TÉCNICO LISBOA

Multivariate Correlations for Early Classification

João Pedro Carriço Beirão

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. Alexandra Sofia Martins de Carvalho
Prof. Paulo Alexandre Carreira Mateus

Examination Committee

Chairperson: Prof. António Manuel Raminhos Cordeiro Grilo
Supervisor: Prof. Alexandra Sofia Martins de Carvalho
Member of the Committee: Prof. Sara Alexandra Cordeiro Madeira

May 2018

Declaração

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Dedicated to my parents.

Acknowledgments

I would like to express my gratitude to my thesis supervisors, Prof. Alexandra Carvalho and Prof. Paulo Mateus, for all their assistance, availability, support and for the opportunity to discover and work in an area for which I have a particular interest. A special acknowledgement to Prof. Dr. Helena Canhão for providing the rheumatoid arthritis dataset from Reuma.pt, to Cátia Botas for the cooperation with the arduous assignment of preprocessing the data, and to Mariano Lemus for the explanations and clarifications regarding the proposed method. I am also thankful to Instituto de Telecomunicações, from Instituto Superior Técnico – Torre Norte, for the possibility of having a grant to work on the FCT (Fundação para a Ciência e Tecnologia) project NEUROCLINOMICS2 (PTDC/EEI-SII/1937/2014).

I would also like to thank my dear colleagues and friends Andreia Chagas, Catarina Gaspar, Christopher Edgley, Francisco Oliveira, Inês Lourenço, João Raposo, Luís Alves, Miguel Vasconcelos, Orlando Vaz, Sofia Quaresma and Tiago Santos. My gratitude for the support, friendship, companionship and for making my academic journey more bearable and truly worthwhile.

Finally, but more importantly, my sincere thanks to all my family, in particular to my mother, Maria de Fátima Beirão, my father, João José Beirão, my sister, Ana Beirão, my brother-in-law, Jorgen Muylaert, my other sister, Inês Bettencourt, my grandparents, António Carriço, Maria Sofia Carriço, João Martins Beirão and Maria de Jesus Beirão, my aunt, Irene Escudeiro, my other aunt, Paula Carvalho, my uncle, Walter Carvalho, and my cousins, Samuel Carvalho and Sara Carvalho. My gratitude for the unconditional support, patience and dedication.

A special acknowledgement for many others that are not mentioned, but whose contribution was remarkable and of great importance. As a famous proverb states, *it takes a village to raise a child*, so my sincere gratitude to all my villagers. Lastly, one final thank to God.

Resumo

As séries temporais multivariadas estão presentes em diferentes aplicações de *Data Mining* (mineração de dados), onde uma ou diversas variáveis são analisadas ao longo do tempo. *Early classification* (classificação antecipada) pode ser entendida como uma extensão do problema de classificação de séries temporais, em cujo objetivo é obter uma previsão confiável o mais cedo possível. Em dados temporais, a ordem das observações é crucial, uma vez que uma determinada sequência de ocorrências pode ser o fator distintivo e discriminante. O mesmo acontece com a classificação antecipada, onde o objetivo é usar a menor quantidade possível de informação, garantindo uma precisão satisfatória. As correlações entre as variáveis das séries temporais multivariadas, associadas a diferentes instantes de tempo, podem fornecer conhecimento acerca de dependências preditivas e relações que podem ser exploradas. E é este o ponto de interseção entre correlações multivariadas e classificação antecipada.

Um método baseado em teoria da informação, que analisa a oportunidade de classificação antecipada num conjunto de séries temporais com a informação relativa às suas respetivas classes, é proposto, implementado e avaliado. O objetivo do algoritmo *Multivariate Correlations for Early Classification* (MCEC - em português, correlações multivariadas para classificação antecipada) é identificar o instante de tempo prematuro para o conjunto total de dados, a partir do qual a restante informação pode ser ignorada, obtendo-se ainda assim uma previsão razoável. Foram realizados testes de validação experimentais em dados sintéticos, simulados e reais. A abordagem proposta obteve bons resultados, os quais foram confirmados com recurso a significância estatística, no que diz respeito ao balanço entre precisão e precocidade, dois dos desafios fundamentais em classificação antecipada. Esta metodologia pode ser considerada uma alternativa relevante, não apenas para o contexto de classificação antecipada, mas também para procedimentos de seleção de atributos.

Palavras-chave: classificação antecipada, correlações multivariadas, informação mútua, entropia condicional, redes Bayesianas, critério de informação de Akaike, descrição de comprimento mínimo

Abstract

Multivariate time series are found in several data mining applications, where one or multiple variables are analysed over time. Early classification arises as an extension of the time series classification problem, in view of obtaining a reliable prediction as soon as possible. In temporal data, the order of the observations is critical, given that a particular sequence of occurrences can be the distinctive and discriminative factor. The same is true for early classification, where the goal is to use as less information as possible, ensuring a decent accuracy. The correlations between the variables of the multivariate time series from different time points may provide insights into predictive dependencies and relationships to be exploited. And here is where multivariate correlations and early classification come together.

An information-theoretic method for investigating the early classification opportunity from a collection of time series together with their respective class labels is proposed, implemented and assessed. The goal of the Multivariate Correlations for Early Classification (MCEC) algorithm is to identify the early classification time point for the entire dataset, from which the remaining information can be neglected and still obtain a satisfactory prediction. Experimental validation tests are performed on synthetic, simulated and real data. The proposed approach achieved good results, which were confirmed with statistical significance, concerning a tradeoff between accuracy and earliness, the two fundamental challenges in early classification. This methodology can be considered a relevant alternative, not only for the early classification context, but also for feature selection procedures.

Keywords: early classification, multivariate correlations, mutual information, conditional entropy, Bayesian networks, Akaike information criterion, minimum description length

Contents

List of Tables

List of Figures

Nomenclature

$\mathcal{B}$	Bayesian network
$\mathcal{N}$	set of nodes
$\mathcal{X}$	alphabet
$\mathcal{S}$	set of structural relations
Φ_{X_i}	set of non-descendants of X_i
Π_{X_i}	set of parents of X_i
Θ	set of conditional probability distributions
A	Accuracy
$BEA(p)$	balance between E and A, according to p
C_i	class label (instance i)
D	dataset
E	Earliness
E	set of directed edges
G	directed acyclic graph
H	entropy
I	mutual information
kNN	k-Nearest-Neighbour
L	length of the time series
N	number of features
n	early classification time point
P	set of probability distributions
p	weight between E and A

R structural relation

S system

T_i time series (instance i)

t_i subsequence of T_i

w number of instances in the dataset

X, Y discrete random variables

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

CART Classification and Regression Trees

CBR Case-Based Reasoning

CSV Comma-Separated Values

DAS Disease Activity Score

DMARD Disease-Modifying Antirheumatic Drug

DTW Dynamic Time Warping

EA-ConvNets Earliness-Aware Deep Convolutional Networks

ECG Electrocardiography

ECM Early Classification Model

ECTS Early Classification on Time Series

EDSC Early Distinctive Shapelet Classification

EPIMTS Early Prediction on Imbalanced Multivariate Time Series

EULAR European League Against Rheumatism

FN *False Negative*

FP *False Positive*

GEFM Generalized Extended F-measure

GSDT Generalized Sequential Decision Tree

HAQ Health Assessment Questionnaire

HMM Hidden Markov Model

ICU Intensive Care Unit

ID3 Iterative Dichotomiser

IPED Interpretable Patterns for Early Diagnosis

KKT Karush-Kuhn-Tucker

LL Log-Likelihood

MCEC Multivariate Correlations for Early Classification

MCFEC Mining Core Feature for Early Classification

MD-MPP Multilevel-Discretized Marked Point Process

MDL Minimum Description Length

MIT Mutual Information Test

MPL Minimum Prediction Length

MSD Multivariate Shapelets Detection

NML Normalized Maximum Likelihood

QBC Query By Committee

QDA Quadratic Discriminant Analysis

RA Rheumatoid Arthritis

REACT Reliable Early Classification

SCR Sequential Classification Rule

SI Silhouette Index

SMOTE Synthetic Minority Oversampling Technique

SPR Sociedade Portuguesa de Reumatologia

SPRINT Scalable Parallelizable Induction of Decision Trees

SVM Support Vector Machine

TCP Transmission Control Protocol

TN *True Negative*

TP *True Positive*

Chapter 1

Introduction

1.1 Motivation

Temporal data, generally known as multivariate time series, consist of measurements or observations acquired and organized sequentially. In this context, one or multiple variables are examined over time, which means that the order of the information plays an important role. This sort of data is commonly found in several data mining application areas such as medicine, economy, meteorology and marketing. Standard sequence classification involves using temporal data for constructing a classifier, which is able to predict the class label of a new given time series, with a satisfactory accuracy. The classification of multivariate time series represents an important problem for time-sensitive applications. A temporal sequence can have multiple components associated to different variables but concerning the same period of time. For example, the medical data of a patient being monitored in an Intensive Care Unit (ICU) may include the blood pressure, the body temperature, the electrocardiography (ECG), among others. Each variable is described as a component (dimension) of a multivariate time series.

Early classification can be viewed as an extension of the time series classification problem and it arises in scenarios where the anticipation of the prediction is beneficial. This matter has been a relevant subject of study in recent past, due to its several time-sensitive applications. The ability to obtain information in advance by having early knowledge about a specific event may be of great utility in many areas. For instance, a medical study [33] described how clinical data revealed that infants who were diagnosed with sepsis disease suffered from an unusual heartbeat twenty-four hours before the diagnosis. In this case, supervising the time series data of the infant's heartbeat and being able to classify it in advance, may lead to an effective early diagnosis and treatment.

In information theory, the idea of correlations between variables is associated to the analysis of the relationships and dependencies among them. In general, correlation can be viewed as a statistical technique for measuring how strong two random variables are related [44]. Moreover, a certain variable described over time through a time series is considered to be serially correlated if there is a statistical dependency between the values from different time periods [37]. Conversely, when two variables or two observations are independent, that means no correlation is verified between them. Mutual information

quantifies the dependency (or the correlation) between variables. It represents the amount of information that one random variable provides about another, that is, how much the knowledge of one variable reduces the uncertainty of another [19]. Therefore, this concept is closely related with entropy, a fundamental measure of information. In fact, the conditional entropy describes the impact that knowing one random variable has on the uncertainty of another one.

Most real-world applications involve a degree of uncertainty, as a result of limitations in the information available and the challenges in modelling complex systems. Bayesian networks consist of probabilistic graphical models commonly used for representing the information contained in a given dataset [44]. Their structures describe statistical dependencies and causal relationships between random variables. Data may contain unexpected correlations and their examination can be useful for providing relevant knowledge to be explored, such as patterns and predictive associations. Considering the previously mentioned example of sepsis disease in infants, the investigation on the correlations among the clinical measurements and the patient's health condition was able to find a meaningful relationship. In fact, there seems to be a correlation between the child's heartbeat and the sepsis disease diagnosis. However, one question arises: how can the correlations from a time series dataset contribute to obtain important information in advance?

1.2 Aims

This thesis attempts to study this question by applying an information-theoretic approach to the early classification context. For such purpose, a theoretical algorithm [47] was implemented and assessed through an empirical investigation.

Consider a multivariate time series which represents the evolution of a collection of variables over time. The objective of early classification is to assign a class label as early as possible, while ensuring that this prediction matches the class that would be assigned for the complete time series. This means that the problem consists of finding a certain early time point from which a classification with a satisfactory accuracy can be performed. Now, the collection of variables described by the multivariate time series can be viewed as discrete random variables, for which a joint probability distribution can be defined. The correlations between the variables from any two time points represent the influence that the information from one time point has on another. Furthermore, the correlations between the time points and the class label indicate the existing relationships and dependencies among them. In fact, given the chronological organization of the multivariate time series, it is relevant to analyse the dependency of the class label on the early states. And here is where multivariate correlations and early classification come together.

Consider a group of multivariate time series and their respective class labels, forming a given dataset. Assume the time series length and the number of variables per time point are fixed and uniform for every instance. Visually, consider this dataset as a cube (Figure 1.1), where the height represents the instances, the length describes the time points and the width denotes the variables under analysis. Moreover, an additional column includes the class labels associated to each component of the height

(instance). The goal of the proposed method is to identify the early classification time point for the entire dataset, which corresponds to a specific instant of the time series (somewhere along the length of the cube), from which the remaining time points are dispensable for a satisfactory classification. For such purpose, the dataset is divided in three sections: the early states (red), portion of the cube from the initial time point until the early time point; the later states (green), portion of the cube from the early time point until the end of the time series; and the class information (blue), column with the class labels for each instance.

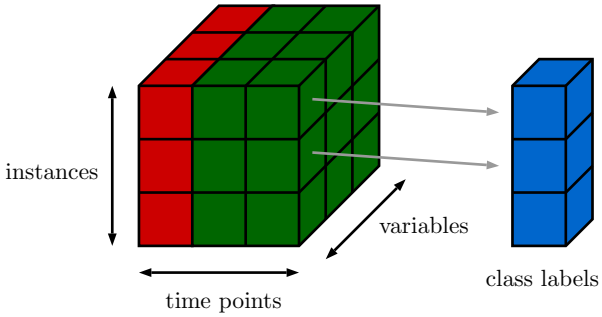


Figure 1.1: Visual representation of a given dataset, containing a collection of multivariate time series and their respective class labels. The height of the cube represents the instances, its length the time points and its width the variables. The blue prism depicts the class labels of each time series (class information). The red section of the cube describes a potential early states portion and the green a potential later states one.

The correlations between the early and the later states, as well as between the early states and the class information are examined. This investigation is performed while varying the early classification time point from the first instant until the end of the time series. The idea is to study the possibility of predicting the class labels using only the information from the early states, that is, neglecting the later time points. Based on a Bayesian network representation, three measures are used to determine the early time point: the difference in entropy (which represents the lack of information for predicting the class labels using only the early states) and two scoring functions, MDL and AIC (which describe the tradeoff between the complexity of the model and its effectiveness in fitting the data). For verification purposes, an investigation on the performance of a set of classifiers is done. Their accuracy, according to the length of the time series, is used as a comparative measure for the experimental results of the proposed algorithm.

This work aims to contextualize the early classification problem in the data mining context, to explain its formulation and applications, and to review current algorithms on this matter. A method based on an information theory approach is proposed, implemented and assessed through experimental tests in synthetic, simulated and real data.

1.3 Claim of contributions

As further exploited in this thesis, several methods addressing the early classification problem have been proposed in the last years. However, it is important to clarify that the algorithm examined in this work is

not intended to be a classifier. The majority of the state-of-the-art approaches present methods which require a learning stage followed by a classification step. This means they are capable of assigning a class label to a single incomplete time series. That is not the case of the methodology considered in this thesis. Conversely, a dataset investigation is proposed, where the information from the entire collection of multivariate time series, as well as their respective class labels, is the subject of study. The early classification opportunity is explored through the analysis of the knowledge contained in the data. Overall, the main contributions of this thesis are:

1. An overview on the data mining context, focusing on data preprocessing and classification. A contextualization and explanation of the early classification problem, as well as a detailed state-of-the-art review on some of the most acknowledged methods in the literature.

2. An information-theoretic algorithm for examining the early classification opportunity in a dataset containing multivariate time series together with their respective class labels. An exposition of the relevant information and probability theory concepts, required to understand the proposed method. The implementation of the algorithm made freely available[1], and an article submitted to an international journal [47]. This is the result of a joint contribution of Mariano J. Lemus, João Pedro Beirão, Prof. Alexandra M. Carvalho, Prof. Paulo Mateus and Prof. Nikola Paunković .

3. An evaluation of the developed method, through experimental tests on synthetic, simulated and real data. That includes a data dimensionality impact analysis, a computation time assay, a comparison with one of the state-of-the-art algorithms, and a statistical significance confirmation concerning a tradeoff between accuracy and earliness, the two fundamental challenges in the early classification context.

1.4 Document Outline

Chapter 2 includes an overview on the data mining background, focusing on the data preprocessing aspects as well as on classification. Concerning the latter, some existing types of classifiers are briefly explained. Moreover, the early classification problem is contextualized and described, its applications are presented and a thorough state-of-the-art review on this subject is expounded.

Chapter 3 introduces some relevant information and probability theory concepts, namely entropy, joint entropy, conditional entropy, mutual information, Bayesian networks and model selection. Then, the proposed method is explained and the specifications of the software implementation are reported.

Chapter 4 describes the experimental results obtained from the assessment of the developed algorithm. A data dimensionality impact analysis as well as a computation time evaluation is included. Furthermore, the experiments on benchmark data (univariate and multivariate time series) and on a real clinical dataset are described. Based on the empirical outcomes, a statistical significance evaluation, regarding the tradeoff between accuracy and earliness, is depicted.

