The centroid approach is similar to the popular "K-Means"algorithm in the literature, which performs well in various domains. However, the technique used to calculate centroids is not satisfactory for handling time series data. In this regard, Krishnan et al.[83] propose adaptations in the clustering algorithm, initializing centroids in the farthest neighbors, contrary to the traditional approach that initializes centroids randomly. Additionally, distance calculations are performed using the DTW algorithm. Finally, the clustering is done using Kohonen maps. This approach showed better performance and scalability than traditional methods. Other approaches, such as hierarchical clustering, can also be applied to time series data. By using a hierarchical clustering method based on

tail dependence coefficients, the authors propose a linkage method based on the tail dependence between clusters merged at each iteration[43]. The proposed linkage method showed superior results compared to traditional approaches. Another relevant approach is presented in [153], where a framework for clustering is proposed, involving obtaining geometric representations of time series data and then performing clustering tasks. The best results were achieved using the UMAP and HDBSCAN methods. The work proposed in [8] evaluates traditional clustering algorithms with three proposed time series similarity measurements, using meteorological and synthetic data, and it is noteworthy that the results between the measurements are similar, and the clustering results are strongly correlated to the parameters. Kang et al. [91] propose the PFD_DTW_HC method for clustering time series. First, Hodrick Prescott (HP) filtering is applied to reduce noise and redundancy in the raw time series, then, the polynomial curve fitting (PCF) is applied to derive a globally continuous function fitting for time series. The time series is transformed into a multi-order derivative feature sequence. The designed polynomial function derivative features-based dynamic time warping (PFD_DTW) algorithm is performed to determine the distance between two equal or unequal granular length time series, and hierarchical clustering is applied. The proposed method has advantages in capturing trend information and interpretability and is more precise than other clustering alternatives, but is sensible to outliers and has high computational complexity.

The clustering of time series data has various applications. For instance, the method proposed in [78] enables the identification of the condition of an aluminum reduction cell using clustering. The method utilizes fuzzy c-means in combination with the DTW distance for clustering the current states of the anodes, along with the pseudo resistance of an anode-cathode path. Finally, a clustering validation index is used to find the ideal number of clusters. Another notable work is the proposal of the "KDetect"algorithm [176], designed for unsupervised anomaly detection in cloud systems. Time series representing the CPU consumption of unknown services are used in a clustering algorithm, enabling anomaly detection. In the experiments, the precision and recall values exceeded 98%. Moreover, the clustering of time series data enabled the analysis of changes caused by the COVID-19 pandemic. Motivated by the analysis of pattern changes in mobility due to COVID-19, Zhang et al. [225] propose the use of the "K-Means"method with the DTW distance measure, achieving more satisfactory results, although with high execution time.

3.1.3 Classification

Unlike clustering, which is an unsupervised learning method, classification tasks allow data labeling based on a pre-identified set. Among the challenges in classification problems is the selection of good features. Lee et al. [103] propose a representation method for open-ended univariate time series that is real-time, normalization-free, and

parameter-tuning-free. Each data point of the time series is converted into a root mean squared error (RMSE) value based on Long Short-Term Memory (LSTM) and a Look-Back and Predict-Forward strategy on the fly. Experiments suggest that the approach is suitable and can be easily deployed on commodity computers and used as a pre-processing tool for real-time time series analysis, such as clustering and classification. Especially in the classification field, a model for selecting temporal features is proposed using a random forest for classification performed by a FCN (Fully Convolutional Network). Several experiments demonstrate the effectiveness of this method [85]. Building on the issue of features in time series data, another method called Adaptive Feature Fusion Network (AFFNet) was created. It combines temporal and data features extracted using a dynamic multiscale CNN model. Distance features are extracted from a prototype distance network model. Finally, a feature fusion model is utilized for classification, and the proposed method achieved superior results compared to the state-of-the-art for many tested datasets [203].

While the utilization of labeled data allows for the construction of robust models, the availability of labeled data poses a challenge for supervised learning tasks. In this context, semi-supervised learning models aim to deal with the scarcity of labeled information in conjunction with the available unlabeled data. However, semi-supervised time series classification models often do not adequately exploit the existing temporal correlation between data and do not fully utilize the unlabeled data. To address these issues, a semi-supervised time series classification model with self-supervised learning was proposed in [210]. This model promises to address the mentioned problems and achieve superior results compared to other approaches. The model proposed by Wei et al. [206] employs a series of unsupervised tasks to capture time-frequency information, and then a multi-task learning framework, considering the consistency between labeled and unlabeled data, to learn common features, and a semi-supervised classification is performed. The use of feature extractors in time series is studied in [66], where the BOSS (Bag-of-SFA-Symbols) feature extractor is applied to different datasets for evaluating the performance of various classification and clustering methods. Although the proposed framework did not surpass the state-of-the-art, it shows great potential in the field. The applications of classification are diverse. For instance, in the study of cattle behavior [93], a shallow CNN is used for the classification of cattle behavior, achieving superior results compared to a stacked autoencoder.

3.1.4 Univariate Time Series Prediction

Indeed, in addition to labeling tasks, time series prediction tasks are essential in today's world. In [68], re-ranking approaches are used for improving prediction tasks in diverse forecast datasets, by proposing a framework with a base kNN regressor, unsupervised re-ranking algorithms, and predictions associated with nearest neighbors weighted based

on their ranking positions. Promising results were shown and, despite the work was not conduced in the time series contexts, the proposal is suitable for time series forecasting.

Predicting the volume of patients in an online healthcare service, for example, is crucial for managing staff and scheduling appointments. In this context, Kazmi et al. [94] conducted a comparative study using various prediction methods to estimate the volume of patients in an online service. In this study, ARIMA and LSTM models showed the best performance. Predictions of echocardiograms assist in performing interventions at the right time and also in preventive measures. For this purpose, a MLP model for echocardiogram forecasting is proposed in [157], achieving accuracy values above 95% in simulated experiments and 98% accuracy in real data. Nguyen et al. [154] propose a bidirectional LSTM model for performing prediction tasks. However, the existence of various neural networks and hyperparameters to be adjusted is a challenging aspect that makes their use more costly.

Nevertheless, neural networks have brought significant advancements in time series prediction. Taking this into account, Li et al. [108] propose a method for automating the search for the best neural network for univariate time series forecasting. The networks discovered by the proposed algorithm showed competitive results compared to other tested methods. The proposal in [184] can be applied in both univariate and multivariate series forecasting. The Mode Decomposition and 2D Convolutional Network (MDCNet) is a long-term time series forecasting architecture based on decomposition and multi-frequency feature extraction with multi-scale 2D convolution. The model is composed of a Variational Mode Decomposition Block, which decomposes series into trend components and stationary modal components at different main frequencies, a Trend Prediction Block, and an Intrinsic Mode Functions Prediction Block, to capture global correlation and hidden dependencies in different frequencies, and a Frequency Enhancement Module, to reduce impact of noise. The results of different modules are merged, to obtain the time series modeling. Performed experiments show that MDCNet reduces errors by 15.1% and 11.5% over previous SOTA methods in multivariate and univariate time series, respectively. In [226], a Stacked Convolution sequence AutoRegressive encoding Network (SCARNet) is proposed for time series prediction, presenting satisfactory results and computational efficiency. The model architecture performs to extract multi-scale information to form the hidden vector for prediction and filter unnecessary information. SCARNet is utilized to predict each dimension of a multivariate time series as an univariate time series.

3.2 Multivariate Time Series

Multivariate time series have gained more prominence in recent years due to the wealth of information these data contain about a particular phenomenon. As these are

time series in which each data point is n-dimensional, models for representation, indexing, retrieval, and other machine learning tasks may differ from univariate time series, although it is not mandatory. Due to this, several models have been proposed over time.

3.2.1 Information Retrieval

One such framework proposed for retrieving multivariate time series is the "Deep Unsupervised Binary Coding Networks (DUBCNs)" [227]. This method consists of an LSTM encoder-decoder with three components: (i) a mechanism for temporal encoding of different sequences, (ii) a clustering loss function in the hidden feature layer to identify hidden features, and (iii) an adversarial loss function based on Generative Adversarial Networks (GANs) to improve the generalization of obtained binary codes. The retrieval tasks are based on the Hamming distance between the binary code of the query time series and the binary code of a candidate time series. The results were superior to the state of the art. Following a similar logic, Song et al. [180] proposed the "Deep r-th root of Rank Supervised Joint Binary Embedding (Deep r-RSJBE)"method for multivariate time series retrieval. LSTM units are used to encode the temporal dynamics of the series, and CNNs are used to encode the interactions between pairs of time series. This information is then incorporated into a binary set. Finally, a ranking loss function based on the r-th root is employed to optimize the accuracy at the top of a ranked list calculated by the Hamming distance. Experiments demonstrate the effectiveness of the method.

Hashing-based strategies are also frequently approached for multivariate time series retrieval. The work developed by Yu et al. [218] focuses on retrieving multivariate time series using a model based on the locality-sensitive hashing (LSH) approximation method, aiming for efficiency. As the number of comparisons between time series is one of the main bottlenecks, this method proposes limiting the search space for a query and using the LSH-based model to identify candidate time series. The second step involves using the LSH hash value to reduce further the number of candidates to be retrieved. Finally, the Euclidean distance or an estimate of DTW between the query time series and the candidate series is calculated. The work in [188], proposes the Unsupervised Transformer-based Binary Coding Networks (UTBCNs) for multivariate time series retrieval. It is composed of a Transformer-based encoder (T-encoder), a Transformer-based decoder (T-decoder), adversarial loss, and hashing loss. The time series are converted to binary coding and the rankings are calculated based on the Hamming Distance. The results outperform the state-of-the-art.

In [125], the Deep Hashing Network for Retrieval-based Anomaly Detection (DHN-RAD) is proposed. First, time series segments are transformed into low-dimensional features using a temporal encoder, built with LSTM units. Then, two hash functions are applied to predict two binary codes with different lengths from each feature. This can

contribute to accelerating the task. Sub-linear search is, then, performed, reaching effective results. A pattern search algorithm is proposed based on the symbolic representation, regular expression, and query expansion (SAXRegEx) in [220]. First, time series are encoded using Symbolic Aggregate approXimation (SAX), enabling using text retrieval techniques. Then, time series are represented as regex, and query expansion is performed to cope with distortions. Finally, the query regex is searched in the time series. Good results were achieved and the proposed algorithm can be applied in any domain. Yu et al. [221], propose the Interactive Pattern Search in Multivariate Time Series with Locality-Sensitive Hashing and Relevance Feedback (PSEUDo), achieving superior results than other tools for interactive MTS analysis. Series are split into windows and Local Sensitive Hashing (LSH) is performed, hashing them into hash table buckets representing the similarity distribution to the query (green means similar and red, dissimilar). Representatives for both similar and dissimilar buckets are drawn, relevance feedback is given and the results are updated.

3.2.2 Clustering

Like in univariate time series, clustering, classification, and forecasting algorithms also play a prominent role in multivariate time series. Ienco et al.[80] propose the "constrained Deep embedding time SEries Clustering (conDetSEC)"framework, based on GRUs, for semi-supervised clustering. This method combines an embedding generation step with a refinement step for the clusterings and shows significant results compared to the state-of-the-art. Another presented framework is "FeatTS"[190], which proposes a semi-supervised clustering method where time series are represented as graphs with various extracted features. This is followed by community detection and the construction of a co-occurrence matrix to select the best clusters.

3.2.3 Classification

Addressing classification in multivariate time series, traditional methods often consider distances in the time domain, ignoring frequency characteristics. In this context, Chen et al. [33] propose a model that takes both characteristics into account. It performs a discrete wavelet decomposition of the time series, generating sub-time series, and extracts features over time and domain. A deep neural network (DNN) is then developed with a learning metric layer that can learn the non-linear characteristics of each level of the representation and also learn semantic similarities between time series. A consistency regularization term between levels is used to improve distance metrics, which are used in a 1-NN classifier. Other networks are widely used in the domain of multivariate time series. Combining the positive characteristics of ESN models, such as high-dimensional random projection and training efficiency, with their limitations in capturing long-term dependency

information, the "Multiscale Echo Self-Attention Memory Network (MESAMN)" [118] was proposed for multivariate time series classification. It consists of a memory encoder, formed by ESN units followed by attention mechanisms, and a memory learner, composed of a multiscale CNN. The results presented are highly effective. Furthermore, in another work, the combination of CNNs and ESNs for the classification of univariate and multivariate time series [82] was proposed, leveraging the strengths of both models. The results in terms of accuracy and sensitivity were superior to other existing algorithms.

Dyformer [213] is an algorithm for multivariate time series classification that incorporates hierarchical pooling to decompose the series into subsequences with different frequency components. Then, the Dyformer modules are employed to achieve adaptive learning strategies for different frequency components based on a dynamic architecture. Also, feature-map-wise attention mechanisms are applied to capture multi-scale temporal dependencies, as well as a joint loss function to facilitate model training. MTS2Graph [217] is a framework for interpreting decisions of convolutional neural network (CNN) models for multivariate time series. It first extracts Multivariate Highly Activated Period (MHAP), that corresponds to representative patterns from the input data that highly activate neurons in the network, and groups MHAPs using K-shape clustering to enhance the generalization. Then, a graph is constructed to obtain the temporal correlation between the MHAPs, where nodes correspond to an MHAP, represented by a cluster median, and edges represent the chronological occurrence of MHAPs. A graph embedding is utilized to learn the new feature representation and a new classifier is trained using the obtained features. The framework presents state-of-the-art performance and helps with interpretability.

In [186], a Deep Learning framework for multivariate time series classification based on embedding temporal discretization is proposed. The proposal is composed of temporal discretization, performing discretization and partial selection of primary features, and model training, that selects more accurate features and performs classification. Du et al. [51] propose a multivariate time series classification method based on fusion features (MTSC_FF). It starts extraction frequency domain features with an attention layer based on continuous wavelet transform. At the same time, it extracts long-range dependency features from the time domain with a sparse self-attention layer and obtains spatial correlations between multivariate time series dimensions with the Kendall coefficient. A Graph Neural Network is used for fusing features, and the classification is performed by a fully connected layer. This method provides an improvement over the interpretability of results with the visualization of the classification-dependent features. The work in [52] proposes a multi-feature-based network (MF-Net) for multivariate time series classification. The first stage consists of extracting global and local features with the global-local block. The second stage is obtaining spatial dependency features from the integrated previous features using the spatial-local block. The classification is implemented using a multilayer perceptron (MLP).

Indeed, there are various applications for the classification of multivariate time series. For instance, Harris et al. [71] considered predicting peaks in the Crimean-Congo Hemorrhagic Fever disease based on climatic data. This disease can cause up to 50% mortality in humans and is of high priority for international health organizations. To predict peaks, they used the state-of-the-art multivariate time series classifier, the "Multivariate Attention Long Short-Term Memory Fully Convolutional Network (MALSTM-FCN),"along with climate data from a specific region. This model achieved an accuracy of over 91% for predictions in the Pakistan region. Another application in the health domain is the prevention of non-communicable diseases. Early detection of diseases like breast cancer is crucial for reducing mortality rates. Zhu et al. [228] propose a home-based breast cancer identification method based on classification. Sensors installed in bras monitor the temperature and texture of the breasts, generating multivariate time series that are classified using various deep learning classifiers. The highest-performing result is selected. One of the challenges faced in this application is the lack of available training data, which is mitigated using data augmentation techniques. Yehuda et al. [216] presents a self-supervised framework for clinical multivariate time series classification, where dynamics of multiple time series are learned as a supervisory signal. The dynamics are learned based on a mathematical framework of Koopman theory, which assumes a high-dimensional embedding space where the operator propagating from one time instant to the next is linear. Experiments on ECGs and EEGs show superior results compared to the state of art classifier without Koopman-aligned optimization for representation.

These examples highlight how multivariate time series classification can be applied to diverse domains, from predicting disease outbreaks based on environmental factors to early detection of life-threatening diseases using wearable sensors. The power of deep learning methods, such as LSTM and CNN-based models, has proven to be effective in these applications, making significant contributions to the fields of healthcare and epidemiology.

3.2.4 Multivariate Time Series Prediction

Applications based on prediction are quite common in the domain of time series. The use of Transformers for time series prediction has proven to be highly efficient, especially for handling long sequences [177]. To improve efficiency, many works have partially combined the architecture of Transformers with CNNs. The work proposed by Shen et al. [177] introduces the concept of tightly-coupled convolutional Transformer (TCCT) and three associated models. Experiments demonstrate that this proposal can yield significant improvements in the state-of-the-art of time series prediction using Transformers. Another neural network-based approach is presented by Long et al. [117], which proposes a model for multivariate time series prediction based on nonlinear spiking neural P (NSNP) systems

and non-subsampled shearlet transform (NSST). A time series is converted to the NSST model, and the NSNP system makes predictions. Furthermore, Chen et al. [32] propose the multichannel fusion graph neural network (MCFGNN) model for multivariate time series prediction. This framework employs three channels to capture meta-features, time variations, and global stabilization graphs. The proposed model outperforms state-of-the-art methods in terms of predictive accuracy.

In [57], sensor data from machinery is utilized to predict their remaining useful life. For this purpose, Neural Turing Machines (NTMs) are used for feature extraction, and two fully connected layers are employed for prediction. Choi et al. [34] propose a framework for time series representation learning suitable for diverse tasks, such as classification, forecasting, and anomaly detection. It generates Positive and negative pairs to establish a multi-task learning setup, then, it formulates contrastive losses to investigate both contextual, temporal, and transformation consistencies, which are optimized, together, to learn general time-series representations. In [81], weather, holidays, and other information are considered for predicting the number of restaurant customers in tourism resorts. The LSTM-based approach showed effective results. Yu et al. [219] propose the time series prediction model NeA3L, employing a framework based on a heterogeneous "encoder-decoder" fusion of Attention mechanism structure for multivariate time series feature extraction. The N-encoder uses N Transformer-encoders for extracting features in the same spatiotemporal context. The Attention layer is used to extract the correlation features and perform the dimension transformation. The Decoder part uses a 3-layer LSTM model, merging the different features and excavating hidden features. The output summarizes cumulative changes of the whole history, and the hidden state carries the final state of the hidden variables. Multiple output layers are used to generate the final prediction results for different time variables.

Functional Relation Field (FRF) [106] proposes a framework for boosting the performance of multivariate time series prediction. This framework can discover the intrinsic graph structure of data, and improve the flow forecasting performance by applying constraint function relationship to the output during training and testing. These constraints can be incorporated into existing backbone networks. Results show that the application of the FRF improves the accuracy of the prediction. Robformer [222] is a decomposition-based Transformer inspired by the polynomial fitting method for long-term multivariate time series forecasting. It consists of three new inner modules to enhance the predictability of Transformers, the robust series decomposition block (RobDecomp), seasonal component adjustment block (SAB), and robust trend forecasting block (RobTF). The first two robustly decompose fluctuating time series, while the RobTF module extracts long-term multiple trend patterns from the view of the polynomial fitting. The method presents improvements over two state-of-the-art methods in five different fields of application (energy, economics, traffic, weather, and disease).

These works illustrate the increasing interest in using deep learning techniques, such as Transformers, LSTM, NSNP, and NTMs, to enhance the efficiency and effectiveness of time series prediction, opening up possibilities for applications in various domains such as healthcare, manufacturing, and demand forecasting.

3.3 Image Based Approaches

Images can encode various information that can be explored for different purposes. Also, images and time series are strongly correlated, where time series can be obtained from images, as in [7], but also the time series can be represented as images, for example, in [204]. In this Section, we intend to discuss both correlations.

3.3.1 Images to Time Series

Remote sensing using satellite images allows obtaining multiple images of the same location over time, enabling the representation of this sequence of images as time series using different methods. Representing this data as time series allows studying the region from a different perspective, both visually and computationally. In the context of information retrieval, a search for images would retrieve a collection of similar images. However, in the context of time series, the objectives are different, as it enables analyzing the area's transformations over time, providing a completely different interpretation of the same initial dataset in comparison to the results obtained from image-based analysis.

One of the applications of this process is in the field of Ecology, particularly in the study of plant phenology. Almeida et al.[7] present a proposal for re-ranking in the context of time series obtained from satellite images for plant phenology study. They use an unsupervised distance learning algorithm that recalculates the distance between two time series considering the number of reciprocal neighbors present in a neighborhood of size k, where the common elements are weighted according to their position in the ranked lists of time series [144]. This method has shown significant improvement in results. In a similar context, in [170], another method for information retrieval from the same dataset is proposed. Time series are represented by images of *Recurrence Plots*. From these images, features are extracted and then textual signatures are obtained from these features. Information retrieval is performed using the boolean model and the vector space model based on the textual information, which has shown efficient results. In another context, in the work of Pailot-Bonnétat et al. [135], an analysis is carried out on an eruption at Mount Etna in 2013. The volcanic plume is crucial information for studying the dispersion of volcanic ash, which is important for identifying hazards to aircraft. By applying the shadow technique, which takes into account shadows projected on the ground and sun geometry to calculate the height of the plume, it was possible to calculate the variation of

plume height over time, based on information retrieval from available data. This enabled documenting the rise and dispersion of the volcanic plume, providing valuable insights for the study of volcanic events.

3.3.2 Time Series to Images

On the one hand, encoding sequences of images into time series yields good results, and on the other hand, encoding time series into images also produces satisfactory outcomes. In the works proposed by Wang et al.[204, 205], time series are transformed into images using the methods *Gramian Angular Fields* and *Markov Transition Fields* and classified by Tiled CNNs, achieving highly competitive results. A similar approach is used for tourist demanding forecast, where time series are imaged using the two previous methods and *Recurrence Plots* and its features are extracted using a CNN. These features are used to feed an LSTM network that will learn the temporal dependencies, during the process of model training [19]. A transfer learning approach for Non-Intrusive Load Monitoring (NILM), proposed on [99], is based on converting low-frequency data into images using GAF, extracting features from these images with VGG16, and then training a classification algorithm to identify the ON/OFF status of the appliances. In [86], a multi-modal fusion transformer is used for multivariate time series classification, reaching superior accuracy than other baselines. GAF is used to convert time series into images, and CNN is used to extract features from 1D time series and 2D images separately. The representations are fused with a transformer encoder and the output feeds a CNN ResNet for classification. In the approach for multivariate time series classification proposed in [104], both time and frequency domains of the time series are considered for the classification task, providing a different perspective for multivariate time series analysis. The proposed model has two modules: the time domain, constructed with CNNs and GCNs, that captures both temporal and spatial dependencies in the time domain, and the frequency domain, where DFT is employed to obtain the frequency spectrum of MTS, and these features are treated as images, transforming the time series classification task into an image classification task in the frequency domain, using a CNN ResNet. The proposed framework considers both information for the classification task.

Additionally, it is also possible to perform time series forecasting using images. Zeng et al. [223] propose representing financial time series as 2D images, where the time series history is formed by a sequence of images, similar to a video. The prediction of future data is done by a CNN, which predicts new images. Thus, time series forecasting is achieved through a video prediction model. This approach yielded superior results compared to various traditional methods such as ARIMA and Prophet. A similar approach is proposed in [181], where time series are represented as images, and the prediction of new images is performed using methods based on *Autoencoders*. Again, the proposed approach

outperforms the ARIMA model, but the results were less satisfactory for financial data and more satisfactory for cyclical data. A third approach with the same idea is presented in [174]. The *ForCNN* method is proposed, which uses convolutional and dense layers to perform time series forecasting. Neural networks based on *VGG-19* and *ResNet-50* models, as well as a self-designed architecture, are tested.

3.4 Literature GAPs

This chapter presented diverse works in the field of time series analysis, encompassing retrieval, classification, clustering, and forecasting for univariate and multivariate time series.

Despite the variety of proposed methods and the reached results, some gaps in the literature can be observed:

1. **Overlook of information retrieval** : The number of works approaching time series information retrieval in the literature is visibly lower than forecasting or classification approaches, despite the importance of retrieval tasks. Our work approaches content-based time series retrieval in Chapters 4 and 5.

2. **High dependence of deep learning approaches**: Deep learning models require large amounts of training data and are also time-consuming. State of art algorithms in time series use, mainly, deep learning approaches. The proposals in this work use pre-trained deep learning models only for image feature extraction, using other tools to perform tasks such as retrieval, classification, and clustering, and reaching competitive results.

3. **Time series feature extraction**: As mentioned, the state of the art in the time series domain is highly attached to deep learning-based methods, and so is the feature extraction. Diverse methods proposed for feature extraction richly explore time series characteristics, but the feature extractors are specifically built for a single deep learning algorithm, which can make difficult use of the feature extraction approach in other tasks. In this work, diverse feature extractors were used in different domains.

4 Re-Ranking and Representations for Time Series Retrieval: A Comparative Study

Despite important, information retrieval in time series is still overlooked in the literature. This work tries to fill this gap by introducing a comparative study that includes different techniques for time series representations and ranking. Special attention is given to comparing the retrieval effectiveness performance of recently proposed re-ranking methods based on unsupervised distance learning in time series search tasks. Conducted experiments included the comparison of 10 distinct representations and four different re-ranking approaches on six diverse time series datasets. Experimental results demonstrate that the proper combination of representation and re-ranking methods can often enhance the overall quality of ranking results, leading to gains on mAP up to 31.78%. We could also observe that no single combination outperforms all others for all datasets. Those results suggest that the choice of time series representation and re-ranking method depends on the specific application context. In summary, our main contributions are:

- Comparative study involving ten different time series representation methods in time series retrieval tasks;
- Comparative study involving four different re-raking methods based on unsupervised distance learning algorithms in time series retrieval tasks;
- Comparative study of different retrieval systems created based on combining different time series representations and unsupervised distance learning approaches.

The rest of the Chapter is organized as follow: In Section 4.1, we define the proposed methodology, while Section 4.2 defines the experimental protocol and Section 4.3 discusses the results. Section 4.4 gives the final considerations about the results.

Findings related to the conducted study were reported in an article submitted to a journal [165].

4.1 Proposed Methodology

The effectiveness of time series retrieval tasks is influenced not only by the representation but also by the distance (or similarity) functions employed to assess how close two feature vectors are. The use of reliable and effective distance measurements between the data objects is of fundamental importance. Generally, the used distance

measures, such as the Euclidean distance or the Dynamic Time Warping [175], consider only pairwise relationships [50], despising the existing contextual information in the whole dataset.

In contrast to pairwise measures, rank-based unsupervised distance learning algorithms are a powerful tool, capable of analyzing the whole dataset structure and considering the neighbor relationships to compute more effective distance measures. Unsupervised and semi-supervised methods are promising as they do not require large volumes of labeled data, which can be expensive and time-consuming to obtain [207]. Different approaches have been proposed in the literature, recently. One research line refers to the use of graphs and hypergraphs, which are utilized to analyze connections between different points and discover new relationships in a dataset [146, 143]. These methods were extensively evaluated in image datasets, achieving competitive results when compared with other ranking algorithms. This work presents a comparative study to assess diverse time series descriptors and a range of re-ranking algorithms.

A schematic view of the conducted steps of this work is presented in Figure 4.1. First, time series are represented through diverse feature extractors, considering both methods that handle time series as 1D signals and those that convert series into 2D representations that are then combined with the ResNet-152 and Vision Transformers deep-learning-based feature extractors (Section 2.5). The methods were:

- Beam Angle Statistics (BAS) [9],
- Discrete Fourier Transform (DFT) [171],
- Discrete Wavelet Transform (DWT) [28],
- Random Convolutional Kernel Transform (ROCKET) [45],
- Gramian Angular Fields (GAF) [204] with CNN ResNet-152 [72],
- Gramian Angular Fields (GAF) [204] with Vision Transformers (ViT) [96],
- Markov Transition Fields (MTF) [204] with CNN ResNet-152 [72],
- Markov Transition Fields (MTF) [204] with Vision Transformers (ViT) [96],
- Recurrence Plot (RP) [53] with CNN ResNet-152 [72],
- Recurrence Plot (RP) [53] with Vision Transformers (ViT) [96].

For different queries, where each time series of a dataset is a different query, collection time series are ranked according to the Euclidean Distance of their representations to the queries. Then, re-ranking methods based on unsupervised distance learning algorithms are employed, as formally described in Section 2.6, which are:

- Log-based Hypergraph of Ranking References (LHRR) [147],
- Breadth-First Search Tree of Ranking References (BFSTREE) [148],
- Rank Diffusion Process with Assured Convergence (RDPAC) [69],
- Rank Flow Embedding (RFE) [193].

The goal is not only to derive the most promising representations and re-ranking methods for different time series retrieval scenarios but also to identify the combinations that would lead to effective results.

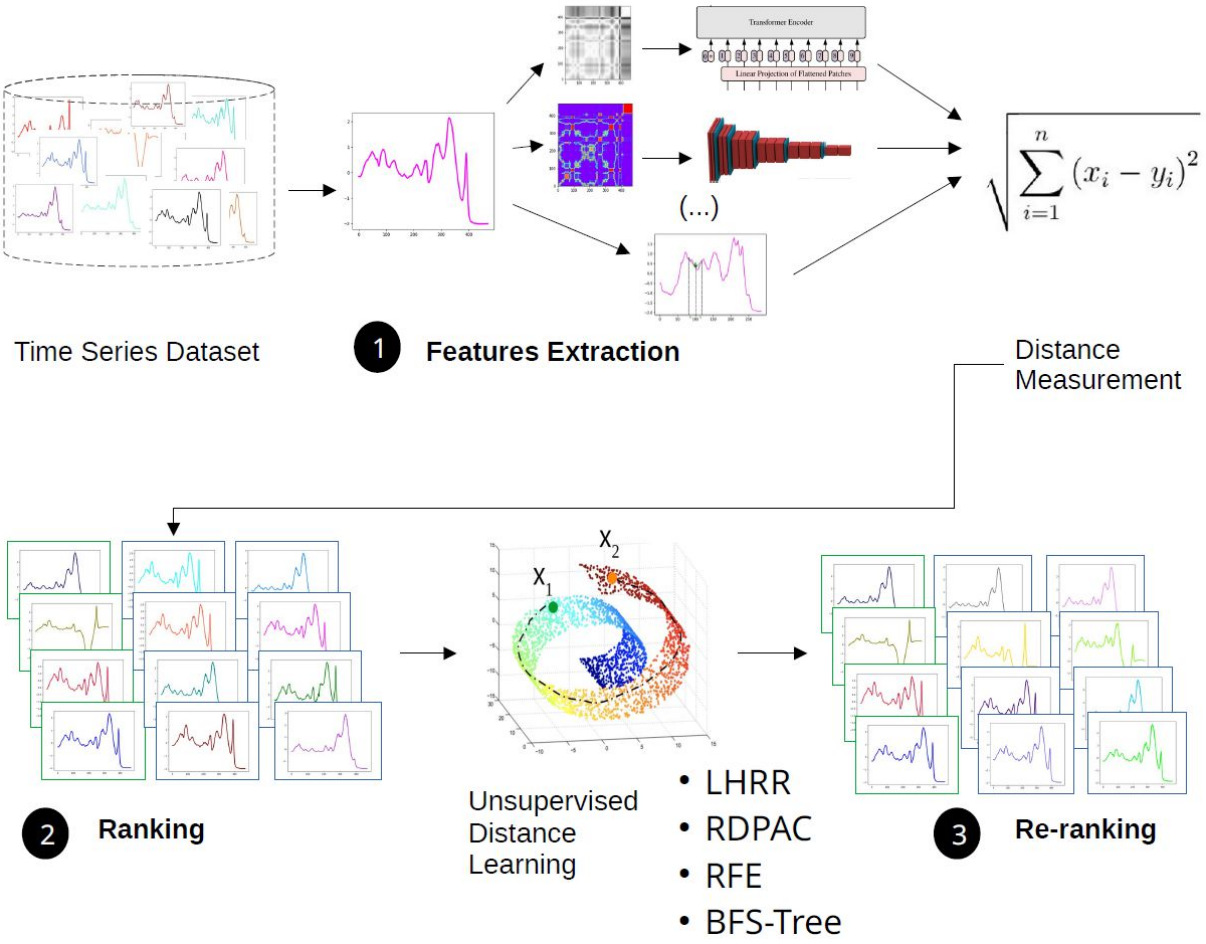


Figure 4.1 – Pipeline employed in the comparative study.

4.2 Experimental Evaluation

This section introduces the adopted experimental protocol and the corresponding outcomes achieved through the proposed comparative study for time series retrieval, considering representation and re-ranking methods. Subsection 4.2.1 outlines the chosen parameterization and the experimental protocol, while Subsection 4.2.2 establishes the

baseline results. The chosen datasets for our experiments are CBF, GunPoint, Beef, Rock, Electric Devices, and Yoga, as detailed in Section 2.7.

4.2.1 Implementation Details and Parameters Definition

This section outlines the parameters, implementation details, and experimental protocol for the time series ranking model. The experiment is composed by three stages: feature extraction, data ranking, and re-ranking. These stages were implemented using Python programming language. Diverse algorithms in this work consider the pyts library [59] implementation, described in Section 2.9. An extensive parameter evaluation was not performed in our experiments.

1. **Feature Extraction**

   All feature extractors from Section 2.5.1 were employed. For time series imaging methods, described in Section 2.5.2, the pyts [59] implementation with default parameters was used, and images were created and saved using the matplotlib library [79]. The colormap parameters were set to *'rainbow'* for GAF and MTF methods and *'binary'* for the RP method. CNN ResNet-152 and Vision Transformers neural networks, presented in Section 2.5.3, were applied to the generated images based on previous research results [204, 19]. For the BAS method, angles were in degrees, and the value of k_b was determined as $k_b = z \times n \times \gamma$, where z is the mean standard deviation of the dataset, calculated from individual time series standard deviations s_i. Here, $\gamma = 0.09$ was uniformly used across most datasets, except Rock, where $k_b = 30$ was chosen empirically because the previous approach did not produce a satisfactory result. The application of the BAS algorithm was also demonstrated in [7]. For the DFT model, the pyts implementation [59] was utilized with parameters: $n\_coefs = 0.4$, $norm\_mean = True$, $norm\_std = True$. PyWavelet [102] library was used for the DWT model with parameters: $wavelet = `haar'$ and $mode = `sym'$, and results were separated into Approximation and Detail. The study by Chan et al. [27] inspired applying DFT and DWT methods. The ROCKET algorithm, a recent method for time series classification, was employed for feature extraction, using the pyts implementation [59] with default parameters.

2. **Ranking**

   After obtaining feature vectors for the time series, we calculate distance matrices using the Euclidean Distance. Each matrix row refers to a dataset element, sorted to form ranked lists, resulting in the set $\mathcal{R}$ as defined in Section 2.6. For each dataset, all the time series were used as queries.

3. **Re-ranking with Unsupervised Distance Learning Algorithms**

The unsupervised distance learning algorithms (discussed in Section 2.6) were applied for post-processing ranking results. We aim to enhance distance measures by considering the dataset structure and improving retrieval outcomes. We used the Unsupervised Distance Learning Framework (UDLF) [194] implementation for these methods. We evaluated retrieval outcomes using Mean Average Precision (mAP), Precision@k, and Recall@k metrics, defined in Section 2.8 before and after re-ranking. Our parameter settings mostly followed defaults, with $K = 15$ for neighborhood analysis. This parameter highly depends on the dataset's number of classes and elements in a class. $K = 15$ covers most of the cases in the utilized datasets. For the ranked list size, denoted by L, we used $L = 1,000$ or the dataset size ($L = n$) when $n < 1,000$. We set $L_{MULT} = 1$ for the RDPAC method to ensure consistent ranked list lengths across re-ranking methods. The effectiveness measures considered the value of k, which ranges from 1 to L in P@k and R@k metrics.

4.2.2 Baselines

We established comparative results to assess the impact of time series representation approaches and re-ranking methods. First, a ranked list is computed based on the time series raw values and the Euclidean distance — without employing any representation approach or re-ranking method. This ranking is referred to as the *initial* ranking. Subsequently, we construct *baselines* for each dataset, starting with the initial ranking and then applying different re-ranking methods.

4.3 Results

This section presents and discusses the results of the proposed methodology. Section 4.3.1 presents a quantitative analysis of the mAP, P@5, P@10, R@5, and R@10 results, while Section 4.3.2 discusses qualitative results through the analysis of produced rankings. Finally, Section 4.3.3 analyzes cases of success and failure.

4.3.1 Retrieval Results – Quantitative Analysis

Tables 4.1, 4.2, 4.3, 4.4, and 4.5 presents the results of mAP, Precision@5 (P@5), Precision@10 (P@10), Recall@5 (R@5), and Recall@10 (R@10), respectively, considering different combinations of time series representation and re-ranking methods on all datasets. Each column refers to a feature descriptor, and the comparisons are made by line. The best results for each feature extractor are highlighted in bold. For each dataset, a line relates to a re-ranking method, and the best results for each dataset are highlighted in red. We performed the Wilcoxon statistical test [209] comparing the best results, highlighted in red, with the others for the same dataset. The symbol '*' is used for results without significant

Table 4.1 – mAP values considering different combinations of time series representations (column) and re-ranking methods (line) on all datasets. Results highlighted in bold refer to the best results for each time series representation across the different re-ranking methods. The result highlighted in red refers to the overall best result for the dataset. The symbol'*' is used for results without significant statistical differences. "No re-rank" results refer to ranking with the Euclidean Distance, while "Baseline" results refer to experiments executed with re-ranking methods applied to the *initial* results. Both are highlighted in different tones of gray. *Initial* results are highlighted in blue and italics

Specification		Descriptor												
Dataset	**UDL**	**Baseline**	**BAS**	**DFT**	**DWT Approx.**	**DWT Detail**	**ROCKET**	**GAF ResNet**	**MTF ResNet**	**RP ResNet**	**GAF ViT**	**MTF ViT**	**RP ViT**	**Mean**
Beef	No re-rank	*47.08**	**46.16**	46.93*	47.03*	47.26*	**34.79**	40.83	**37.53**	**40.86**	42.75	40.02	45.60*	*43.07*
	LHRR	46.48	44.49	46.50	46.38	48.25	33.86	38.83	36.69	39.94	42.08	43.44	49.40	*43.03*
	RDPAC	**47.49**	44.26	**47.64**	**47.57**	49.58*	33.23	39.16	36.50	39.16	42.68	40.51	**50.18***	*43.16*
	BFSTREE	46.83	44.50	46.78	46.86	49.05	33.78	**41.79**	36.56	39.93	42.81	43.92	48.69	***43.46***
	RFE	45.33	44.80	46.12	45.38	**50.99**	32.77	39.82	36.97	40.03	**44.16**	**44.52**	50.07*	*43.41*
CBF	No re-rank	*64.47*	56.72	66.04	65.93	42.43	87.19	60.29	55.44	70.50	50.20	48.10	63.64	*60.91*
	LHRR	70.19	66.13	70.88	71.11	41.94	89.92	67.15	61.56	77.84	57.38	54.93	73.56	*66.88*
	RDPAC	**81.45**	**68.23**	**84.04**	**84.92**	38.09	**95.43**	**73.48**	**66.69**	**84.22**	**62.92**	**59.98**	**78.74**	***73.18***
	BFSTREE	73.30	66.28	75.36	76.33	**46.59**	90.71	68.98	63.06	78.72	58.14	56.03	75.17	*69.06*
	RFE	68.92	64.10	69.63	69.71	43.60	87.94	66.17	60.23	75.94	55.80	54.35	71.43	*65.65*
GunPoint	No re-rank	*62.41*	63.98	62.44	62.43	64.44	60.85	76.89	67.89	78.99	71.96	62.79	71.77	*67.24*
	LHRR	63.86	65.98	63.86	63.88	67.29	61.71	77.25	68.26	81.93	77.35	63.37	75.18	*69.16*
	RDPAC	**64.70**	**67.72**	64.61	64.58	66.73	61.90	79.17	**70.85**	**86.28**	78.66	63.57	77.25	***70.50***
	BFSTREE	63.97	66.73	64.14	64.05	**69.84**	**62.22**	**79.70**	69.33	80.84	**80.09**	**64.12**	**77.70**	*70.23*
	RFE	64.46	65.73	**64.73**	**64.73**	67.87	61.97	75.40	67.66	82.16	75.68	61.36	76.06	*68.98*
Rock	No re-rank	***58.71***	69.09	61.31	**58.71**	62.93	**58.06**	59.24	55.00	62.77	58.12	54.36	65.83	*60.34*
	LHRR	54.90	**81.11***	**66.76**	54.90	**82.93**	53.34	64.40	61.37	62.94	62.94	59.23	**65.99**	*64.23*
	RDPAC	56.26	71.15	64.72	56.26	79.01*	53.05	63.21	60.46	63.33	**64.54**	**61.53**	65.03	*63.21*
	BFSTREE	57.06	80.27*	65.41	57.06	79.59	54.46	**65.92**	**61.82**	**63.68**	62.35	58.98	65.27	***64.32***
	RFE	56.29	78.03	65.81	56.29	67.84	54.11	65.57	58.09	62.39	63.55	54.90	64.65	*62.29*
EDev	No re-rank	*20.08*	24.66	24.94	23.54	14.80	37.56	41.17	34.97	40.24	40.05	37.01	35.97	*31.25*
	LHRR	24.75	21.30	23.95	23.24	14.52	35.24	39.13	34.30	39.00	38.16	34.50	36.69	*30.40*
	RDPAC	21.06	25.50	26.01	25.86	15.12	**37.78**	**41.96**	**35.86**	40.86	**41.00**	**37.48**	37.01	*32.12*
	BFSTREE	21.12	25.20	25.64	25.79	15.10	37.52	41.56	35.48	40.55	40.61	37.13	36.72	*31.87*
	RFE	**24.79**	**26.05**	**28.05**	**27.27**	**17.17**	37.60	41.63	35.82	**40.89**	40.77	36.81	**37.51**	***32.86***
Yoga	No re-rank	*30.30*	31.58	30.30	30.30	30.23	29.89*	30.50	28.77	30.60	30.17	27.93	29.83	*30.03*
	LHRR	30.41	31.24	30.41	30.41	29.82	30.18	30.81	28.86	30.52	30.77	28.57	30.00	*30.17*
	RDPAC	**30.76**	31.79*	**30.75**	**30.76**	30.32	**30.96**	**31.45**	**29.50**	**31.44**	**31.03***	**28.79**	**30.54**	***30.67***
	BFSTREE	30.59*	31.71	30.58*	30.58*	30.38	30.56	31.33*	29.34	31.32*	30.95*	28.63	30.38	*30.53*
	RFE	30.47	**31.86**	30.46	30.47	**30.41**	30.19	31.18*	29.19	31.15*	30.73	28.62	30.35	*30.42*
	Mean	*48.62*	*51.21*	*50.83*	*49.41*	*46.47*	*51.29*	*53.46*	*48.47*	***55.63***	*51.61*	*47.18*	*54.87*	

statistical differences. The results related to the *initial* ranking, i.e., results computed based on the raw time series values and ranking with Euclidean Distance, are highlighted in italics and blue for comparison. There are reported mean values (last column and last line) for the descriptors and re-ranking approaches.

- **mAP Results**

For the Beef dataset, the most favorable outcome was observed for the combination of DWT-Detail and RFE (highlighted in red). For this dataset, the highest re-ranking gains, 11.24%, were observed for the combination of MTF-ViT and RFE when compared with the MTF-ViT results with no re-ranking method. Searches on the CBF dataset predominantly benefited from the combination of the RDPAC re-ranking method with the ROCKET feature extractor and achieved the best results when considering the combination of DWT-Approximation and RDPAC (28.80% gain over the no re-rankings results). In the case of the GunPoint dataset, image-based methods proved superior. RP-ResNet coupled

with RDPAC led to the best results, while GAF-ViT in conjunction with BFSTREE yielded the most considerable improvement, an 11.30% gain. The combination of DWT-Detail and LHRR in the Rock dataset consistently delivered the best results, with the highest observed gains over all datasets (31.78%). The most effective combination for the Electric Devices dataset was GAF-ResNet with RDPAC, while the combination of the baseline and RFE delivered a gain of 23.46%. This is particularly interesting as it is the only worth-noting result involving using the baseline set of features. The gains in the Yoga dataset were modest, at 3.58%, achieved with the ROCKET and RDPAC combination. Here, BAS and RFE yielded the best results.

Notably, the RDPAC method consistently produced reliable results across the experiments on different datasets, while RFE yielded the best results for two datasets (Beef and Yoga). While there was no consensus regarding the ideal feature extractor, some trends emerged. In conjunction with ResNet-152, RP yielded the highest average results, and DWT-Detail excelled in two datasets (Beef and Rock). The highest gains were observed in the Rock dataset.

Table 4.2 – P@5 values.

Specification		Descriptor												
Dataset	**UDL**	**Baseline**	**BAS**	**DFT**	**DWT Approx.**	**DWT Detail**	**ROCKET**	**GAF ResNet**	**MTF ResNet**	**RP ResNet**	**GAF ViT**	**MTF ViT**	**RP ViT**	**Mean**
Beef	No re-rank	*56.67**	**54.67***	57.00*	56.67*	**60.00***	**40.33**	**50.00**	41.00	48.67	51.67	44.33	57.00*	*51.50*
	LHRR	53.00	51.33	53.00	52.67*	59.33	39.00	48.67	39.67	47.67	51.33	52.67	60.00*	*50.70*
	RDPAC	**57.00***	52.33	**57.33***	**57.00***	59.67*	35.33	48.67	41.33	**51.00**	51.67	50.33	**61.00**	***51.89***
	BFSTREE	55.33*	49.67	55.00*	54.67*	57.33*	36.00	49.00	39.67	47.33	50.67	51.33	58.00*	*50.33*
	RFE	52.67	50.67	52.33	53.33	59.00*	34.67	**50.00**	**41.67**	44.67	**52.33**	**54.00**	57.67*	*50.25*
CBF	No re-rank	*98.39*	89.16	99.89*	99.81*	48.69	99.85*	91.76	88.69	96.26	84.19	81.55	93.38	*89.30*
	LHRR	**100.0**	**96.97**	**100.0**	99.91*	65.12	99.83*	91.38	**90.19**	96.06	84.60	82.09	94.80	***91.75***
	RDPAC	99.94*	96.58	**100.0**	**99.94***	60.11	**99.94***	**92.52**	90.17	**96.37**	**84.80**	**82.28**	**95.03**	*91.47*
	BFSTREE	**100.0**	96.04	**100.0**	99.91*	**65.48**	99.91*	91.78	89.59	95.57	84.37	81.66	94.32	*91.55*
	RFE	**100.0**	96.32	**100.0**	99.91*	64.26	99.91*	91.46	89.78	95.53	83.48	81.38	94.73	*91.40*
GunPoint	No re-rank	*93.1*	**90.7**	93.2	93.2	91.0	94.5	98.7*	**88.7**	99.7*	98.0*	81.4	99.2*	*93.45*
	LHRR	**93.4**	90.0	**94.1**	**93.8**	91.5	93.3	**99.5***	87.0	**99.8**	98.8*	**82.7**	99.2*	***93.59***
	RDPAC	92.4	89.1	92.7	92.8	**92.6**	**94.7**	99.2*	87.5	**99.8**	99.2*	81.2	**99.7***	*93.41*
	BFSTREE	92.2	90.2	93.0	92.8	**92.6**	93.8	99.3*	86.5	99.4*	**99.3***	81.3	99.4*	*93.32*
	RFE	92.7	90.3	92.7	93.1	91.7	94.5	97.9*	87.7	99.5*	98.6*	82.5	98.6*	*93.32*
Rock	No re-rank	***78.00***	81.43*	80.86*	**78.00**	80.57	**74.00**	75.43	66.00	**76.57**	79.71	69.43	**79.43**	*76.62*
	LHRR	74.29	86.29*	82.00*	74.29	**87.43**	65.71	80.00*	**70.29**	74.86	**82.57***	66.57	78.29	*76.88*
	RDPAC	74.86	82.86*	82.86*	74.86	86.00*	69.43	77.71	69.43	73.43	79.71	**73.14**	76.00	*76.69*
	BFSTREE	75.71	86.00*	81.43*	75.71	84.57*	69.43	79.14*	70.00	75.14	82.29*	68.00	78.00	***77.12***
	RFE	75.43	**86.86***	**83.14***	75.43	83.43	69.43	**81.43***	69.14	74.00	80.00*	66.29	77.71	*76.86*
EDev	No re-rank	***74.48***	70.64	72.30	73.14	61.91	**81.36**	84.94*	78.80	**84.55**	83.40	78.26	83.70	*77.29*
	LHRR	74.27	71.85	73.31	73.35	**66.53**	80.95	**85.30**	78.82	84.54	**83.69**	**78.38**	**83.90**	***77.91***
	RDPAC	74.07	71.09	72.41	72.67	64.73	80.74	85.04	78.86	84.41	83.60	78.03	83.80	*77.45*
	BFSTREE	74.12	71.71	73.45	**73.73**	65.65	80.86	**85.30**	**78.98**	84.51	83.51	78.43	83.86	*77.84*
	RFE	74.28	**71.78**	**73.46**	73.72	65.30	80.79	85.19*	78.63	84.43	83.47	78.37	83.83	*77.77*
Yoga	No re-rank	***92.69***	**90.79**	**92.69**	**92.70**	88.02	**94.19**	**88.92**	82.31	90.23	89.03	82.77	90.47	*89.57*
	LHRR	92.42	90.65	92.47	92.41	91.14	93.98*	88.70	83.06	90.36	88.68	83.68	90.36	*89.83*
	RDPAC	92.48	90.76	92.45	92.38	90.59	94.05*	88.90	83.14	**90.55**	**89.24**	**83.73**	90.41	*89.89*
	BFSTREE	92.55	90.70	92.65	92.62	**91.42**	94.03*	88.44	**83.35**	90.48	88.73	83.30	**90.68**	***89.91***
	RFE	92.64	**90.79**	92.67	92.55	90.69	93.87*	88.60	82.80	90.39	88.87	83.16	90.19	*89.77*
	Mean	*81.64*	*80.61*	*82.61*	*81.57*	*75.21*	*79.28*	*82.10*	*74.76*	*82.19*	*81.32*	*74.07*	***84.09***	

- **Precision and Recall Results**

It is possible to observe that the results between P@5, P@10, R@5, and R@10 are similar, i.e., the best-performing scores are observed for the same combinations of representations and re-ranking methods. The results, however, are different from those presented in Table 4.1, which suggests that the definition of the metric for determining the best-performing combination is essential. For the Beef dataset, the best results were

Table 4.3 – P@10 values.

Specification		Descriptor												
Dataset	**UDL**	**Baseline**	**BAS**	**DFT**	**DWT Approx.**	**DWT Detail**	**ROCKET**	**GAF ResNet**	**MTF ResNet**	**RP ResNet**	**GAF ViT**	**MTF ViT**	**RP ViT**	**Mean**
Beef	No re-rank	*43.17**	**41.33**	43.17*	43.17*	42.50	27.50	34.83	31.17	33.50	36.83	33.33	42.83*	*37.78*
	LHRR	42.17	40.17	42.00	42.17	47.33*	**28.17**	35.33	29.67	34.33	38.50	39.00	48.17*	*38.92*
	RDPAC	**44.00***	40.33	**44.33***	**44.33***	**48.50***	26.50	34.67	**31.83**	33.83	38.33	33.50	**49.33**	***39.12***
	BFSTREE	43.50*	38.83	43.33*	43.50*	47.83*	27.00	**37.83**	29.50	**34.67**	37.67	38.50	46.67*	*39.07*
	RFE	41.83	39.50	41.67	41.33	48.33*	25.67	34.50	28.17	33.17	**38.83**	**39.33**	47.17*	*38.29*
CBF	No re-rank	*97.44*	85.45	99.54*	99.48*	42.46	99.33	88.04	84.62	94.18	78.84	75.89	89.87	*86.26*
	LHRR	99.95*	95.66	99.95*	99.81*	58.83	99.59*	89.26	86.61	94.67	79.92	76.51	**93.08**	***89.49***
	RDPAC	99.86*	**95.96**	**100.00**	**99.86***	51.91	**99.74***	**90.00**	**87.02**	**95.53**	**80.35**	**77.34**	92.96	*89.21*
	BFSTREE	99.99*	95.40	**100.00**	99.82*	**59.83**	**99.74***	89.04	86.20	94.17	79.56	76.12	92.20	*89.34*
	RFE	**100.00**	95.11	99.88*	99.73*	58.03	**99.74***	88.49	85.69	94.05	78.71	75.67	92.98	*89.01*
GunPoint	No re-rank	*87.20*	85.50	87.30	87.45	86.85	88.85	97.40	83.80	98.85*	96.35	75.10	98.25*	*89.41*
	LHRR	88.80	84.85	88.80	89.05	89.55	90.05	**98.85***	84.00	99.55*	98.45*	76.70	98.40*	*90.59*
	RDPAC	**88.85**	85.25	88.20	88.05	89.55	**91.65**	98.70*	83.95	**99.65**	**99.30***	**76.75**	**99.55***	*90.79*
	BFSTREE	**88.85**	**85.70**	89.25	89.05	**91.50**	90.75	**98.85***	**84.05**	99.50*	98.85*	75.90	98.50*	***90.90***
	RFE	88.50	84.30	**88.95**	**89.15**	89.50	90.65	95.25	83.55	98.90*	98.45*	74.65	96.90	*89.90*
Rock	No re-rank	***60.29***	69.43	68.57	**60.29**	65.14	**58.00**	64.00	54.57	**65.29**	63.43	54.86	68.43	*62.69*
	LHRR	57.29	**82.00***	**77.43***	57.29	**83.29**	56.14	70.29	57.86	65.14	**68.86**	59.57	**71.71**	*67.24*
	RDPAC	56.86	75.14	74.57*	56.86	80.14	55.00	67.86	58.86	64.86	68.43	**63.29**	69.14	*65.92*
	BFSTREE	59.14	80.86*	75.29*	59.14	79.57*	55.57	**70.86**	**60.43**	65.14	68.43	59.86	70.43	***67.06***
	RFE	58.00	80.14*	74.86*	58.00	68.71	55.71	**70.86**	54.57	64.00	69.43	56.14	70.00	*65.04*
EDev	No re-rank	*68.57*	64.08	65.83	67.17	53.98	**77.29**	81.47	**75.11**	80.58	79.52	**74.52**	79.82	*72.33*
	LHRR	68.83	65.79	67.91	68.05	**60.27**	76.82	81.79*	74.89	81.02	**80.08**	74.22	80.28	***73.33***
	RDPAC	68.55	65.11	66.75	67.45	57.10	76.51	81.74*	74.87	80.87	80.03	73.99	80.25	*72.77*
	BFSTREE	**69.03**	**65.86**	67.89	**68.64**	58.05	76.91	**81.87**	74.87	**81.09**	79.96	74.23	**80.33**	*73.23*
	RFE	68.73	65.53	**67.96**	68.31	58.80	76.67	81.62*	74.46	80.67	79.66	74.14	80.03	*73.05*
Yoga	No re-rank	*88.99*	86.63	89.00	88.99	83.40	90.94*	84.38	76.73	85.73	84.57	77.20	85.91	*85.21*
	LHRR	89.28	86.92	89.28	89.26	87.68	91.17*	84.67	78.27	86.95	85.05	78.88	86.98	*86.2*
	RDPAC	89.21	87.02	89.26	89.16	86.96	91.25*	**85.15**	78.30	**87.26**	85.34	**79.35**	87.19	*86.29*
	BFSTREE	89.48	**87.63**	89.52	89.48	**87.87**	**91.28**	84.82	**78.57**	86.96	**85.36**	78.89	**87.31**	***86.43***
	RFE	**89.49**	87.35	**89.56**	**89.49**	87.11	90.92	84.59	77.59	86.61	84.88	78.35	86.64	*86.05*
	Mean	*74.53*	*74.76*	*77.00*	*74.45*	*68.35*	*73.50*	*76.23*	*68.33*	*76.69*	*74.73*	*66.73*	***79.04***	

Table 4.4 – R@5 values.