Lastly, Chapter 5 encloses the final conclusions, the achievements and suggestions for future work.

[1] `https://github.com/joaopbeirao/MCEC-algorithm`

Chapter 2

Background

As pointed out by Larose [45], tremendous amounts of data are collected daily for a variety of applications such as science, health, finances, marketing and security. However, as John Naisbitt mentioned in his book [58], *we are drowning in information but starved for knowledge*. The easiness in storing information led to the necessity of developing automatic tools for transforming the available data into meaningful knowledge.

2.1 Data Mining

Data Mining can be viewed as the response to this problem, since it consists in the process of discovering relevant patterns and relationships in large datasets. This search for valuable information usually includes steps such as data preprocessing, patterns discovery, patterns evaluation and knowledge presentation. The information repositories may be of different forms such as databases, data warehouses, transactional data, and advanced data types (e.g. time-related or sequence data, spatial data, data streams and others).

In general, data mining functionalities comprise two categories of tasks: descriptive, where patterns, trends and other properties of a target dataset are analysed in the interest of searching for interpretations and explanations; and predictive, where from the analysis of the available data, inductions, estimations and predictions are performed. Some examples of data mining functionalities include data characterization and discrimination; the mining of frequent patterns, associations, and correlations; classification and regression; clustering; and outlier analysis. [38]

As an interdisciplinary field, data mining is closely related to a variety of domains, namely statistics, machine learning, pattern recognition, database and data warehouse systems, information retrieval, visualization, algorithms, high performance computing, amongst others.

2.1.1 Data Preprocessing

Real-world data typically demand a preprocessing phase as a way to improve its quality. Since the information repositories tend to be noisy, incomplete and inconsistent, data preprocessing techniques

are applied in the interest of improving the performance, accuracy and efficiency of the data mining processes.

The inaccuracies in data may be due to faulty instruments for data collection, inconsistencies in data formats or in naming conventions, incomplete information, and errors during data transmission, data aggregation or the data entry process [45].

Data preprocessing is considered an important stage in the data mining process and it usually comprises four techniques: data cleaning, data integration, data reduction and data transformation [38]. These procedures are intended to improve the quality of the data, which depends on the designated use, and they do not require a specific order since they complement each other.

Data Cleaning

Data cleaning operations attempt to deal with incomplete, noisy and inconsistent information in the data repository. The existence of missing values is frequently adverse for data mining techniques. Even though some methods are capable of dealing with incomplete data, their approach may be neither the most appropriate nor robust for a concrete situation. These are the most common methods for handling missing values [38, 45]:

- Omit the records or fields with missing values (risk of ignoring important information);
- Replace missing values manually (impracticable in the case of large datasets);
- Replace missing values with constant;
- Replace missing values with descriptive statistics of the field (e.g. mean, median or mode);
- Replace missing values with randomly generated values based on the analysis of the existing distributions (e.g. regression, decision tree induction or Bayesian inference).

This step of data preprocessing is also responsible for smoothing out noise from data, by taking into account the adjacent values. Noise consists of random errors or variances in a given variable; and binning methods and regression are examples of data smoothing techniques commonly used to correct discrepancies in the data [38]. Another procedure related to noise handling corresponds to the identification of outliers. These are values abnormally distant from the remaining, which in some cases may be out of the data ranges, or evidence an inconsistent deviation from the trends. Outliers do not always represent errors in data entries, however, their detection can be crucial for the efficiency of the data mining methods. Clustering, histograms of variables and scatter plots are strategies typically used for identifying outliers [45].

Data Integration

When combining information from multiple data repositories, the resulting merged dataset may contain redundancies and inconsistencies. A thorough data integration may be important for dealing with problems such as semantic heterogeneity and structure of the data. This phase may involve procedures such

as entity identification, redundancy and correlation analysis, field duplication detection and data value conflict detection and resolution [38]. In some cases, data integration may be followed by an additional data cleaning in order to avoid inconsistencies caused by the combination of data from different sources.

Data Reduction

If on the one hand, data integration allows the combination of data from multiple repositories, on the other, in some situations, a reduction on the amount of information is desired. The large dimensions of the data repository may hamper its analysis, and the use of data mining methods may become impractical or infeasible. Data reduction techniques are helpful in these circumstances, since they aim for acquiring a diminished dataset without losing relevant information. This means that the application of the data mining functionalities on the compressed dataset is expected to achieve a more efficient performance, yet accomplish similar results.

The methods for data reduction include dimensionality reduction, numerosity reduction and data compression. In the process of dimensionality reduction, the number of random variables or features is diminished according to certain specifications. Wavelet transforms, principal component analysis, feature selection and feature construction are examples of dimensionality reduction methods. Numerosity reduction techniques attempt a reduced representation of the data through parametric models (e.g. regression or log-linear models) or nonparametric models (e.g. histograms, clustering, sampling, or data cube aggregation). The data compression methodologies consist of transformations applied in the interest of achieving a diminished dataset. [38]

Data Transformation

Some data mining functionalities require data in certain forms as a way to achieve a more efficient performance. For instance, the measurement units of a specific variable may have an undesired impact on the results, particularly in distance-based methods, by modifying the relevance of a given feature. For this reason, data transformation is useful for converting the data into a more convenient form, according to the designated data mining technique. Data transformation strategies include normalization, smoothing, feature construction, aggregation, discretization and concept hierarchy generation for nominal data.

2.1.2 Classification

Data mining functionalities generally comprise two categories: supervised and unsupervised [45]. In unsupervised methods, the algorithms search for relationships, patterns, trends and structure among unlabelled data. This means that no external knowledge is provided concerning the target or output variables. Clustering is an example of an unsupervised method, where the objects in data are organized into groups (clusters) according to their similarities. On the other hand, the majority of data mining techniques consist of supervised methods [45]. In this case, the goal is to develop a mapping model from labelled data, that is able to predict the target or output variables of new given data.

Classification corresponds to a supervised method, in which models are known as classifiers and the target or output variable is called the class label attribute. This data mining method is usually separated in two steps: the learning step (also referred to as training phase) and the classification step (or test phase) [38]. In the first step, the algorithm is provided with a dataset (named training set) that includes, amongst all data, the information regarding the predefined target or output variable (class labels). Based on the analysis of the training set, the algorithm is expected to learn and build a mapping model (classifier) that describes and specifies the existing data classes. Classification rules, decision trees and mathematical formulas, are examples of typical representations for classification models. These specifications may provide a better understanding of the available data and they can be used to assign class labels to new records. In the classification step, the performance of the developed model is examined using a distinct dataset named test set. Similarly to the training set, apart from all the variables, the test set includes the class label attribute. However, in this step, the class label assignment (classification) is performed based on the model built on the learning step. Then, the effectiveness of the classification is evaluated by comparing the class label predictions with the expected values. An acquainted issue in classification is when the classifier is excessively conformed to the training set. This is called overfitting and it occurs when there is an imbalance between the complexity of the mapping model and its ability to generalize [45]. In these situations, the predictive performance of the classification is affected since the classifier has incorporated certain irregularities from the training set.

Classification is a subject of study common to a variety of fields closely related to data mining, such as machine learning, pattern recognition and statistics. Applications comprise banking, education, fraud detection, target marketing, manufacturing, medical diagnosis, amongst others. [38]

Evaluation

In classification tasks, the evaluation of a classifier estimates its ability to effectively predict, by analysing the classification results produced. In general, the evaluation of the model's quality is accomplished with the use of a test set consisting of data distinct from the training set. These two datasets may be generated through several methods for splitting the labelled data into a training set and a testing set [38]. Cross-validation is one of the existing strategies not only for dealing with the classification datasets but also to provide more reliable evaluation results of the classifier. In k-fold cross validation, from the initially available data, k complementary subsets are randomly generated, all with approximately the same dimensions. Then, the learning step is performed with data from the $k-1$ subsets (training set) and the remaining subset is used for the classification step. The training and test phases are executed k times, thus allowing for each subset to work as test set. The evaluation measures are calculated based on the average of the overall k classification results. When the k subsets preserve approximately the same class distribution of instances as in the original data, the method is referred to as stratified cross validation.

A variation of the k-fold cross validation consists of the leave-one-out method (also known as jackknife) [53]. In this case, only one instance is used as the test set, for each iteration, while the others are responsible for the training of the classifier. This method may not be suitable for large datasets due to

its computational demand.

Four distinct types of classification results are typically used to compare the classifier's prediction with the expected outcome [53]:

- *True Positive* (TP) - when an instance that belongs to a certain class is classified as such;
- *True Negative* (TN) - when an instance that does not belong to a certain class is not classified as such;
- *False Positive* (FP) - when an instance does not belong to a certain class but it is classified as such;
- *False Negative* (FN) - when an instance belongs to a certain class but it is not classified as such.

		Predicted Class	
		Positive Class	**Negative Class**
Actual Class	**Positive Class**	True Positive (TP)	False Positive (FP)
	Negative Class	False Negative (FN)	True Negative (TN)

Table 2.1: Confusion Matrix for a binary classification problem.

These terms are usually organized in a table called confusion matrix (Table 2.1), whose dimensions vary according to the number of existing classes. The confusion matrix gives information about the ability of the classifier to analyse the instances of different classes [38]. For an effective classifier, the diagonal of this matrix (TP and TN) includes the majority of the instances and the remaining entries (FN and FP) are close to zero. The evaluation of a given classifier is usually based on several measures, including:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{2.1}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \text{ and} \tag{2.2}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{2.3}$$

The Accuracy in Equation (2.1) corresponds to the percentage of correctly classified instances and it indicates how precisely the classifier is capable of predicting the various classes as a whole. This measure is not so rigorous for class imbalanced data, where the existing classes are not proportionally represented [38]. In this case, Sensitivity, in Equation (2.2), and Specificity, in Equation (2.3), are more appropriate evaluation measures, since they reflect the proportion of positive or negative instances, respectively, which are properly classified [53].

Class imbalanced data correspond to a non-similar distribution of instances from the existing classes, and they can represent a problem in the learning performance of the algorithms. This is considerably frequent in some real-world datasets, where the samples associated to a specific class are in insufficient number. The solutions include, among others, under-sampling methods, where data is removed; and over-sampling methods, where instances are added to the dataset [42]. The Synthetic Minority Over-sampling Technique (SMOTE) [16] is an example of a procedure for dealing with imbalanced data. This

approach combines the over-sampling of the minority class with the under-sampling of the majority one. The use of methodologies for imbalanced learning can be significantly beneficial to the classification performance [35].

Distance-based classifiers

The group of classification methods known as distance-based classifiers use a measure of proximity (distance) to assign a class label to a new instance. Euclidean distance, absolute difference, maximum distance metric and Dynamic Time Warping (DTW), are examples of proximity measures for analysing the distance of the unknown instance to the existing classes [53].

In k-Nearest Neighbours (kNN) [21], all the instances of the training set are stored and used to represent their respective class. For this method, the class labels of the k closest neighbours determine the classification of the incoming instance. This means that, after computing the distance between the unknown sample and each record of the training set, the class label is assigned according to the k closest cases. [45]

Some considerations about this algorithm need to be taken into account, namely the choice of the distance measure and the value of k. Both specifications affect the performance of kNN, since the size of the data can influence some proximity metrics, small k's may generate noisy results and large k's can create ambiguous decisions for classification [53].

Due to its computational expense, particularly for large datasets, alternatives to the kNN technique include the Exemplar-based Nearest Neighbours [59], where each class is represented by one sample, named the exemplar. The features of the representative instance are usually computed as the average of the overall samples in the same class.

Bayes Classifiers

Bayes classifiers are considered statistical classifiers, seeing that they build a probabilistic model of the features in the learning step, and based on that model they predict the classification of a new instance, through probabilities' computation [53]. Considering the general case, the *a posteriori* probability $P(C_i|X)$ represents the probability of the instance X, of unknown class, belonging to class C_i. In these classification methods, this probability is computed through the Bayes' Theorem:

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}, \tag{2.4}$$

where $P(X|C_i)$ is the probability of the class C_i containing the instance X, $P(C_i)$ corresponds to the probability of an instance belonging to class C_i, and $P(X)$ consists of the prior probability of X.

Naïve Bayes classifier [66] is one of the most appreciated probabilistic classifiers because of its simplicity and effectiveness. This algorithm is known not only for building models easily but also for making rapid predictions [53]. The connotation of "naïve" is related to the assumption that the features are class conditionally independent [38]. This means that the influence a certain feature has on a given class C_i is independent of all the other features' values [57]. Even if some features are dependent or

related to each other, they are considered as properties that individually contribute to the *a posteriori* probability.

For a given training set of multiple instances, each one is represented by a feature vector $X = \{x_1, x_2, ..., x_N\}$ and its respective class C_i. A new incoming instance X is classified according to the class with the maximum a posteriori probability. In Equation (2.4), based on the assumption of class conditional independence and since $P(X)$ is equal for all classes, $P(X|C_i)$ is calculated as:

$$P(X|C_i) \approx \prod_{k=1}^{N} P(x_k|C_i), \tag{2.5}$$

and $P(C_i)$ may be computed from the training set as the quotient between the number of instances that belong to class C_i and the total number of records. Since x_k corresponds to the value of the feature k of the instance X, the probabilities from Equation (2.5) can be estimated from the training set. However, according to the type of feature, distinct procedures are used. While for categorical variables, $P(x_k|C_i)$ can be computed through frequency calculation, for the continuous case, it is obtained from the probability density function of the feature [57].

Decision Trees

Some classification methods comprise the construction of a decision tree. This consists of a diagram of decision nodes, that represent evaluations on features, linked by branches, which denote the evaluations' results [68]. The origin or highest node is known as the root node and the terminal nodes, also called leaf nodes, indicate the class labels. For an incoming instance of an unknown class, its features are tested by the decision tree. Starting from the root node, the features are evaluated on the decision nodes. According to the outcome, the respective branch is chosen, which leads to another decision node or to a leaf node, allowing the classification [45].

In these type of classifiers, the decision tree is constructed from the training set and their respective class labels. These methods are considered simple and fast, since they consist of an intuitive knowledge representation, and they usually involve a top-down recursive divide-and-conquer approach [38]. This means that the dataset is gradually sectioned into subsets, using feature selection techniques. These correspond to heuristic procedures for identifying the features that most effectively divide the existing instances into distinct classes. During the tree construction, feature selection measures are employed, such as the information gain or the Gini index [45]. While some of them produce binary trees, where each decision node generates only two branches, others enable non-binary decisions. Another issue that arises with decision trees consists of handling noise and outliers in the training data. Tree pruning is a response to this problem through the identification and removal of branches that are the result of noisy data.

ID3 (Iterative Dichotomiser) [62], C4.5 [63], CART (Classification ad Regression Trees) [71] and SPRINT (Scalable Parallelizable Indution of Decision Trees) [70] are examples of decision tree algorithms. The differences between them are mainly related with the feature selection techniques and the tree pruning mechanisms.

Support Vector Machines

According to Mitsa [53], Support Vector Machines (SVMs) consist of one of the most accurate and robust data mining methods, since they are not affected by the number of dimensions nor require the use of the entire training set in the learning step. However, the computational complexity and, consequently, the demanding training time are usually identified as disadvantages.

SVMs [72] assume that the data from the training set can be separated by class, even if that involves a transformation into higher dimensions. The goal is to construct a decision boundary, known as maximum marginal hyperplane, that separates the training set in two classes, while maximizes the margin between both of them. Hence, this method is mainly designed for binary class conditions, where the hyperplane corresponds, for instance, to a separating line or plane for the two or three dimensions context, respectively.

The margin can be seen as the distance between the closest elements of the two classes, which means that it represents the separation between classes. For this reason, the larger the margin, the better the quality of the division and, consequently, the more effective the classifier at classifying new incoming instances. The classification step is performed by testing the unknown instance in the mathematical representation. According to the sign of the result, the class label is assigned.

The hyperplane is found through the support vectors, which consist of the elements closest to the decision boundary. They are commonly referred to as the essential or critical training instances, seeing that for a new case, where all the other records were removed, the constructed hyperplane would be the same [38].

In terms of computation, the maximal margin hyperplane can be found through solving quadratic optimization problems, Lagrangian formulations or using Karush-Kuhn-Tucker (KKT) conditions. Through the kernel trick [73], SVMs are able to perform non-linear classification by transforming the data from the training set into an appropriate higher dimensional space, in the interest of having a linearly separable situation. In this case, in order to decrease the complexity of the calculations, instead of computing the inner products in the new space, a kernel function may be applied to the original space. Polynomial kernel, Gaussian radial basis function kernel and Sigmoid kernel, are examples of existing kernel functions.

For the multi-class context, SVM classifiers may be used through several strategies such as the one proposed by Aiolli et al. [3], where a classifier is trained per class.