Specification		Descriptor												
Dataset	**UDL**	**Baseline**	**BAS**	**DFT**	**DWT Approx.**	**DWT Detail**	**ROCKET**	**GAF ResNet**	**MTF ResNet**	**RP ResNet**	**GAF ViT**	**MTF ViT**	**RP ViT**	**Mean**
Beef	No re-rank	*23.61**	**22.78***	23.75*	23.61*	**25.00***	**16.81**	**20.83**	17.08	20.28	21.53	18.47	23.75*	*21.46*
	LHRR	22.08	21.39	22.08	21.94	24.72*	16.25	20.28	16.53	19.86	21.39	21.94*	25.00*	*21.12*
	RDPAC	**23.75***	21.81	**23.89***	**23.75***	24.86*	14.72	20.28	17.22	**21.25**	21.53	20.97	**25.42**	***21.62***
	BFSTREE	23.06*	20.69	22.92*	22.78*	23.89*	15.00	20.42	16.53	19.72	21.11	21.39	24.17*	*20.97*
	RFE	21.94	21.11	21.81*	22.22*	24.58*	14.44	**20.83**	**17.36**	18.61	**21.81***	**22.50***	24.03*	*20.94*
CBF	No re-rank	*1.59*	1.44	**1.61**	**1.61**	0.79	**1.61**	1.48	1.43	**1.55**	1.36	1.32	1.51	*1.44*
	LHRR	**1.61**	**1.56**	**1.61**	**1.61**	1.05	**1.61**	1.47	**1.45**	**1.55**	1.36	1.32	**1.53**	*1.48*
	RDPAC	**1.61**	**1.56**	**1.61**	**1.61**	0.97	**1.61**	**1.49**	**1.45**	**1.55**	**1.37**	**1.33**	**1.53**	*1.47*
	BFSTREE	**1.61**	1.55	**1.61**	**1.61**	**1.06**	**1.61**	1.48	**1.45**	1.54	1.36	1.32	1.52	***1.48***
	RFE	**1.61**	1.55	**1.61**	**1.61**	1.04	**1.61**	1.48	**1.45**	1.54	1.35	1.31	**1.53**	*1.47*
GunPoint	No re-rank	*4.65*	**4.53**	4.66	4.66	4.55	**4.73**	4.94*	**4.43**	**4.99**	4.90*	4.07	4.96*	*4.67*
	LHRR	**4.67**	4.50	**4.70**	**4.69**	4.57	4.66	**4.98***	4.35	**4.99**	4.94*	**4.13**	4.96*	***4.68***
	RDPAC	4.62	4.45	4.63	4.64	**4.63**	**4.73**	4.96*	4.37	**4.99**	4.96*	4.06	**4.99**	*4.67*
	BFSTREE	4.61	4.51	4.65	4.64	**4.63**	4.69	4.97*	4.32	4.97*	**4.97***	4.06	4.97*	*4.67*
	RFE	4.64	4.51	4.64	4.65	4.58	4.72	4.90*	4.38	4.98*	4.93*	4.12	4.93*	*4.66*
Rock	No re-rank	***24.70***	25.35	25.36	**24.70**	25.18	**21.81**	23.66	22.01	**24.47**	23.76	23.10	23.81	*23.99*
	LHRR	23.36	28.06*	**26.01***	23.36	**28.39**	18.65	24.73*	**23.55**	23.33	**24.93***	21.76	23.81	*24.16*
	RDPAC	23.97	27.50*	25.81*	23.97	28.11*	20.27	23.37	23.25	23.18	24.42	**23.34**	**24.20**	***24.28***
	BFSTREE	24.09	28.06*	25.25*	24.09	27.28*	20.24	23.70*	23.30	23.46	24.39*	22.84	**24.20**	*24.24*
	RFE	23.97	**28.12***	25.61*	23.97	27.48*	20.46	**25.14***	22.23	23.24	23.83	21.08	23.34	*24.04*
EDev	No re-rank	***0.17***	0.16	**0.17**	**0.17**	0.14	**0.20**	**0.21**	**0.19**	**0.21**	**0.20**	**0.19**	**0.21**	***0.19***
	LHRR	**0.17**	**0.17**	**0.17**	**0.17**	**0.15**	0.19	**0.21**	**0.19**	**0.21**	**0.20**	**0.19**	**0.21**	***0.19***
	RDPAC	**0.17**	**0.17**	**0.17**	**0.17**	**0.15**	0.19	**0.21**	**0.19**	**0.21**	**0.20**	**0.19**	**0.21**	***0.19***
	BFSTREE	**0.17**	**0.17**	**0.17**	**0.17**	**0.15**	0.19	**0.21**	**0.19**	**0.21**	**0.20**	**0.19**	**0.21**	***0.19***
	RFE	**0.17**	**0.17**	**0.17**	**0.17**	**0.15**	0.19	**0.21**	**0.19**	**0.21**	**0.20**	**0.19**	**0.21**	***0.19***
Yoga	No re-rank	***0.28***	**0.28**	**0.28**	**0.28**	0.27	**0.29**	**0.27**	**0.25**	0.27	**0.27**	**0.25**	0.27	***0.27***
	LHRR	**0.28**	**0.28**	**0.28**	**0.28**	**0.28**	**0.29**	**0.27**	**0.25**	0.27	**0.27**	**0.25**	0.27	***0.27***
	RDPAC	**0.28**	**0.28**	**0.28**	**0.28**	**0.28**	**0.29**	**0.27**	**0.25**	**0.28**	**0.27**	**0.25**	**0.28**	***0.27***
	BFSTREE	**0.28**	**0.28**	**0.28**	**0.28**	**0.28**	**0.29**	**0.27**	**0.25**	**0.28**	**0.27**	**0.25**	**0.28**	***0.27***
	RFE	**0.28**	**0.28**	**0.28**	**0.28**	**0.28**	**0.29**	**0.27**	**0.25**	0.27	**0.27**	**0.25**	0.27	***0.27***
	Mean	*8.93*	*9.24*	*9.20*	*8.93*	***9.65***	*7.09*	*8.59*	*7.68*	*8.41*	*8.75*	*8.22*	*9.22*	

observed for RP-ViT combined with RDPAC, while the highest gains were observed from the combination with MTF-ViT and RFE. For the CBF dataset, the best results were observed for the Baseline and DFT, considering various re-ranking methods and, for R@5, DWT-App and ROCKET are also highlights. The highest gains were observed in

Table 4.5 – R@10 values.

Specification		Descriptor												
Dataset	**UDL**	**Baseline**	**BAS**	**DFT**	**DWT Approx.**	**DWT Detail**	**ROCKET**	**GAF ResNet**	**MTF ResNet**	**RP ResNet**	**GAF ViT**	**MTF ViT**	**RP ViT**	**Mean**
Beef	No re-rank	*35.97**	**34.44**	35.97*	35.97*	35.42	22.92	29.03	25.97	27.92	30.69	27.78	35.69*	***31.48***
	LHRR	35.14	33.47	35.00	35.14	39.44*	**23.47**	29.44	24.72	28.61	32.08	32.50	40.14*	*32.43*
	RDPAC	**36.67***	33.61	**36.94***	**36.94***	**40.42***	22.08	28.89	**26.53**	28.19	31.94	27.92	**41.11**	***32.60***
	BFSTREE	36.25*	32.36	36.11*	36.25*	39.86*	22.50	**31.53**	24.58	**28.89**	31.39	32.08	38.89*	*32.56*
	RFE	34.86	32.92	34.72	34.44	40.28*	21.39	28.75	23.47	27.64	**32.36**	**32.78**	39.31*	*31.91*
CBF	No re-rank	*3.14*	2.76	3.21*	3.21*	1.37	3.20	2.84	2.73	3.04	2.54	2.45	2.90	*2.78*
	LHRR	3.22*	3.09	3.22*	**3.22***	1.90	3.21*	2.88	2.79	3.05	2.58	2.47	**3.00**	***2.89***
	RDPAC	3.22*	**3.10***	**3.23**	**3.22***	1.67	**3.22***	**2.90**	**2.81**	**3.08**	**2.59**	**2.49**	**3.00**	*2.88*
	BFSTREE	**3.23**	3.08	**3.23**	**3.22***	**1.93**	**3.22***	2.87	2.78	3.04	2.57	2.46	2.97	*2.88*
	RFE	**3.23**	3.07	3.22*	**3.22***	1.87	**3.22***	2.85	2.76	3.03	2.54	2.44	**3.00**	*2.87*
GunPoint	No re-rank	*8.72*	8.55	8.73	8.74	8.68	8.88	9.74	8.38	9.89*	9.64	7.51	9.83*	*8.94*
	LHRR	8.88	8.48	8.88	8.90	8.95	9.00	**9.89***	**8.40**	9.96*	9.85*	7.67	9.84*	*9.06*
	RDPAC	8.88	8.52	8.82	8.80	8.96	**9.16**	9.87*	8.39	**9.97**	**9.93***	**7.68**	**9.96***	*9.08*
	BFSTREE	**8.89**	**8.57**	**8.92**	8.90	**9.15**	9.07	**9.89***	**8.40**	9.95*	9.89*	7.59	9.85*	***9.09***
	RFE	8.85	8.43	8.89	**8.91**	8.95	9.06	9.53	8.35	9.89*	9.85*	7.46	9.69	*8.99*
Rock	No re-rank	***35.03***	38.88	42.27	**35.03**	36.28	31.21	38.98	35.75	**40.18**	36.57	35.98	40.01	*37.18*
	LHRR	33.46	**52.52***	**47.17***	33.46	**53.94**	31.22	**41.82**	38.68	39.95	**38.18**	37.31	**43.19**	***40.91***
	RDPAC	33.67	50.23	44.52*	33.67	52.02*	30.56	39.12	38.94	40.01	37.82	**39.35**	42.21	*40.18*
	BFSTREE	34.73	52.08*	45.23*	34.73	51.96	30.87	41.19	**40.21**	39.64	37.73	38.00	42.69	*40.76*
	RFE	33.99	51.00*	44.03*	33.99	46.21	**31.32**	**41.82**	35.35	38.51	37.78	35.49	42.72	*39.35*
EDev	No re-rank	*0.30*	0.28	0.29	0.30	0.23	**0.36**	**0.39**	**0.36**	**0.39**	**0.38**	**0.35**	**0.39**	***0.34***
	LHRR	**0.31**	**0.30**	**0.31**	0.30	**0.26**	0.35	**0.39**	0.35	**0.39**	**0.38**	**0.35**	**0.39**	***0.34***
	RDPAC	**0.31**	0.29	0.30	0.30	0.25	0.35	**0.39**	0.35	**0.39**	**0.38**	0.34	**0.39**	***0.34***
	BFSTREE	**0.31**	0.29	**0.31**	**0.31**	0.25	**0.36**	**0.39**	0.35	**0.39**	**0.38**	**0.35**	**0.39**	***0.34***
	RFE	0.30	0.29	**0.31**	**0.31**	**0.26**	0.35	**0.39**	0.35	**0.39**	**0.38**	**0.35**	**0.39**	***0.34***
Yoga	No re-rank	***0.54***	0.52	**0.54**	**0.54**	0.50	0.55*	0.51	0.46	0.52	0.51	0.47	0.52	*0.52*
	LHRR	**0.54**	**0.53**	**0.54**	**0.54**	**0.53**	0.55*	**0.52**	**0.48**	**0.53**	**0.52**	**0.48**	**0.53**	*0.52*
	RDPAC	**0.54**	**0.53**	**0.54**	**0.54**	**0.53**	**0.56**	**0.52**	**0.48**	**0.53**	**0.52**	**0.48**	**0.53**	***0.53***
	BFSTREE	**0.54**	**0.53**	**0.54**	**0.54**	**0.53**	**0.56**	**0.52**	**0.48**	**0.53**	**0.52**	**0.48**	**0.53**	***0.53***
	RFE	**0.54**	**0.53**	**0.54**	**0.54**	**0.53**	0.55*	0.51	0.47	**0.53**	**0.52**	**0.48**	**0.53**	*0.52*
	Mean	*13.81*	*15.77*	*15.55*	*13.81*	***16.44***	*11.11*	*13.95*	*12.47*	*13.63*	*13.77*	*13.12*	*15.82*	

DWT-Detail with BFSTREE. The best results were RP-ResNet152 for the GunPoint dataset, considering diverse re-ranking methods. In Table 4.4, we can also observe a highlight in RP-ViT associated with RDPAC. The highest gains were observed in DWT-detail with BFSTREE. Highest gains with RDPAC can also be outlined for P@5 and R@5. In Table 4.1, it is possible to observe that RP-ResNet152 performed well, and the results on the Rock dataset were similar, considering precision, recall, and mAP values. The combination of DWT-Detail and LHRR produced the best results and highest gains. The best overall results were observed on Electric Devices for GAF-ResNet152. For recall results, we can also highlight RP-ResNet152 and RP-ViT. The highest gains were observed on DWT-Detail. The best results for the Yoga dataset were observed for the ROCKET descriptor. The highest gains were observed for DWT-Detail. Considering R@5, RP-ResNet152 and RP-ViT also provide the highest gains.

Frequently, DWT-Detail is associated with the highest gains. For precision results (Tables 4.2 and 4.3), the feature with the highest values, in general, was RP-ViT, while the best re-rank method was LHRR for P@5 and BFSTREE for P@10. For recall (Tables 4.4 and 4.5), DWT-Detail also arouses the feature with the highest values. RDPAC led to the best general re-ranking results, as in mAP. For R@5, also positive results were observed for LHRR and, for R@10, BFSTREE.

4.3.2 Retrieval Results – Qualitative Analysis

Figures 4.2, 4.3, 4.4, and 4.5 show examples of qualitative results of a query for the Rock dataset. The following images represent the time series classes encoded in different colors. The queries are highlighted with green boxes, while red boxes are used for the non-relevant results.

It is possible to observe that some representations led to improved results while others (time series raw values) did not. Also, the performance of the re-ranking algorithms was quite dependent on the representation used.

For example, for the cases in which the ranking produced by the representation contains enough relevant time series at the top-ranked positions (Figures 4.2 and 4.3), the use of the re-ranking methods led to very positive gains, with more relevant time series ranked at the top positions, as the ranking produced by the employed representations (DWT-Detail and BAS) already contained relevant time series. On the other hand, Figures 4.4 and 4.5 show examples in which the ranking produced by the representation does not contain much relevant time series at the top-ranked positions (raw time series values and RP-ViT), the use of the re-ranking methods did not lead to significant improvements.

Figure 4.2 – Visual ranking results on Rock [14] dataset for DWT-Detail representations. The query is highlighted with green boxes, while red boxes are used for the non-relevant results.

Figure 4.3 – Visual ranking results on Rock [14] dataset for BAS representations.

Figure 4.4 – Visual ranking results on Rock [14] dataset when the time series raw values are used.

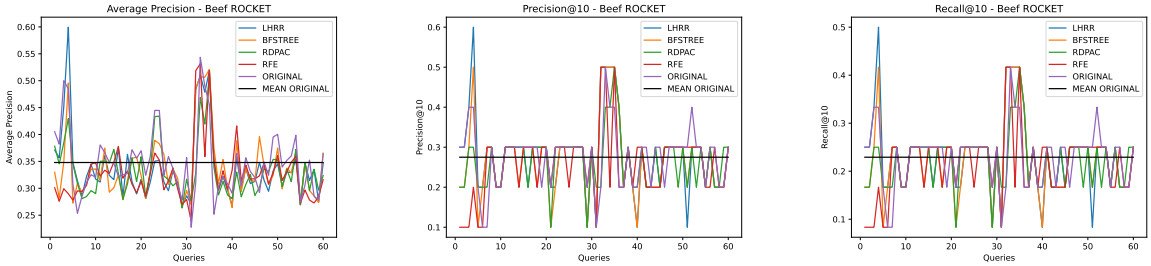
Figure 4.5 – Visual ranking results on Rock [14] dataset for RP-ViT representations.

4.3.3 Retrieval Results – Analysis of Cases of Success and Failure

In Table 4.1, it is possible to observe that some methods exhibit better outcomes than others depending on the dataset. This section sheds light on some successful and unsuccessful cases identified.

A comparison between methods was performed on the Beef and GunPoint datasets, which are similar datasets composed of a small number of time series. Re-ranking methods on the GunPoint dataset consistently delivered positive and noteworthy improvements. On the other hand, the results observed on the Beef dataset were not always positive for re-ranking methods. The analysis considers each dataset query's Precision@10, Recall@10, and Average Precision metrics.

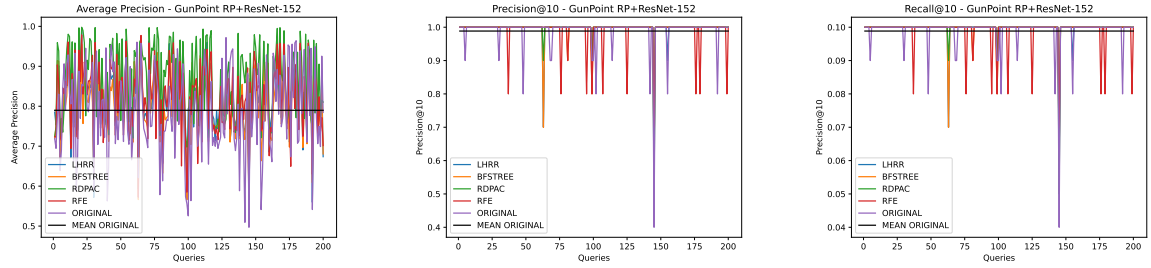
In Figure 4.6, we analyze the performance of the ROCKET feature extractor on the Beef dataset. In Table 4.1, it is possible to observe that there were no gains on mAP after re-ranking. When using this representation method, we observed a decline in mAP across all re-ranking methods. It is worth noting that all metrics maintained an average of around 30%. The initial 10 queries, observed in the x-axis, and queries between 30 and 40 display a better, above-average performance, especially when considering LHRR and RFE methods. Drops are observed throughout the remaining queries, with RDPAC performing noticeably inferior. The below-average performance in most queries, associated with low metric values, may explain why the re-ranking methods did not perform as expected.



(a) Average Precision (b) Precision@10 (c) Recall@10

Figure 4.6 – Measurements for the Beef dataset, considering the ROCKET feature extractor.

In Figure 4.7, we present the metrics obtained from ranking the GunPoint dataset using the RP-ResNet152 combination for feature extraction. Observing Table 4.1, we see that this combination resulted in the best mAP value when considering the re-ranking with RDPAC. With this representation method, we not only observe overall gains in all re-ranking scenarios, in which most queries presented P@10, R@10, and mAP with above-average values, but also note that the highest mAP value was achieved when using RDPAC. The average results for mAP and precision metrics were consistently high, and there were various peaks and drops for the mAP metric. Precision and recall metrics exhibit stability, with very few declines. Consistently, the re-ranking methods delivered superior results compared to the Euclidean Distance ranking. The observed stability may be an important factor for the gains after re-ranking.

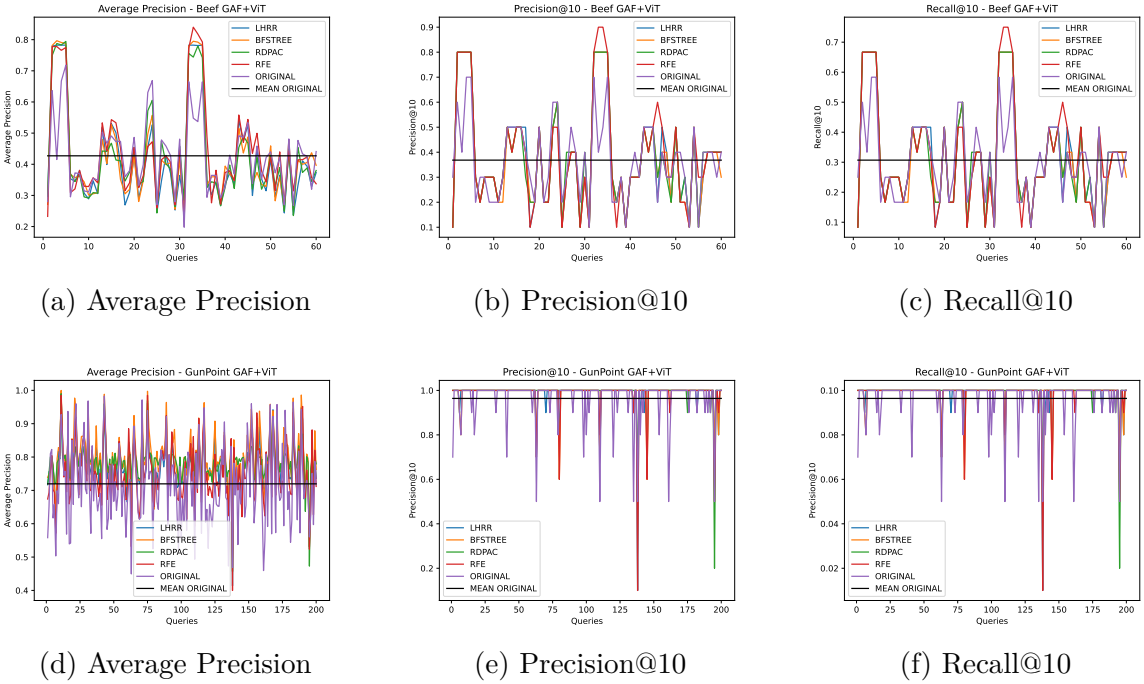


(a) Average Precision (b) Precision@10 (c) Recall@10

Figure 4.7 – Measurements for GunPoint dataset, considering the RP+ResNet feature extractor.

In Figure 4.8, we compare results obtained using the GAF-ViT feature extraction methods, which led to positive mAP gains in both datasets. Interestingly, we observe similar patterns to those seen in Figures 4.6 and 4.7. Notably, the average results for the Beef dataset were higher than the previous example, and all unsupervised distance learning methods performed better than the original, especially RFE. For the GunPoint dataset, BFSTREE yielded better results, while RFE and RDPAC produced some losses for a few queries despite the consistently average positive results of all re-ranking methods.

The obtained results suggest that positive outcomes are achieved through effective re-ranking when sufficient contextual information is available in the input ranked lists



(a) Average Precision (b) Precision@10 (c) Recall@10

(d) Average Precision (e) Precision@10 (f) Recall@10

Figure 4.8 – Comparision between datasets for GAF+ViT feature extractor.

computed based on the Euclidean distance. Otherwise, obtaining relevant gains is challenging when the original ranking presents low values of precision, recall, and average precision, i.e., when the input ranked lists do not contain much relevant time series at the top positions. In those cases, there might not be enough information available for re-ranking methods to enhance the outcomes.

4.4 Final Remarks

This study extensively examined time series representations and re-ranking methods for time series retrieval tasks. The findings indicate that a proper combination of time series representation and re-ranking methods holds promise for time series retrieval, with favorable outcomes observed in most scenarios. Nonetheless, it is worth noting that while results were mostly positive, the use of certain re-ranking methods led to minor losses (e.g., Beef and Rock datasets, considering the ROCKET feature extractor). Upon closer investigation, these losses appear to be attributed to limited contextual information available for substantial re-ranking improvements. No combination of time series retrieval and re-ranking methods yielded superior results for all datasets. These results suggest that their selection is application-dependent.

5 A Ranked-Based Framework based on Manifold Learning for Multivariate Time Series Retrieval and Classification

Multivariate time series find applications in various domains, and it is crucial to have tools for effectively manipulating this type of data. Currently, state-of-the-art algorithms for multivariate series primarily rely on deep learning approaches, which require large amounts of training data, significant processing time, and computational power. These algorithms are also built for specific tasks. However, traditional approaches for univariate time series may not be suitable due to the distinct dimensions of multivariate data. To overcome this challenge, we propose utilizing contextual rank aggregation, which combines multiple sets of ranked lists of a dataset to create a single, more effective ranked list. By considering each dimension of a multivariate time series as a separate univariate time series and extracting features for each dimension, as a sequel of the proposal presented in Chapter 4, we can utilize rank aggregation to merge the similarity between time series of different dimensions. This process enables us to obtain a single-dimensional similarity representation of a multivariate time series. Experiments were conducted on eight datasets, performing retrieval and classification tasks with the obtained representations. The results indicate that the proposed approach is suitable for the proposal and is competitive with state-of-the-art approaches. The main contributions of our work are listed below:

- The proposal of a versatile pipeline for both multivariate time series retrieval and classification, where any feature extractor, rank aggregation method, and classifier can be utilized in the experiments, providing a flexible model that can be adapted to various datasets and contexts;
- To the best of our knowledge, this is the first time that rank aggregation algorithms are utilized in the context of multivariate time series. This approach enables a thorough contextual analysis, effectively capturing the relationships between different dimensions of the series;
- Possibility to extend the framework's applicability to other information domains that encode multichannel data, e.g., hyperspectral images. This extension facilitates the aggregation of diverse dimensions within these data types, enhancing their analysis and interpretation.

The Chapter is organized as follows: In Section 5.1, we define the proposed

methodology, while Section 5.2 defines the experimental protocol and discusses the obtained results. Section 5.3 gives the conclusions about the results.

Findings related to the conducted study are reported in an article submitted to a journal [162].

5.1 Proposed Method

Defining similarity between representations is a challenging aspect in many scenarios, especially when there are different visions of the same data available, such as the different dimensions of multivariate time series, for example. Contextual rank aggregation methods combine different dimensions of similarity and also consider contextual relationships codified in datasets, generating a new similarity measure between the elements.

The concept of similarity in a feature space approaches both retrieval and classification task. In this work, the new similarity measure obtained with rank aggregation is used for retrieval tasks and classification with kNN. This Section details the proposed methodology.

Figure 5.1 shows the proposed methodology for multivariate time series retrieval and classification. In the first step, each dimension of the multivariate time series is split and considered as a univariate time series. Step two employs feature extraction methods for each dimension of the time series. The considered methods were:

- Beam Angle Statistics (BAS) [9]: a well-stabilished shape-based feature extractor,
- Random Convolutional Kernel Transform (ROCKET) [45]: a recent approach in the time series field,
- Recurrence Plot (RP) [53] with CNN ResNet-152 [72]: the combination of time series imaging methods with neural networks as feature extractors are popular tools in the area [204, 205, 212, 87],

all described in Section 2.5. Then, in the third step, a contextual rank aggregation process, defined in Section 2.6, is performed, generating rankings for a multivariate time series dataset. The following methods were chosen, based on recent approaches and code availability:

- Log-based Hypergraph of Ranking References (LHRR) [147],
- Rank Diffusion Process with Assured Convergence (RDPAC) [69],
- Rank Flow Embedding (RFE) [193],

detailed in Section 2.6. In the final step, a kNN classification is performed based on ranking information. The k-nearest neighbors are utilized for classification, where an element is labeled based on the most common class among the k nearest elements [61].

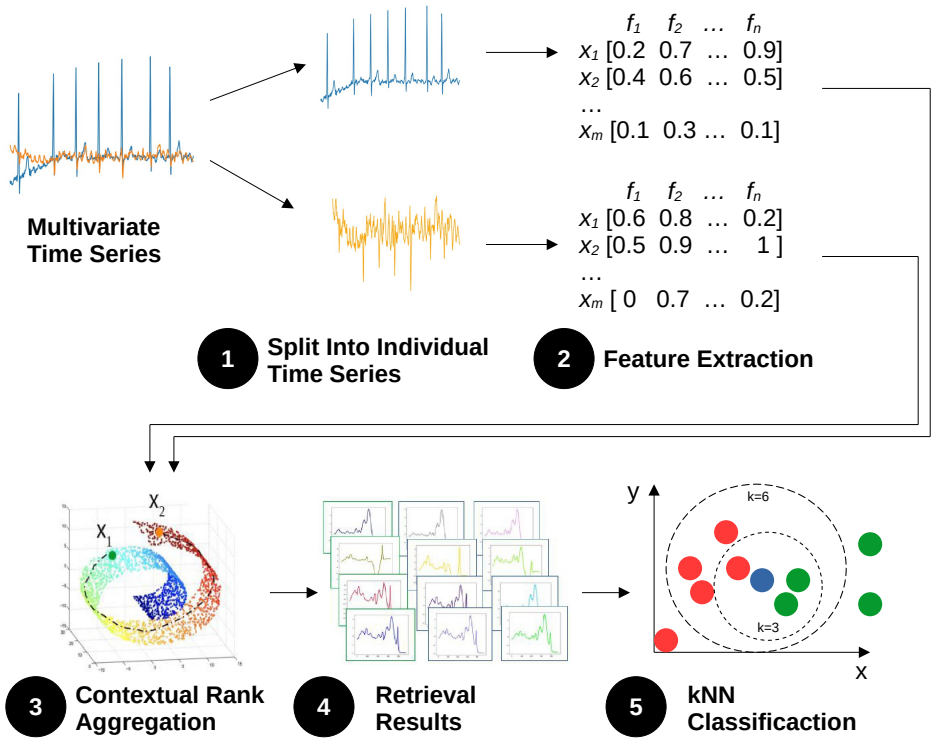


Figure 5.1 – Proposal for Multivariate Time Series retrieval and classification

5.2 Experimental Evaluation

In this Section, we detail the datasets utilized in the experiments, outline the experimental protocol, discuss the impact of parameters, and present the retrieval results and classification results, comparing them with recent baselines.

5.2.1 Datasets

This section details the 8 multivariate time series datasets utilized in our experiments, sourced from the UCR archive [11]. The datasets have several sizes, dimensions, number of classes, time series lengths, and domains. Not all datasets were contemplated due to restricted computational resources and time. Technical information and a brief description are included.

• **ArticularyWordRecognition (AWR) [202]:** Contains 575 time series with 9 dimensions and 25 classes, where 275 elements compose the train set. The time series

length is 144. Based on the analysis of tongue and lips movement during a speech by Motion Track using an Electromagnetic Articulograph (EMA).

• **AtrialFibrillation (AF) [126]:** Has 30 time series with a length of 640 and 2 dimensions, divided into 3 classes. This dataset is composed of ECG signals of atrial fibrillation. The train/test split of this dataset is 50%/50%.

• **BasicMotions (BM) [Clements]:** This dataset has 80 6-dimensional time series with lengths 100 and 4 classes that correspond with smartwatch signals of four basic activities. Half of the dataset is used for training.

• **Heartbeat (HB) [114]:** 61-dimensional dataset with 2 classes and 409 elements, and 204 elements compose the training set. Each element has a size of 405. Composed of heart sounds from healthy and pathological patients, recorded from different body locations.

• **MotorImagery (MI) [100]:** Composed by 378 64-dimensional time series with length 3000 and divided into 2 classes, of which 278 series are in the train set and 100 series in the test set. EEG dataset of imagined movements of the left small finger or the tongue.

• **NATOPS (N) [65]:** 360 time series with size 51, 24 dimensions and divided into 6 classes. The data is obtained by sensors placed on different body parts, and the intended purpose is to differentiate several actions. Half of the data composes the train set.

• **PenDigits (PD) [6]:** 10 classes of time series with 2 dimensions and length 8. There are 7494 train time series and 3498 test time series. It is a handwritten digit classification task.

• **StandWalkJump (SWJ) [15]:** 27 4-dimensional time series divided into 3 classes, with the size of 2500. 12 of the time series are in the train set. It comprises ECG samples from a healthy 25-year-old male standing, walking, and jumping.