Neural Networks

Neural networks are computational models that resemble the human brain because of their structure and operation [52]. They consist of a collection of nodes (comparable to neurons) connected through weighted arcs. The multilayer feed-forward is a common form of neural networks that organizes the nodes in a structure with an input layer, one or more hidden layers, and an output layer. While the input layer corresponds to the features of each instance from the training set, the output layer consists of their respective class labels. Although the nodes of each layer are not linked together, they are connected

to all the nodes of the following layer. The term feed-forward means that the information progresses between layers, from left-to-right, in only one direction (no cycles back to nodes from previous layers). The output of each node is calculated by summing the weighted outputs from the previous nodes and applying a non-linear activation function to this sum.

These models are known as adaptive systems since their structure is modified according to the available information. Before the training step, a decision has to be made regarding the network topology: number of nodes in the input layer, number of hidden layers, number of nodes in each hidden layer, and number of nodes in the output layer. Afterwards, each node has its weight adjusted in the interest of minimizing the error of the class label prediction for the training set instances.

Backpropagation is a well-known neural network algorithm used to perform the training phase. It consists of an iterative process where the training set is used to compare the predicted classifications for each instance with their expected correct class labels. Since, initially, the weight assigned to each arc is a random value, during this learning step, these weights are modified in order to reduce the difference between the predicted and the expected classification results. These adjustments occur from the output layer to the input layer (from right-to-left), through a sort of trial-and-error process.

For the classification step, a new incoming instance is input to the neural network, which means that its features are used in the input layer, and the calculations output the predicted class label (in the output layer). In spite of its robustness to noise and its ability to classify patterns, neural network classification is usually criticized for its difficult interpretation and its time consuming training step [38]. Several different neural network algorithms have been proposed with different structures, methodologies and activation functions [53].

2.1.3 Sequence Classification

As previously mentioned in this chapter, the information repositories, in which data mining methods are applied, may be of different forms. If on the one hand, data mining can be defined as the process of automatically analysing data and extracting relevant information or knowledge from existing patterns, trends or relationships; on the other hand, the field of temporal data mining is interested in applying the same process to sequence data [46]. Taking into consideration that a sequence corresponds to an ordered list of events, this type of data may be categorized into five groups [78]:

- *simple symbolic sequence*, where the events consist of alphabetic symbols (e.g. DNA sequence, formed by the amino acids A, C, G, T);

- *complex symbolic sequence*, where each event is a vector of symbolic values (e.g. list of items bought by a customer in a store, during one year);

- *simple (or univariate) time series*, which represents a sequence of numeric values organized in regular time intervals (e.g. ECG measurements of a certain patient, measured each second, during one hour);

- *multivariate time series*, which represents a sequence of numerical vectors, that contain the information of more than one variable, also organized in regular time intervals (e.g. measurements of some gases concentrations in the air of a certain room, collected each hour, during one day);

- *complex event sequence*, where each event is a vector of multiple data types - numerical, categorical and others (e.g. patient's health record from monthly medical appointments, during one year). In the scope of the thesis, this type of sequence data is considered an heterogeneous multivariate time series.

In temporal data mining, the order of the events is informative and important for the description and the modelling of the data [46]. This means that there are positional or temporal dependencies and the data is organized with respect to a particular index. For instance, in gene sequences, the nucleotides follow a specific order within a DNA molecule, while in time series, the events are explicitly indexed by time as a collection of chronological observations [28].

The main tasks in temporal data mining are described by Laxman et al. [46] and correspond to: prediction, clustering, search and retrieval, pattern discovery, and classification. In the first one, the predictive model constructed for the data uses the information from previous records to forecast future values of the sequence. In clustering, collections of sequences are organized in groups according to their similarities. Sequence search and retrieval methods are useful in the case of large datasets, since they are interested in identifying concrete subsequences among data. Pattern discovery techniques are concerned with searching hidden local structures of interest (patterns), which may represent knowledge within the context. Finally, the last task suggests that a sequence may have a class label assigned.

In sequence classification, the training set corresponds to a collection of sequences with the information about their respective class labels. For example, a multivariate time series with the measurements of some gases concentrations in the air of a certain room, collected every hour, during one day, may indicate polluted or not polluted environment. Three categories organize the existing sequence classification techniques, according to their specifications [38, 78]. In feature-based classification (e.g. decision trees and neural networks), conventional classification methods are applied to the sequence data considered as features. For numeric data, discretization is required, which may originate the loss of information. In sequence distance-based classification (e.g. kNN and SVM), conventional classifications methods are applied, yet the distance function measures the similarity between sequences and dictates the quality of the classification. In model-based classification (e.g. Naïve Bayes classifier and Hidden Markov Models), statistical models are used to accomplish the classification of sequences. These techniques are based on generative models that describe the probability distribution of the sequences in each of the existing classes.

As proven by the already mentioned examples, sequence classification has a wide range of real-world applications. For instance, in speech recognition, gesture recognition, genomic analysis, information retrieval, health informatics, stock market analysis, economic and sales forecasting, as well as process and quality control [38, 78].

2.2 Early Classification

Recent research has been focusing on an extension of the sequence classification problem, known as early classification. Seeing that earliness is intuitively related with temporal data, this problem deals with observations collected over time, generally referred to as time series. In this sort of data, the information is acquired and organized sequentially, which means that the order of the measurements has significance and their values are highly correlated. This is the case in electronic medical records, when the patient's health condition is monitored in each appointment and the information collected is structured as chronological records.

Consider a dataset D composed of a collection of pairs:

$$(T_i, c_i) : i \in \{1, ..., w\}, \tag{2.6}$$

where T_i consists of a time series, c_i corresponds to its respective class label and w represents the number of instances in D.

In general, a time series can be defined as a vector of length L:

$$T_i = \left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, ..., \mathbf{x}_L^{(i)}\right), \tag{2.7}$$

where each component:

$$\mathbf{x}_k^{(i)} = \left(x_{k_1}^{(i)}, x_{k_2}^{(i)}, ..., x_{k_N}^{(i)}\right), \tag{2.8}$$

consists of a set of N features measured at time point k.

In time series classification, a class label c_i is associated to each T_i through the relation:

$$Class(T_i) = c_i. \tag{2.9}$$

The standard time series classification goal is to construct a classifier from a training set, capable of assigning a class label to a new time series, with the maximum accuracy possible. Beyond optimizing the accuracy of the classification, in some applications it is beneficial to classify data as early as possible [79]. The amount of available time points is associated with more complete information about the time series, which, in general, is expected to allow a more accurate class label prediction. In addition, the anticipation of the classification implicates fewer time points and, consequently, less knowledge from the time series, which may have an effect on the accuracy of the outcomes [55]. Therefore, one of the fundamental challenges is the tradeoff between accuracy and earliness, since it is desirable to obtain a class label prediction without waiting for the end of the sequence, while ensuring an acceptable classification accuracy [79]. Early classification of time series aims for making predictions as soon as enough data is available, and it is relevant in contexts where the collection of data has a cost associated or the delay of the predictions is adverse [55].

The work from Xing et al. [79] was one of the first to formulate the problem of early classification. For

a time series T_i, Equation (2.7), the subsequence:

$$t_i = \left(\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, ..., \mathbf{x}_n^{(i)}\right) \tag{2.10}$$

describes the section from the beginning until a time point $n \in \{1, ..., L\}$. This variable represents the timestamp from which the information of the time series can be neglected. An early classifier is able to find the time point n and perform an accurate classification based on t_i, such that:

$$Class(T_i) = Class(t_i) = c_i. \tag{2.11}$$

The possibility of acquiring reliable information in advance can be determinant in some situations. For example, when deciding if a tumour is benign or cancerous, the diagnosis may take a certain amount of time, since the evolution, the characteristics and the modifications over time dictate the final verdict. The anticipation of this decision may not be urgent for the benign case, however, it can be crucial in the event of being cancer [60]. The advantages of early classification in medical applications is summarized by Ghalwash et al. [30] when they say that *providing correct and timely diagnosis saves lives*. The early and accurate prediction of a patient's health condition, based on the available information, enables the beginning of the necessary treatment before the disease is completely active [30].

As stated by Xing et al. [80], it is important to distinguish early classification from classic time series prediction, where the goal is to forecast values at some point in the future. Early classification of temporal data consists of anticipating the classification by using only a portion of the available information, without compromising the prediction quality. The goal is to predict the class label of a time series as early as possible provided that the classification accuracy is close to the one using the complete data.

2.2.1 Applications

Early classification may have a variety of applications, such as disease diagnosis in health informatics, anomaly detection, disaster prediction and process control. In many different applications for disease diagnosis prediction, using clinical data, the amount of available information may improve the accuracy of the classification results. However, using only part of the data for an early classification with an acceptable accuracy may be of great interest. Being able to predict the evolution of a clinical case, in an early stage, can be crucial for taking the right decisions.

An application for early classification addressed by Hatami et al. [39] consists of the detection and recognition of odours in the environment. In places such as laboratories, warehouses and space stations, the monitoring of the air conditions inside is important to identify the presence of toxic or dangerous chemicals. These compounds can be prejudicial to human health and its early detection may be determinant in minimizing the exposure damages.

In situations of natural catastrophes, early classification can also have a relevant impact. Earthquakes and tsunamis are known to be preceded by some indications related with seismic activity. Monitoring the convenient data and being able to anticipate the prediction of natural disasters, may be crucial

in avoiding greater damage.

Network security and traffic engineering can also benefit with early classification. In order to protect network resources and restrict illicit use, it is important to identify the application associated with the traffic flow. A technique for accurate traffic classification is proposed by Bernaille et at. [9], using only the information from the first five packets of a TCP (Transmission Control Protocol) connection. This allows an early detection of the application associated, since their identification occurs before the end of the TCP flow. Intrusions or malicious attacks to the network may be prevented, as well as anomalies detected, with an accurate early traffic classification.

In general, the benefits of early classification are related with the possibility of saving time, due to having the information beforehand and acting accordingly in advance.

2.2.2 Related Work

Literature complies with the fact that Rodriguez et al. [67] were the first to mention early classification. They propose a time series classification method based on relative and region predicates that describe temporal intervals. Classifiers are constructed using these predicates, through a boosting approach, which consists of a method that generates ensembles of classifiers. Since the classification uses the linear combination of the predicates, omitting the ones with an unknown result, this approach can be used with variable length time series and can classify incomplete data. They point the advantage of this partial time series classification in situations where information is needed as early as possible. However, they also mention the necessity of verifying the evolution of the variables in order to confirm the precision of the classification. Therefore, although they focus on early prediction based on incomplete data, the unavailable information is ignored, which means that the reliability and the accuracy of the early classification are not taken into consideration. Nevertheless, this initial approach suggested the possibility of adapting boosting methods for addressing early classification [69].

The application of early classification in supervision and diagnosis of dynamic systems was introduced by Bregón et al. [11]. They use a Case Based Reasoning (CBR) methodology to detect faults, as early as possible, in a continuous process. Each time a fault is detected, the CBR system is expected to identify the most likely cause, out of a certain group of alternatives. The k-nearest neighbour (kNN) algorithm is used for the learning process, with three different similarity measures: the Euclidean distance, the Manhattan distance and the Dynamic Time Warping (DTW). Moreover, the kNN algorithm consists of the classifier that allows the classification of time series with different lengths. Likewise in the work from Rodriguez et al. [67], in spite of focusing on the classification of incomplete data as a way to achieve early classification, the optimization of its accuracy and reliability is not a subject of study.

While referring to the existing standard sequence classification methods, Xing et al. [77] pointed that the general approaches were only concerned with improving the accuracy of the classification. In general, most methods consisted of extracting features from complete sequences and developing classification models based on a set of features. Hence, they studied the problem of early classification on sequence data and proposed two methods: the Sequential Classification Rule (SCR) and the Gen-

eralized Sequential Decision Tree (GSDT). The goal consisted of finding sequence classifiers able to predict the class label of a new sequence, without using its entire length, while ensuring an expected accuracy. At first, in the SCR method, a set of features with effective characteristics for early prediction is extracted from the training data. These characteristics, such as frequency, distinctiveness and earliness, are identified through the utility measure, which is based on the information gain and on the weighted support of the feature. Then, from the extracted features and based on both the expected accuracy and the prediction cost, a set of sequential classification rules is formed and used as the classifier.

In the GSDT method, instead of an association rule, a decision tree is built, using a set of features as an attribute. Likewise in the SCR method, the features used are effective for early prediction and the expected accuracy is taken into account. Unlike Rodriguez et al. [67] and Bregón et al. [11], while attempting early classification, Xing et al. [77] also focused on the accuracy of the prediction. On the other hand, the application of the proposed sequence classification methods on time series involves discretization of the data. However, their results demonstrate that the SCR and the GSDT methods do not achieve good performances in this type of data. That could be related with the potential loss of information during discretization or with the fact that these approaches are designed specifically for symbolic sequences.

Due to the wide time-sensitive applications for early classification, Xing et al. [79] analysed some of the existing standard time series classification methods. They pointed that in feature-based methods, time series need to be transformed into a set of features usually through discretization or symbolic transformation. On the other hand, instance-based methods do not require these sort of modifications to the data, since they use the information obtained from the time series of the training set to classify the new data. Therefore, Xing et al. [79] propose an extension of an instance-based method based on the 1-nearest neighbour (1NN) classifier, with the Euclidean distance, for early classification on time series data. Based on the idea of not using the entire time series, but instead, finding a previous location for which the accuracy of the classification is maintained, the Early Classification on Time Series (ECTS) method is proposed. They identified two important requirements for an early classifier: being able to indicate the earliest time location of accurate classification; and ensuring an accuracy close to the case of using the full length time series. Thus, their approach involves a training phase, where the Minimum Prediction Length (MPL) is computed for each time series, yet based on a cluster of identical time series. This variable represents the time location (timestamp) from which the information of a time series can be discarded. Seeing that the classification is not considerably modified by using the data from a specific point on, the amount of information can be compressed, allowing early classification. Then, in the classification phase, for a new time series to be classified, if its 1-nearest neighbour from the training set has a MPL at most equal to the current timestamp being analysed, then the same class label is assigned to the new time series. In the work from Xing et at. [79], the decision upon the tradeoff between earliness and accuracy is analysed, since the reliability of the 1NN is evaluated while anticipating the classification.

Considering some of the existing approaches for early classification on time series and, in particular, the ECTS method, Xing et al. [80] identified one limitation: the interpretability. They point the importance of not only providing classification results, but also extracting interpretable features from time series, for

early classification purposes. These features allow the identification of relevant patterns, useful for application domain experts to obtain additional and summarized information from data. Based on time series shapelets, introduced by Ye et al. [82], the Early Distinctive Shapelet Classification (EDSC) method is proposed. This feature-based approach consists of extracting subsequences of time series (shapelets), which can distinctly point to the target class, and then selecting the ones more effective for early classification. Alternatively to the information gain criteria suggested by Ye et al. [82], they indicate a feature extraction method that uses density estimation or Chebyshev's inequality for computing the distance threshold. However, since the set of extracted shapelets may be extremely large or contain redundant subsequences, a utility rank is assigned to each shapelet according to its earliness, frequency and distinctiveness (extension of the F-measure method). Choosing a reduced subset of the extracted shapelets, according to the above mentioned restrictions, avoids overfitting and ensures the use of the most effective features for early classification. These shapelets' subset is used as a classifier since when a new time series is scanned, the earliest feature match is searched. In spite of its effective performance, the proposed method is restricted to univariate time series [30].

As an extension of EDSC, Ghalwash et al. [29] propose the Multivariate Shapelets Detection (MSD) method. The idea of early classification on time series is maintained, however, it is generalized into a multivariate context. An N-dimensional shapelet is described as a set of multiple extracted subsequences, each of them associated to one specific dimension. Likewise EDSC, the goal is to allow the classification of a time series using only a portion of it. Yet, Ghalwash et al. [29] use the information gain not only for feature extraction but also for feature selection. They point some limitations to the Chebyshev's inequality method and, instead, compute the distance threshold for a shapelet by choosing the one that maximizes the information gain (as suggested initially by Ye et al. [82]). In addition, regarding the shapelets' utility rank, they introduce the use of weighted information gain in place of the extended F-measure, proposed by Xing et al. [80]. A theorem is shown stating that between two shapelets, the one with the highest weighted information gain is the one with better earliness, for the same accuracy performance [29]. Based on this result, a utility rank is assigned to each shapelet, and that allows the selection of a reduced subset with the most effective ones for classification. Likewise EDSC, a new time series is classified by finding the earliest covering shapelet from the generated subset. One drawback of shapelets pointed by Mueen et al. [56] is the significant computation time to extract them. This is a relevant issue also related with the MSD method, since its computation time increases considerably with the amount of data.