5.2.2 Experimental Protocol

Our experiments were conducted based on the proposed methodology. All n-dimensional time series from each dataset above-described were split into n unidimensional time series. Feature extraction was performed for each dimension. To execute the contextual rank aggregation, it was necessary to get an Euclidean Distance matrix between all elements in each time series dimension. After the rank aggregation, the ranking is evaluated considering the Precision@10 and Mean Average Precision (mAP), described in Section 2.8, and the ranking information is used for the k-nearest Neighbors (kNN) classification, using the train/test split defined in the literature for each dataset. The classification task is evaluated based on the accuracy of the results, and three recent

baselines are considered for comparison.

5.2.3 Impact of Parameters

In this Section, we discuss the impact of the neighborhood parameter on the experiments. The k number of neighbors has a crucial impact on the retrieval results using unsupervised distance learning algorithms and classification tasks, considering the k-Nearest Neighbors classifier. Considering this, an experimental pipeline was conducted to evaluate the impact of k on the accuracy of classification results in each dataset. Following the experimental protocol, we performed the retrieval and classification, variating the value of k from $[5, x]$, with step 5, where x is the last multiple of five before the number of elements per class in a dataset. In Figure 5.2, the three most significant graphic examples are presented for datasets AtrialFibrilation, Heartbeat, and MotorImagery, based on the characteristics of the datasets such as number of series in a dataset, length of time series, number of elements per class and variation of the accuracy considering the variation of the parameter k.

Considering the general results and the two smaller datasets, AttrialFibrilation and StandWalkJump, we delimited $k = 10$, which seems to be a value that attends all of the proposed datasets in our experiments.

5.2.4 Retrieval Results

Tables 5.1 and 5.2 present the results for retrieval tasks, considering the combinations of representation and ranking aggregation methods presented in each line of the Tables. Each column corresponds to a dataset where the Precision@10 and mAP metrics are reported, with one metric in each Table. The best results for each dataset are highlighted in bold.

Table 5.1 – Retrieval results: P@10

Method Combination	AWR	AF	BM	HB	MI	N	PD	SWJ
BAS + LHRR	82.45%	**47.33%**	94.50%	65.87%	**56.32%**	67.39%	52.85%	39.63%
BAS + RDPAC	84.23%	46.00%	74.25%	62.08%	55.29%	70.31%	55.04%	**43.70%**
BAS + RFE	73.95%	45.00%	86.87%	63.69%	54.34%	64.61%	51.32%	37.41%
ROCKET + LHRR	95.81%	36.00%	99.37%	67.53%	55.05%	74.72%	95.14%	37.41%
ROCKET + RDPAC	**98.21%**	34.67%	97.12%	68.22%	54.81%	**77.47%**	**99.13%**	37.41%
ROCKET + RFE	92.75%	37.33%	**100%**	67.80%	54.95%	71.06%	98.68%	40.74%
RP + LHRR	93.48%	44.00%	95.63%	67.70%	55.74%	69.83%	87.04%	41.85%
RP + RDPAC	96.37%	41.33%	95.63%	**69.12%**	55.77%	67.78%	93.18%	39.63%
RP + RFE	87.51%	39.67%	97.12%	65.89%	55.90%	66.64%	91.36%	36.30%

In ArticularyWordRecognition, it is possible to note that RDPAC and LHRR provided the best results of P@10 and mAP. At the same time, ROCKET and RP + ResNet proved to be superior features for this dataset. A particular highlight is the combination of ROCKET and RDPAC. AttrialFibrilation presented a different behavior

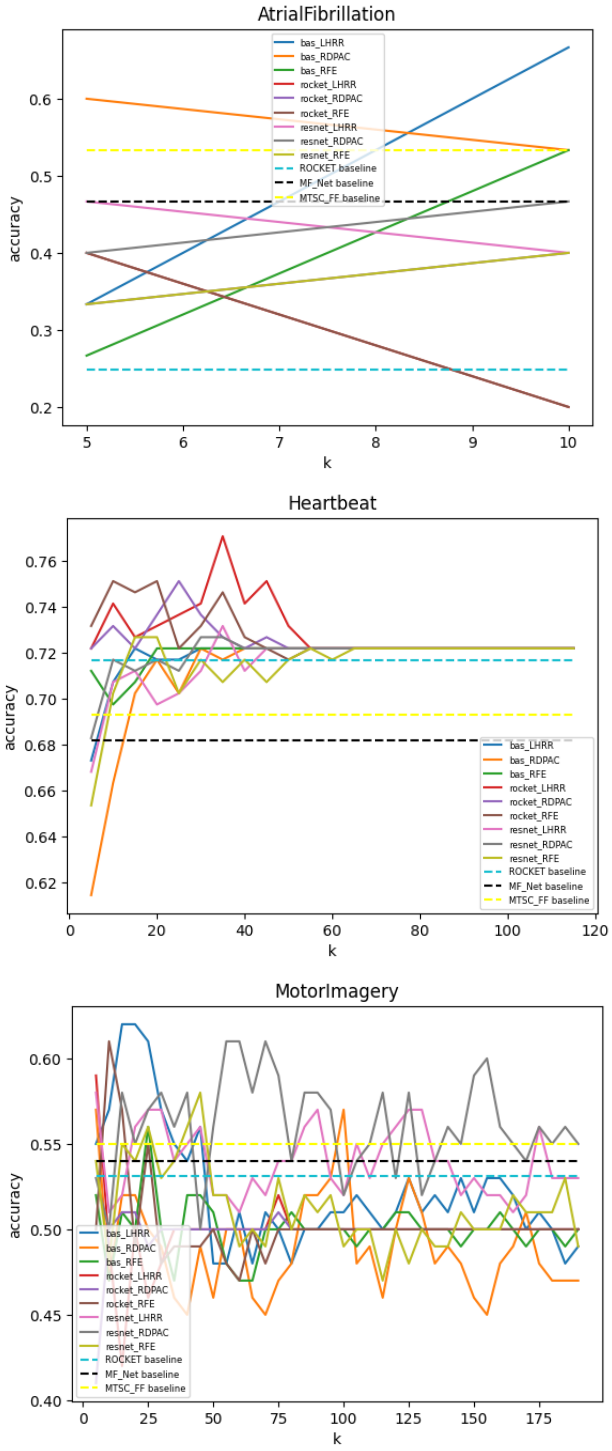


Figure 5.2 – Accuracy varying over the value of k.

from the previous dataset, where the best combination for P@10 (BAS and LHRR) is not the same as for mAP (BAS and RDPAC). Nevertheless, BAS has proven to be a suitable feature extractor for this dataset. BasicMotions presented very high results, reaching 100% of P@10 and 99.09% of mAP. ROCKET and RP + ResNet are, again, highlights of the feature extraction, but this time, the combination of ROCKET and RFE led to the best results. Heartbeat dataset does not present a highlight for the feature extractor,

Table 5.2 – Retrieval results: mAP

Method Combination	AWR	AF	BM	HB	MI	N	PD	SWJ
BAS + LHRR	69.24%	56.15%	93.38%	61.60%	**51.89%**	48.24%	03.31%	51.74%
BAS + RDPAC	74.07%	**57.39%**	79.36%	59.36%	51.58%	57.08%	07.40%	**55.95%**
BAS + RFE	58.26%	53.32%	86.32%	61.53%	51.38%	41.02%	05.42%	49.96%
ROCKET + LHRR	91.93%	47.62%	98.50%	63.62%	51.47%	67.12%	36.86%	49.66%
ROCKET + RDPAC	**97.01%**	45.50%	97.67%	**63.68%**	51.43%	**74.11%**	**48.94%**	50.85%
ROCKET + RFE	85.24%	46.90%	**99.09%**	63.13%	51.44%	62.71%	44.18%	51.81%
RP + LHRR	88.37%	53.96%	95.28%	62.90%	51.53%	58.38%	22.82%	54.56%
RP + RDPAC	94.80%	51.91%	96.01%	63.51%	51.58%	61.92%	39.01%	51.90%
RP + RFE	76.97%	50.38%	95.18%	62.37%	51.52%	52.93%	29.74%	52.58%

but RDPAC provided the best results with RP + ResNet feature extraction (P@10) and ROCKET (mAP). Significative differences between the results cannot be seen in the MotorImagery dataset, but the combination of BAS and LHRR produced a higher P@10 and mAP. In NATOPS, ROCKET provided the best results, especially when combined with RDPAC. The same behavior can be observed in the PenDigits dataset. Finally, in StandWalkJump, similar to MotorImagery, significative differences between the results cannot be seen, but BAS associated with RDPAC yielded the best results.

Except for AttrialFibrilation and Heartbeat, the highlights for P@10 and mAP in each dataset are seen in the same descriptor and ranking aggregation method. Combining the ROCKET feature extractor and RDPAC rank aggregation generally yielded the best results.

5.2.5 Classification and Comparison with Other Approaches

In Table 5.3, we present the accuracy results for the classification tasks. Our results are compared with three different recent methods for time series classification. Lines refer to datasets, and columns refer to classification methods. For each dataset, the best accuracy results are highlighted in bold. The win counts for each classifier are included at the bottom of the Table.

Table 5.3 – Accuracy comparison results

	Baselines			Proposed Approaches								
Dataset	ROCKET [45]	MF-Net [52]	MTSC-FF [51]	BAS+ LHRR	BAS+ RDPAC	BAS+ RFE	ROCKET+ LHRR	ROCKET+ RDPAC	ROCKET+ RFE	RP+ ResNet+ LHRR	RP+ ResNet+ RDPAC	RP+ ResNet+ RFE
AWR	**99.60%**	98.30%	98.30%	84.70%	81.70%	75.33%	97.00%	98.00%	95.00%	95.00%	95.70%	90.70%
AF	24.90%	46.60%	53.30%	**66.70%**	53.33%	53.33%	20.00%	40.00%	20.00%	40.00%	46.70%	40.00%
BM	99.00%	95.00%	95.00%	87.50%	67.50%	90.00%	97.50%	95.00%	**100%**	**100%**	97.50%	**100%**
HB	71.70%	68.20%	69.30%	70.73%	66.34%	69.80%	74.15%	73.20%	**75.12%**	70.73%	71.71%	70.24%
MI	53.10%	54.00%	55.00%	57.00%	47.00%	47.00%	49.00%	50.00%	**61.00%**	51.00%	50.00%	48.00%
N	88.50%	**92.70%**	88.90%	63.33%	65.00%	56.70%	72.22%	76.11%	70.00%	71.11%	63.90%	67.22%
PD	**99.60%**	98.30%	97.90%	59.20%	55.63%	53.52%	93.85%	98.37%	97.31%	88.91%	93.05%	90.71%
SWJ	45.60%	40.00%	46.70%	46.70%	**53.33%**	26.70%	33.33%	20.00%	33.33%	**53.33%**	40.00%	33.33%
Wins	2	1	0	1	1	0	0	0	**3**	2	0	1

The datasets ArticularyWordRecognition and PenDigits presented similar behavior, where ROCKET has the accuracy highlight. However, the other methods note similar accuracy results, excluding those obtained with the BAS feature extractor. NATOPS had better results with the MF-Net classifier, and our approach seems less suitable for this

dataset. StandWalkJump had two interesting highlights, with BAS + RDPAC and RP + ResNet + LHRR providing superior results compared to the baselines. MotorImagery and HeartBeat are highlighted by a combination of ROCKET and RFE. At the same time, the BasicMotions dataset is not only highlighted for this combination, but also for RP + ResNet with LHRR and RFE, with 100% accuracy. Finally, AttrialFibrilation obtained the best results with BAS and LHRR.

From 8 datasets, our methods performed better than the baselines in 5 cases. Despite not having the same highlights, the P@10 and mAP results complement the accuracy results. We can highlight the combination of ROCKET, RFE, and kNN with three best results. It is noteworthy that the accuracy reached 100% in the BasicMotions dataset. It is not a rule that using ROCKET as a feature extractor combined with a rank aggregation method and a kNN classifier will always generate better results when compared with the classification using only ROCKET classifier.

5.3 Final Remarks

In this work, we proposed a versatile pipeline suitable for multivariate time series retrieval and classification. The individual dimensions are fully explored to find better similarity representations for multivariate time series datasets. The experimental evaluation indicates interesting results for retrieval tasks and promising results for classification, reaching competitive results with recent baselines.

6 Framework for Data Analysis by Temporal Graph Encoding

Temporal graphs have been successfully employed to model complex spatiotemporal phenomena. This Chapter introduces a new unsupervised methodology to leverage a domain-specific analysis by using temporal graphs and time segmentation criteria. The proposal was validated in a football analysis context, where we analyze ball possessions in football matches based on clustering techniques. This work relies on representing the domain-specific data as temporal graphs based on changes over time. The goal is to identify relevant spatiotemporal patterns found in data. First, our data (ball possessions) are represented by temporal graphs, which are later encoded into graph visual rhythms by considering time segmentation criteria (changes in the team with the ball possession). Feature extraction is then performed utilizing pre-trained deep learning models, and dimensionality reduction methods are applied in the vector representation. The final step concerns the identification of patterns based on cluster analysis. Experiments were conducted on data from 10 Brazilian First Division Championship matches, utilizing diverse methods for temporal graph representation, pre-trained deep learning models, dimensionality reduction approaches, and clustering techniques. Experimental results suggest that our methodology is suitable for ball possession analysis and any domain-specific data with temporal correlation. The best-performing components led to a clear separability of ball possession clusters associated with relevant tactical properties.

In short, our main contributions are:

- A flexible pipeline for unsupervised analysis of domain-specific data;
- Comparative study involving temporal graph representations, complex network measurements, pre-trained deep learning models, dimensionality reduction methods, and clustering techniques, aiming at identifying the best-performing combination for spatiotemporal data analysis.

This work was developed in partnership with Professor Ricardo da Silva Torres and Professor Felipe Arruda Moura, in the Wageningen University & Research. The work is in the process of writing and submission of a paper in a journal [Rozin et al.].

The rest of the Chapter is organized as follows: Section 6.1 discusses the proposed methodology and relevant background concepts, while Section 6.2 overviews the experimental protocol, while Section 6.3 presents and discusses obtained results. Our conclusions are presented in Section 6.4.

6.1 Materials and Methods

The proposed approach is based on clustering domain-specific data based on temporal segmentation criteria. We analyzed offensive sequences of ball possessions from football matches. Data is modeled as temporal networks from which measurements are computed. This allows for identifying patterns in data associated with relevant characteristics or events. Figure 6.1 summarizes the proposal of our framework:

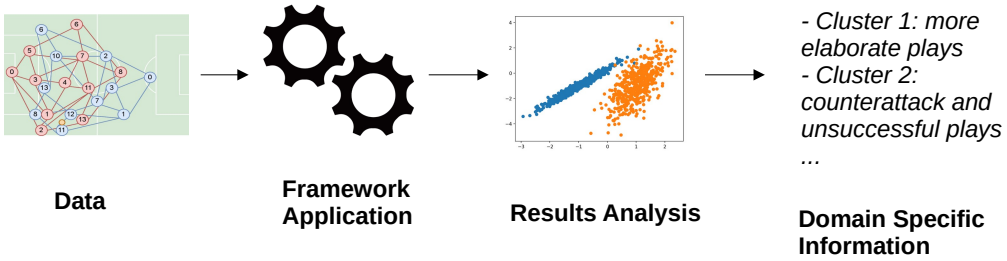


Figure 6.1 – This Figure summarizes the objective of our proposal. Data is inserted into our pipeline, generating clustering results, where valuable information about initial data can be obtained. In this example, information about ball possessions is extracted

Figure 6.1 presents an overview of the proposed framework methodology. We use data collected from football matches, which shows players' positions over time. This data is used to build graphs that encode players and define their interactions in terms of their proximity. After representing a match as a temporal network, the graph measurements for each time stamp are extracted, as shown in step 1. This can be interpreted as a multivariate time series. Then, considering each ball possession, the time series are split and codified as graph visual rhythm images in step 2. We perform feature extraction, in step 3, from the images using transfer learning, i.e., using pre-trained deep learning models. Next, we perform dimensionality reduction to obtain 2D representations, as presented in step 4. Then, finally, in step 5, we cluster the ball possession according to their temporal-graph-based patterns. Each step is detailed in the next sections.

6.1.1 Temporal Graph Creation

The available data consists of players' coordinates of the players over time. Considering a team, for each time stamp t_i, we construct a graph where each node corresponds to a player, and each edge encodes information regarding the proximity of players. Edges are, therefore, weighted by the distance between two players. We also follow the strategy of Rodrigues et al. [160] to remove edges from the graph if an adversary player is between two teammates. The graph changes as the players' positions change along the timestamps. Changes are encoded into the graph if a player has been sent off or substituted. Those changes over time characterize the temporal graph. Figure 6.3 exemplifies how the

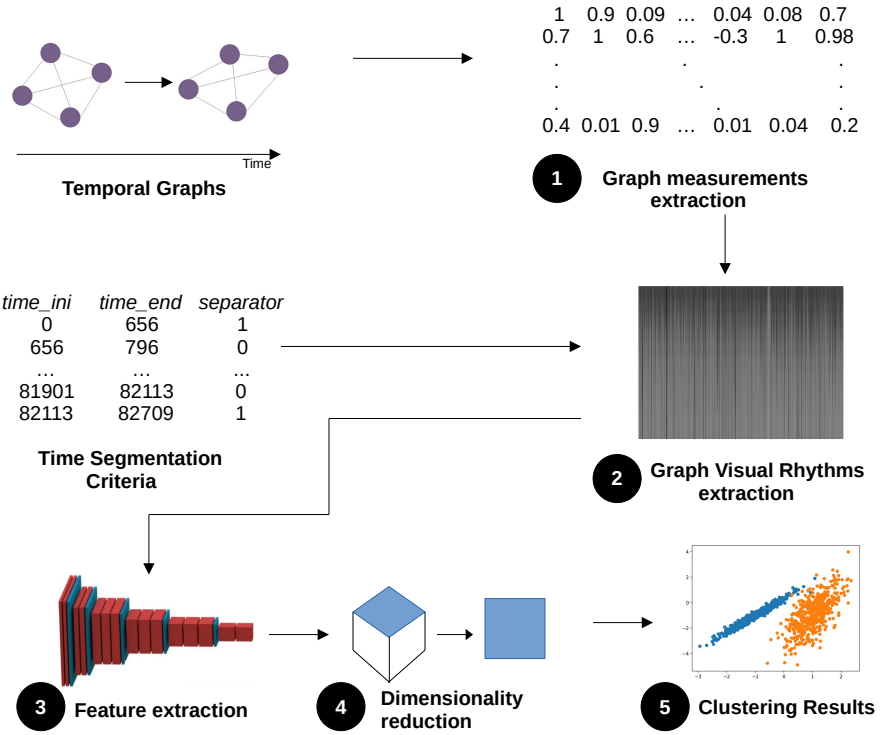


Figure 6.2 – Generic pipeline for data characterization based graph visual rhythm representations and transfer learning procedures. The methodology comprises five steps. First, graph measurements are extracted from temporal graphs (1). Later, those graph measurements are used to create graph visual rhythms (GVRs) related to temporal segmentation criteria (2). Then, feature vectors are extracted from GVRs based on pre-trained deep learning methods (3). Dimensionality reduction is then employed for projecting features into 2D spaces (4). The final step concerns clustering data based on their 2D representations (5).

graph changes over time. Red and Blue graphs correspond to different teams, while the yellow point represents the ball.

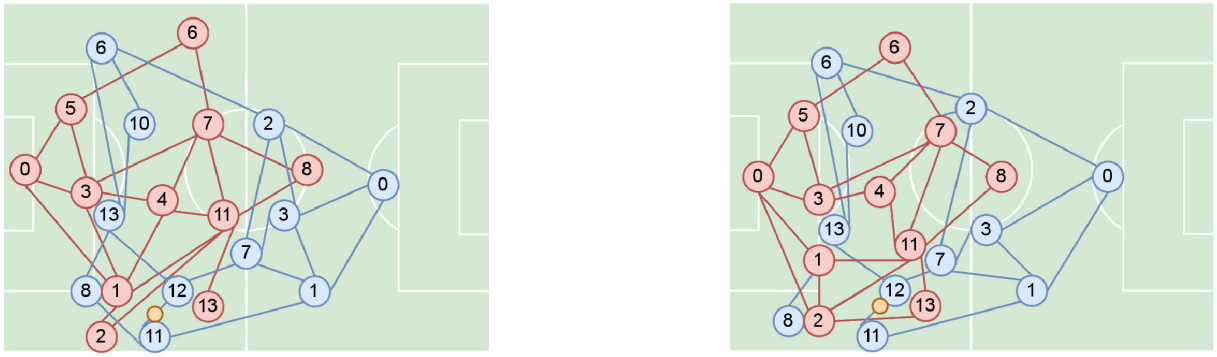
6.1.2 Graph Measurements

Graph measurements give diverse information about graphs and are crucial for their analysis. Our study investigates five graph measurements explored in the context of football match analysis [183].

- **Average Path Lenght:**

Also called the average shortest path, it defines the average distance between any pair of nodes in a graph [adn]. The average path length is described by Equation 6.1:

$$a = \frac{1}{n(n-1)} \times \sum_{\substack{s,t \in V \\ s \neq t}} d(s,t), \tag{6.1}$$



(a) Timestamp t_1 (b) Timestamp t_2

Figure 6.3 – Graph changes over time. Red and Blue graphs correspond to different teams, while the yellow point represents the ball.

where n is the number of nodes in a graph G, V is the set of nodes in G, and $d(s,t)$ is the shortest path from node s to node t.

The average path length measures how dispersed a player is concerning the rest of the team. Higher values mean that the player is more isolated from the rest of the team, while lower values indicate higher compactness of teammates.

- **Betweeness Centrality:**

Let $\sigma(s,t)$ denote the count of shortest paths from s to t, while $\sigma(s,t|v)$ represents the number of shortest paths from s to t that passes through v. The betweenness centrality $c_b(v)$ of a node v is computed by summing $\sigma(s,t|v)$ across all unique node pairs s and t, and then dividing by the total count of shortest paths [22].

$$c_b(v) = \sum_{s,t \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}, \tag{6.2}$$

where

$$\begin{cases} \sigma(s,t) = 1, & \text{if } s = t \\ \sigma(s,t|v) = 0, & \text{if } v \in \{s,t\} \end{cases} \tag{6.3}$$

Below is a commonly employed normalization method for betweenness centrality, where n is the total number of vertices.

$$c_{b_n}(v) = \frac{c_b(v)}{(n-1)(n-2)} \tag{6.4}$$

The betweenness centrality shows how important a player is in play constructions. Higher values indicate a greater availability to interact in further plays and do passes.

- **Eccentricity:**

The eccentricity of the vertex v of a graph corresponds to the maximum distance between v and any other node. The distance $d(v, u)$ between the nodes u and v corresponds to the shortest path length between them [208]. The eccentricity is defined in Equation 6.5:

$$e(v) = max(d(v, v_j)), \forall v_j \in V \tag{6.5}$$

The eccentricity represents how easily the ball can reach a player, considering the entire network.

- **Local Efficiency:**

For a vertex v, we consider a subgraph $\hat{G}_v$, composed of the neighbors of v and their edges. The local efficiency of a vertex v corresponds to the fault tolerance by accessing the local communication, i.e., the direct neighbors, when a vertex and all of its corresponding edges are removed [101]. Considering $deg(v)$ the degree of a vertex v, Equation 6.6 defines the local efficiency:

$$E_{loc}(v) = \frac{1}{deg(v)} \sum_{v_j \in \hat{G}_v} E(v_j) \tag{6.6}$$

The local efficiency shows the importance of the players in plays regarding the impact on the nearby teammates.

- **Vulnerability:**

The vulnerability of a node v considers global and local efficiency to compute the efficiency drop after removing v [75]. Equation 6.7 defines the vulnerability:

$$V(v) = 1 - \frac{E_{loc}(v)}{E(v)} \tag{6.7}$$

Vulnerability shows the impact on the team wherever a given player is removed. Negative values indicate less efficiency without the player.

- **Global Efficiency:**

The global efficiency of a vertex v is the measurement of the information transmission capacity of the graph through v and indicates the efficiency for sending information amidst vertices. It is proportional to the inverse distance between v and other vertex j [101]. Equation 6.8 defined the global efficiency:

$$E(v) = \frac{\sum_{j \neq v}^{j \in G} \frac{1}{d(v-j)}}{(n-1)} \tag{6.8}$$

The global efficiency indicates how well-positioned the players are so they can be available to make and receive passes.

6.1.3 Graph Visual Rhythms (GVR)

In this work, temporal graphs, where each graph corresponds to one of the teams in a match, are represented considering the graph measurements and the ball possessions. For each timestamp, graph measurements are obtained for each node. The final result can be interpreted as an 11-dimensional multivariate time series. A GVR [160] is an image representation in which each line refers to a player, and each column represents a time stamp. Formally, given a temporal graph $\mathbb{G} =< G_i, G_i + 1, ..., G_j >$, where i is the frame of the match where the ball possessions begin and j is the frame where the ball possession ends, a set of graph measurements is computed for each vertex (player) in the graph. The temporal graph $\mathbb{G}$ is converted to a map $\mathbb{F}$ where:

$$\mathbb{F} = \begin{matrix} <<F_{0,i}, F_{0,i+1}..., F_{0,j}> \\ <F_{1,i}, F_{1,i+1}..., F_{1,j}> \\ ... \\ <F_{m,i}, F_{m,i+1}..., F_{m,j}>> \end{matrix} \tag{6.9}$$

For a soccer match, $m = 10$, considering the number of 11 players. From $\mathbb{F}$, an image is created by considering each value $F(i, j)$ as the intensity of the pixel at the position (i, j). For standardization, the values of graph measurements are sorted, for each timestamp. Figure 6.4 presents an example of a GVR representation.

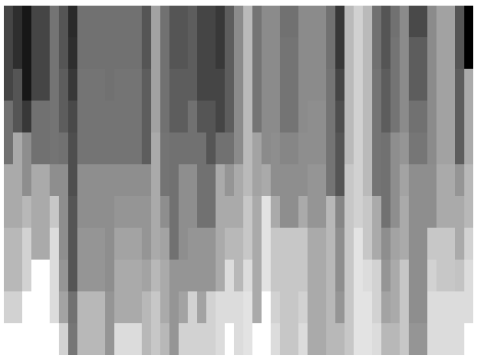


Figure 6.4 – Example of Graph Visual Rhythm representation

6.1.4 Image Feature Extraction

Images hold diverse information, and using pre-trained neural networks via transfer learning for feature extraction has exhibited potential in diverse domains due to robust generalization and feature learning [115]. Here, we employ a transfer learning procedure based on five different neural networks, pre-trained on ImageNet [46], for feature extraction. The neural networks are:

- CNN ResNet-152 [72]

- Vision Transformers [96]
- VGG-19 [179]
- EfficientNet V2 [187]
- DINO V2 [134]

Considered models are described in Section 2.5.3. The networks were selected based on previous literature results for diverse machine learning tasks [72, 96, 179, 187, 134].

6.1.5 Dimensionality Reduction

The above-described feature extractors produce high-dimensional features. Dimensionality reduction methods are employed to visualize high-dimensional data in lower dimensions and avoid the curse of dimensionality in machine learning tasks [17, 16]. Three dimensionality reduction methods were selected, based on a traditional approach [64], and in *de facto* [38] standards. In this work, data was reduced to two dimensions. Below, the dimensionality reduction methods employed in our study are described.

- **Principal Component Analysis (PCA)**

PCA [64] is a statistical dimensionality reduction method built to retain as much original information as possible. It starts by centering the data to ensure the new coordinate system is centered at the origin. Next, it computes the covariance matrix of the centered data, which measures the relationships between pairs of features. PCA, then, performs eigenvalue decomposition on the covariance matrix to find its eigenvectors and eigenvalues. The principal components, representing the directions of maximum variance in the data, are ranked based on their corresponding eigenvalues. Typically, a subset of principal components is selected to retain the desired amount of variance. Finally, the original data is projected onto the selected principal components to obtain a lower-dimensional representation.

- **t-distributed Stochastic Neighbor Embedding (t-SNE)**

t-SNE [196] is a statistical dimensionality reduction method commonly used for visualizing high-dimensional data in lower-dimensional space, typically two or three dimensions. First, the algorithm computes pairwise similarities between data points in the high-dimensional space, often using a Gaussian kernel function based on Euclidean distances. Next, it constructs a probability distribution over pairs of points in both the high-dimensional and lower-dimensional spaces using a heavy-tailed Student's t-distribution. This distribution assigns higher probabilities to similar data points and lower probabilities to dissimilar ones. The lower-dimensional embeddings are then optimized to minimize the Kullback-Leibler divergence (KL divergence) between these two distributions.

This optimization process involves iteratively adjusting the positions of points in the lower-dimensional space using gradient descent. Finally, t-SNE arranges similar data points close to each other in the lower-dimensional space while pushing dissimilar points farther apart, revealing the underlying structure and relationships within the data.

- **Uniform Manifold Approximation and Projection (UMAP)**

UMAP [121] is an alternative dimensionality reduction algorithm to t-SNE. It constructs a high-dimensional graph representation of the data, connecting each data point to its nearest neighbors. Unlike t-SNE, which primarily preserves local structure, UMAP aims to capture both local and global structure. It creates a fuzzy topological representation of the data, representing each point as a fuzzy set of its neighbors to flexibly capture relationships. Then, the cross entropy between high and low-dimensional data is minimized by adjusting the positions of the points, creating a low-dimensional embedding that captures both local and global data structures.

6.1.6 Clustering

Three different clustering methods were considered for our study. We selected methods that do not require determining the number of clusters beforehand.

- **Density-Based Spatial Clustering of Applications with Noise (DBSCAN)**

DBSCAN [56] is a robust density-based clustering algorithm that defines a cluster as an area of high density separated by areas of low density. Starting from an arbitrary dataset point, all points in an ϵ-distance are found, where ϵ is a radius within which the points are considered neighbors. If the number of points within this distance exceeds the predetermined threshold (MinPts), the selected point is designated as a core point. This process is applied recursively to neighboring points until all border points, which lack sufficient neighbors to be core points, are identified. The algorithm continues recursively until all points in the dataset have been assigned to a cluster or labeled as noise points if they are not within the ϵ-distance of any core point.

- **Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)**

HDBSCAN [25], an extension of the DBSCAN algorithm, that automates parameter selection and introduces a hierarchical clustering approach. Unlike its predecessor, HDBSCAN dynamically determines essential parameters, eliminating the need for manual tuning. By constructing a hierarchical clustering of the dataset, HDBSCAN captures clusters at multiple levels of granularity, allowing for the identification of clusters with varying densities and shapes. Due to its robustness to noise and improved outlier detection, HDBSCAN frequently outperforms DBSCAN.

- **Affinity Propagation**

Affinity Propagation [63] utilizes a similarity matrix to iteratively navigate between two prescribed energy-based functions termed "messages" among dataset elements. In this iterative process, each data point is considered a possible cluster center, while a responsibility function is used to see how well each data point supports other data points being cluster centers. Specifically, a message sent from element i to candidate cluster center c_c measures the suitability of c_c to function as the cluster center for element i. Reciprocally, an availability function emanating from a candidate cluster center c_c to an element i evaluates the appropriateness of element i, selecting c_c as its cluster center. Upon convergence, the algorithm reveals a set of robust cluster centers and a refined clustering arrangement for the dataset. This clustering algorithm relies on diverse similarity measures between data samples, diverging from conventional prototype-based clustering methods like k-means, which necessitate an explicit vector space structure.

6.2 Experimental Protocol

This section presents the adopted evaluation protocol, including information about the evaluation measures and the dataset.

6.2.1 Clustering Quality Measurements

We chose intrinsic methods to evaluate the clustering tasks, i.e., methods that do not require a ground truth to evaluate the results. These methods assess the separability and compactness of computed clusters [int]. The applied methods are described below.

- **Silhouette Score**

The Silhouette Score [161] measures how similar an object is to its cluster (cohesion) compared to other clusters (separation). A higher silhouette score indicates that the object is well-matched to its cluster and poorly matched to neighboring clusters. The Silhouette Score is represented by Equation 6.10:

$$Sil = \frac{m_{nc} - m_{ic}}{max(m_{ic}, m_{nc})}, \tag{6.10}$$

where m_{ic} is the mean intra-cluster distance and m_{nc} is the mean nearest-cluster distance for each sample.

- **Davies-Bouldin Index**

The Davies-Bouldin Index (DBI) [40] rates the clustering quality by measuring the similarity between each cluster and its most similar neighboring cluster relative to the average dissimilarity within each cluster. First, it computes the centroid of each cluster and

then calculates the similarity between centroids of all pairs of clusters. For each cluster, the neighboring cluster with the highest similarity is identified. The DB index for each cluster is then computed based on the Equation 6.11:

$$DBI = \frac{S_i + S_j}{d(C_i, C_j)}, \tag{6.11}$$

where S_i is the average dissimilarity of cluster i to all other clusters, S_j is the average dissimilarity of the neighboring cluster of cluster i to all other clusters, and $d(C_i, C_j)$ is the similarity between centroids of clusters i and j. The overall DB index is the average of all cluster-specific DB values. A lower DB index indicates better clustering, signifying well-separated and compact clusters. At the same time, higher values indicate poorer clustering with clusters that may be too spread out or too close to each other.

- **Calisnki-Harabasz Index**

The Calinski-Harabasz Index (CHI) [23] evaluates the clustering quality by assessing the ratio of between-cluster dispersion to within-cluster dispersion, measuring the compactness and separation of clusters. It begins by calculating the total dispersion of data points, representing the sum of squared distances between each point and the overall centroid of the dataset. Subsequently, it computes the dispersion within clusters (D_W), representing the sum of squared distances between each point and its centroid within its cluster, and the dispersion between clusters (D_B), representing the sum of squared distances between each cluster centroid and the overall dataset centroid. The CHI is then computed using Equation 6.12:

$$CHI = \frac{\frac{D_B}{(k_c-1)}}{\frac{D_W}{(n-k_c)}}, \tag{6.12}$$

where k_c is the number of clusters and n, is the total number of data points.

A higher CHI suggests better clustering, with well-separated and compact clusters, while lower values indicate poorer clustering, characterized by potential overlap or excessive cluster spread.

6.2.2 Domain Specific Indicators

Our evaluation also considers football performance measurements extracted from the matches. Beyond statistics from the football match, e.g., mean number of passes per cluster, we also consider two concepts, team spread and sample entropy, detailed below.

- **Team Spread [128]**

In each timestamp, t_k, a symmetric Euclidean Distance matrix $D(t_k)$, of order 11 (11 players) is calculated between each player and his teammates. Considering the characteristics of the matrices, where $d_{ii} = 0$, because the distance between a player and himself is 0, and $d_{ij} = d_{ji}$, only the lower triangular matrix L_t is considered. The Frobenius

norm is frequently used in linear algebra, providing a measure of distance [67], and it is applied in L_t for each timestamp t_k to obtain the team spread. This is described in Equation 6.13.

$$||L_t(t_k)||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}|l_{ij}|^2} \tag{6.13}$$

Lower values indicate that the players are closer together, and higher values indicate that the players are more spread out across the football field.

The minimum spread refers to the minimum value of the spread, the maximum spread refers to the maximum value of the spread, and the range spread refers to the difference between the maximum and the minimum spread.

- **Sample Entropy [159]**

Sample entropy (SampEn) is a statistical measure of the complexity of time series signals. Let an embedding dimension m, tolerance r and a time series $T = \{t_1, t_2, ..., t_n\}$, with n number of points, we select segments $T_m(i) = \{t_i, t_{i+1}, t_{i+2}, ..., t_{i+m-1}\}$, and calculate B, point by point, which corresponds to the number of cases where $D[T_m(i), T_m(j)](i \neq j) < r$, and D is a distance function. We also want to calculate C, point by point, which corresponds to the number of cases where $D[T_{m+1}(i), T_{m+1}(j)](i \neq j) < r$. The Sample Entropy is described by the negative natural logarithm of this conditional probability and is represented by Equation 6.14.

$$SampEn = -\ln \frac{C}{B} \tag{6.14}$$

Higher values indicate more complex time series with frequent changes and less predictability.

- **Number of Passes:** Refers to the number of passes performed during a ball possession.
- **Number of Dribbles:** Refers to the number of dribbles performed during a ball possession.
- **Reached the Final Third of the Field:** An indicator if the plays with the ball during a ball possession reached the final third of the field.
- **Attack Duration:** Refers to the time duration of a ball possession.
- **Distance Travelled by Ball:** Refers to all the distance traveled by the ball during a ball possession.

6.2.3 Dataset

The study analyzed 1279 ball possessions from 10 official Brazilian First Division Championship matches. Filming was conducted using up to six digital cameras, each operating at a rate of 7.5 Hz or 30 Hz, depending on the match, positioned at elevated stadium locations. These cameras collectively covered different sections of the football pitch, with some overlapping areas. Post-match, the images were transferred to computers and synchronized with event identification in overlapping regions, such as player kicks or ball impacts. Player positions over time were then tracked using an automatic tracking method. Subsequently, a Matlab® algorithm was developed to analyze the available data and determine the presence or absence of ball possession during the matches [129, 127].

The ball possessions were selected based on the following criteria:

- Length equal or superior to four seconds;
- Sample Entropy of the Spread variable is superior to zero;
- If the number of dribbles and passes equals zero, the ball possession is excluded from the analysis.

Following these criteria, from 1514 ball possessions, 1279 were selected.

6.3 Results

This section presents and discusses quantitative and qualitative results.

6.3.1 Quantitative Results

Tables 6.1, 6.2, 6.3, 6.4, and 6.5 present the clustering results for each graph measurement. Rows relate to the feature extraction methods, while columns refer to the clustering evaluation metrics. The results are grouped in the rows by the dimensionality reduction method and in the columns by the clustering method. The best results for each clustering metric are highlighted in bold in each table.

Table 6.1 details the results relative to the Average Path Lenght graph measurement. It is possible to observe that HDBSCAN outperforms the other two clustering methods in the Silhouette Score and Daves-Bouldin Measurements. At the same time, Affinity Propagation provides the best Calinsky-Harabaz Index results. Usually, feature extraction with ResNet-152 provides better results than the other neural networks. Dimensionality reduction with UMAP provides better results.

Table 6.2 presents the Betweenness Centrality results. The patterns observed about HDBSCAN, Affinity Propagation, and UMAP methods are also observed in this set of

Table 6.1 – Results for the Average Path Length graph measurement. Rows relate to the feature extraction methods, while columns refer to the clustering evaluation metrics. The results are grouped in the rows by the dimensionality reduction method and in the columns by the clustering method. DB refers to the DBSCAN clustering method, AP to Affinity Propagation, and HDB to HDBSCAN. Sil refers to the Silhouette Score, DBI to the Davies-Bouldin Index, and CHI to the Calinski-Harabasz Index. EffNV2 refers to EfficientNet V2, DV2 Refers to DINO V2, and RN152 refers to ResNet-152. The best results for each clustering metric are highlighted in bold.

Dim. Red.	Feature	DB			AP			HDB		
		Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑
t-SNE	**ViT**	0.38	1.47	871.44	0.39	0.83	3585.39	0.55	0.53	1318.28
	VGG19	0.24	1.81	716.95	0.38	0.84	3942.43	0.59	0.50	1468.68
	EffNV2	-0.20	3.98	150.09	0.39	0.80	2975.40	0.47	0.59	587.04
	DV2	0.13	1.93	771.98	0.39	0.84	3451.50	0.42	0.66	449.48
	RN152	0.26	1.43	788.99	0.40	0.84	3853.87	0.60	0.48	1598.95
PCA	**ViT**	0.35	1.47	419.92	0.35	0.82	2385.68	-0.09	1.46	47.37
	VGG19	0.67	1.06	2456.64	0.38	0.83	5288.42	0.61	0.77	2609.53
	EffNV2	0.53	1.27	610.65	0.35	0.81	6031.40	0.42	0.54	621.40
	DV2	0.45	2.40	351.76	0.34	0.82	2628.79	0.56	1.47	1128.10
	RN152	0.33	1.52	259.51	0.36	0.86	4320.09	0.40	0.58	263.06
UMAP	**ViT**	0.02	1.87	2661.48	0.54	0.62	18748.19	0.81	0.24	6815.97
	VGG19	0.78	0.25	7969.78	0.49	0.75	21438.78	0.78	0.25	7969.78
	EffNV2	0.11	1.99	2304.33	0.42	0.74	8417.38	0.71	0.30	1976.69
	DV2	0.45	2.90	9513.53	0.51	0.64	23417.57	0.83	0.22	14727.08
	RN152	0.40	1.20	7392.90	0.61	0.54	**26025.36**	**0.86**	**0.17**	11575.26

Table 6.2 – Results for the Betweenness Centrality graph measurement.

Dim. Reduction	Feature	DB			AP			HDB		
		Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑
t-SNE	**ViT**	0.14	3.04	289.52	0.37	0.83	2411.54	0.05	2.22	294.94
	VGG19	0.06	1.86	336.48	0.37	0.85	2859.06	0.50	0.56	667.61
	EffNV2	-0.09	4.75	229.29	0.40	0.78	2690.83	0.49	0.56	686.03
	DV2	-0.02	0.84	294.97	0.37	0.84	2307.75	0.53	0.51	873.56
	RN152	0.14	1.15	593.30	0.37	0.83	2597.58	0.58	0.44	1158.92
PCA	**ViT**	0.08	3.81	19.26	0.34	0.82	1530.34	0.06	2.42	27.63
	VGG19	0.31	1.29	511.30	0.35	0.87	3157.44	0.59	0.44	1016.65
	EffNV2	0.35	3.78	372.51	0.35	0.82	3677.16	0.50	0.91	546.56
	DV2	0.25	2.52	300.10	0.34	0.82	1851.45	0.57	0.54	938.41
	RN152	0.45	1.15	353.75	0.33	0.89	2052.36	0.44	1.08	536.63
UMAP	**ViT**	-0.18	3.23	389.46	0.40	0.79	3273.87	0.50	0.50	1020.22
	VGG19	0.80	0.20	5161.03	0.41	0.78	**12129.37**	0.80	0.20	5161.03
	EffNV2	0.18	0.72	1934.74	0.44	0.72	10722.81	0.78	0.22	4075.77
	DV2	0.04	4.19	3984.03	0.40	0.79	11557.96	**0.84**	**0.17**	7968.57
	RN152	0.24	1.13	2913.28	0.40	0.80	10730.22	0.82	**0.17**	7008.48

results. The feature extraction with DINO V2, combined with HDBSCAN and UMAP, provided the best Silhouette and DBI results. Also, ResNet-152, with the same combination of dimensionality reduction and clustering method, provided another highlight of DBI. The combination of VGG19, Affinity Propagation, and UMAP provided the best CHI results.

In Table 6.3, it is possible to observe the Eccentricity results. Again, UMAP,

Table 6.3 – Results for the Eccentricity graph measurement.

Dim. Reduction	Feature	DB			AP			HDB		
		Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑
t-SNE	**ViT**	0.26	1.68	590.12	0.43	0.73	3246.59	0.28	0.91	472.87
	VGG19	-0.04	1.97	317.50	0.36	0.85	2526.23	0.54	0.51	909.72
	EffNV2	0.23	1.28	710.39	0.45	0.75	3332.67	0.55	0.48	969.95
	DV2	0.22	0.94	907.42	0.42	0.77	3760.87	0.48	0.46	1359.18
	RN152	0.24	1.37	944.71	0.42	0.77	3139.80	0.53	0.51	861.16
PCA	**ViT**	0.34	0.83	243.90	0.36	0.80	1686.40	0.31	1.40	75.78
	VGG19	-0.00	3.11	159.74	0.36	0.83	2366.89	0.25	3.49	240.78
	EffNV2	0.46	1.68	828.80	0.37	0.82	3426.35	0.69	0.32	1915.09
	DV2	0.07	2.34	7.89	0.35	0.82	1426.87	0.06	3.70	6.28
	RN152	-0.17	5.75	3.24	0.36	0.79	1458.34	-0.16	1.52	21.37
UMAP	**ViT**	0.15	1.03	2938.09	0.49	0.68	10678.55	0.21	1.63	85.96
	VGG19	**0.76**	**0.25**	3569.40	0.42	0.81	8732.51	**0.76**	**0.25**	3569.40
	EffNV2	0.33	1.08	2240.56	0.50	0.66	**12214.62**	0.28	1.19	654.87
	DV2	0.16	1.89	2400.45	0.47	0.71	11689.61	0.65	0.30	3982.46
	RN152	0.47	1.04	2570.59	0.48	0.67	10524.53	0.26	2.06	418.53

Table 6.4 – Results for the Local Efficiency graph measurement.

Dim. Reduction	Feature	DB			AP			HDB		
		Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑
t-SNE	**ViT**	-0.05	0.72	736.54	0.37	0.81	3186.85	0.65	0.41	2880.62
	VGG19	0.26	8.54	728.70	0.39	0.84	2893.14	0.53	0.57	1457.96
	EffNV2	0.11	1.17	633.67	0.40	0.83	3017.94	0.52	0.49	926.99
	DV2	0.37	1.51	1481.70	0.41	0.77	3632.54	0.62	0.45	2230.55
	RN152	0.51	1.76	1326.92	0.41	0.81	3808.82	0.60	0.47	2045.86
PCA	**ViT**	0.36	1.57	646.01	0.35	0.83	2838.28	0.64	0.38	1528.72
	VGG19	0.17	1.48	72.50	0.35	0.83	1605.61	-0.13	3.96	47.79
	EffNV2	-0.18	3.44	3.35	0.34	0.81	2650.25	0.04	1.79	264.16
	DV2	0.38	1.88	15.68	0.34	0.82	1805.64	0.31	1.82	51.94
	RN152	0.45	2.13	818.71	0.35	0.86	2195.84	0.43	1.63	821.02
UMAP	**ViT**	0.29	0.51	8781.97	0.41	0.76	**17509.27**	**0.86**	**0.16**	17508.97
	VGG19	0.28	1.71	4527.87	0.45	0.72	12645.84	0.79	0.26	6502.94
	EffNV2	0.39	0.90	1546.30	0.43	0.77	7390.98	0.38	1.48	754.08
	DV2	-0.14	2.34	2505.65	0.45	0.73	11965.81	0.75	1.48	2171.72
	RN152	0.25	1.25	2807.86	0.51	0.68	13569.27	0.81	0.24	7629.58

HDBSCAN, and Affinity Propagation are highlighted. VGG19 with UMAP led to the best Silhouette and DBI results in HDBSCAN and DBSCAN. The equal metrics presented in both clustering methods are one of the few cases where the clustering configuration generated by both methods is the same. EfficientNet V2 with UMAP and Affinity Propagation are highlighted for the CHI.

Table 6.4 presents the Local Efficiency results, and the same patterns about the dimensionality reduction and clustering methods are observed again. Here, features obtained with Vision Transformers are highlights, where the combination of HDBSCAN and UMAP provides the best Silhouette and DBI results, and Affinity Propagation provides the best CHI results.

Table 6.5 shows the Vulnerability results, where similar highlights of Table 6.1 are

Table 6.5 – Results for the Vulnerability graph measurement.

Dim. Reduction	Feature	DB			AP			HDB		
		Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑	Sil ↑	DBI ↓	CHI ↑
t-SNE	**ViT**	-0.14	5.59	7.46	0.35	0.83	1666.91	-0.07	1.61	71.03
	VGG19	-0.10	1.10	216.85	0.37	0.82	2375.39	-0.09	1.36	216.91
	EffNV2	-0.27	1.81	181.06	0.41	0.81	2993.02	0.51	0.54	703.93
	DV2	0.03	1.75	267.53	0.38	0.81	2541.50	0.52	0.53	786.01
	RN152	0.29	1.79	850.53	0.38	0.80	2390.92	0.56	0.46	1131.73
PCA	**ViT**	0.20	2.29	20.43	0.34	0.81	1234.69	-0.20	1.81	14.60
	VGG19	0.11	3.66	24.72	0.32	0.88	1463.20	-0.15	1.62	28.90
	EffNV2	0.17	2.97	235.69	0.38	0.79	3429.41	-0.09	1.91	43.07
	DV2	0.42	2.13	62.07	0.33	0.81	1462.12	0.48	1.66	76.24
	RN152	0.39	1.29	51.17	0.35	0.84	1161.95	-0.21	1.98	13.45
UMAP	**ViT**	-0.08	1.19	6.78	0.41	0.75	2041.53	0.05	2.45	47.09
	VGG19	0.00	1.10	310.29	0.43	0.75	4343.54	0.60	0.38	1535.73
	EffNV2	-0.00	1.00	859.23	0.45	0.70	7608.01	0.49	0.83	707.47
	DV2	0.01	1.11	1038.10	0.42	0.77	6068.98	0.70	0.32	2069.85
	RN152	-0.02	2.13	1757.07	0.41	0.79	**10135.10**	**0.80**	**0.20**	5249.43

observed. UMAP combined with ResNet-152 generated features where the clustering with HDBSCAN provided the best Silhouette and DBI results and the clustering with Affinity Propagation provided the best CHI results.

The results show that the UMAP dimensionality reduction provided far superior results to the other two methods. Affinity Propagation Clustering led to effective clustering results when considering the Calinsky-Harabasz index, while HDBSCAN performed best for the Silhouette Score and Daves-Bouldin index. This indicates that the clusters obtained in HDBSCAN are compact and well-spread, i.e., samples fit the cluster to which they belong. It is also worth noting that no feature extractor is suitable for all scenarios. Using ResNet-152 led to the best results observed for two different graph measurements (e.g., Average Path Length and Vulnerability), while Vision Transformers provided the best results for Local Efficiency.

6.3.2 Qualitative Results

We also performed a qualitative analysis based on sample distributions. We computed three sets of boxplot graphs with measurements relative to the matches by cluster. The objective is to characterize clusters considering the intrinsic properties of their samples, i.e., the properties of the ball possessions. We present the statistics of the number of passes, the number of dribbles, the percentage of times that the ball reached the final third of the field, the attack duration, total distance traveled by the ball in a ball possession, maximum, minimum and range spread, and sample entropy.

We selected three clustering configurations based on the resulting number of clusters and the numerical results presented in the previous section. Wilcoxon statistical tests [209] were performed between them with $p < 0.05$. A ball possession projection for raw data

and clustered data is also presented for each set of graphs.

In Figure 6.5, statistics of the cluster are generated by the combination of Betweenness Centrality, DINO v2, UMAP, and HDBSCAN. This configuration led to clustering the dataset into two groups, as shown in Figure 6.5(b). Figure 6.5(a) presents the data projection, where data is separated into two groups. Observing Figures 6.5(c) and (d), we can notice the biggest differences between the two clusters are the duration of the ball possessions, the number of passes, and the distance traveled by the ball, where the cluster 1 (black) presents a higher mean number of passes, longer ball possessions and higher distance traveled by the ball, indicating more well-elaborated ball possessions. Meanwhile, in cluster 0 (blue), these statistics are lower and may indicate counterattacks or failures. Wilcoxon statistical tests between this clustering configuration and the ones presented in Figures 6.6 and 6.7 present p with values $1.81e-154$ and $2.17e-204$, respectively.

Figure 6.6 presents the results from the combination of Local Efficiency, ResNet 152, UMAP, and HDBSCAN. This configuration led to three clusters, as seen in Figure 6.6(b). Figure 6.6(a) shows the data points placed in the two-dimensional space already separated into three groups. This clustering configuration also distinguishes the clusters by the duration of the ball possessions, the number of passes, and the distance traveled by the ball. Cluster 0 (blue) presents more elaborate plays, with higher statistics, while cluster 1 (black) presents intermediate values, and cluster 2 (red), lower values, probably indicating less elaborate plays or unsuccessful attacks. Statistical tests between this clustering configuration and the one previously discussed led to $p = 1.81e-154$, while the test between this configuration and the one presented in Figure 6.7 led to $p = 7.56e-120$.

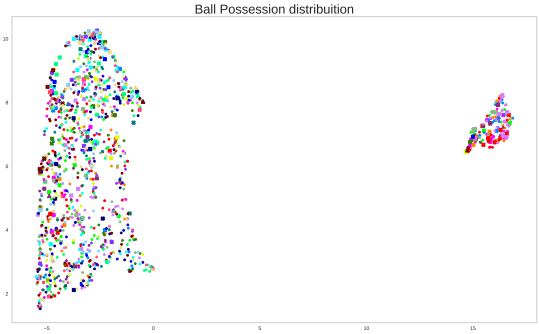
Those cluster configurations have been associated with high-quality produced clusters in terms of their compatibility and separability presented in Tables 6.2 and 6.4. However, properties of produced clusters did not relate to tactical behaviors (e.g., the spread and sample entropy measures) in the ball possessions except for minimum spread. The results suggest that clusters can be characterized based on the length of the ball possession, i.e., produced clusters are associated with either shorter sequences or longer ones.

In Figure 6.7, otherwise, the presented results also demonstrate phenomena of tactical behaviors associated with the offensive sequence length. This clustering configuration, generated by the combination of Average Path Length, ResNet 152, UMAP, and Affinity Propagation, provided a good distinction between the ball possessions, and we can draw attention to clusters that present distinguished behaviors, enabling a clear performance analysis. Data was separated into six clusters (Figure 6.7(b)) and we can observe that data was initially separated into three distinctive groups (Figure 6.7(a)). Clusters 2 and 4 (in red and yellow) present ball possessions with a lower number of passes, also a smaller length, a smaller distance traveled by the ball, and the plays reached

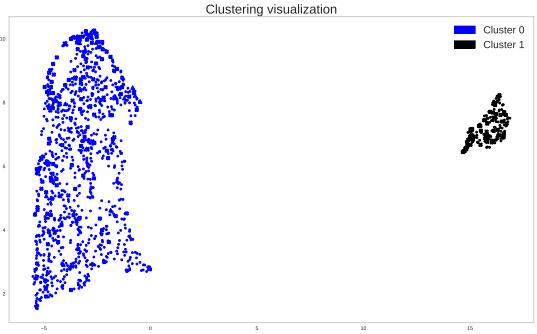
the final third of the field fewer times, indicating counterattacks or small unsuccessful plays. On the other hand, cluster 5 (in purple) presents very distinctive ball possessions, with a higher number of passes, longer attacks, and bigger distance traveled by the ball, which translates into a tactical behavior where the team is more spread on the pitch, and there is more variability and unpredictability during the offensive sequences, which can be observed for the Sample Entropy. It is also noteworthy the high percentage of reaching the final third of the field. This cluster translates to more elaborate plays with higher variability of tactical behavior. Wilcoxon tests between this configuration and the ones in Figure 6.5 and Figure 6.6 led to, respectively, $p = 2.17e - 204$ and $p = 7.56e - 120$.

6.4 Final Remarks

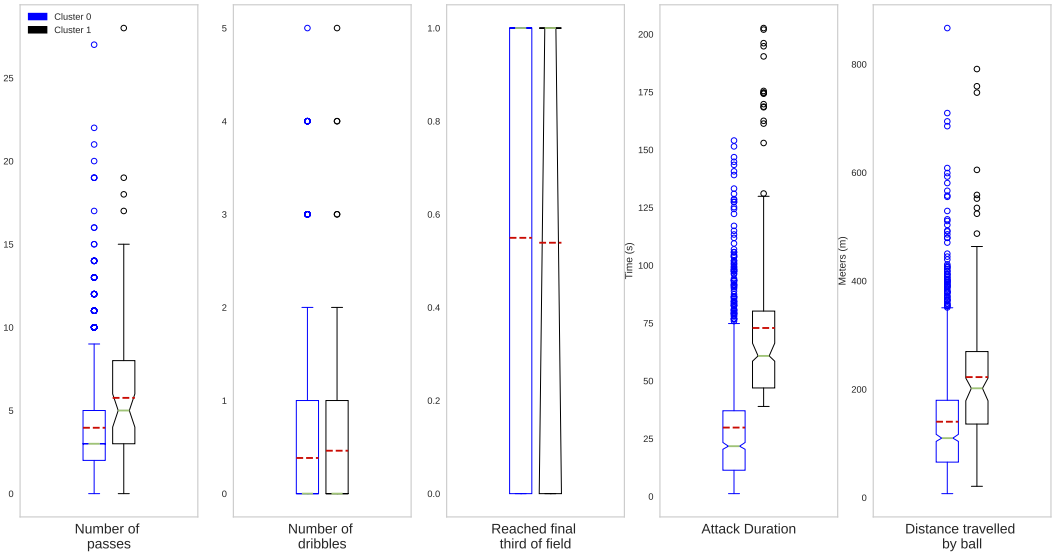
This work introduces a spatiotemporal methodology for domain-specific analysis by using temporal graphs and time segmentation criteria. The proposal was validated in a football analysis context, focusing on ball possession clustering based on temporal graphs' data representation. Our findings indicate that a proper combination of methods leads to satisfactory computational results and presents a clear distinction between tactical phenomena, allowing the study of ball possessions with specific patterns. Data characteristics such as team spread, duration of ball possession, players' position in the field, e.g., players in the final third of the field, associated with patterns obtained in the feature extraction step, used in the clustering algorithms, are useful tools for studying successful and unsuccessful attacks, counterattacks, among other events during a football match. These information can be used for applications such as video retrieval, for obtaining match highlights and for study purposes.