More specifically focused on biomedical applications, Ghalwash et al. [32] recognized the efficiency of Hidden Markov Models (HMMs) in dealing with multivariate biomedical temporal data. Due to their dynamic modelling capability as well as their flexibility in handling missing values, these models have been used in speech recognition, language processing and gene expression analysis. In their work [32], they propose the combination of an HMM with a Support Vector Machine (SVM) for the early classification context. An hybrid model named Early Classification Model (ECM) is proposed, in which the HMM is responsible for learning the distribution of the patterns in the training time series, that are then used by the SVM as features for classification. Similarly to the MSD method [29], the ECM also

deals with multivariate time series and the classification consists of gradually analysing a portion of the new time series and checking the ability to predict its label. When comparing both approaches, Ghalwash et al. [32] claim to have significantly surpass the method proposed by the other work [29], yet the interpretability of the results is not provided. As mentioned by Xing et al. [80], particularly in medical applications, the information of the factors that explain a specific prediction is considered useful for the physicians.

Also aiming for medical applications, specifically early disease diagnosis, Ghalwash et al. [30] propose an optimization-based approach for constructing predictive models, through the extraction of multivariate Interpretable Patterns for Early Diagnostics (IPED) from multivariate time series data. This method is organized in three steps: first, the time series data (training set) is transformed into a binary matrix with all extracted subsequences (shapelets) of different lengths, from each dimension; second, a multivariate shapelet is extracted from the binary matrix, for each class, through a convex-concave optimization problem; and third, the dimensionality is reduced and interpretable key shapelets are extracted, both accomplished through a mixed integer optimization formulation. These key shapelets represent each class and, during the classification step, they are compared with the incoming time series. As the time points are being analysed, in the occurrence of a match, the class label is assigned, performing early classification. The main improvement of this method over the ECM approach [32] is that in IPED [30], similarly to EDSC [80], they are concerned with providing interpretable results and consider that to be of great relevance, particularly in medical applications.

Furthermore, He et al. [40] acknowledged the wide utility of early classification on multivariate time series and analysed the existing classification methods for this type of data. They identified two limitations of the MSD method [29]: the inability to extract time-independent subsequences from each dimension for the same multivariate shapelet, and the impossibility of dealing with dimensions of different length. In most applications, patterns of interest for early prediction may appear in different intervals for distinct components of study. In addition, in some cases, these components do not necessarily correspond to time series of uniform length, which results in dimensions with variable temporal sizes. As an attempt to address these issues, a new shapelet's quality evaluation approach is proposed by He et al. [40], as well as the Mining Core Feature for Early Classification (MCFEC) method. They introduce concepts related with the shapelets such as similarity degree, precision, recall and earliness; and they use them in the two steps of the MCFEC method: feature extraction and feature selection. Initially, for each dimension independently, potential shapelets, with minimal precision and recall values, are extracted from the multivariate time series training data. Then, an algorithm selects, from the potential shapelets, the most distinctive and stable ones that will be used in generating the classifier. Instead of using the information gain as in the work of Ghalwash et al. [29], they organize the potential shapelets of each dimension in clusters, according to the similarity degree, using the Silhouette Index (SI) method. Then, a Generalized Extended F-measure (GEFM) is used as the shapelet's quality evaluation for selecting core features of each dimension. Regarding the classification of a new multivariate time series, He et al. [40] propose two methods for generating the classifier for early prediction of its class. The MCFEC-rule classifier, similarly to the SCR method [77], creates an association rule based on the selected core features

(shapelets) from different dimensions. Concerning the other classification method, named MCFEC-QBC classifier, a Query By Committee (QBC) approach is used by matching the new multivariate time series with the core features in order to find the predominant class. The methodology suggested by He et al. [40] focus on early classification of multivariate time series and it is flexible in dealing with the relevant information of distinct dimensions. In comparison with the MSD method [29], a significant progress is achieved in terms of the computation time of the training phase.

As an attempt to focusing on the reliability of the classification decision, Parrish et al. [60] propose an approach based on a decision rule that uses linear or quadratic classifiers. Since, in some applications, there can be a cost associated with obtaining data, they suggest a method for classification of incomplete data. The reliability of the decision is analysed with the probability that the classification performed with the incomplete data would be the same as the one performed with the complete data. This probability corresponds to the degree of confidence that defines the threshold from which the existing data is sufficient for an admissible decision. Three set construction methods are used for computing the reliability threshold: the Chebyshev set, the Gaussian Näive Bayes Quadratic set and the Gaussian Näive Bayes box set. Regarding the estimation of means and covariances of the complete and incomplete information, the joint Gaussian estimation and the Gaussian Mixture Model (GMM) estimation methods are analysed. As classifiers, both the local Quadratic Discriminant Analysis (QDA) and the linear support vector machine are used to perform the classification of incomplete data. The reliability threshold can be compared with the MPL parameter from the ECTS method [79], which is used to control the earliness of the classifier. The advantage of the parameter suggested by Parrish et al. [60] is the assurance on the reliability of the obtained decision, since the classification is performed only when the criterion is met. A similar measure is used in the model proposed by Ghalwash et al. [31], providing an uncertainty estimation of the predictions. Moreover, they mention the importance for users to have an evaluation of the quality of the classification for better interpretation of the outcomes.

Other solutions in the literature proposed multiple approaches for the early classification problem. The work by Wang et al. [75] introduces a deep feature learning method integrated with a non-linear classification model for the early time series classification context. It is named Earliness-Aware Deep Convolutional Networks (EA-ConvNets) and uses a neural network architecture to learn highly discriminative shapelets from time series data for making early class label predictions on incoming instances. In the work from He et al. [41], the issue of early classification in imbalanced data is examined. In real-world datasets used for classification, the number of instances per class may be considerably different. This can affect the precision of the classifiers, since they learn from the available training set. The Early Prediction on Imbalanced Multivariate Time Series (EPIMTS) method, that uses an under-sampling technique, is presented by He et al. [41]. The work of Li et al. [48] proposes an approach for time-critical early decision making, that focus on modelling two aspects of multivariate time series: temporal dynamics and sequential cues. A statistical learning process is suggested, including a Multilevel-Discretized Marked Point Process (MD-MPP) model for the representation of the time series, and probabilistic suffix tree for the characterization of the existing sequential patterns [48]. In the work from Hatami et al. [39], the proposed method is based on a set of classifiers used sequentially in an iterative manner. Each

classifier makes predictions with the portion of the time series available, but it also has a reject option in the case of an unsatisfactory classification. In this case, the decision upon the class label prediction is passed to the next classifier, ensuring, at the end of the process, not only an early but also confident classification. The methodology from Lin et al. [50] is called Reliable Early Classification (REACT) and it generalizes the early classification study for multivariate time series with numerical and categorical features.

One of the most recent approaches proposed for the early classification on time series problem is presented by Mori et al. [55]. Similarly to the ECTS method [79], the accuracy and the earliness of the predictions are identified as the main objectives of early classification on time series, and optimizing the tradeoff between both is perceived to be one of its fundamental challenges. The same goal is identified: predict class labels as early as possible, provided that the level of accuracy is suitable. As an attempt to tackle the problem of these two conflicting objectives, an early classification method based on probabilistic classifiers is proposed and called ECDIRE [55]. They analysed some of the existing methods in the literature and developed an approach capable of dealing with three aspects: avoid unnecessary calculations (specifically, forecasting and checking at all time points), control the reliability of the classifications (for instance, in the case of outliers), and measure the uncertainty of the predictions (a quantitative and interpretable evaluation). The learning step is organized in three phases: first, a procedure is designed for analysing the training set and identifying, for each class, the time points from whence the predictions are suitable; second, a threshold is defined, for each class, to control the reliability of the class label predictions and evaluate their quality; and third, the probabilistic classifiers are trained for performing early classification. In the classification step, unknown time series are classified having into account the information obtained from the learning process. Thus, excessive calculations related with premature predictions are avoided, and the precision of the classification is controlled by dealing with unreliable predictions and providing a quantitative and interpretable measure of the quality of the outcomes. Although some of these issues focused by Mori et al. [55] have already been dealt with in other approaches, they propose a method that aims to comprise all of them simultaneously.

Chapter 3

Proposed Method

The proposed method for early classification is explained in this chapter. This approach for multivariate correlations is the result of a joint contribution of Mariano J. Lemus, João Pedro Beirão, Prof. Alexandra M. Carvalho, Prof. Paulo Mateus and Prof. Nikola Paunković [47]. An introduction to some information and probability theory concepts is included, in the interest of contextualizing the multivariate correlations methodology for the early classification problem. Then, the proposed method is expounded and the aspects concerning the implementation are described.

3.1 Information Theory

Information theory is concerned with the study of information measures, their properties and their applications [25]. It studies the transmission, processing, extraction and usage of information. The concepts from information theory have been used in many different fields of science and technology, such as biology, neurobiology, chemistry, economy, computer science, bioinformatics, web search, cryptography, pattern recognition, anomaly detection, wireless communication and video compression [25, 18]. Information theory is considered to be the intersection of mathematics, physics, statistics, probability theory and engineering, with applications in problems that deal with manipulation, acquisition, storage and transmission of information. Therefore, this subject of study is closely related with data mining techniques and, in particular, it can be of great benefit for the early classification context.

Information theory deals with a variety of information or communication sources, including the Discrete Memoryless Sources [20]. These consist of independent random variables from a finite range of symbols (alphabet) and their respective probability distributions.

Entropy The concept of entropy corresponds to a fundamental measure in information theory, which quantifies the average uncertainty of a random variable. Considering the discrete random variable X, with a set of symbols (alphabet $\mathcal{X}$) and probability mass function $p(x) = P(X = x)$, where $x \in \mathcal{X}$, its entropy is defined by:

$$H(X) = -\sum_{x} p(x) \log_2 p(x). \tag{3.1}$$

It is important to note that this measure does not depend on the symbols from the alphabet $\mathcal{X}$ of the random variable X, but it is a function of their probabilities. The base of the logarithm in Equation (3.1) determines the unit in which entropy is expressed. The most common is the binary logarithm which causes the entropy to be measured in bits. This unit can be used to define entropy as the average number of bits needed to describe the outcome of a random variable [18]. Some of the properties of $H(X)$ include its non negativity $(H(X) \geq 0)$, its upper bound based on the number of symbols (N) in the alphabet $(H(X) \leq \log_2 N)$, and the indetermination convention $0 \log_2 0 = 0$ [25]. In general, the entropy of a random variable is maximum when all symbols have the same probability, meaning that the uncertainty of the outcome is high, since the events are equally likely. On the other hand, if one of the symbols has probability equal to one, and all the others have zero probability to occur, there is no uncertainty in the outcome of the random variable, which means that the entropy is zero.

Joint Entropy Considering two discrete random variables X and Y and their joint probability $p(x, y) = P(X = x, Y = y)$, where $x \in \mathcal{X}, y \in \mathcal{Y}$, their joint entropy is defined by:

$$H(X, Y) = -\sum_x \sum_y p(x, y) \log_2 p(x, y). \tag{3.2}$$

Note that if X and Y are independent, their joint probability is $p(x, y) = p(x)p(y)$ and, consequently, their joint entropy is equal to the sum of the two individual entropies: $H(X, Y) = H(X) + H(Y)$.

Conditional Entropy The conditional entropy, $H(X|Y)$, corresponds to the uncertainty of the random variable X, knowing the outcome of the random variable Y. Its definition is:

$$H(X|Y) = -\sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(y)}, \tag{3.3}$$

and it measures the amount of information required to describe the outcome of X, given that the value of Y is known. Note that, according to the Bayes' theorem, the conditional probability is defined by $p(x|y)p(y) = p(x, y)$ and the conditional entropy corresponds to $H(X|Y) = H(X, Y) - H(Y)$. In addition, if X and Y are independent, then $H(X|Y) = H(X)$, which means that the knowledge about Y has no impact on the uncertainty of the random variable X.

Mutual Information The dependence between two random variables can be measured using the mutual information. This concept quantifies the amount of information that one random variable gives about another and it is defined by:

$$I(X; Y) = \sum_x \sum_y p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)}. \tag{3.4}$$

This quantity is non-negative $(I(X; Y) \geq 0)$ and symmetric for X and Y: $I(X; Y) = I(Y; X)$, seeing that it represents the information shared by both variables. Based on the Bayes' theorem for entropies, the mutual information can be expressed as $I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$. Since $I(X; Y)$

measures how much the knowledge about X reduces the uncertainty about Y (or the other way around), if both random variables are independent, the mutual information is equal to zero, because they do not give any information of each other. Thorough details and demonstrations about these concepts can be found in information theory literature [18, 25].

3.2 Probabilistic Graphical Models

Uncertainty is an inevitable aspect of most real-world applications, as a consequence of the limitations in the information available, in the ability to model the systems and, in their non-deterministic nature [44]. Probabilistic graphical models attempt to describe the behaviour of complex systems using a graph-based framework for representing the probability distributions. This type of representation provide a model framework which is transparent (easy to understand and explain), effective for inference (knowledge obtained from the distribution), and data-driven (constructed by learning from data) [44].

Bayesian networks

Bayesian networks are probabilistic graphical models for describing complex domains, and they can be used to represent the information about an uncertain system [44]. The Bayesian network representation consists of a directed acyclic graph G, characterized by a set of nodes $\mathcal{N} = \{X_1, X_2, ..., X_n\}$ and a set of directed edges E. Figure 3.1 includes an example of a $G = (\mathcal{N}, E)$, where each node (vertex) corresponds to a random variable X_i, and the edges (arrows), that connect the nodes in a specific direction, describe the probabilistic dependencies between the random variables. For example, if the nodes X_1 and X_2 are connected through an edge from the first to the latter, this means that there is a statistical dependence between those two random variables. In particular, the outcome of the random variable X_2 is dependent on the value of X_1. In this case, X_1 is considered a parent of X_2, and X_2 is called a descendant of X_1.

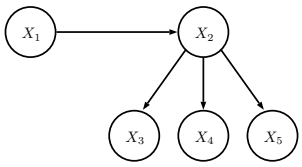


Figure 3.1: Example of a Bayesian network.

For each node, two sets can be defined: the set of parents Π_{X_i} (nodes from which the current node is connected to) and the set of non-descendants Φ_{X_i} (variables in the graph that are not connected from the current node). The structure of a Bayesian network is based on the independence assumption that each node X_i is conditionally independent of the set of non-descendants Φ_{X_i}, provided that the set of parents Π_{X_i} is known [15]. The group of local probability models, representing the dependence of each variable X_i on Π_{X_i}, specifies the parameters for quantifying the network structure [8]. These form the

set of conditional probability distributions $\Theta = \left\{\theta_{X_i|\Pi_{X_i}}\right\}_{i\in\{1,\ldots,n\}}$, where:

$$\theta_{X_i|\Pi_{X_i}} = P(X_i = x_i|\Pi_{X_i} = \omega_i), \tag{3.5}$$

associated to each node X_i and conditioned on its set of parents Π_{X_i}.

A Bayesian network $\mathcal{B} = (G, \Theta)$ consists of the direct acyclic graph structure G together with the set of parameters Θ. The unique joint probability distribution of a given collection of random variables, defined by this representation, is calculated as [44]:

$$P_{\mathcal{B}}(X_1, ..., X_n) = \prod_{i=1}^{n} P_{\mathcal{B}}(X_i|\Pi_{X_i}) = \prod_{i=1}^{n} \theta_{X_i|\Pi_{X_i}}. \tag{3.6}$$

For a given dataset D, the problem of learning a Bayesian network consists of designing the $\mathcal{B} = (G, \Theta)$ that best represents D, according to a scoring function. The scoring function corresponds to the search guide parameter for evaluating the effectiveness of the network in representing the data. Some of the scoring algorithms used for learning Bayesian networks are based on information theory concepts, such as Log-Likelihood (LL), Normalized Maximum Likelihood (NML), Akaike Information Criterion (AIC), Mutual Information Test (MIT), and Bayesian Information Criterion (BIC), the latter also referred to as Minimum Description Length (MDL) [15].

When the structure of the network is fixed, the parameters Θ that maximize all the above scores for a given dataset are those described by the observed frequency estimates.

Theorem 1 ([44])**.** Consider the direct acyclic graph structure G and the dataset D. The Bayesian network $\mathcal{B} = (G, \Theta)$, with a fixed structured G, that maximizes the likelihood of observing D is such that:

$$\hat{P}_{\mathcal{B}}(X_i = x_i|\Pi_{X_i} = \omega_i) = \frac{|D_{x_i,\omega_i}|}{|D_{\omega_i}|}, \tag{3.7}$$

for which $|D_{x_i,\omega_i}|$ represents the number of instances in D, where X_i takes the value x_i and its parents (Π_{X_i}) take the value ω_i. Similarly, $|D_{\omega_i}|$ denotes the number of instances in D, where the parents of X_i take the value w_i.