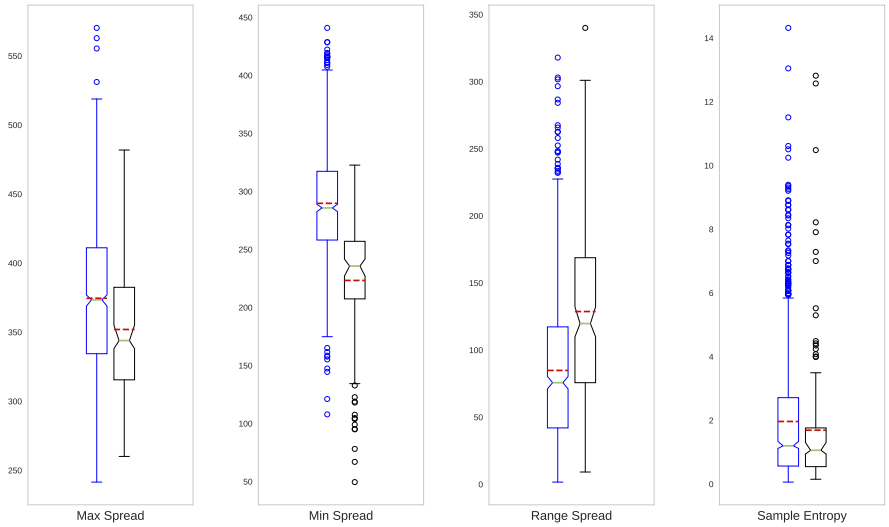
(a) Ball Possession Projection



(b) Clustering

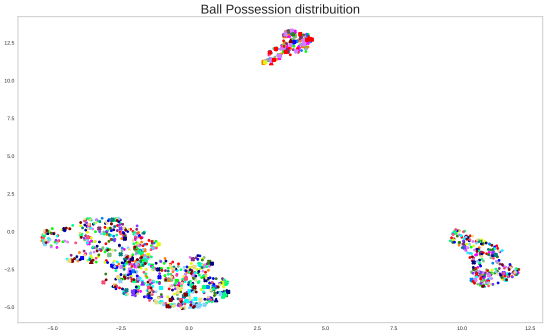


(c) Statistics for produced clusters

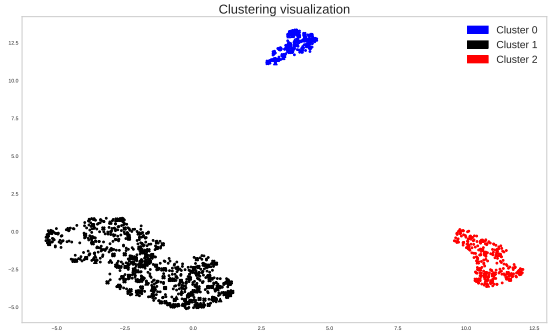


(d) Statistics for produced clusters

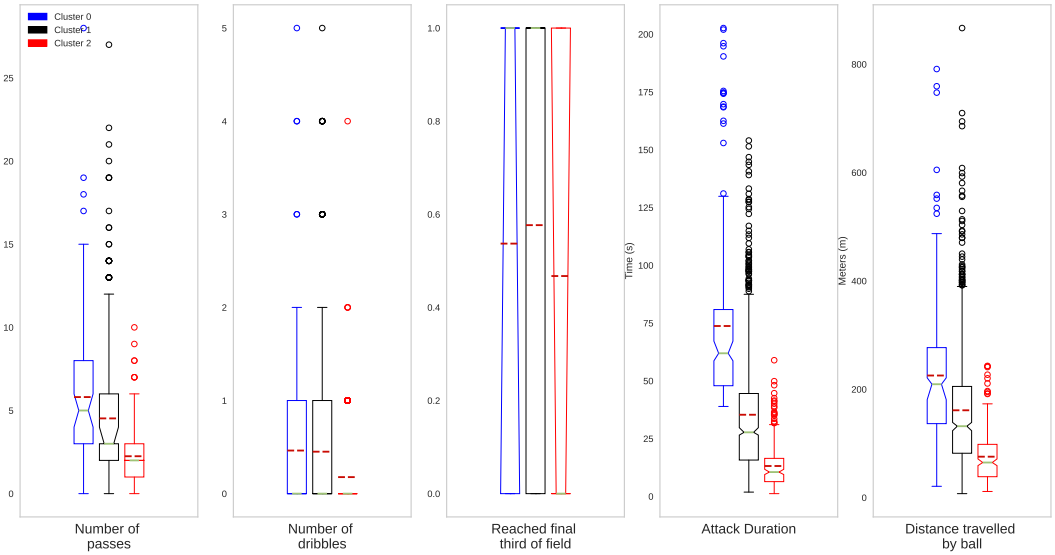
Figure 6.5 – Statistics for clusters created based on the configuration, including the use of Betweenness Centrality, DINO V2, UMAP, and HDBSCAN.



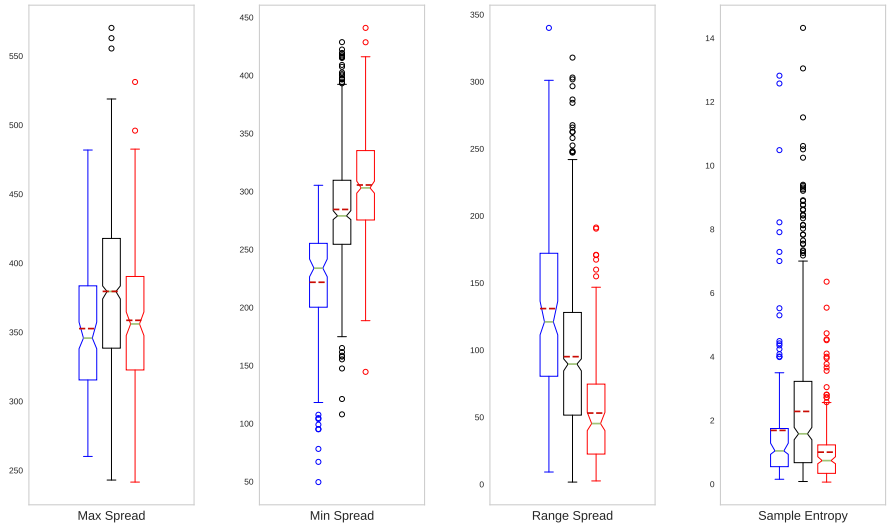
(a) Ball Possession Projection



(b) Clustering

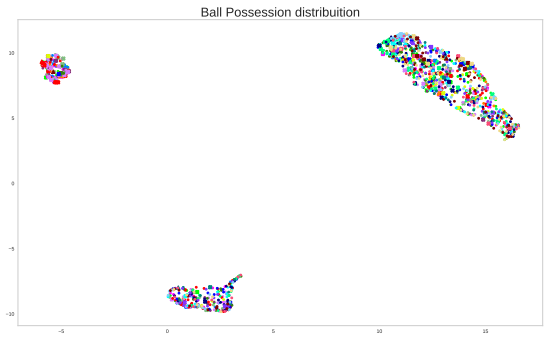


(c) Statistics for produced clusters

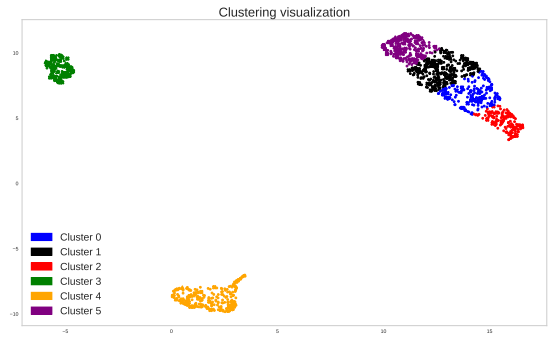


(d) Statistics for produced clusters

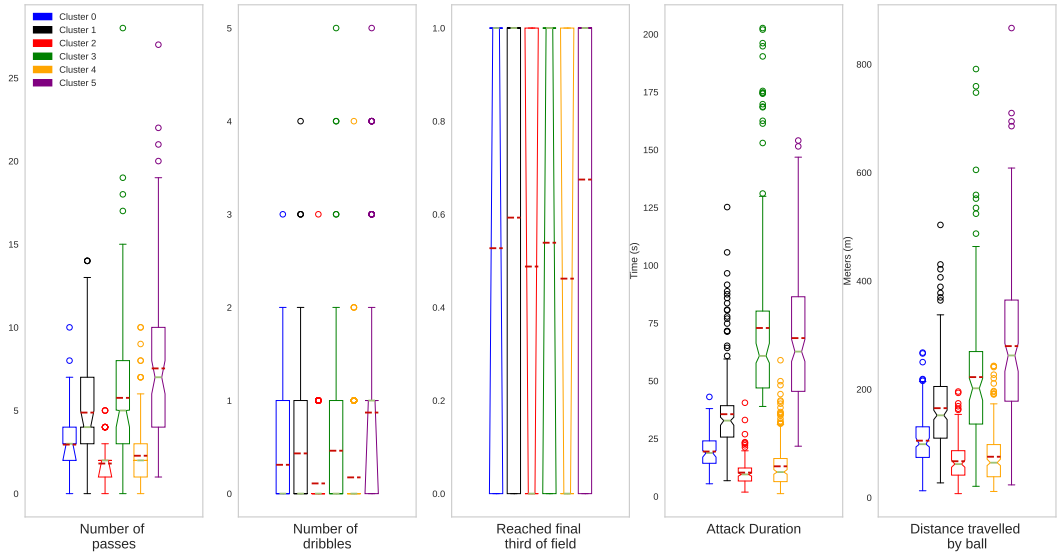
Figure 6.6 – Statistics for clusters created based on the configuration, including using Local Efficiency, ResNet-152, UMAP, and HDBSCAN.



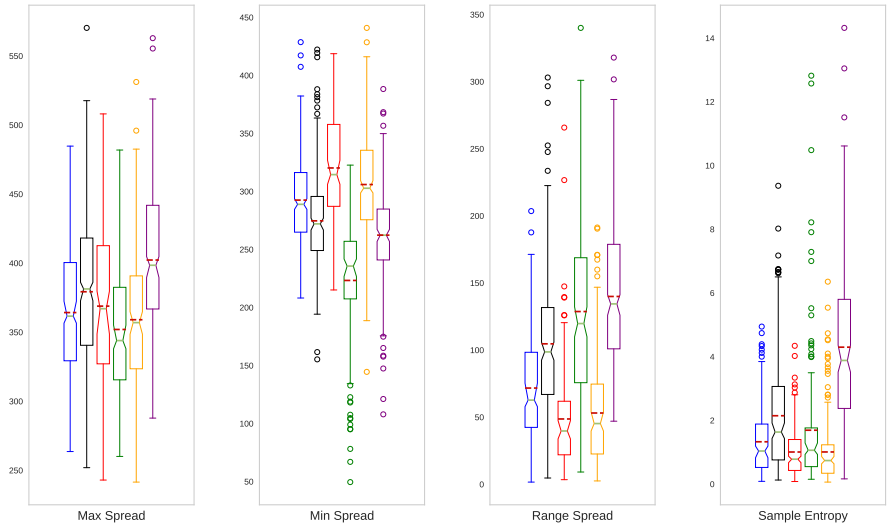
(a) Ball Possession Projection



(b) Clustering



(c) Statistics for produced clusters



(d) Statistics for produced clusters

Figure 6.7 – Statistics for clusters created based on the configuration, including the use of Average Path Lenght, ResNet-152, UMAP, and Affinity Propagation.

7 Semi-Supervised Time Series Classification through Image Representations

One of the greatest challenges nowadays is dealing with great amounts of available data. While vast quantities of data are available, labeling it, on the other hand, is laborious, expensive, and time-consuming. In particular, for time series, finding annotations for data can be difficult due to the necessity of a professional or the excessive amount of instances in a dataset [207]. In these scenarios, algorithms that deal with reduced amounts of labeled data emerge. One example is Semi-Supervised Learning (SSL), which can explore labeled and unlabeled data for tasks such as classification. This work uses a kNN graph-based transductive SSL approach for time series classification. A feature extraction step, based on imaging time series and obtaining features using deep neural networks, is performed before the classification step. An extensive evaluation is conducted over four datasets, and a parametric analysis of the nearest neighbors is performed. Also, a statistical analysis of the distances obtained is undertaken. Results suggest that our methods are suitable for classification and competitive with supervised and unsupervised baselines in some datasets.

Our main contributions are:

- Studying different approaches for imaging of time series, extracting features from generated images using neural networks originally trained on a large image dataset, and then using the obtained features to construct a similarity graph for SSL.
- Our classification results show that the proposed methods suit time series classification. We also observe that the imaging method impacts results more meaningfully than the feature extraction methods.
- A statistical analysis of distances and similarities generated via the proposed method to understand classification results.

We have an overview of the work in Section 7.1, a theoretical foundation for semi-supervised classification is given in Section 7.2, the experimental protocol is defined in Section 7.3, and the results are discussed in Section 7.4.

7.1 Overview

Semi-supervised learning methods are essential in machine learning, considering the increasing amount of data available nowadays and the scarcity of labels. Our approach

is based on applying a Transductive Semi-Supervised Classification model to four time series datasets, considering the raw representation of data and the representation based on images. Transductive learning is well applied in domain-specific tasks, where generalization is not necessary, unlike inductive settings [173]. In this work, labeling is performed based on specific examples using the Label Propagation algorithm.

The proposed methodology is represented in Figure 7.1 and explained below. Gramian Angular Fields, Markov Transition Fields, and Recurrence Plots were utilized for imaging time series, as presented in step 1. The feature extraction of the images, in step 2, was performed using CNN ResNet-152 and Vision Transformers, described in Section 2.5.3, with both models pre-trained with the ImageNet dataset. The process of imaging time series is detailed in Section 2.5.2. A similarity graph based on the Euclidean Distance between the features of the elements is then, built (steps 3 and 4), and used for Semi-Supervised Classification with Label Propagation, in step 5. The obtained results were competitive with SOTA and suggest that our approach is suitable for time series classification. This work was first published in the *23rd International Conference on Computational Science and Its Applications (ICCSA 2023) by Springer Nature, pgs. 48–65* [163], reproduced with permission from Springer Nature.

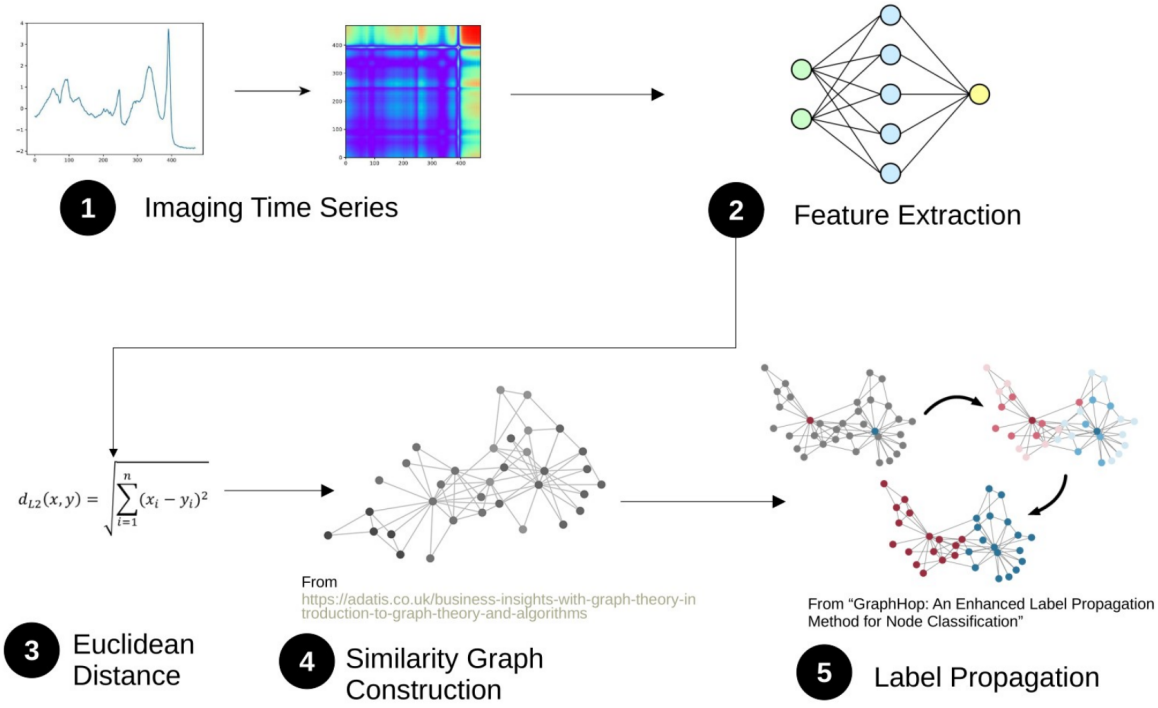


Figure 7.1 – Proposed methodology for time series semi-supervised classification

7.2 Semi-Supervised Classification

The semi-supervised classification problem involves determining labels for a dataset with a few labeled and many unlabeled samples. The transductive setting aims to label the entire dataset, while the inductive setting learns a classification function [197]. We use a graph-based Transductive Semi-Supervised Learning (SSL) algorithm with two steps: graph construction and inference [197]. The graph is constructed to represent class relations between similar nodes. Commonly, Gaussian similarity is used to weigh the graph edges based on the Euclidean distance [44, 197]. The Gaussian Similarity is expressed in Equation 7.1:

$$w_{g_{i,j}} = \exp\left\{\frac{-D^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right\}, \tag{7.1}$$

where D is the Euclidean Distance. After the weighting, a sparsification procedure is performed, where the process begins with a fully connected weighted graph, and then edges between less similar nodes are removed. The utilized approach is to keep only the $k-$nearest neighbors connected, with $k = \log_2 N$ [44, 18]. In this case, the parameter σ is set as defined in Equation 7.2 [84]:

$$\sigma = \frac{1}{3n}\sum_{i=1}^{n} D(\mathbf{x}_i, \mathbf{x}_{i_k}), \tag{7.2}$$

where i_k is the $k-$nearest neighbor of the $i-$th element. Label propagation is applied on the graph to classify unlabeled elements using information from labeled ones. The well-established Gaussian Random Fields (GRF) [230, 197] method is commonly used for label propagation, where probabilities of nodes belonging to different classes are updated iteratively based on their neighbors' probabilities. The optimization process minimizes a cost function H_f under the constraints of known labels for labeled nodes. The function H_f is defined in Equation 7.3

$$H_f = -\sum_{i<j} w_{g_{i,j}} \sum_{s} \psi_{i,s}\psi_{j,s}, \tag{7.3}$$

where $\psi_{i,s}$ represents the probability of the $i-$th node belonging to the $s-$th class, and $w_{g_{i,j}}$ is the pairwise similarity among nodes. The objective is to minimize H_f while satisfying constraints on the probabilities of labeled nodes. If $i \leq L_e$, the probabilities are fixed to known labels.

$$\psi_{i,s} = \begin{cases} 1, & \text{if } s = y_i \\ 0, & \text{otherwise.} \end{cases} \tag{7.4}$$

The optimization process involves a propagation process on graph G, where each probability at iteration t is updated as a weighted average of its neighbors in the previous iteration, as in Equation 7.5.

$$\psi_{i,s}^{(t+1)} \propto \sum_{j \neq i} w_{g_{i,j}} \psi_{j,s}^{(t)} \tag{7.5}$$

Then, the individual probabilities are normalized to sum to 1. The described method can be iterated until reaching a maximum number of iterations t_{max} or an upper bound e for the difference between consecutive iterations. After convergence, an unknown label $\hat{y}_i$ (with $i > L_e$) is determined based on the highest probability, as defined in Equation 7.6.

$$\hat{y}_i = arg \max_s \psi_{i,s}. \tag{7.6}$$

7.3 Evaluation

This section outlines the experimental protocol for semi-supervised time series classification. The datasets utilized in the experiments are CBF, ECG5000, Yoga, and Electric Devices, described in Section 2.7.

7.3.1 Experimental Protocol

The process of time series imaging was performed using Python 3.10, applying the methods described in Section 2.5.2, and maintaining the proposed parameters from the pyts library [58]. Images were generated and saved using the matplotlib library [79]. The GAF and MTF methods used a 'rainbow' colormap, while the Recurrence Plot used a 'binary' colormap. The generated images were fed into the CNN ResNet-152 and ViT architectures, outlined in Section 2.5.3, for feature extraction. Subsequently, the Euclidean distance was computed among pairs of instances to construct the similarity matrix for label propagation. The inference step used $t_{max} = 10^3$ and $e = 10^{-3}$. To evaluate the methods, each dataset $\mathcal{D}$ was split into a labeled set $\mathcal{D}_l$ and an unlabeled set $\mathcal{D}_u$, ensuring that the second always contains representatives of each class. This procedure was repeated for different sizes of $\mathcal{D}_u$, and the classification experiments were averaged over ten equally sized splits of $\mathcal{D}$. The effect of the number of neighbors on classification results was evaluated by varying k under a fixed size for $\mathcal{D}_u$ and then averaging oversplits.

The accuracy and adjusted mutual information (AMI) measurements, both defined in Section 2.8, were used to assess the classification results, following the experimental protocol proposed in [18]. The first is a traditional classification metric, and the second is a clustering measurement. Using AMI in this context indicates the correlation between

classification and clustering in Semi-Supervised learning tasks, where labeled and unlabeled information are utilized.

Comparisons with previous works utilized supervised classification methods as baselines, particularly those from [92] and [112]. The accuracy results were then compared with the two best methods on each dataset. Also, the AMI baseline was obtained by clustering data using the combination of DTW distance measure (Section 2.1) and HDBSCAN clustering (Section 6.1.6).

Additionally, a statistical analysis was conducted on the extracted features to compare feature extraction methods to understand the distribution of pairwise distances and similarities and their coefficient of variation (defined as the standard deviation divided by the mean).

7.4 Results and discussion

The Section presents the empirical evaluation of various feature extraction methods in semi-supervised classification. It discusses the performance of these methods and their suitability for label propagation. The Section then introduces a statistical analysis of pairwise distances to understand the distinctions between the evaluated feature extraction methods.

7.4.1 Classification results

The results demonstrate that the proposed methods for feature extraction are well-suited for semi-supervised classification using label propagation. Accuracy and adjusted mutual information (AMI) show a strong relationship in all studied datasets, indicating a connection between clustering and classification.

The Electric Devices dataset yields intriguing results, as shown in Figure 7.2. While supervised baselines trained with a high rate of labeled data achieve high accuracy, some proposed methods are not far behind, even with a lower labeled data ratio. Notably, GAF and RP outperform methods using LSTM+FCN in standard splits when $r_l > 0.06$. Additionally, using raw data leads to the worst results for this dataset, which suggests that the dataset may differ from others and provide an opportunity for improvement. AMI results easily outperform our unsupervised baseline.

Except for Electric Devices, raw data methods consistently outperform methods that use features extracted via neural networks. This observation may be related to a Gaussian property of CBF, ECG5000, and Yoga datasets. However, further investigation is required to confirm this hypothesis.

Comparison with previous works, in Figure 7.3, reveals that our methods fall

slightly short of achieving the same accuracy as neural network-based and BOSS methods on the CBF dataset, while except GAF and MTF associated with ViT, our method outperforms the HDBSCAN. On the ECG5000 dataset, our methods perform almost as well as LSTM+FCN, and except for MTF-based features, surpass the unsupervised baseline, as shown in Figure 7.4. Yoga is the most challenging dataset, with our methods requiring a higher labeled data ratio to achieve competitive accuracy. Meanwhile, our results easily outperform the clustering baseline. Figure 7.5 presents the results.

Methods relying on MTF for time series imaging tend to be less accurate than those using GAF and RP. The underlying reason for this behavior remains unclear and warrants further investigation. On the other hand, in most cases, our methods surpass the unsupervised baseline, which suggests that the information contained in the unlabeled data alone is insufficient to label the elements in the proposed datasets.

The role of graph construction on classification results is explored, and there is evidence of a plateau in performance metrics as the number of nearest neighbors increases, as shown in Figure 7.6. The proposed feature extraction approach significantly contributes to Electric Devices' performance, confirming that imaging time series can provide a more discriminative data representation among classes.

Choosing the value of k in the method affects computational complexity. Based on experimental results, the initial choice of $k = \log_2 N$ appears to strike a good balance between speed and accuracy.

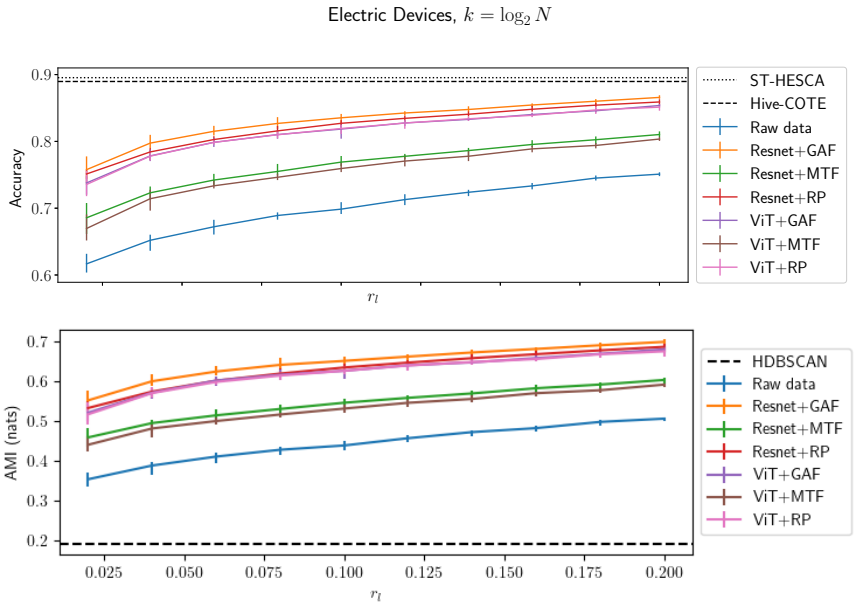


Figure 7.2 – Results for the ElectricDevices dataset as a function of r_l. Supervised baselines were obtained from 8926 labeled points, corresponding to $r_l \approx 0.54$.

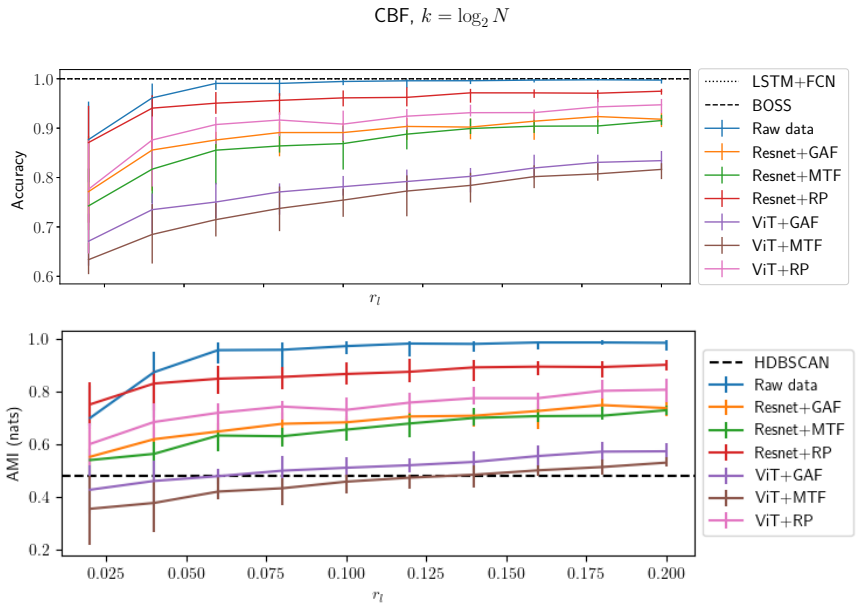


Figure 7.3 – Results for the CBF dataset as functions of the number of nearest neighbors k (up) and the rate of labeled data r_l (down). Supervised baselines were obtained with initially 30 labeled points, corresponding to $r_l \approx 0.03$.

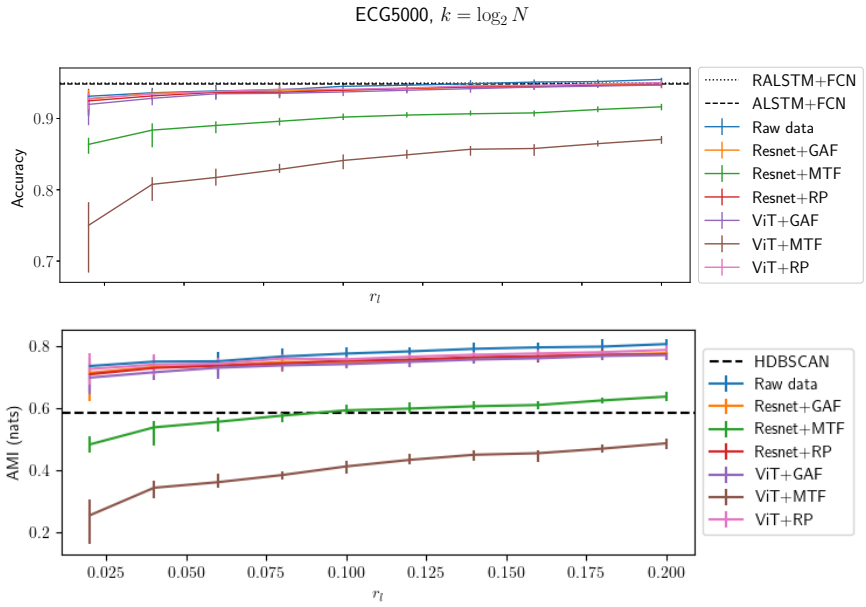


Figure 7.4 – Results for the ECG5000 dataset as a function of r_l. Supervised baselines were obtained with initially 500 labeled points, corresponding to $r_l = 0.1$.

7.4.2 Statistical analysis of pairwise distances and similarities

The analysis focuses on the distribution of pairwise distances and similarities for the CBF, ECG5000, and Yoga datasets, as the Electric Devices dataset's size makes it impractical for this analysis.