This probabilistic graphical model is useful for real-world applications that deal with a certain amount of uncertainty. In most situations, the available observations are insufficient for an undeniable decision about the real state of the system. Medical diagnosis is an example of this issue, since the same symptoms may be related with multiple diseases, and the prognosis is never part of the observations [44]. In the context of time series classification and particularly in early classification, the uncertainty is not only associated with the prediction of the class labels, but also with the reliability of the anticipated outcome.

Model Selection

The problem of model selection consists of using a given data for choosing the best model from a set of alternatives [1]. In general, the true model (i.e. the process which generated the data) is unknown

and the goal is to find the most suitable candidate (with estimated parameters) to the data from the list of options (multi-model inference), using a model selection criterion [13]. This criterion measures the quality of each model in fitting the data, while taking the complexity into account. Although the increase of the complexity is expected to improve the model suitability to the data, its generalization is hampered as well as its ability to deal with noise (overfitting). Overall, the ideal goal is not to model the data, but instead, to model the information in the data [13]. Two model selection estimators are considered in the scope of this thesis: Bayesian Information Criterion (BIC), also known as Minimum Description Length (MDL), and Akaike Information Criterion (AIC).

Minimum Description Length (MDL) This is an important concept in information theory, with applications in probability theory, specifically in the context of model selection. The MDL principle is known as an Occam's razor approach to select, for a given dataset, the best fitting model and its parameters. It states that, for a certain data and a number of alternative models, the best option corresponds to the simplest model [18].

In the problem of learning a Bayesian network, the Bayesian Information Criterion (BIC) is also known as the MDL scoring function because of their coincident results. It is concerned with analysing the tradeoff between the log-likelihood of the dataset D (the effectiveness of the fit to the data) and the complexity of the model $\mathcal{B}$. Due to the similarity with the Minimum Description Length, from this point on, the reference used is MDL score. This scoring function is defined as [44]:

$$MDL(D|\mathcal{B}) = LL(D|\mathcal{B}) - \frac{\log_2 N}{2}|\mathcal{B}|, \tag{3.8}$$

where N corresponds to the size of the data, and $|\mathcal{B}|$ represents the model dimension (number of independent parameters in $\mathcal{B}$). The log-likelihood term quantifies the amount of information required to describe the dataset D, using the set of conditional probability distributions Θ. Conversely, the penalty term measures the amount of information needed to encode the model $\mathcal{B}$, which represents the size of the representation $\mathcal{B}$ [15]. It is desired the most effective fit to the dataset, provided that the complexity of the model is as low as possible. For that reason, the dependence of a variable on its parents increases the $MDL(D|\mathcal{B})$, and the complexity of the network decreases this score [44]. Thus, the optimal model complexity is obtained with the maximization of Equation (3.8) over the list of candidate models.

Akaike Information Criterion (AIC) Similarly to the MDL scoring function, the Akaike Information Criterion (AIC) [4] corresponds to a measure of the quality of statistical models for describing a given dataset. Since it provides a model selection estimator, the AIC is a commonly used scoring algorithm for analysing the tradeoff between the model quality of the fit to the data and the complexity of the model [44]. In the problem of learning a Bayesian network, the difference between MDL and AIC is associated to the penalty applied to the number of independent parameters $|\mathcal{B}|$. The AIC scoring function can be defined as [15]:

$$AIC(D|\mathcal{B}) = LL(D|\mathcal{B}) - |\mathcal{B}|. \tag{3.9}$$

In Equation (3.8), the second term quantifies the amount of information required to encode the model $\mathcal{B}$, where each parameter in the set Θ is considered to use $\frac{1}{2}\log_2 N$ bits. Conversely, in Equation (3.9) each parameter of Θ is considered to use 1 bit. This means that the penalization on the number of independent parameters is stronger in the MDL scoring function than in the AIC score. Likewise, the best model corresponds to the one that maximizes Equation (3.9).

Comparison between MDL and AIC Literature complies with the fact that these two criteria demonstrate different properties for model selection and that they are appropriate according to specific conditions [13, 81, 74].

According to Vrieze [74], MDL is considered to be consistent in selecting the true model, with probability close to one, given that the true model is in the set of candidate models. Consistency is a property of model selection criteria that include a complexity penalty which varies with the dimensionality of the data (number of instances). Since AIC has a constant penalty, a more general estimation of the true model is selected, with probability different than zero. On the other hand, if the true model is not in the set of alternatives, AIC is considered to be effective, since it selects the model that minimizes the mean squared error of the estimation.

Regarding their derivation perspectives, Burnham et al. [13] pointed out that MDL is usually preferred because of its Bayesian approach. Nevertheless, they state that both criteria can be justified and derived within either a Bayesian or a non-Bayesian (frequentist) framework. From their point of view, what distinguishes MDL from AIC is related with the objective true model. In general, MDL is effective for a fixed and finite dimensional true model, while AIC is convenient for true models with complex parameters [74]. However, both criteria are unsuitable for dealing with low dimensional datasets for which the number of instances is close to the number of parameters to estimate [65].

Burnham et al. [13], conclude that the comparison between these two model selection criteria should depend on the context (nature of the true model) and on the conditions (performance measures).

3.3 Multivariate Correlations

From a statistical point of view, the concept of correlation between variables attempts to measure the relationships and dependencies among them. According to Koller and Friedman [44], the correlation between two variables is associated with two situations: when one variable causes the other, or when both variables result of the same origin. The knowledge of how the variables are related, as well as of what inferences can be made about their causal relationships, is useful for drawing conclusions about potential predictive relationships to be analysed and exploited. For example, the electrical energy consumed by a given house may be influenced by the weather conditions outside. Extreme temperatures may cause a more significant electricity demand for heating or cooling the house. This relation between electrical energy consumption and weather conditions indicates a correlation between these two variables, meaning that a variation in one quantity has an impact on the other. In real-world applications, the study and the awareness of these relationships can provide relevant information.

For a finite set of discrete random variables $S = \{X_i\}_{i=1,...,n}$, with joint probability distribution $P_S(X_1, ..., X_n)$, the total correlation between those variables can be defined as [18]:

$$I(X_1, ..., X_n) = \sum_{i=1}^{n} H(X_i) - H(X_1, ..., X_n). \tag{3.10}$$

In this case, the mutual information measures the dependencies among the variables, i.e. the amount of information that these quantities give about each other.

Let a structural relation R be a subset of the system S. Its joint probability distribution corresponds to the marginal distribution from S:

$$P_R(X_{R_1}, ..., X_{R_k}) = \sum_{X_i \notin R} P_S(X_1, ..., X_n), \tag{3.11}$$

where k is the number of elements in R.

Definition 1. A structure associated to the system S with underlying joint probability P_S is a pair $(\mathscr{S}, P_{\mathscr{S}})$ where $\mathscr{S} = \{R_j\}_{j=1,...,k}$ is a collection of structural relations and $P_{\mathscr{S}}$ is (another) joint probability distribution over S such that:

1. No structural relation $R_i \in \mathscr{S}$ is contained in another ($\forall_{i,j}\ R_i \nsubseteq R_j$);
2. Every $X_i \in S$ is included in at least one $R_j \in \mathscr{S}$;
3. $P_{\mathscr{S}}$ consists of the solution to the optimization problem:

$$\begin{aligned} &\underset{P \in \mathcal{P}}{\text{maximize}} \quad H(P) \\ &\text{subject to} \quad \sum_{X_i \notin R_j} P_{\mathscr{S}}(X_1, ..., X_n) = \sum_{X_i \notin R_j} P_S(X_1, ..., X_n) \quad \forall R_j \in \mathscr{S}, \end{aligned}$$

where $\mathcal{P}$ is the set of probability distributions of the variables from S.

For example, from the set of discrete random variables $S = \{X_1, X_2, X_3, X_4\}$, some admissible structures $\mathscr{S}$ correspond to $\{\{X_1, X_2, X_3\}, \{X_4\}\}$, $\{\{X_1, X_2\}, \{X_3, X_4\}\}$ or $\{\{X_1, X_2\}, \{X_1, X_4\}, \{X_2, X_3, X_4\}\}$. Conversely, $\mathscr{S} = \{\{X_1, X_2, X_3\}, \{X_1, X_3\}, \{X_4\}\}$ is not an acceptable structure since the relation between X_1 and X_3 is included in two structural relations, which represents a transgression of the first property. Similarly, $\mathscr{S} = \{\{X_1, X_2\}, \{X_2, X_4\}\}$ does not consist of a proper structure because the variable $X_3 \in S$ is not part of any structural relation from $\mathscr{S}$, as required by the second statement.

The probability distributions from the set $\mathcal{P}$, associated to the multiple structures of the system S, take into consideration the existing correlations in all the relations $R_j \in \mathscr{S}$. From Equation (3.10), for a finite set of random variables, the total correlation is defined as a function of their joint distribution. The constraint of the optimization problem (third property of a structure) guarantees that the probability distributions of each $R_j \in \mathscr{S}$ corresponds to the respective marginal distribution from S.

For a given system $S = \{X_i\}_{i=1,...,n}$ and an associated set of structural relations $\mathscr{S} = \{R_j\}_{j=1,...,k}$, the mutual information $I(S)$ represents the maximum amount of information that the variables X_i from

S provide about each other. On the other hand, $I(\mathscr{S})$ quantifies the information described by the correlations inside the structural relations R_j. The difference $I(S) - I(\mathscr{S})$ measures the knowledge of the dependencies and relationships between the variables of S that are not included in the relations that compose $\mathscr{S}$. Through Equation (3.10), this value can be described by the difference in entropy:

$$I(S) - I(\mathscr{S}) = \left(\sum_{i=1}^{n} H(X_i) - H(S)\right) - \left(\sum_{i=1}^{n} H(X_i) - H(\mathscr{S})\right) = H(\mathscr{S}) - H(S). \tag{3.12}$$

Seeing that the entropy quantifies the average uncertainty of a random variable, $H(\mathscr{S}) - H(S)$ is always non-negative, because $H(S)$ consists of the lowest possible average number of bits required to describe the random variables from S. Similarly, this difference represents the information given by the existing correlations in S, that is not incorporated in the structural relations from $\mathscr{S}$. In the last property of a structure, the optimization problem consists of maximizing the entropy of the probability distribution, in the interest of finding the $P_{\mathscr{S}}$ that corresponds to the least correlated probability distribution that takes into consideration the structural relations $R_j \in \mathscr{S}$.

Multivariate Correlations for Early Classification

In the context of sequence classification, consider a time series T (Equation (2.7)), representing the evolution of the variable X over time, and its respective class label C (Equation (2.9)), acting as another variable correlated with T. The set of X_k can be viewed as a collection of time dependent discrete random variables, for which a joint probability distribution can be defined. The correlation between any two variables (e.g. X_1 and X_2) measures the influence that the value of X at one time point has on the value of X at another instant (e.g. the dependence of X_2 on X_1). Note that, since a time series is chronologically organized, it is relevant to analyse the dependency of variables on their early states, i.e. the degree of dependence of X at a certain time point on the value observed at a previous instant. Similarly, the correlation between C and X_k quantifies the influence that the variable X at time point k has on the class label.

In sequence classification, the analysis of the relation between these variables is of great interest. Particularly in the early classification context, the focus is to study systems where the class labels verify a high dependence on a certain amount of early states of X_k, while the remaining time points are dispensable for a satisfactory classification.

Consider the finite set of discrete random variables S to be composed of the time series T together with its respective class label C. The system, with an associated joint probability distribution $P_S(X_1, X_2, ..., X_L, C)$, where L represents the time series length, is defined as:

$$S = \{X_1, X_2, ..., X_n, X_{n+1}, ...X_L, C\}, \tag{3.13}$$

for which n corresponds to a specific instant in the time series, designated early time point. The goal is

to find the value n that describes the distribution $P_S(X_1, X_2, ..., X_n, C)$ such that:

$$P_S(C|X_1, X_2, ..., X_L) \approx P_S^n(C|X_1, X_2, ..., X_n). \tag{3.14}$$

The conditional probability $P(X|Y)$ measures the likelihood of the event X, given that the event Y is observed. Therefore, $P_S^n(C|X_1, X_2, ..., X_n)$ and $P_S(C|X_1, X_2, ..., X_L)$ describe the probability of the class label C occurring, provided that some or all variables of T are known, respectively. Seeing that $n < L$, Equation (3.14) denotes that the variables $\{X_1, X_2, ..., X_n\}$ characterize the class label of the time series almost as accurately as using the entire T. In addition, a criterion is required for identifying the optimal n according to which the complexity of the model defined by P_S^n is low, provided that the majority of correlations from P_S are considered.

In general, sequence classification methods are performed in a collection of time series T_i with their respective class labels C_i, organized in a dataset D (Equation (2.6)). In some cases, the joint probability distribution P_S is not known in advance, thus it has to be computed from the data, through maximum likelihood estimation. In particular, given a dataset D, with size w, as the system S, the distribution P_S that maximizes the likelihood of D is such that:

$$\hat{P}_S(X_1 = x_1, \ldots, X_L = x_L, C = c) = \frac{|D_{x_1,\ldots,x_L,c}|}{w} \tag{3.15}$$

for which $|D_{x_1,\ldots,x_L,c}|$ is the number of instances in D, where each X_i takes the value x_i and C takes the value c.

Given the system S, described in Equation (3.13), the set of structural relations, defined by:

$$\mathscr{S}_n = \{\{X_1, ..., X_n, X_{n+1}, ..., X_L\}, \{X_1, ..., X_n, C\}\}, \tag{3.16}$$

depends on the value of n and it corresponds to a structure that respects the previously described properties. Considering $A_n = \{X_1, ..., X_n\}$, $B_n = \{X_{n+1}, ..., X_L\}$ and $C = \{C\}$[1], the structure is represented as:

$$\mathscr{S}_n = \{\{A_n, B_n\}, \{A_n, C\}\}. \tag{3.17}$$

The structural relation A_n contains the information about the evolution of the variable X until the time point n, i.e. the early states of the collection of time series. On the other hand, B_n describes the remaining instants of T_i which can be viewed as the knowledge about the later states of the variable X. Finally, C represents the class label information from the collection of time series. The structure $\mathscr{S}_n$ can be seen as a simplified model of the system S. It is expected to include the correlations between the early and the later information about the time series (A_n and B_n), as well as between the early states of T_i and the knowledge about their classes (A_n and C). Conversely, the correlations between B_n and C are not preserved because the idea is to study the possibility of describing the class from the early states A_n, while neglecting the information from B_n. The probability distribution of $\mathscr{S}_n$ is obtained based

[1]The notation is purposely overloaded, as it is common in probability theory to represent a singleton random vector with the element of the singleton.

on Theorem 2 and considering the Bayesian network represented in Figure 3.2.

Theorem 2 ([47])**.** Consider the Bayesian network $\mathcal{B}_n = (G_n, \Theta_n)$ with G_n given by Figure 3.2 and Θ_n calculated according to Theorem 1. Let $\mathcal{B}_n$ represent the dataset D as the system S, with underlying probability given by $\hat{P}_S$, as in Equation (3.15). The structure $(\mathscr{S}_n, P_{\mathscr{S}_n})$ over S has a probability distribution equal to the joint probability distribution of $\mathcal{B}_n$, that is, $P_{\mathscr{S}_n} = P_{\mathcal{B}_n}$.

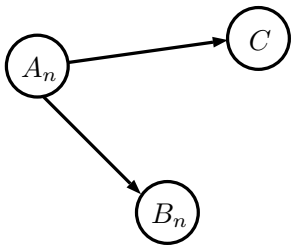


Figure 3.2: Bayesian network representation of the structure $\mathscr{S}_n$ from the system S.

Thus, from the set of parents $\Pi_{A_n} = \emptyset$, $\Pi_{B_n} = \{A_n\}$ and $\Pi_C = \{A_n\}$, and through Equation (3.6), the joint probability distribution is given by:

$$P_{\mathscr{S}_n} = P(A_n)P(B_n|A_n)P(C|A_n). \tag{3.18}$$

From Equation (3.12) and for each value of n, the difference of entropy applied to these context can be represented as:

$$I(S) - I(\mathscr{S}_n) = H(\mathscr{S}_n) - H(S) = H(C|A_n) - H(C|A_n \cup B_n). \tag{3.19}$$

The conditional entropy is used to quantify the uncertainty about the classes of the collection of time series, given that T_i is fully or partially known. On the one hand, $H(C|A_n)$ consists of the amount of information required to predict the class labels, provided that the time series are known until the time point n. On the other hand, $H(C|A_n \cup B_n)$ corresponds to the amount of information needed to describe C_i, based on the knowledge of the entire T_i. The difference between these two conditional entropies measures the knowledge that the whole time series provides about the classes (i.e. the correlation between C and $A_n \cup B_n$) which is not represented by the incomplete data (i.e. the correlation between C and A_n). Thus, Equation (3.19) can be viewed as the lack of information caused by describing the structural relation C from A_n, i.e. the loss of knowledge for using the collection of time series only until the early time point, in the classification process.

In addition to earliness in predicting the classes, the goal consists of finding the value n for which $\mathscr{S}_n$ represents the system S with a reasonable complexity. Since this can be seen as a problem of learning the Bayesian network from Figure 3.2, both the MDL and the AIC scoring functions are applied to the multivariate correlations for early classification approach, in the interest of finding the best fitting model. These scores are used as two criteria for choosing the early time point, such that the selection of the model takes its simplicity into consideration. From Equation (3.8) and considering the probability

distribution $P_{\mathscr{S}_n}$, described in Equation (3.18), the MDL scoring function is defined as:

$$MDL(D|\mathscr{S}_n) = \frac{\log_2 w}{2}|\mathscr{S}_n| - \sum_{i=1}^{w} \log_2 \left[p(C|A_n)p(B_n|A_n)p(A_n)\right], \tag{3.20}$$

where w is the number of instances in the dataset D, $|\mathscr{S}_n|$ denotes the number of independent parameters in the model, and $P_{\mathscr{S}_n}$ is associated to the model $\mathscr{S}_n$, which describes the system S as a representation of the given data. Similarly, the AIC score, applied to this context, is defined as:

$$AIC(D|\mathscr{S}_n) = |\mathscr{S}_n| - \sum_{i=1}^{w} \log_2 \left[p(C|A_n)p(B_n|A_n)p(A_n)\right]. \tag{3.21}$$

As represented in the direct acyclic graph structure from Figure 3.2, the goal is to analyse how the structural relation A_n is able to describe C, while the correlation between B_n and C is neglected. For this reason, the computation of the network complexity only considers the relation between the early states and the class labels:

$$|\mathscr{S}_n| = |\{A_n, C\}| = ||A_n|| - 1 + (||C|| - 1)\,||A_n|| = ||A_n|| \times ||C|| - 1, \tag{3.22}$$

where $||A_n||$ and $||C||$ denote the number of distinct observations in the structural relation A_n and C, respectively. In Equations (3.20) and (3.21), the first term quantifies the complexity of the model, i.e. the amount of information required to encode not only $\mathscr{S}_n$, but also the data given $\mathscr{S}_n$. The second term measures the log-likelihood of the data based on the model, i.e. the amount of information needed to represent the dataset D according to the probability distribution $P_{\mathscr{S}_n}$. While n increases, the size of A_n becomes larger, the number of correlations is higher and, consequently, the complexity of the model increases. In addition, the more information about the time series there is, the better the correlations describe the data, which means a decrease in the number of bits needed to describe C from A_n. The difference between these two terms describes the tradeoff between the model complexity and the effectiveness of the fit to the data.

Note that Equations (3.20) and (3.21), correspond to the symmetric of the definitions described in Equations (3.8) and (3.9), respectively. This means that, the best model is the one that minimizes the scoring functions. The simplest model, that is able to use the least amount of correlations while it maintains a distribution as close to the original as possible, is found through minimizing both $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$.

Early classification analysis

In the interest of verifying the reliability of this early classification approach, an investigation on the performance of multiple classifiers is done, while varying the length of the time series. Seven classifiers are considered (Table 3.1), using the default parameters and stratified cross-validation with 10 folds.

Classifier	Method	Description
NB	Probabilistic Bayes classifier	Naïve Bayes
BN	Probabilistic Bayes classifier	Bayes Net
SMO	Support vector machines	Sequential Minimal Optimization
J48	Decision tree classifier	C4.5 decision tree
REPTree	Decision tree classifier	Reduces Error Pruning Tree
RandFor	Decision tree classifier	Forest of multiple random trees
kNN	Distance-based classifier	k-Nearest-Neighbor

Table 3.1: Description of the classifiers used for comparing with the proposed method.

As explained in Section 2.1, Naïve Bayes (NB[1]) and Bayes Net (BN[2]) are both statistical classifiers. In the learning step, they build a probabilistic model with the data attributes. Then, based on that model, they perform classification, through the computation of probabilities. The strategy used by the Support Vector Machines (SVMs) is to construct a decision boundary (maximum marginal hyperplane) that separates the training set in two classes, while maximizing the margin between both of them. A mathematical representation is computed and used for testing the instances, in the classification step. SMO[3] corresponds to the implementation of a sequential minimal optimization algorithm [61] for training an SVM classifier. Note that the default conditions include the kernel exponent set to one (linear), and pairwise classification is used in the case of multi-class datasets. Other classifiers include the construction of a decision tree, which is a diagram of feature evaluation nodes, linked by outcome branches. A new instance has its attributes tested in the tree and, according to the branch results, a class label is assigned. J48[4] generates a decision tree based on the C4.5 algorithm [64]. REPTree[5] (Reduced Error Pruning Tree) is considered a fast decision tree learner because it uses information gain as splitting criterion [43]. RandForest[6] constructs a collection of decision trees and the classification step is based on the combination of all results [12]. Finally, distance-based classifiers use a measure of proximity, such as Euclidean distance or Dynamic Time Warping (DTW), to assign class labels to new instances. kNN[7] uses one nearest-neighbour ($k = 1$) for performing classification [2].

3.4 Implementation

The proposed algorithm is implemented in Java language, using some functionalities of Weka Data Mining Software[8] [36]. The Multivariate Correlations for Early Classification (MCEC) program[9], summarized in Algorithm 1, receives as input a comma-separated values (CSV) file, containing the time series and the respective class labels. Each line is expected to correspond to one instance, for which the last col-

[1] http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html
[2] http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/BayesNet.html
[3] http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html
[4] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html
[5] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/REPTree.html
[6] http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html
[7] http://weka.sourceforge.net/doc.dev/weka/classifiers/lazy/IBk.html
[8] https://www.cs.waikato.ac.nz/ml/weka/
[9] https://github.com/joaopbeirao/MCEC-algorithm

umn corresponds to the class attribute. The columns must include the features grouped per time point, chronologically organized. In addition, the number of attributes (dimensions) is also required as input. The time series can be univariate or multivariate, however, they must be of fixed length. Features are allowed to be categorical or numeric, but the dataset cannot contain missing values, since the algorithm is not provided with any imputation procedure. Furthermore, the numeric attributes must be discretized.

The outcomes of the difference in entropy, log-likelihood, MDL score, AIC score and classification accuracy, all for $n \in \{1, ..., L\}$, are outputted from the Java program in text files. An additional Matlab script is provided for generating the five graphs for representing the results.

The Appendix A includes a detailed explanation of the proposed method applied to a synthetically generated dataset. For clarification purposes, the functioning of the algorithm is expounded through calculation descriptions and graph analysis.

Algorithm 1 Multivariate Correlations for Early Classification (MCEC).

Input:

D: dataset with time series and respective class labels;
N: number of dimensions (features).

Output:

Vector with the difference in entropy values;
Vector with the log-likelihood values;
Vector with the MDL score values;
Vector with the AIC score values;
Vector with the early classification accuracy analysis values.

1: **for** $n \in \{1, ..., L\}$ **do**
2: Separation of data from D in five groups: $\{A_n\}$, $\{C\}$, $\{A_n, C\}$, $\{A_n, B_n\}$ and $\{A_n, B_n, C\}$
3: Count number of occurrences of each case in each group
4: Calculate the probability values: $P(A_n = a)$, $P(A_n = a, B_n = b)$, $P(A_n = a, C = c)$ and $P(A_n = a, B_n = b, C = c)$ according to Equation (A.5)
5: Compute $H(C|A_n) - H(C|A_nB_n)$ according to Equation (A.4)
6: Count number of independent parameters: $||A_n||$ and $||C||$
7: Compute $|\mathscr{S}_n|$ and $LL(D|\mathscr{S}_n)$ according to Equation (3.22) and (A.15)
8: Compute $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$ according to Equation (3.20) and (3.21)
9: Compute classification accuracy with the time series until n
10: Output the five vectors

Chapter 4

Experimental Results

In the attempt to validate the proposed method, this chapter presents the experiments performed with the MCEC algorithm. At first, synthetically generated datasets were used for examining the impact of the data dimensionality variation on the behaviour of this method, as well as on its computation time. Then, the performance results on datasets from online repositories were depicted and analysed. These databases comprise univariate and multivariate time series. For the first type of data, a comparison with a state-of-the-art approach from the early classification literature is enclosed. In addition, based on all experiments, a statistical study on the tradeoff between the two fundamental challenges in early classification is described. Finally, the MCEC method was applied to a real case scenario: a clinical dataset with the information about patients suffering from Rheumatoid Arthritis. All the experiments included in this chapter were conducted using a PC computer with an Intel Core i7-2677M @ 1.80GHz CPU and with 4GB RAM memory.

4.1 Synthetic data

This section describes the empirical study of the proposed method on synthetically generated datasets. In order to evaluate the ability of the MCEC algorithm, the methodology used in Appendix A, for producing synthetic data, is replicated. The procedure is based on the exclusive disjunction and it allows an interpretation of the results in comparison with the expected outcomes. The parametrization of the data generator enables the variation on common time series dataset aspects: number of features per time point (N), length of the time series (L) and number of instances (w). Moreover, two additional variables are included: the number of randomly generated columns (x) and the percentage of noise in the dataset ($pNoise$). Recall the data type is boolean for all features and all datasets contain 2 classes. According to the specified parameters, a database is created, with w time series, each with N attributes per time point and length equal to L. The value of x represents the number of initial instants that are randomly generated. The following time points are computed as the XOR of the x previous ones. For the multivariate case, where each instant is composed of a set of features, the process is maintained for each attribute independently. The class labels are computed with the same use of the exclusive disjunction,

however, for $N \geq 2$, another XOR is applied to the collection of features, in order to obtain only one value for the class attribute. Aiming for providing more realistic data, the noise percentage causes a number of arbitrary positions to be changed, i.e. 0 becomes 1 and vice versa.

An explanatory example of these randomly generated datasets is provided in Table A.1. The idea is to produce a set of time series where the class labels are a function of the x initial time points, in the interest of analysing if the proposed algorithm is able to recognize this correlation and consequently the early classification opportunity.

Variation of data size

At a first stage, the impact of the dimensional parameters variation on the MCEC method behaviour is analysed. Seeing that the two model selection criteria used in the proposed approach are sensitive to the data size, the variation of w is studied, under different conditions. Therefore, the output graphs are examined except for the classification accuracy, since the intention is to explore how the system is affected by modifications in the dimensionality of the dataset. The absolute values of the log-likelihood and of both scoring functions increase in proportion with w. Because of that, feature scaling normalization, given by:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{4.1}$$

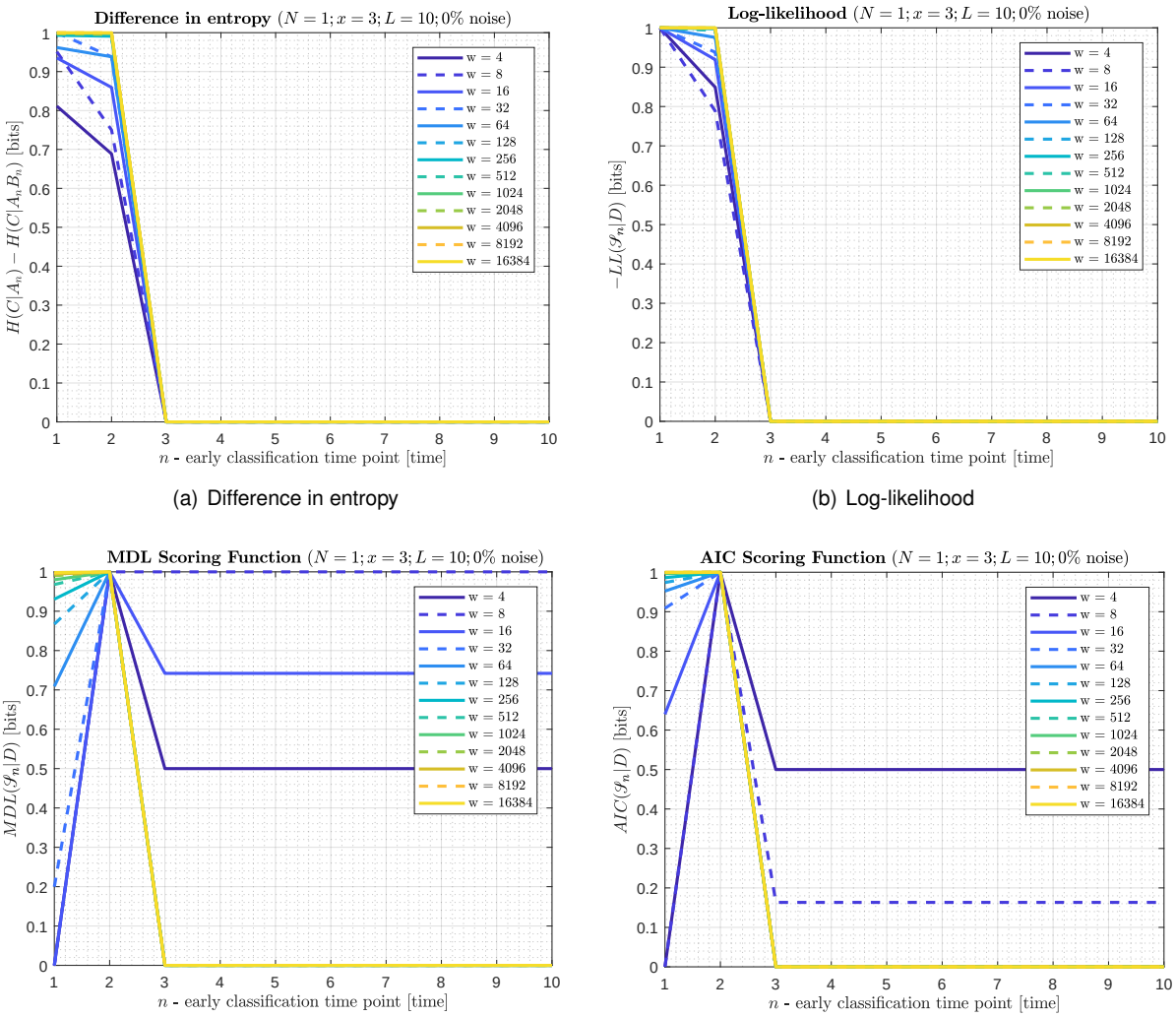
is applied to the results, for comparing the relative behaviour of the quantities. Only the entropy graph includes the non-normalized values, on account of being a difference between two variables of the same order of magnitude.

Figure 4.1 represents the behaviour of the four measures under investigation, for datasets with different number of instances. Three columns are randomly generated ($x = 3$), which means that a feasible prediction of the class labels is expected using only the first three time points. This figure describes the univariate case ($N = 1$), for a fixed time series length ($L = 10$), with no addition of noise ($pNoise = 0\%$). In order to explore a dimensional range for the data size, the values for w comprise a set of powers of two: $\{2^2, 2^3, ..., 2^{14}\}$.

As previously mentioned, $H(C|A_n) - H(C|A_nB_n)$ quantifies the lack of knowledge caused by describing the classes using the time series in the dataset only until time point n. From Figure 4.1(a), the variation of w does not extensively affect the difference in entropy. Since $H(C|A_n) - H(C|A_nB_n) = 0$ for $n \geq 3$, there seems to be enough information to predict the class labels, with the first three time points. The variation of entropy from $n = 1$ to $n = 2$ is sharper for lower values of w, and it becomes null while the number of instances increases.

Furthermore, $-LL(D|\mathscr{S}_n)$ describes the amount of information needed to represent the dataset D using the model $\mathscr{S}_n$. Figure 4.1(b) demonstrates that the data is completely depicted by the structure $\mathscr{S}_3$, seeing that the log-likelihood is zero from $n = 3$ forward. The behaviour of this measure is very similar to the difference in entropy, since they both quantify how good the model fits the data.