Figure 7.7 presents kernel density estimation of the distribution of distances, revealing that features extracted using ViT networks yield smaller pairwise distances than

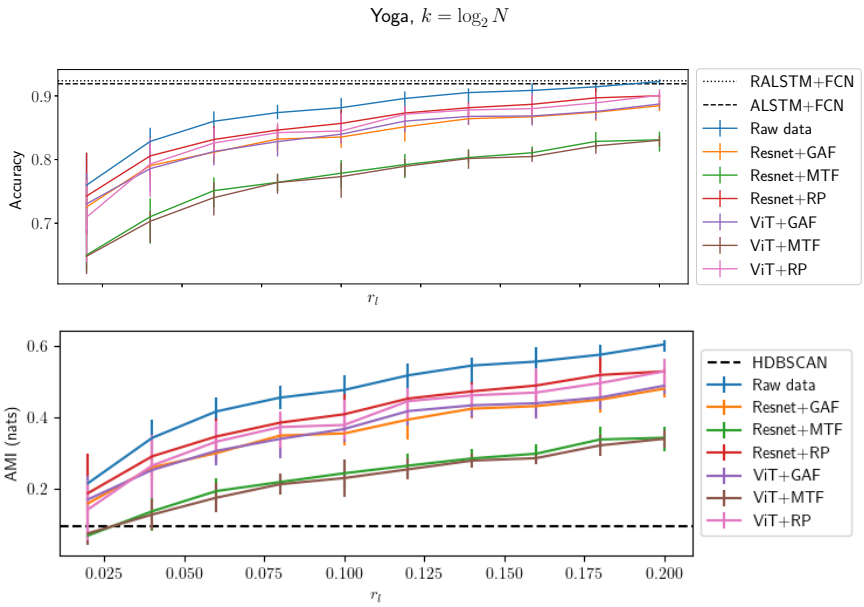


Figure 7.5 – Results for the Yoga dataset as a function of r_l. Supervised baselines were obtained with initially 300 labeled points, corresponding to $r_l \approx 0.09$.

other methods.

Notably, the distribution of pairwise distances for the most accurate methods does not resemble a Gaussian. Some distributions even exhibit a double-peaked pattern, suggesting a separation between intraclass and outer class distances. This phenomenon is observed with raw data in ECG5000 and Resnet+RP in CBF.

Furthermore, combining imaging with feature extraction from neural networks tends to smooth the distribution of distances. However, ResNet and ViT achieve this smoothing differently, with ViT producing smaller distances.

The coefficient of variation of similarities is analyzed, revealing that the distribution for raw data in CBF exhibits the lowest variation. This and previous analyses suggest that similarities may reflect intraclass proximity in this dataset.

On the other hand, for the Yoga dataset, similarities calculated from MTF imaging have a lower variation. Similarly, ECG5000 and Yoga are the datasets where MTF performs poorly for imaging. The single-peaked distribution of their kernel densities implies that the pairwise distances of these methods suffer from an indistinguishability phenomenon.

7.5 Final Remarks

This study investigated the application of imaging time series followed by feature extraction from pre-trained deep neural networks for semi-supervised classification. The results suggest that this approach is effective for time series classification, showing competitive performance on most datasets.

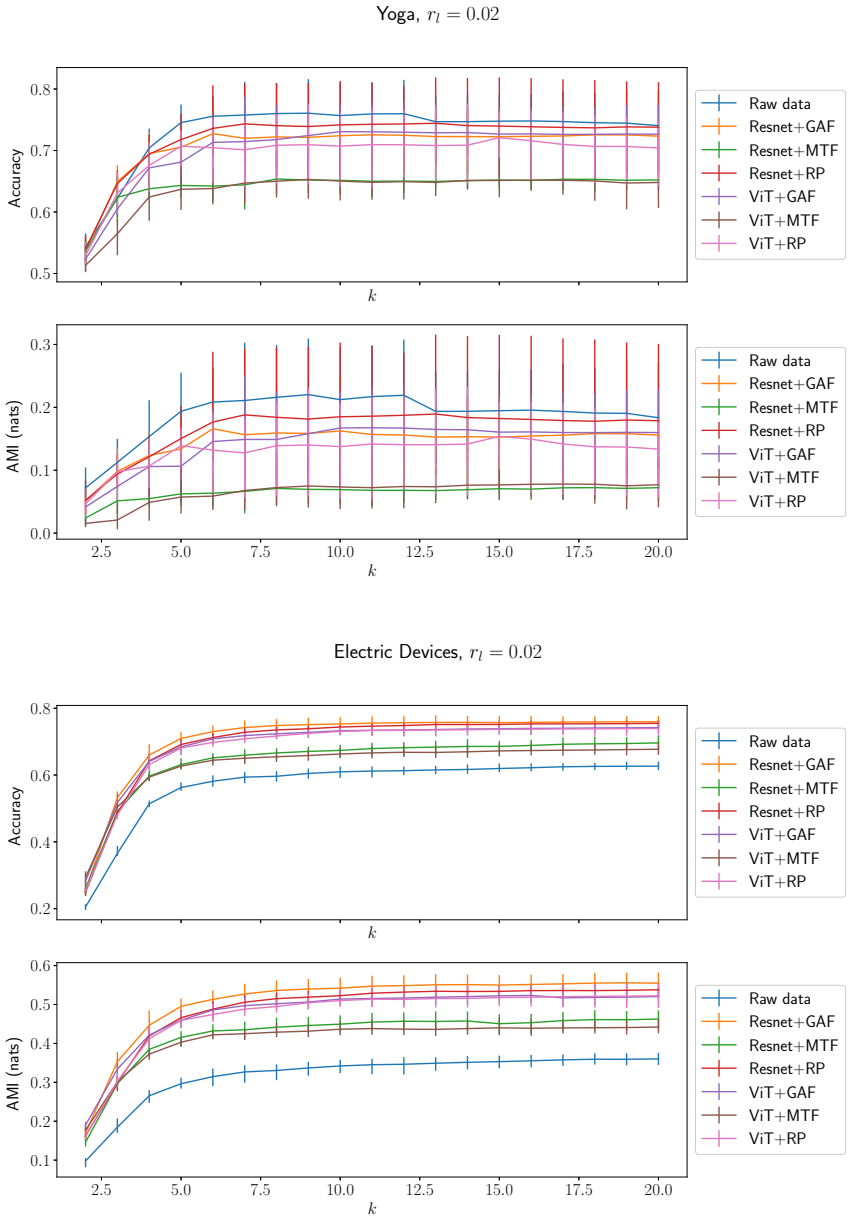


Figure 7.6 – Classification results as a function of k for the Yoga (up) and Electric Devices (down) datasets.

However, the empirical evaluation raised several questions. Using raw data instead of imaging for datasets CBF, ECG5000, and Yoga proved to be the most effective approach. On the other hand, imaging with GAF and RP, followed by MTF, provided the best results for Electric Devices.

Interestingly, the imaging methods significantly impacted classification results more than the choice of neural network architectures. Among the imaging methods, MTF performed poorly compared to GAF and RP.

The statistical analysis of pairwise distances and similarities provided further insights into the classification results. It revealed interesting properties of features extracted

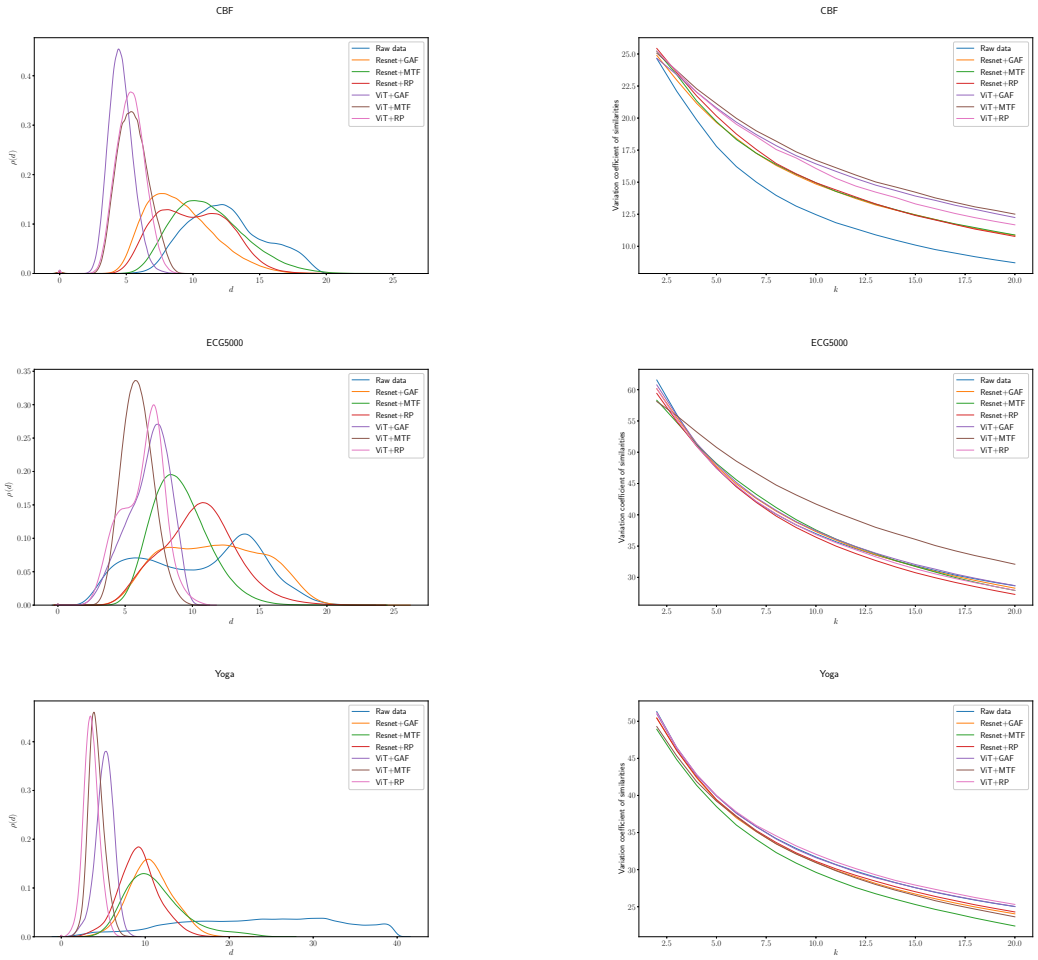


Figure 7.7 – Kernel density estimation of pairwise distances (left) and coefficient of variation of pairwise similarities as a function of k (right) for CBF, ECG5000, and Yoga.

from neural networks, such as Vision Transformers producing more homogeneous and smaller distances.

8 Conclusions

This Chapter presents the final considerations of this work. Four approaches for time series analysis were proposed, two for univariate series and two for multivariate series, exploiting, in all scenarios, different strategies for feature extraction and highlighting the impact of the time series representation method in the final result. Two approaches use contextual similarity measures, one for univariate and one for multivariate time series. Supervised, semi-supervised and unsupervised training were considered, approaching classification, retrieval and clustering for time series analysis.

Chapter 4 presents results for univariate time series retrieval, conducting experiments for diverse feature extractors and re-ranking methods, and presenting positive mAP, precision and recall results, while Chapter 7 explores label propagation for univariate time series semi-supervised classification, investigating the impact of imaging methods in this task, and comparing results with SOTA methods, presenting interesting results and valuable insights.

Two proposals in this work approached multivariate time series. In Chapter 5, rank aggregation methods were used to create a distance matrix for multivariate time series datasets, considering features extracted from each dimension of the series, individually. Retrieval and classification results were evaluated, presenting positive outcomes. Chapter 6 proposes an unsupervised framework for data analysis based on time segmentation criteria, where the processed data originates a multivariate time series dataset. Image-based representations, dimensionality reduction and clustering methods were explored, and results were mainly evaluated based on domain-specific variables. This framework was validated considering ball possessions from football matches and shows promising results.

The rest of the Chapter is organized as follows: Section 8.1 discusses the relationships between research questions and contributions of this work, while Section 8.3 discusses possible future works.

8.1 Contributions and Research Questions

This section interconnects the contributions and research questions. Below, we present a discussion:

1. *How can we extract meaningful features for a required task?*

   Time series encodes diverse information and many methods can be used to extract and explore them. All of the proposals in this work explore the feature extraction in

time series, with meaningful results in the proposed tasks.

2. *How the information contained in the time series image representations can be explored?*

   Again, this question was addressed in all Chapters, and the information contained in the images was explored using feature extraction techniques, with diverse neural network models.

3. *Which time series features are more suitable for a machine learning or information retrieval task?*

   In all Chapters, this challenge was addressed, and there is no consensus about the best feature extractor for time series that is suitable for every task. It is dependent on: *(i)* the application and *(ii)* dataset's characteristics.

4. *How to find distance measurements that translate effectively the relationship between the data points of time series datasets and how to explore more global relationships between time series to perform information retrieval?* Traditional distance metrics often measure only pairwise relationships, ignoring the neighborhood and, consequently, contextual information in a dataset. Exploring these relationships helps to rearrange the dataset and to find better ranks, as it was made in Chapters 4 and 5.

5. *How to effectively explore information encoded in the dimensions of a time series?*

   In Chapter 5, we propose processing each dimension of a multivariate time series individually and merging similarity information through using contextual rank aggregation techniques. It ensures that information on each dimension is considered in the evaluation.

6. *How to properly interpret the clustering results of an unlabeled time series dataset?*

   Often, clustering results are evaluated using extrinsic methods, and intrinsic methods may not give valuable insights into the interpretability of results. In Chapter 6, we propose a clustering-based framework for data analysis considering time segmentation criteria. It transforms the problem into a time series clustering task. This, associated with domain-specific variables, helps with the interpretability of results.

7. *How to explore supervised and unsupervised information in time series data properly?*

   Chapter 7 proposes the representation of time series using imaging methods associated with feature extraction using Neural Networks and semi-supervised classification using the well-known label propagation, assuring both labeled and unlabeled information is explored in a graph configuration.

8.2 Publications and International Fellowship

During the period of the master, four publications and submissions were produced, where the student is the first author. This section lists all publications, submissions, and works under review submitted to both conferences and international scientific journals. The publications, associated with Chapters 4 through 7 of this text, are:

1. **B. Rozin,** E. Bergamim, D. C. G. Pedronette, and F. A. Breve, "Semi-supervised time series classification through image representations," in International Conference on Computational Science and Its Applications – ICCSA 2023, O. Gervasi, B. Murgante, D. Taniar, B. O. Apduhan, A. C. Braga, C. Garau, and A. Stratigea, Eds. Cham: Springer Nature Switzerland, 2023, pp. 48–65 [163].

   Status: Published

2. **B. Rozin,** D. C. G. Pedronette, and R. S. Torres, "Re-Ranking and Representations for Time Series Retrieval: A Comparative Study" [165].

   Status: Submitted

3. **B. Rozin,** and D. C. G. Pedronette, "A Ranked-Based Framework based on Manifold Learning for Multivariate Time Series Retrieval and Classification" [162].

   Status: Submitted

4. **B. Rozin,** R. S. Torres, F. A. Moura, and D. C. G. Pedronette, "Ball Possession Analysis based on Temporal Network Properties" [Rozin et al.].

   Status: In Submission

- **International Fellowship**

During the course of the masters, the student received a "*Bolsa Estágio de Pesquisa no Exterior (BEPE)*", conducted under the supervision of:

- Professor Ricardo da Silva Torres from Wageningen University & Research (WUR) (Wageningen, The Netherlands), which contributed to the research projects involving Chapters 4 and 6.

The fellowship covered a 3-month stay in Wageningen from October to December 2023 to conduct the research at WUR.

8.3 Future Works

Four research lines were followed during the elaboration of this work and diverse future works can be proposed:

- **Clustering-based Analysis of Temporal Graphs**
  - Validate the pipeline in other domains;
  - Design and evaluate pipelines for prediction and information retrieval tasks of ball possessions in football matches;
  - Investigate using the most promising combinations in video searches based on tactical properties associated with ball possession. This research venue can benefit from recent results related to implementing search systems based on projection methods [105].
- **Univariate Time Series Retrieval**
  - Investigate if an extensive parametric analysis can reach higher gains;
  - Employ aggregation methods for ranking and correlation analysis. The starting point will be the aggregation methods associated with re-ranking formulations [193, 69, 146];
  - Using optimization or learning methods to discover the best combination of descriptor and re-ranking methods automatically. The Genetic Programming framework investigated in [131] is a promising starting point.
- **Time Series Semi-Supervised Classification**
  - Study the proprieties of the pairwise distances and similarities and their impacts on the results.
  - Test different similarity measurements to build the graph for label propagation.
- **Multivariate Time Series Retrieval and Classification**
  - Deepen the analysis of the impact of the parameter K on the results;
  - Test different classifiers in the pipeline, such as weighted kNN and graph-based approaches [136];
  - Explore the pipeline in other multichannel data.
- **All Research Lines**

- Evaluation and analysis of the efficiency of the algorithms, based on computational complexity;
- Use of texture-based feature extraction approaches for image representations of time series.

Bibliography

[adn] Average path length - an overview | sciencedirect topics. `https://www.sciencedirect.com/topics/computer-science/average-path-length`. Accessed: 2024-02-24.

[int] Clustering quality - an overview | sciencedirect topics. `https://www.sciencedirect.com/topics/computer-science/clustering-quality`. Accessed: 2024-02-24.

[3] Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Man, D.; Monga, R.; Moore, S.; Murray, D.; Shlens, J.; Steiner, B.; Sutskever, I.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Vinyals, O.; Warden, P.; Wicke, M.; Yu, Y.; and Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *None*.

[4] Abdoli, A.; Murillo, A. C.; Yeh, C.-C. M.; Gerry, A. C.; and Keogh, E. J. (2018). Time series classification to improve poultry welfare. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 635–642.

[5] Al-Jowder, O.; Kemsley, E.; and Wilson, R. (2002). Detection of adulteration in cooked meat products by mid-infrared spectroscopy. *Journal of Agricultural and Food Chemistry*, 50(6):1325–1329.

[6] Alimoglu, F. and Alpaydin, E. (1997). Combining multiple representations and classifiers for pen-based handwritten digit recognition. *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, 2:637–640 vol.2.

[7] Almeida, J.; Pedronette, D. C. G.; Alberton, B. C.; Morellato, L. P. C.; and Torres, R. d. S. (2016). Unsupervised distance learning for plant species identification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(12):5325–5338.

[8] Alonso, A.; Gabirondo, P.; and Scotto, M. (2024). Clustering time series by extremal dependence. *International Journal of Data Science and Analytics*.

[9] Arica, N. and Vural, F. T. Y. (2003). BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24(9-10):1627–1639.

[10] Arica, N. and Yarman-Vural, F. (2003). Bas: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24:1627–1639.

[11] Bagnall, A.; Dau, H. A.; Lines, J.; Flynn, M.; Large, J.; Bostrom, A.; Southam, P.; and Keogh, E. (2018). The uea multivariate time series classification archive, 2018.

[12] Bagnall, A.; Davis, L.; Hills, J.; and Lines, J. (2012). Transformation based ensembles for time series classification. *Proc. 12th SDM*.

[13] Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31:606–660.

[14] Baldridge, A.; Hook, S.; Grove, C.; and Rivera, G. (2009). The aster spectral library version 2.0. *Remote Sensing of Environment*, 113(4):711–715.

[15] Behravan, V.; Glover, N. E.; Farry, R.; Chiang, P. Y.; and Shoaib, M. (2015). Rate-adaptive compressed-sensing and sparsity variance of biomedical signals. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6.

[16] Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press.

[17] Bellman, R.; Corporation, R.; and Collection, K. M. R. (1957). *Dynamic Programming*. Rand Corporation research study. Princeton University Press.

[18] Bergamim, E. and Breve, F. (2022). On tuning a mean-field model for semi-supervised classification. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(5):053402.

[19] Bi, J.-W.; Li, H.; and Fan, Z.-P. (2021). Tourism demand forecasting with time series imaging: A deep learning model. *Annals of Tourism Research*, 90:103255.

[20] Boser, B. E.; Guyon, I. M.; and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Workshop on Computational Learning Theory*, COLT '92, pages 144–152.

[21] Bovolo, F.; Demir, B.; and Bruzzone, L. (2015). A cluster-based appraoch to content based time series retrieval (cbtsr). pages 2793–2796.

[22] Brandes, U. (2001). A faster algorithm for betweenness centrality*. *The Journal of Mathematical Sociology*, 25(2):163–177.

[23] Caliński, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27.

[24] Campbell, J. Y. and Mankiw, N. G. (1989). Consumption, Income and Interest Rates: Reinterpreting the Time Series Evidence. In *NBER Macroeconomics Annual 1989, V. 4*, pages 185–246. National Bureau of Economic Research, Inc.

[25] Campello, R. J. G. B.; Moulavi, D.; and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In Pei, J.; Tseng, V. S.; Cao, L.; Motoda, H.; and Xu, G., editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg. Springer Berlin Heidelberg.

[26] Cao, L.; Mees, A.; and Judd, K. (1998). Dynamics from multivariate time series. *Physica D: Nonlinear Phenomena*, 121(1):75–88.

[27] Chan, K.-P. and Fu, A. W.-C. (1999). Efficient time series matching by wavelets. In *Proceedings 15th International Conference on Data Engineering (Cat. No.99CB36337)*, pages 126–133.

[28] Chaovalit, P.; Gangopadhyay, A.; Karabatis, G.; and Chen, Z. (2011). Discrete wavelet transform-based time series analysis and mining. *ACM Comput. Surv.*, 43(2).

[29] Chapelle, O.; Schölkopf, B.; and Zien, A., editors (2006). *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

[30] Chen, L.; Chen, D.; Yang, F.; and Sun, J. (2021a). A deep multi-task representation learning method for time series classification and retrieval. *Information Sciences*, 555:17–32.

[31] Chen, Y.; Hao, Y.; Rakthanmanon, T.; Zakaria, J.; Hu, B.; and Keogh, E. (2015). A general framework for never-ending learning from time series streams. *Data Mining and Knowledge Discovery*, 29.

[32] Chen, Y. and Xie, Z. (2022). Multi-channel fusion graph neural network for multivariate time series forecasting. *Journal of Computational Science*, 64:101862.

[33] Chen, Z.; Liu, Y.; Zhu, J.; Zhang, Y.; Jin, R.; He, X.; Tao, J.; and Chen, L. (2021b). Time-frequency deep metric learning for multivariate time series classification. *Neurocomputing*, 462:221–237.

[34] Choi, H. and Kang, P. (2024). Multi-task self-supervised time-series representation learning. *Information Sciences*, 671:120654.

[Clements] Clements, J. Dataset: Basicmotions. `https://www.timeseriesclassification.com/description.php?Dataset=BasicMotions`. Accessed: 2024-06-09.

[36] Costa, B.; Freire, J.; Cavalcante, H.; Homci, M.; Castro, A.; Viégas Jr, R.; Meiguins, B.; and Morais, J. (2017). Fault classification on transmission lines using knn-dtw. pages 174–187.

[37] da S. Torres, R. and Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada (RITA)*, 13(2):161–185.

[38] Damrich, S.; Böhm, J. N.; Hamprecht, F. A.; and Kobak, D. (2023). From t-sne to umap with contrastive learning.

[39] Dau, H. A.; Keogh, E.; Kamgar, K.; Yeh, C.-C. M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C. A.; Yanping; Hu, B.; Begum, N.; Bagnall, A.; Mueen, A.; Batista, G.; and Hexagon-ML (2018). The ucr time series classification archive. `https://www.cs.ucr.edu/~eamonn/time_series_data_2018/`.

[40] Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227.

[41] de Abreu Campos, V. and Pedronette, D. C. G. (2019). A framework for speaker retrieval and identification through unsupervised learning. *Comput. Speech Lang.*, 58:153–174.

[42] de Almeida, L. B.; Valem, L. P.; and Pedronette, D. C. G. (2022). Graph convolutional networks and manifold ranking for multimodal video retrieval. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2811–2815.

[43] De Luca, G. and Zuccolotto, P. (2021). Hierarchical time series clustering on tail dependence with linkage based on a multivariate copula approach. *International Journal of Approximate Reasoning*, 139:88–103.

[44] de Sousa, C. A. R.; Rezende, S. O.; and Batista, G. E. (2013). Influence of graph construction on semi-supervised learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*, pages 160–175. Springer.

[45] Dempster, A.; Petitjean, F.; and Webb, G. I. (2020). ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495.

[46] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[47] Deselaers, T.; Pimenidis, L.; and Ney, H. (2008). Bag-of-visual-words models for adult image classification and filtering. In *2008 19th International Conference on Pattern Recognition*, pages 1–4.

[48] Dias, D.; Dias, U.; Menini, N.; Lamparelli, R.; Maire, G. L.; and da S Torres, R. (2020a). Image-based time series representations for pixelwise eucalyptus region classification: A comparative study. *IEEE Geoscience and Remote Sensing Letters*, 17(8):1450–1454.

[49] Dias, D.; Pinto, A.; Dias, U.; Lamparelli, R.; Maire, G. L.; and da Silva Torres, R. (2020b). A multirepresentational fusion of time series for pixelwise classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (JSTARS)*, 13:4399–4409.

[50] Ding, H.; Trajcevski, G.; Scheuermann, P.; Wang, X.; and Keogh, E. (2008). Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proc. VLDB Endow.*, 1(2):1542–1552.

[51] Du, M.; Wei, Y.; Hu, Y.; Zheng, X.; and Ji, C. (2024). Multivariate time series classification based on fusion features. *Expert Systems with Applications*, 248:123452.

[52] Du, M.; Wei, Y.; Zheng, X.; and Ji, C. (2023). Multi-feature based network for multivariate time series classification. *Information Sciences*, 639:119009.

[53] Eckmann, J.-P.; Kamphorst, S.; and Ruelle, D. (1987). Recurrence plots of dynamical systems. *Europhysics Letters (epl)*, 4:973–977.

[54] Ensafi, Y.; Amin, S. H.; Zhang, G.; and Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – a comparative analysis. *International Journal of Information Management Data Insights*, 2(1):100058.

[55] Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Comput. Surv.*, 45(1).

[56] Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

[57] Falcon, A.; D'Agostino, G.; Lanz, O.; Brajnik, G.; Tasso, C.; and Serra, G. (2022). Neural turing machines for the remaining useful life estimation problem. *Computers in Industry*, 143:103762.

[58] Faouzi, J. and Janati, H. (2020a). pyts: A python package for time series classification. *Journal of Machine Learning Research*, 21(46):1–6.

[59] Faouzi, J. and Janati, H. (2020b). pyts: A python package for time series classification. *Journal of Machine Learning Research*, 21(46):1–6.

[60] Faria, F. A.; Almeida, J.; Alberton, B.; Morellato, L. P. C.; and da S. Torres, R. (2016). Fusion of time series representations for plant recognition in phenology studies. *Pattern Recognition Letters*, 83(Part 2):205 – 214.

[61] Fix, E. and Hodges, J. (1951). *Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine.

[62] Freedman, D. (2009). *Statistical Models: Theory and Practice*. Cambridge University Press.

[63] Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315(5814):972–976.

[64] F.R.S., K. P. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

[65] Ghouaiel, N.; Marteau, P.-F.; and Dupont, M. (2017). Continuous pattern detection and recognition in stream - a benchmark for online gesture recognition. *International Journal of Applied Pattern Recognition*, 4:146.

[66] Glenis, A. and Vouros, G. A. (2022). Scale-boss: A framework for scalable time-series classification using symbolic representations. In *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*, SETN '22, New York, NY, USA. Association for Computing Machinery.

[67] Golub, G. H. and Van Loan, C. F. (1989). *Matrix Computations*. The Johns Hopkins University Press, second edition.

[68] Gonçalves, F. M. F.; Pedronette, D. C. G.; and da Silva Torres, R. (2023). Regression by re-ranking. *Pattern Recognition*, 140:109577.

[69] Guimarães Pedronette, D. C.; Pascotti Valem, L.; and Latecki, L. J. (2021). Efficient rank-based diffusion process with assured convergence. *Journal of Imaging*, 7(3).

[70] Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

[71] Harris, J.; Munasinghe, T.; Tubbs, H.; and Anyamba, A. (2022). Predicting crimean-congo hemorrhagic fever outbreaks via multivariate time-series classification of climate data. In *Proceedings of the 6th International Conference on Medical and Health Informatics*, ICMHI '22, page 215–218, New York, NY, USA. Association for Computing Machinery.

[72] He, K.; Zhang, X.; Ren, S.; and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

[73] Hewage, P.; Behera, A.; Trovati, M.; Pereira, E.; Ghahremani, M.; Palmieri, F.; and Liu, Y. (2020). Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24(21):16453–16482.

[74] Hinton, G.; Vinyals, O.; and Dean, J. (2015). Distilling the knowledge in a neural network.

[75] Holme, P.; Kim, B. J.; Yoon, C. N.; and Han, S. K. (2002). Attack vulnerability of complex networks. *Phys. Rev. E*, 65:056109.

[76] Huang, J.; Kumar, S. R.; Mitra, M.; Zhu, W.-J.; and Zabih, R. (1997). Image indexing using color correlograms. In *CVPR*, pages 762–768.