Considering the scoring functions, they both describe a tradeoff between the complexity of the model and its suitability for representing the data. In the early classification context, the lowest value of both



(a) Difference in entropy

(b) Log-likelihood

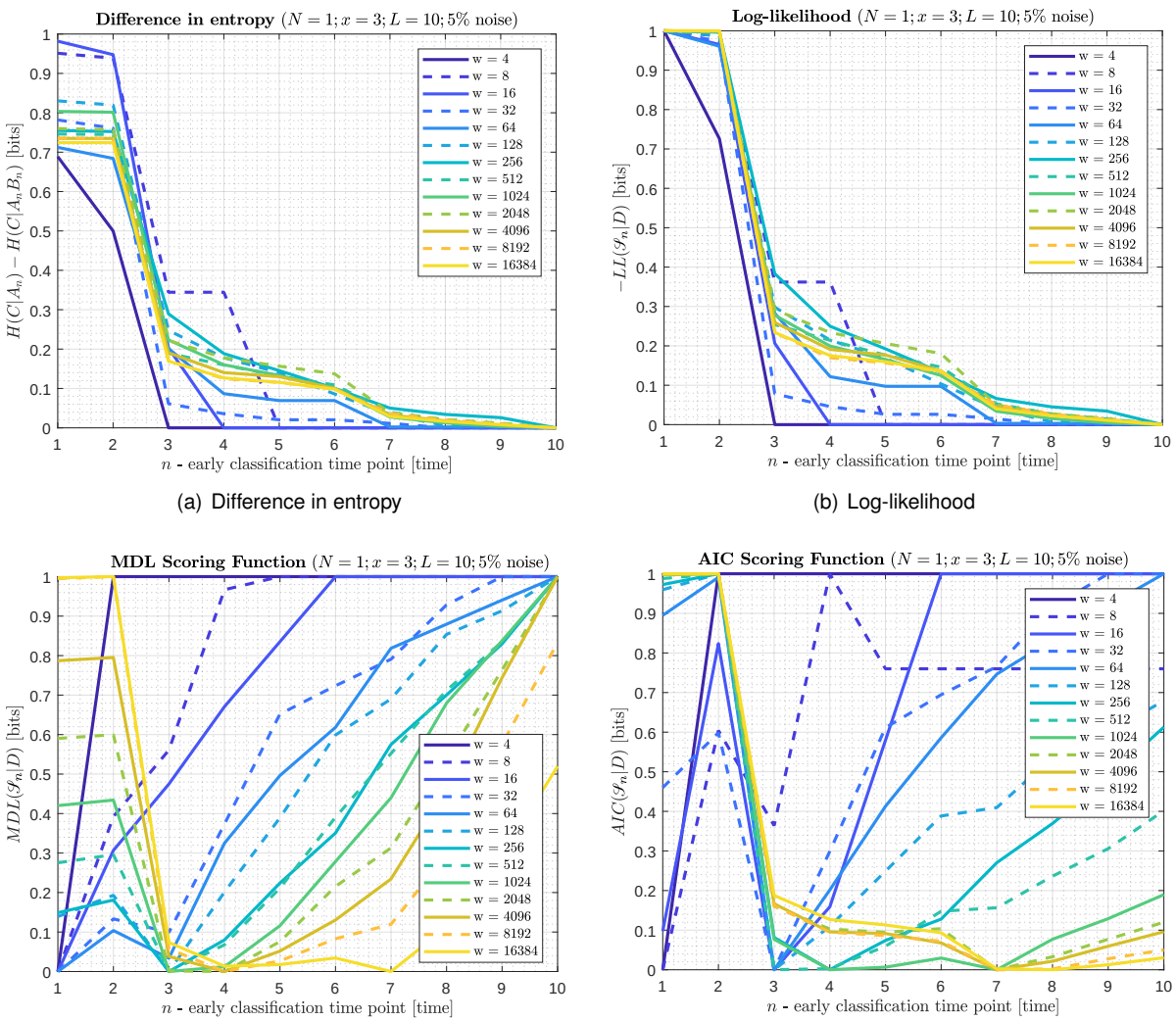
(c) MDL

(d) AIC

Figure 4.1: Variation of w for synthetic datasets with $N = 1$, $L = 10$, $x = 3$, $pNoise = 0\%$.

scores corresponds to the time point from which additional information can be disregarded. While the graph from Figure 4.1(c) shows that MDL has a minimum at $n \geq 3$ for $w \geq 32$, the one from Figure 4.1(d) displays AIC achieving it for $w \geq 16$. In the two cases, the scores are constant from the point where they attain the lowest value on. This is because each time point is a function of what is behind, since it consists of the exclusive disjunction of the three previous instants. Consequently, the number of independent parameters in group A_n ($||A_n||$) is constant for $n \geq 3$, i.e. the number of distinct cases in the list $\mathcal{A}_n$ stabilizes from that point on.

Figure 4.2 describes the experimental tests in datasets with the same parameters than the ones used in Figure 4.1, except for the percentage of noise. With $pNoise = 5\%$, a more realistic environment is simulated. The difference in entropy (Figure 4.2(a)) and the log-likelihood (Figure 4.2(b)) have a smoother decreasing behaviour and more difficulty in reaching zero, in particular for higher values of w. However, in general, the most expressive reduction in both cases is verified from $n = 2$ to $n = 3$. This indicates that, although the lack of information is minimized as more of the time series is observed, for a certain threshold the graphs show an early classification opportunity. Similarly to the 0% noise case, the



(a) Difference in entropy

(b) Log-likelihood

(c) MDL

(d) AIC

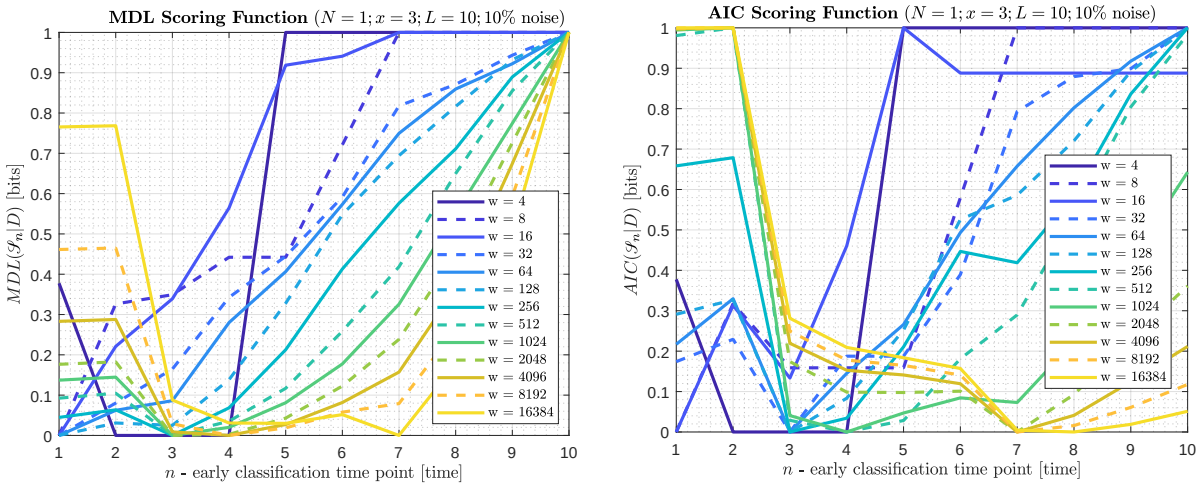
Figure 4.2: Variation of w for synthetic datasets with $N = 1$, $L = 10$, $x = 3$, $pNoise = 5\%$.

increase on the number of instances is followed by a stabilization from $n = 1$ to $n = 2$, which means that few knowledge is gained with the use of only the first two time points. The behaviour of both measures seems to be convergent for $w \to \infty$.

Regarding the scoring functions, Figure 4.2(c) and Figure 4.2(d) show a high variance of MDL and AIC with the data size, respectively. For few instances, the lowest value is obtained at $n = 1$, which means that the model, considered the best in terms of complexity and fitness to the data, is the one with merely the first time point. In this case, a proper model selection is impracticable, since the samples available are insufficient to conveniently represent the dataset, that is, the data does not contain enough information. MDL displays a minimum at $n \geq 3$ for $w \geq 128$ (higher than for 0% noise) and AIC for $w \geq 16$ (the same as for 0% noise). The lowest value of $MDL(D|\mathscr{S}_n)$ is attained at $n = 3$ for $w \in \{128, ..., 1024\}$, at $n = 4$ for $w \in \{2048, ..., 8192\}$, and at $n = 7$ for $w = 16384$. The minimum of $AIC(D|\mathscr{S}_n)$ is reached at $n = 3$ for $w \in \{16, ..., 128\} \cup \{512\}$, at $n = 4$ for $w = 256$, at $n = 7$ for $w \in \{1024, ..., 8192\}$, and at $n = 8$ for $w = 16384$. For the experimental range of sample size, the results suggest that, although the AIC score contains a minimum at $n \geq 3$ for lower values of w, in general, it has a greater tendency to obtain values

of n further from the expected outcome ($n = 3$). In both cases, the increase on the number of instances causes the best model (early classification time point) to have a more significant deviation from the true distribution.

In order to examine the impact of the noise in the inferences drawn about the model selection criteria, similar experiments were performed on datasets with $pNoise$ equal to 10% and 25%. Concerning the difference in entropy and the log-likelihood measures, the decreasing behaviour is preserved, although the variation becomes less accentuated with noisier data. Moreover, since noise causes higher uncertainty, the jump from $n = 2$ to $n = 3$ is not so expressive, and consequently, the early classification opportunity at $n = 3$ is less obvious. For $pNoise = 10\%$, while the lowest value of MDL (Figure 4.3(a)) at $n \geq 3$ is attained for $w \geq 256$ (higher than for 5% noise), in AIC (Figure 4.3(b)) this minimum is reached for $w \geq 32$ (higher than for 5% noise). Note that, in Figure 4.3, the curve $w = 4$ also displays a minimum for



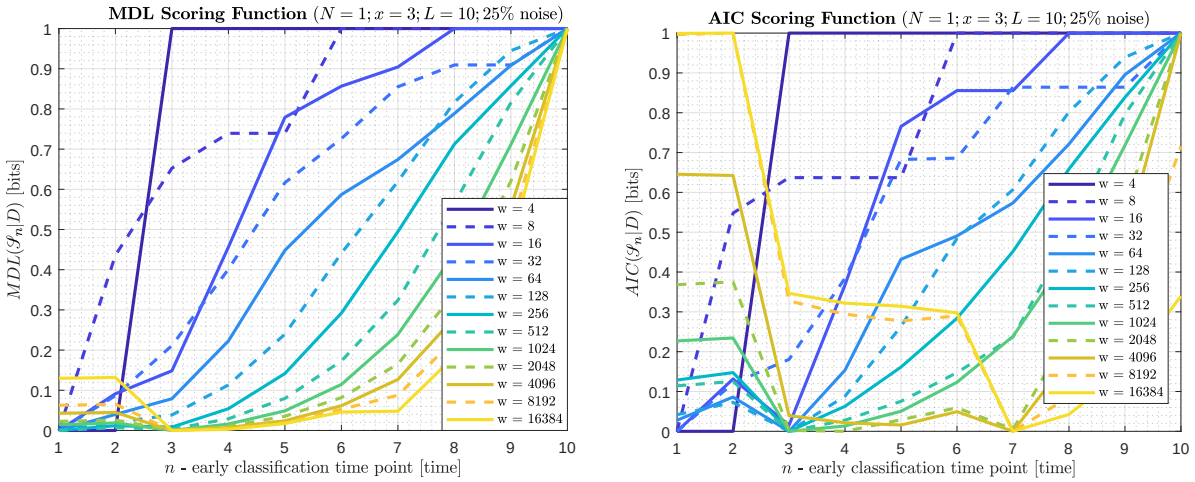
(a) MDL

(b) AIC

Figure 4.3: Variation of w for synthetic datasets with $N = 1$, $L = 10$, $x = 3$, $pNoise = 10\%$.

$n \in \{2, ..., 4\}$. This event should not be considered relevant, since the dataset is so reduced that noise has an unbalanced influence in the results. Proof of that is, for example, the curve $w = 8$, which does not have a minimum at $n \geq 3$. For $pNoise = 25\%$, the lowest value of MDL (Figure 4.4(a)) at $n \geq 3$ is attained for $w \geq 1024$ (higher than for 10% noise), whereas in AIC (Figure 4.4(b)) it is reached for $w \geq 64$ (higher than for 10% noise). While Figure 4.3(a) ($pNoise = 10\%$) shows the MDL graph with some ambiguity in selecting the true model for larger values of w, Figure 4.4(a) ($pNoise = 25\%$) describes the same score identifying $n = 3$ as the early time point with zero error. Furthermore, a lower deviation from the true distribution is also observed in AIC, for $pNoise = 25\%$ (Figure 4.4(b)), in comparison with the case with $pNoise = 10\%$ (Figure 4.3(b)).

A few considerations about the response of the MCEC method to variations on the dimensionality of the dataset can be referred. Firstly, with regard to the univariate context and for given datasets with time series of fixed length, the number of instances has a significant impact on both scoring functions and a not so expressive influence in the difference in entropy and the log-likelihood measures. In addition, the results suggest there is a value of w from which the minimization of the model selection criteria is



(a) MDL

(b) AIC

Figure 4.4: Variation of w for synthetic datasets with $N = 1$, $L = 10$, $x = 3$, $pNoise = 25\%$.

achieved at $n = 3$. This indicates that the number of instances in a dataset influences the effectiveness of the scoring functions in selecting the true distribution. As mentioned in Section 3.2, both model selection criteria are unsuitable for reduced datasets, where the number of instances is not considerably higher than the number of estimated model parameters, leading to overfitting.

Concerning the comparison between both criteria, the experiments demonstrate that, in general, AIC outperforms MDL, for more reduced datasets. However, for larger values of w, the AIC evidences a greater deviation from the true distribution, having the tendency to choose more complex models than MDL. This fact verifies the MDL reputation of being more consistent than AIC in selecting the underlying model among the candidates, provided that the true model is in the set of alternatives.

In general, the sharp decreases in $H(C|A_n) - H(C|A_nB_n)$ and in $LL(D|\mathcal{S}_n)$ for $n = 3$, together with the minimum values depicted in both scores, give confidence in the early classification potential of the proposed method. On the other hand, the experiments demonstrate that the decision upon the early time point (n) is not always unanimous among the three measures that compose the MCEC algorithm. This means that, in some cases, the instant from which the remaining of the time series in the dataset can be neglected is not trivially identified.

Additional experiments were performed to the proposed method in order to analyse the impact of the variation of two other parameters: the number of features (N) and the time series length (L). The objective consists of not only examine the early prediction opportunity, but also continue the investigation on how the dimensionality of the dataset influences the minimization of the model selection criteria.

Seeing that the algorithm is capable of handling multivariate time series ($N \geq 2$), the study involves randomly generated datasets with $N \in \{1, 2, 3, 5\}$, while $x = 3$, $L = 10$ and $pNoise \in \{0\%, 5\%\}$. With regard to the difference in entropy and the log-likelihood, the decreasing behaviour of these measures is not substantially affected by the variation on the number of features per time point. In general, the reduction within $n \in \{2, ..., 4\}$ is expressive, which indicates that, in this time period, there occurs a distinguishable decrease on the amount of information needed to predict the time series classes of the dataset.

Considering the scoring functions, the value of w from which both criteria display a minimum at $n \geq 3$ increases with N. Table 4.1 confirms this inference by describing the variation on the number of instances for which the minimum of MDL and AIC is attained at $n \geq 3$, according to the number of features per time point. In fact, for all experiments, the minimums were reached for $n = 3$. Moreover, AIC

N	$pNoise$	w	
		MDL	AIC
1	0%	32	16
	5%	128	16
2	0%	1024	128
	5%	2048	256
3	0%	8192	1024
	5%	32768	4096
5	0%	> 131072	65536
	5%	> 131072	> 131072

Table 4.1: Values of w from which the scoring functions display a minimum at $n \geq 3$. Parameters: $x = 3$, $L = 10$, $pNoise \in \{0\%, 5\%\}$ and $N \in \{1, 2, 3, 5\}$.

seems to be less conditioned by N than MDL, since its values of w are always lower. This suggests that, though the dataset size impacts the effectiveness of the model selection criteria, the early classification time point is identified with substantial consistency, under certain conditions.

Another parameter examined was the length of the time series in the dataset. Although the proposed method requires the data to have a fixed L, this value can vary from database to database. In order to evaluate how the variation of the time series length affects the MCEC algorithm, several experiments were performed with $L \in \{6, 10, 18, 38, 78, 158\}$, for the following fixed parameters: $x = 3$, $N = 1$ and $pNoise \in \{0\%, 5\%\}$. Similarly to the study on multiple values of N, the analysis also focused on investigating if the early prediction is practicable and if the dataset size conditions the minimization of the two criteria.

Concerning the curves from $H(C|A_n) - H(C|A_nB_n)$ and $LL(D|\mathscr{S}_n)$, the impact of the variation of L is not significant. In most cases, an emphatic decrease is verified afresh around $n \in \{2, ..4\}$, which represents a reduction on the information needed for predicting the classes, over that time period. Table 4.2 includes the values of w from which both scoring functions show a minimum at $n \geq 3$. Unlike the results from Table 4.1, the lowest values of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$ were not consistently obtained for $n = 3$, but instead, they deviated from the true distribution ($n \in \{4, 6, 7, 8\}$) with the increase of the number of instances. The results demonstrate that the time series length does not considerably condition the ability of both criteria to select the best model, since the values of w in Table 4.2 do not significantly change with the variation of L. Although not always according to the expected model ($n = 3$), and occasionally in a non-unanimous decision situation, the early classification opportunity is observable in the majority of the cases.