[77] Huang, Y.; Wanga, Z.; Zhang, T.; Xu, C.; and Lianga, H. (2023). Multi-modal simultaneous machine translation fusion of image information. *Journal of Engineering Research*, 11(2):100085.

[78] Huang, Z.; Yang, C.; Chen, X.; Zhou, X.; and Gui, W. (2022). Time series clustering method with cluster validation to identify unknown local cell conditions in the aluminum reduction cell. *Computers Industrial Engineering*, 174:108790.

[79] Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

[80] Ienco, D. and Interdonato, R. (2023). Deep semi-supervised clustering for multi-variate time-series. *Neurocomputing*, 516:36–47.

[81] Iha, T.; Kawamitsu, I.; Ohshiro, A.; and Nakamura, M. (2021). An lstm-based multivariate time series predicting method for number of restaurant customers in tourism resorts. In *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, pages 1–4.

[82] Incardona, S.; Tripodo, G.; Buscemi, M.; Shahvar, M.; and Marsella, G. (2022). A new neural network architecture for time series classification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, page 167818.

[83] Jayanth Krishnan, K. and Mitra, K. (2022). A modified kohonen map algorithm for clustering time series data. *Expert Systems with Applications*, 201:117249.

[84] Jebara, T.; Wang, J.; and Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448.

[85] Ji, C.; Du, M.; Hu, Y.; Liu, S.; Pan, L.; and Zheng, X. (2022). Time series classification based on temporal features. *Applied Soft Computing*, 128:109494.

[86] Jiang, H.; Liu, L.; and Lian, C. (2022). Multi-modal fusion transformer for multivariate time series classification. In *2022 14th International Conference on Advanced Computational Intelligence (ICACI)*, pages 284–288.

[87] Jiang, J.-R. and Chen, H.-C. (2023). Manufacturing quality prediction based on deep learning in conjunction with gramian angular and markov transition fields. In *2023 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan)*, pages 439–440.

[88] Joachims, T. (2000). Transductive inference for text classification using support vector machines. In *Proceedings of the 20th international conference on machine learning*.

[89] Jogin, M.; Mohana; Madhulika, M. S.; Divya, G. D.; Meghana, R. K.; and Apoorva, S. (2018). Feature extraction using convolution neural networks (cnn) and deep learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, pages 2319–2323.

[90] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edition.

[91] Kang, Y.; Wu, C.; and Yu, B. (2024). Time series clustering based on polynomial fitting and multi-order trend features. *Information Sciences*, 678:120939.

[92] Karim, F.; Majumdar, S.; Darabi, H.; and Chen, S. (2017). Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669.

[93] Kasfi, K. T.; Hellicar, A.; and Rahman, A. (2016). Convolutional neural network for time series cattle behaviour classification. In *Proceedings of the Workshop on Time Series Analytics and Applications*, TSAA '16, page 8–12, New York, NY, USA. Association for Computing Machinery.

[94] Kazmi, S.; Bozanta, A.; and Cevik, M. (2021). Time series forecasting for patient arrivals in online health services. In *Proceedings of the 31st Annual International Conference on Computer Science and Software Engineering*, CASCON '21, page 43–52, USA. IBM Corp.

[95] Kim, J.; Shim, K.; Kim, J.; and Shim, B. (2023). Vision transformer-based feature extraction for generalized zero-shot learning.

[96] Kolesnikov, A.; Dosovitskiy, A.; Weissenborn, D.; Heigold, G.; Uszkoreit, J.; Beyer, L.; Minderer, M.; Dehghani, M.; Houlsby, N.; Gelly, S.; Unterthiner, T.; and Zhai, X. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

[97] Kotsiantis, S. B. and Pintelas, P. E. (2004). Recent Advances in Clustering: A Brief Survey. *WSEAS Transactions on Information Science and Applications*.

[98] Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley.

[99] Kyrkou, L.; Nalmpantis, C.; and Vrakas, D. (2019). Imaging time-series for nilm. In Macintyre, J.; Iliadis, L.; Maglogiannis, I.; and Jayne, C., editors, *Engineering Applications of Neural Networks*, pages 188–196, Cham. Springer International Publishing.

[100] Lal, T. N.; Hinterberger, T.; Widman, G.; Schröder, M.; Hill, J.; Rosenstiel, W.; Elger, C. E.; Schölkopf, B.; and Birbaumer, N. (2004). Methods towards invasive human brain computer interfaces. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, page 737–744, Cambridge, MA, USA. MIT Press.

[101] Latora, V. and Marchiori, M. (2001). Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701.

[102] Lee, G.; Gommers, R.; Waselewski, F.; Wohlfahrt, K.; and Aaron (2019). Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4:1237.

[103] Lee, M.-C.; Lin, J.-C.; and Stolz, V. (2023). Np-free: A real-time normalization-free and parameter-tuning-free representation approach for open-ended time series. In *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 334–339.

[104] Lei, T.; Li, J.; and Yang, K. (2024). Time and frequency-domain feature fusion network for multivariate time series classification. *Expert Systems with Applications*, 252:124155.

[105] Leticio, G. R.; Kawai, V. S.; Valem, L. P.; aes Pedronette, D. C. G.; and da S. Torres, R. (2024). Manifold information through neighbor embedding projection for image retrieval. *Pattern Recognition Letters*, 183:17–25.

[106] Li, T.; Yu, B.; Li, J.; and Zhu, Z. (2024). Functional relation field: A model-agnostic framework for multivariate time series forecasting. *Artificial Intelligence*, 334:104158.

[107] Li, X.; Yang, J.; and Ma, J. (2021). Recent developments of content-based image retrieval (cbir). *Neurocomputing*, 452:675–689.

[108] Li, Y.; Liu, J.; and Teng, Y. (2022a). A decomposition-based memetic neural architecture search algorithm for univariate time series forecasting. *Applied Soft Computing*, 130:109714.

[109] Li, Y.; Shen, D.; Nie, T.; and Kou, Y. (2022b). A new shape-based clustering algorithm for time series. *Information Sciences*, 609:411–428.

[110] Liang, H.; Sun, X.; Yunlei, S.; and Gao, Y. (2017). Text feature extraction based on deep learning: a review. *EURASIP Journal on Wireless Communications and Networking*, 2017.

[111] Lin, J.; Khade, R.; and Li, Y. (2012). Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39:287–315.

[112] Lines, J.; Taylor, S.; and Bagnall, A. (2018). Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM transactions on knowledge discovery from data*, 12(5).

[113] Ling, H.; Yang, X.; and Latecki, L. J. (2010). Balancing deformability and discriminability for shape matching. In *ECCV*, volume 3, pages 411–424.

[114] Liu, C.; Springer, D.; Li, Q.; Moody, B.; Abad, R.; Chorro, F.; Castells Ramon, F.; Roig, J.; Silva, I.; Johnson, A.; Syed, Z.; Schmidt, S.; Papadaniil, C.; Hadjileontiadis, L.; Naseri, H.; Moukadem, A.; Dieterlen, A.; Brandt, C.; Tang, H.; and Clifford, G. (2016). An open access database for the evaluation of heart sound algorithms. *Physiological Measurement*, 37:2181–2213.

[115] Liu, Y.; Pu, H.; and Sun, D.-W. (2021). Efficient extraction of deep image features using convolutional neural network (cnn) for applications in detecting and analysing complex food matrices. *Trends in Food Science Technology*, 113:193–204.

[116] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

[117] Long, L.; Liu, Q.; Peng, H.; Wang, J.; and Yang, Q. (2022). Multivariate time series forecasting method based on nonlinear spiking neural p systems and non-subsampled shearlet transform. *Neural Networks*, 152:300–310.

[118] Lyu, H.; Huang, D.; Li, S.; Ng, W. W.; and Ma, Q. (2023). Multiscale echo self-attention memory network for multivariate time series classification. *Neurocomputing*, 520:60–72.

[119] Manning, C. D.; Raghavan, P.; and Schtze, H. (2008a). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

[120] Manning, C. D.; Raghavan, P.; and Schütze, H. (2008b). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

[121] McInnes, L.; Healy, J.; and Melville, J. (2020). Umap: Uniform manifold approximation and projection for dimension reduction.

[122] Menini, N.; Almeida, A. E.; Lamparelli, R. A. C.; Maire, G. L.; Santos, J. A.; Pedrini, H.; Hirota, M.; and da S. Torres, R. (2019). A soft computing framework for image classification based on recurrence plots. *IEEE Geoscience and Remote Sensing Letters*, 16(2):320–324.

[123] Mishra, S.; Bordin, C.; Taharaguchi, K.; and Palu, I. (2020). Comparison of deep learning models for multivariate prediction of time series wind power generation and temperature. *Energy Reports*, 6:273–286.

[124] Mitrović, D.; Zeppelzauer, M.; and Breiteneder, C. (2010). Chapter 3 - features for content-based audio retrieval. In *Advances in Computers: Improving the Web*, volume 78 of *Advances in Computers*, pages 71–150. Elsevier.

[125] Mizoguchi, T.; Kobayashi, Y.; and Ajiro, Y. (2023). Unsupervised retrieval based multivariate time series anomaly detection and diagnosis with deep binary coding models. In *4th Asia Pacific Conference of the Prognostics and Health Management*, Tokyo, Japan.

[126] Moody, G. (2004). Spontaneous termination of atrial fibrillation: A challenge from physionet and computers in cardiology 2004. volume 31, pages 101– 104.

[127] Moura, F.; Emmerik, R.; Exel, J.; Martins, L. E. B.; Barros, R.; and Cunha, S. (2016). Coordination analysis of players' distribution in football using cross-correlation and vector coding techniques. *Journal of Sports Sciences*, 1:1–9.

[128] Moura, F.; Martins, L. E. B.; Anido, R.; Barros, R.; and Cunha, S. (2012). Quantitative analysis of brazilian football players' organisation on the pitch. *Sports biomechanics / International Society of Biomechanics in Sports*, 11:85–96.

[129] Moura, F.; Martins, L. E. B.; Anido, R.; Ruffino, P.; Barros, R.; and Cunha, S. (2013). A spectral analysis of team dynamics and tactics in brazilian football. *Journal of sports sciences*, 31:1568 – 1577.

[130] Mudassir, M.; Bennbaia, S.; Ünal, D.; and Hammoudeh, M. (2020). Time-series forecasting of bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*.

[131] Muñoz, J. A. V.; da Silva Torres, R.; and Gonçalves, M. A. (2015). A soft computing approach for learning to aggregate rankings. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM*, pages 83–92, Melbourne, VIC, Australia.

[132] Nawaz, R.; Cheah, K. H.; Nisar, H.; and Yap, V. V. (2020). Comparison of different feature extraction methods for eeg-based emotion recognition. *Biocybernetics and Biomedical Engineering*, 40(3):910–926.

[133] Ojala, T.; Pietikainen, M.; and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987.

[134] Oquab, M.; Darcet, T.; Moutakanni, T.; Vo, H. V.; Szafraniec, M.; Khalidov, V.; Fernandez, P.; Haziza, D.; Massa, F.; El-Nouby, A.; Howes, R.; Huang, P.-Y.; Xu, H.; Sharma, V.; Li, S.-W.; Galuba, W.; Rabbat, M.; Assran, M.; Ballas, N.; Synnaeve, G.; Misra, I.; Jegou, H.; Mairal, J.; Labatut, P.; Joulin, A.; and Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision.

[135] Pailot-Bonnétat, S.; Harris, A. J. L.; Calvari, S.; De Michele, M.; and Gurioli, L. (2020). Plume height time-series retrieval using shadow in single spatial resolution satellite images. *Remote Sensing*, 12(23).

[136] Papa, J.; Falcão, A.; and Suzuki, C. (2009). Supervised pattern classification based on optimum-path forest. *Int. J. Imaging Syst. Technol.*, 19(2):120–131.

[137] Passalis, N. and Tefas, A. (2016). Entropy optimized feature-based bag-of-words representation for information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1664–1677.

[138] Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

[139] Patil, S. and Talbar, S. (2012). *Content Based Image Retrieval Using Various Distance Metrics*, pages 154–161. Springer Berlin Heidelberg, Berlin, Heidelberg.

[140] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[141] Pedronette, D. C. G. and da S. Torres, R. (2010). Shape retrieval using contour features and distance optmization. In *VISAPP*, volume 1, pages 197 – 202.

[142] Pedronette, D. C. G. and da S. Torres, R. (2011). Exploiting clustering approaches for image re-ranking. *Journal of Visual Languages & Computing*, 22(6):453–466.

[143] Pedronette, D. C. G.; Gonçalves, F. M. F.; and Guilherme, I. R. (2018). Unsupervised manifold learning through reciprocal knn graph and connected components for image retrieval tasks. *Pattern Recognition*, 75:161–174. Distance Metric Learning for Pattern Recognition.

[144] Pedronette, D. C. G.; Penatti, O. A. B.; Calumby, R. T.; and da S. Torres, R. (2014). Unsupervised distance learning by reciprocal knn distance for image retrieval. In *International Conference on Multimedia Retrieval (ICMR'14)*.

[145] Pedronette, D. C. G.; Torres, R. d.; Borin, E.; and Breternitz, M. (2012). Efficient image re-ranking computation on gpus. In *2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, pages 95–102.

[146] Pedronette, D. C. G.; Valem, L. P.; Almeida, J.; and da S. Torres, R. (2019). Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Transactions on Image Processing*, 28(12):5824–5838.

[147] Pedronette, D. C. G.; Valem, L. P.; Almeida, J.; and Torres, R. d. S. (2019). Multimedia retrieval through unsupervised hypergraph-based manifold ranking. *IEEE Transactions on Image Processing*, 28(12):5824–5838.

[148] Pedronette, D. C. G.; Valem, L. P.; and da S. Torres, R. (2021). A bfs-tree of ranking references for unsupervised manifold learning. *Pattern Recognition*, 111:107666.

[149] Pino, F. A. (2014). Sazonalidade na agricultura. *Revista De Economia Agrícola (Impresso)*, v. 61:p. 63–93.

[150] Pisani, F.; Valem, L. P.; Pedronette, D. C. G.; da Silva Torres, R.; Borin, E.; and Jr., M. B. (2020). A unified model for accelerating unsupervised iterative re-ranking algorithms. *Concurrency and Computation: Practice and Experience*, 32(14):e5702.

[151] pyts. Discrete fourier transform. Acesso: 06/12/2022.

[152] pyts. Random convolutional kernel transform. Acesso: 06/12/2022.

[153] Péalat, C.; Bouleux, G.; and Cheutet, V. (2022). Improved time series clustering based on new geometric frameworks. *Pattern Recognition*, 124:108423.

[154] Quoc Nguyen, D.; Nguyet Phan, M.; and Zelinka, I. (2021). Periodic time series forecasting with bidirectional long short-term memory: Periodic time series forecasting with bidirectional lstm. In *2021 The 5th International Conference on Machine Learning and Soft Computing*, ICMLSC'21, page 60–64, New York, NY, USA. Association for Computing Machinery.

[155] Rakthanmanon, T.; Campana, B.; Mueen, A.; Batista, G.; Westover, B.; Zhu, Q.; Zakaria, J.; and Keogh, E. (2012). Searching and mining trillions of time series subsequences under dynamic time warping. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 262–270, New York, NY, USA. Association for Computing Machinery.

[156] Ratanamahatana, C. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong.

[157] Ratna Prakarsha, K. and Sharma, G. (2022). Time series signal forecasting using artificial neural networks: An application on ecg signal. *Biomedical Signal Processing and Control*, 76:103705.

[158] Ricardo Baeza-Yates, B. R.-N. (2013). *Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca*. Editora Bookman.

[159] Richman, J. S. and Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049. PMID: 10843903.

[160] Rodrigues, D. C. U. M.; Moura, F. A.; Cunha, S. A.; and da S. Torres, R. (2019). Graph visual rhythms in temporal network analyses. *Graphical Models*, 103:101021.

[161] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65.

[162] Rozin, B. and aes Pedronette, D. C. G. (2024). A ranked-based framework based on manifold learning for multivariate time series retrieval and classification (submitted).

[163] Rozin, B.; Bergamim, E.; Pedronette, D. C. G.; and Breve, F. A. (2023). Semi-supervised time series classification through image representations. In Gervasi, O.; Murgante, B.; Taniar, D.; Apduhan, B. O.; Braga, A. C.; Garau, C.; and Stratigea, A., editors, *International Conference on Computational Science and Its Applications – ICCSA 2023*, pages 48–65, Cham. Springer Nature Switzerland.

[164] Rozin, B. and Pedronette, D. (2021). Time series classification using shape features based on angle statistics. In *Anais do XVIII Encontro Nacional de Inteligência Artificial e Computacional*, pages 470–481, Porto Alegre, RS, Brasil. SBC.

[165] Rozin, B.; Pedronette, D. C. G.; and Torres, R. S. (2024). Re-ranking and representations for time series retrieval: A comparative study (submitted).

[Rozin et al.] Rozin, B.; Torres, R. d. S.; Moura, F. A.; and Pedronette, D. C. G. Ball possession analysis based on temporal network properties (not published - in process of writing).

[167] Saito, N. (1994). *Local feature extraction and its applications using a library of bases*. Yale University.

[168] Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, 10:e0118432.

[169] Samiee, K.; Kovács, P.; and Gabbouj, M. (2014). Epileptic seizure classification of eeg time-series using rational discrete short-time fourier transform. *IEEE transactions on bio-medical engineering*, 62.

[170] Santos, E. S.; Alberton, B.; Morellato, L. P.; and da Silva Torres, R. (2019). Pixelwise time series retrieval in phenological studies. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 6586–6589.

[171] Schäfer, P. and Högqvist, M. (2012). Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, page 516–527, New York, NY, USA. Association for Computing Machinery.

[172] Schlegl, T.; Schlegl, S.; Tomaselli, D.; West, N.; and Deuse, J. (2022). Adaptive similarity search for the retrieval of rare events from large time series databases. *Advanced Engineering Informatics*, 52:101629.

[173] Schneider, P. and Xhafa, F. (2022). Chapter 8 - machine learning: Ml for ehealth systems. In Schneider, P. and Xhafa, F., editors, *Anomaly Detection and Complex Event Processing over IoT Data Streams*, pages 149–191. Academic Press.

[174] Semenoglou, A.-A.; Spiliotis, E.; and Assimakopoulos, V. (2023). Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, 157:39–53.

[175] Senin, P. (2009). Dynamic time warping algorithm review.

[176] Sharma, S.; Diarra, A.; Alvares, F.; and Ropars, T. (2020). Kdetect: Unsupervised anomaly detection for cloud systems based on time series clustering. In *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, SNTA '20, page 3–10, New York, NY, USA. Association for Computing Machinery.

[177] Shen, L. and Wang, Y. (2022). Tcct: Tightly-coupled convolutional transformer on time series forecasting. *Neurocomputing*, 480:131–145.

[178] Shih, S.-Y.; Sun, F.-K.; and Lee, H.-y. (2019). Temporal pattern attention for multivariate time series forecasting. *Mach. Learn.*, 108(8–9):1421–1441.

[179] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*.

[180] Song, D.; Xia, N.; Cheng, W.; Chen, H.; and Tao, D. (2018). Deep r -th root of rank supervised joint binary embedding for multivariate time series retrieval. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery amp; Data*

Mining, KDD '18, page 2229–2238, New York, NY, USA. Association for Computing Machinery.

[181] Sood, S.; Zeng, Z.; Cohen, N.; Balch, T.; and Veloso, M. (2022). Visual time series forecasting: An image-driven approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21, New York, NY, USA. Association for Computing Machinery.

[182] Stehling, R. O.; Nascimento, M. A.; and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM*, pages 102–109.

[183] Stival, L.; Pinto, A.; Andrade, F. d. S. P. d.; Santiago, P. R. P.; Biermann, H.; Torres, R. d. S.; and Dias, U. (2023). Using machine learning pipeline to predict entry into the attack zone in football. *PLOS ONE*, 18(1):1–24.

[184] Su, J.; Xie, D.; Duan, Y.; Zhou, Y.; Hu, X.; and Duan, S. (2024). Mdcnet: Long-term time series forecasting with mode decomposition and 2d convolution. *Knowledge-Based Systems*, 299:111986.

[185] Swain, P. H. and Hauska, H. (1977). The decision tree classifier: Design and potential. *IEEE Transactions on Geoscience Electronics*, 15(3):142–147.

[186] Tahan, M. H.; Ghasemzadeh, M.; and Asadi, S. (2023). A novel embedded discretization-based deep learning architecture for multivariate time series classification. *IEEE Transactions on Industrial Informatics*, 19(4):5976–5984.

[187] Tan, M. and Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training. *CoRR*, abs/2104.00298.

[188] Tan, Z.; Zhao, M.; Wang, Y.; and Yang, W. (2023). Multivariate time series retrieval with binary coding from transformer. In Tanveer, M.; Agarwal, S.; Ozawa, S.; Ekbal, A.; and Jatowt, A., editors, *Neural Information Processing*, pages 397–408, Singapore. Springer Nature Singapore.

[189] Tao, B. and Dickinson, B. W. (2000). Texture recognition and image retrieval using gradient indexing. *JVCIR*, 11(3):327–342.

[190] Tiano, D.; Bonifati, A.; and Ng, R. (2021). Featts: Feature-based time series clustering. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, page 2784–2788, New York, NY, USA. Association for Computing Machinery.

[191] Torres, R. D. S. and Falcão, A. X. (2006). Content-based image retrieval: Theory and applications. *Revista de Informática Teórica e Aplicada*, 13:161–185.

[192] Torres, R. d. S.; Hasegawa, M.; Tabbone, S.; Almeida, J.; dos Santos, J. A.; Alberton, B.; and Morellato, L. P. C. (2013). Shape-based time series analysis for remote phenology studies. In *IEEE Int. Geoscience and Remote Sensing Symposium*, pages 3598–3601.

[193] Valem, L. P.; Pedronette, D. C. G.; and Latecki, L. J. (2023). Rank flow embedding for unsupervised and semi-supervised manifold learning. *IEEE Transactions on Image Processing*, 32:2811–2826.

[194] Valem, L. P. and Pedronette, D. C. G. a. (2017). An unsupervised distance learning framework for multimedia retrieval. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, pages 107–111, New York, NY, USA. ACM.

[195] Valem, L. P.; Pedronette, D. C. G. a.; Torres, R. d. S.; Borin, E.; and Almeida, J. (2015). Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, page 51–58, New York, NY, USA. Association for Computing Machinery.

[196] van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

[197] Van Engelen, J. E. and Hoos, H. H. (2020a). A survey on semi-supervised learning. *Machine learning*, 109(2):373–440.

[198] Van Engelen, J. E. and Hoos, H. H. (2020b). A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

[199] Vinh, N. X.; Epps, J.; and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11:2837–2854.

[200] Volna, E.; Kotyrba, M.; and Habiballa, H. (2015). Ecg prediction based on classification via neural networks and linguistic fuzzy logic forecaster. *The Scientific World Journal*, 2015:205749.

[201] Wan, G. G. and Liu, Z. (2008). Content-based information retrieval and digital libraries. *Information Technology and Libraries*, 27(1):41–47.

[202] Wang, J.; Balasubramanian, A.; Mojica de la Vega, L.; Green, J. R.; Samal, A.; and Prabhakaran, B. (2013). Word recognition from continuous articulatory movement time-series data using symbolic representations. In Alexandersson, J.; Ljunglöf, P.; McCoy, K. F.; Portet, F.; Roark, B.; Rudzicz, F.; and Vacher, M., editors, *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies*, pages 119–127, Grenoble, France. Association for Computational Linguistics.

[203] Wang, T.; Liu, Z.; Zhang, T.; Hussain, S. F.; Waqas, M.; and Li, Y. (2022). Adaptive feature fusion for time series classification. *Knowledge-Based Systems*, 243:108459.

[204] Wang, Z. and Oates, T. (2015a). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. In *Workshops at the twenty-ninth AAAI conference on artificial intelligence*.

[205] Wang, Z. and Oates, T. (2015b). Imaging time-series to improve classification and imputation. In *Proceedings of the 24th International Conference on Artificial Intelligence*, page 3939–3945. AAAI Press.

[206] Wei, C.; Wang, Z.; Yuan, J.; Li, C.; and Chen, S. (2023). Time-frequency based multi-task learning for semi-supervised time series classification. *Information Sciences*, 619:762–780.

[207] Wei, L. and Keogh, E. (2006). Semi-supervised time series classification. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 748–753.

[208] West, D. (2001). *Introduction to Graph Theory*. Featured Titles for Graph Theory. Prentice Hall.

[209] Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

[210] Xi, L.; Yun, Z.; Liu, H.; Wang, R.; Huang, X.; and Fan, H. (2022). Semi-supervised time series classification model with self-supervised learning. *Engineering Applications of Artificial Intelligence*, 116:105331.

[211] Xu, G.; Ren, M.; Wang, Z.; and Li, G. (2024). Memf: Multi-entity multimodal fusion framework for sales prediction in live streaming commerce. *Decision Support Systems*, 184:114277.

[212] Xu, H.; Li, J.; Yuan, H.; Liu, Q.; Fan, S.; Li, T.; and Sun, X. (2020). Human activity recognition based on gramian angular field and deep convolutional neural network. *IEEE Access*, 8:199393–199405.

[213] Yang, C.; Wang, X.; Yao, L.; Long, G.; and Xu, G. (2024). Dyformer: A dynamic transformer-based architecture for multivariate time series classification. *Information Sciences*, 656:119881.

[214] Yang, X. and Latecki, L. J. (2011). Affinity learning on a tensor product graph with applications to shape and image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2011)*, pages 2369–2376.

[215] Ye, L. and Keogh, E. (2011). Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. *Data Mining Know. Discovery*, 22:149–182.

[216] Yehuda, Y.; Freedman, D.; and Radinsky, K. (2023). Self-supervised classification of clinical multivariate time series using time series dynamics. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5416–5427, New York, NY, USA. Association for Computing Machinery.

[217] Younis, R.; Hakmeh, A.; and Ahmadi, Z. (2024). Mts2graph: Interpretable multivariate time series classification with temporal evolving graphs. *Pattern Recognition*, 152:110486.

[218] Yu, C.; Luo, L.; Chan, L. L.-H.; Rakthanmanon, T.; and Nutanong, S. (2019). A fast lsh-based similarity search method for multivariate time series. *Information Sciences*, 476:337–356.

[219] Yu, X.; Wang, H.; Wang, J.; and Wang, X. (2024a). A common feature-driven prediction model for multivariate time series data. *Information Sciences*, 677:120967.

[220] Yu, Y.; Becker, T.; Trinh, L. M.; and Behrisch, M. (2023a). Saxregex: Multivariate time series pattern search with symbolic representation, regular expression, and query expansion. *Computers Graphics*, 112:13–21.

[221] Yu, Y.; Kruyff, D.; Jiao, J.; Becker, T.; and Behrisch, M. (2023b). Pseudo: Interactive pattern search in multivariate time series with locality-sensitive hashing and relevance feedback. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):33–42.

[222] Yu, Y.; Ma, R.; and Ma, Z. (2024b). Robformer: A robust decomposition transformer for long-term time series forecasting. *Pattern Recognition*, 153:110552.

[223] Zeng, Z.; Balch, T.; and Veloso, M. (2022). Deep video prediction for time series forecasting. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21, New York, NY, USA. Association for Computing Machinery.

[224] Zhang, Y.; Xia, C.; Cao, G.; Zhao, T.; and Zhao, Y. (2024). Pattern recognition of hand movements based on multi-channel mechanomyography in the condition of one-time collection and sensor doffing and donning. *Biomedical Signal Processing and Control*, 91:106078.

[225] Zhang, Z.; Li, D.; Zhang, Z.; and Duffield, N. (2021). A time-series clustering algorithm for analyzing the changes of mobility pattern caused by covid-19. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Animal Movement Ecology and Human Mobility*, HANIMOB '21, page 13–17, New York, NY, USA. Association for Computing Machinery.

[226] Zhao, S.; Kong, M.; Li, R.; Hounye, A.; Ri, S.; Hou, M.; and Cao, C. (2024). Scarnet: using convolution neural network to predict time series with time-varying variance. *Multimedia Tools and Applications*, pages 1–27.

[227] Zhu, D.; Song, D.; Chen, Y.; Lumezanu, C.; Cheng, W.; Zong, B.; Ni, J.; Mizoguchi, T.; Yang, T.; and Chen, H. (2020). Deep unsupervised binary coding networks for multivariate time series retrieval. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(02):1403–1411.

[228] Zhu, H.; Zhao, P.; Chan, Y.-P.; Kang, H.; and Lee, D. L. (2022). Breast cancer early detection with time series classification. In *Proceedings of the 31st ACM International Conference on Information amp; Knowledge Management*, CIKM '22, page 3735–3745, New York, NY, USA. Association for Computing Machinery.

[229] Zhu, X. (2005). Semi-Supervised Learning Literature Survey - TR 1530. *Sciences-New York*.

[230] Zhu, X. and Ghahramani, Z. (2002). Learning from labeled and unlabeled data with label propagation.

[231] Zuva, K. and Zuva, T. (2012). Evaluation of information retrieval systems. *International Journal of Computer Science and Information Technology*, 4:35–43.