In sum, these are the conclusions that can be drawn from the performed experiments based on the variation of the dataset dimensionality:

L	$pNoise$	w	
		MDL	AIC
6	0%	32	16
	5%	64	32
10	0%	32	16
	5%	128	16
18	0%	32	16
	5%	128	16
38	0%	32	16
	5%	32	32
78	0%	32	16
	5%	64	32
158	0%	32	16
	5%	64	16

Table 4.2: Values of w from which the scoring functions display a minimum at $n \geq 3$. Parameters: $x = 3$, $N = 1$, $pNoise \in \{0\%, 5\%\}$ and $L \in \{6, 10, 18, 38, 78, 158\}$.

1. The number of instances (w) and the number of features per time point (N) have a significant impact on both model selection criteria and a not so expressive influence in the difference in entropy and log-likelihood measures.

2. The time series length (L) does not considerably affect none of the four measures.

3. The number of instances (w) in a dataset conditions the effectiveness of the scoring functions in selecting the true model, meaning the early classification time point (n).

4. AIC is less conditioned by w than MDL, but the latter identifies the true model more consistently than the first score.

5. The decision upon the early classification time point can be ambiguous, that is, the three main measures that compose the MCEC algorithm can propose distinct values of n.

Computation time

Concerning the computation time analysis of the MCEC algorithm performance, Table 4.3 includes an empirical study on the variation of three data size parameters: number of features (N), time series length (L) and number of instances (w). Each value is calculated as the average time duration of the proposed method on ten synthetically produced datasets. In all experiments, the first three time points are randomly generated ($x = 3$) and no noise is added ($pNoise = 0\%$). The range of values for the three dimensional parameters was intended to describe a comprehensive, representative and uniform set of experiments, for comparing the impact of the data size on the MCEC computation time.

For the variation on the number of features, the constant parameters are set as $L = 10$ and $w = 500$. The values of N allow a comparison between the univariate context ($N = 1$) and higher dimensions ($N \in \{10, 100, 1000\}$). The cases $N = 10000$ and $N = 100000$ are not considered due to their computationally

Parameters	Values					
	1	10	100	1000	10000	100000
Number of features (N)	0.576 s	0.868 s	4.071 s	58.840 s	–	–
Time series length (L)	–	0.133 s	4.577 s	319.215 s	–	–
Number of instances (w)	–	0.047 s	0.081 s	0.613 s	2.259 s	16.093 s

Table 4.3: Computation time of the MCEC algorithm performance on synthetically generated datasets. Fixed parameters: $x = 3$ and $pNoise = 0\%$. Constant parameters: $L = 10$ and $w = 500$, for the variation of N; $N = 1$ and $w = 500$, for the variation of L; and $N = 1$ and $L = 10$, for the variation of w.

demanding requirements. Table 4.3 shows an increase of the computation time with the number of features. From $N = 1$ to $N = 10$, the difference is subtle, however, for $N = 100$ and $N = 1000$ a significant growth is verified. In the MCEC algorithm each dimension is computed in block, i.e. each time point is represented by all its attribute values. For this reason, the variation on the time results is mainly related with the data file reading and with the formation of groups A_n, B_n and C.

With regard to the time series length analysis, the datasets hold $N = 1$ and $w = 500$. The range of L provides insight on short ($L = 10$), medium ($L = 100$) and long ($L = 1000$) time series situations. Both $L = 10000$ and $L = 100000$ events are not examined for the same reason as in the number of features analysis. Moreover, a time series with only one time point is not applicable in this synthetically generated dataset context, therefore, $L = 1$ is not verified. In Table 4.3, the results demonstrate that the time series length extensively impacts the computation time of the proposed method. Seeing that the early classification time point (n) is examined from 1 until L, the entire time series is analysed. Thus, an increase on L represents an enlargement on the size of the observation window, where the early classification opportunity is being investigated.

Regarding the experiments on datasets with different number of instances, the constant parameters are defined as $N = 1$ and $L = 10$. Datasets with distinct amounts of available information are represented in the collection of values for $w \in \{10, ..., 100000\}$. The results from Table 4.3 suggest that the algorithm is fairly robust with regard to the number of instances. All samples are analysed together, since this method measures the amount of information contained in the entire database, and the groups are constructed according to time points, not instances. Therefore, the computation time is mostly due to the dataset scanning.

4.2 Benchmark data

This section includes the experiments of the MCEC algorithm on datasets from repositories available online. Two types of time series were tested: univariate and multivariate. For the first, twenty datasets were examined and the outcomes compared with a state-of-the-art method in the early classification literature [79]. Regarding the multivariate case, six datasets were analysed. Lastly, the obtained results were confirmed with statistical significance, concerning the tradeoff between earliness and accuracy.

4.2.1 Univariate Time Series

The UEA & UCR Time Series Classification Repository[1] [6] provides more than 90 time series datasets for research into time series classification. For analysing the performance of the MCEC algorithm, Table 4.4 lists the experimental results on 20 benchmark datasets from the referred repository. This subset of examples is considered comprehensive and representative, since it comprises a diverse range of both dimensional parameters and classification conditions. Each dataset is composed of numeric univariate time series ($N = 1$) with a fixed length, and their respective class labels. For each example, a training set and a test set are provided separately. The preprocessing of the data included the aggregation of both subsets in one single dataset (data integration). In addition, a supervised discretization by Fayyad & Irani's MDL method [24] was performed to the numeric attributes (data transformation), using the filter from the Weka Data Mining Software in Java. None of the datasets contained missing values, therefore no imputation was required.

From the MCEC algorithm, for each dataset, three values for the early classification time point (n) were extracted. As previously mentioned, n represents the time point from which the information contained in the time series is considered expendable. The first value is obtained from the difference in entropy measure: n such that $H(C|A_n) - H(C|A_nB_n) = 0.3 \times [H(C|A_1) - H(C|A_1B_1)]$, which means that n corresponds to the time point where a reduction of 70% from the initial value of entropy is verified, henceforth called $CH-70$. The second and third values are a result of the minimization of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$, respectively, i.e. n consists of the time point where the criteria is minimum. A percentage value is associated with the early classification time point:

$$\text{Earliness}[\%] = \frac{n}{L} \times 100. \tag{4.2}$$

This measure quantifies the amount of the time series considered necessary for a satisfactory prediction, with respect to its total length (L). The lower the value of Earliness, the less time points are considered required, and the earlier the classification is expected to be performed.

The data classification was performed through stratified cross-validation with 10 folds, using seven different classifiers (Table 3.1), set with default parameters, as explained in Section 3.3. The classifier with the highest percentage of correctly classified instances (Equation (2.1) at page 9) was selected. The three measures from the MCEC algorithm determine the instant from which the information in the time series can be neglected. Based on the three values of n, the selected classifier was used for the classification of the data. At most, three derivative subsets were considered, each with L defined as one of the early classification time points computed by the proposed method. The accuracy results obtained for the chosen classifiers, together with the earliness outcomes (Equation (4.2)) are included in Table 4.4. The results from Table 4.4 describe the MCEC algorithm effort in attempting early classification, based on the analysis of the information contained in the datasets. The column "Full" contains the outcomes for the complete time series and it is used as a reference framework. Moreover, the "MCEC algorithm" columns indicate the results for the incomplete time series, where L is defined according to the values

[1] www.timeseriesclassification.com

of n. The symbol (*) means that more than one classifier achieved the best percentage of correctly classified instances. In addition, the column "ECTS" includes the results from the state-of-the-art method [79] used for performance comparison.

Concerning the "Full" column, the percentage of correctly classified instances is higher or equal than 90% in 14 of the 20 cases, reaching 100% in 2 datasets ("Coffee" and "Meat"). The worst examples in terms of accuracy correspond to "Computers" and "FiftyWords", with a percentage lower than 70%. In the first situation, given the number of classes, the classifier is not capable of performing a decent classification with the data available. However, in the latter, an accuracy of 67.62% is not completely unsatisfactory, provided that there are 50 distinct class labels. Therefore, the results from the "Full" column of Table 4.4 confirm not only the quality of most databases for classification purposes, but also the ability of the selected classifiers in classifying the full-length time series data.

Dataset		MCEC algorithm			Full	ECTS
		$CH-70$	MDL	AIC		
Adiac	n	14	1	1	–	–
37 classes	Earliness	7.95%	0.57%	0.57%	–	70.05%
$L = 176$	Accuracy	41.23%	17.54%	17.54%	77.47%	65.43%
$w = 781$	Classifier	SMO	SMO*	SMO*	SMO	–
ArrowHead	n	37	1	4	–	–
3 classes	Earliness	14.74%	0.40%	1.59%	–	78.20%
$L = 251$	Accuracy	68.25%	53.56%	56.87%	93.37%	89.55%
$w = 211$	Classifier	RandFor	RandFor*	kNN	RandFor	–
Beef	n	118	1	5	–	–
5 classes	Earliness	25.11%	0.21%	1.06%	–	60.18%
$L = 470$	Accuracy	60.00%	40.00%	48.33%	75.00%	55.00%
$w = 60$	Classifier	kNN*	kNN*	kNN*	kNN*	–
BeetleFly	n	431	107	333	–	–
2 classes	Earliness	84.18%	20.90%	65.04%	–	78.68%
$L = 512$	Accuracy	85.00%	67.50%	87.50%	95.00%	60.00%
$w = 40$	Classifier	RandFor*	NB*	RandFor*	RandFor*	–
BirdChicken	n	267	201	202	–	–
2 classes	Earliness	52.15%	39.26%	39.45%	–	56.41%
$L = 512$	Accuracy	77.50%	70.00%	75.00%	90.00%	82.50%
$w = 40$	Classifier	RandFor	NB*	NB*	NB*	–
Car	n	127	1	27	–	–
4 classes	Earliness	22.01%	0.17%	4.68%	–	75.62%
$L = 577$	Accuracy	69.17%	34.17%	42.50%	83.33%	77.50%
$w = 120$	Classifier	RandFor	kNN*	kNN*	kNN	–
CBF	n	8	1	3	–	–
3 classes	Earliness	6.25%	0.78%	2.34%	–	88.55%
$L = 128$	Accuracy	52.15%	44.84%	48.39%	99.68%	98.92%
$w = 930$	Classifier	SMO	NB*	NB*	SMO	–
ChlorineConc	n	48	1	38	–	–
3 classes	Earliness	28.92%	0.60%	22.89%	–	23.88%
$L = 166$	Accuracy	82.05%	54.89%	74.86%	98.98%	93.10%
$w = 4307$	Classifier	RandFor	RandFor*	RandFor	RandFor	–
Coffee	n	43	23	26	–	–
2 classes	Earliness	15.04%	8.04%	9.09%	–	72.50%
$L = 286$	Accuracy	89.29%	76.79%	80.36%	100.00%	92.67%
$w = 56$	Classifier	RandFor	RandFor*	NB*	RandFor*	–

Continued on next page

Dataset		MCEC algorithm			Full	ECTS
		$CH-70$	MDL	AIC		
Computers	n	303	1	2	–	–
2 classes	Earliness	42.08%	0.14%	0.28%	–	92.27%
$L = 720$	Accuracy	67.60%	58.20%	65.80%	66.00%	59.40%
$w = 500$	Classifier	RandFor	RandFor*	RandFor*	RandFor	–
Earthquakes	n	18	2	4	–	–
2 classes	Earliness	3.52%	0.39%	0.78%	–	99.13%
$L = 512$	Accuracy	89.93%	79.14%	79.86%	99.28%	70.74%
$w = 278$	Classifier	RandFor	RandFor*	NB*	RandFor	–
ECG200	n	16	3	6	–	–
2 classes	Earliness	16.67%	3.13%	6.25%	–	67.34%
$L = 96$	Accuracy	81.00%	66.50%	81.00%	90.50%	89.00%
$w = 200$	Classifier	BN	kNN*	SMO	kNN	–
FiftyWords	n	38	7	7	–	–
50 classes	Earliness	14.07%	2.59%	2.59%	–	77.33%
$L = 270$	Accuracy	32.16%	15.69%	15.69%	67.62%	66.96%
$w = 905$	Classifier	SMO	SMO*	SMO*	SMO	–
GunPoint	n	36	1	23	–	–
2 classes	Earliness	24.00%	0.67%	15.33%	–	57.39%
$L = 150$	Accuracy	92.00%	71.50%	83.00%	99.50%	92.00%
$w = 200$	Classifier	SMO*	SMO*	RandFor	SMO	–
Meat	n	72	1	11	–	–
3 classes	Earliness	16.07%	0.22%	2.46%	–	54.65%
$L = 448$	Accuracy	90.00%	66.67%	75.00%	100.00%	97.50%
$w = 120$	Classifier	REPTree	RandFor	SMO*	SMO*	–
OliveOil	n	55	3	6	–	–
4 classes	Earliness	9.65%	0.53%	1.05%	–	78.12%
$L = 570$	Accuracy	68.33%	55.00%	58.33%	96.67%	86.67%
$w = 60$	Classifier	RandFor	SMO*	NB*	SMO*	–
SwedishLeaf	n	7	1	2	–	–
15 classes	Earliness	5.47%	0.78%	1.56%	–	79.62%
$L = 128$	Accuracy	56.53%	29.60%	37.78%	91.02%	80.09%
$w = 1125$	Classifier	RandFor	SMO*	SMO*	SMO	–
SyntheticControl	n	5	1	2	–	–
6 classes	Earliness	8.33%	1.67%	3.33%	–	93.13%
$L = 6$	Accuracy	82.17%	49.67%	69.00%	98.83%	90.33%
$w = 600$	Classifier	BN	BN*	BN	BN	–
TwoPatterns	n	95	1	11	–	–
4 classes	Earliness	74.22%	0.78%	8.59%	–	83.80%
$L = 128$	Accuracy	56.36%	26.28%	29.20%	75.18%	96.46%
$w = 5000$	Classifier	RandFor	RandFor*	kNN	RandFor	–
Wafer	n	11	2	3	–	–
2 classes	Earliness	7.24%	1.32%	1.97%	–	65.78%
$L = 152$	Accuracy	97.91%	97.60%	97.66%	99.85%	99.68%
$w = 7164$	Classifier	kNN	RandFor*	kNN	RandFor	–

Table 4.4: Experimental results of the MCEC algorithm in 20 benchmark archive datasets, containing numeric univariate time series ($N = 1$). Three early classification time points (n) are identified. The first corresponds to the time point where a reduction of 70% from the initial value of $H(C|A) - H(C|AB)$ is verified ($CH - 70$). The second and third consist of the time point associated with the minimum value of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$, respectively. The classification of the data is performed through stratified cross-validation with 10 folds, for a collection of classifiers, set with default parameters (Table 3.1). At each experiment, the one with the best accuracy for the given data is selected. While column "Full" contains the outcomes for the complete time series, the "MCEC algorithm" columns indicate the results for the incomplete time series, where L is defined according to the values of n. The symbol (*) means that more than one classifier achieved the best percentage of correctly classified instances. For spacing reasons, "ChlorineConc" is short for "ChlorineConcentration". "ECTS" [79] is a benchmark method in the early classification literature, included for performance comparison.

For all datasets, a reduction of 70% from the initial value of $H(C|A) - H(C|AB)$ is verified, for a value of n lower than L, that is, Earliness (Equation (4.2)) is always beneath 100%. In particular, this percentage is lower than 30% in 16 of the 20 cases, and under 10% for 7 datasets. Among those 16 examples, in 8 occurrences ("ChlorineConcentration", "Coffee", "Earthquakes", "ECG200", "GunPoint", "Meat", "SyntheticControl" and "Wafer") the classifiers achieve an accuracy higher or equal than 80%, even though always beneath the same percentage for the full-length data. In fact, concerning the difference in entropy results (first column of the "MCEC algorithm"), the classification accuracy with less time points outperforms the reference value ("Full" column) only for the "Computers" dataset. This example suggests that it is possible to obtain a better classification performance using only part of the time series from the data.

Regarding the 8 cases with Earliness $< 30\%$ and Accuracy $> 80\%$, when comparing the percentage of correctly classified instances for the full-length and for $CH - 70$, the difference is beneath 20% in all cases. This means that, in these experiments, with fewer time points analysed (earlier in time), the loss in terms of classification accuracy can be diminished. For instance, in the "Wafer" outcomes, using only 7.24% of the time series, an accuracy of 97.91% is achieved, which consists of -1.94% in comparison with the full-length result.

With regard to the results of the scoring functions, the balance between the complexity of the model and its effectiveness in fitting the data is found for $n < L$ in all experiments, i.e., the second and third columns of "MCEC algorithm" denote Earliness $< 100\%$. MDL and AIC indicated the same best model (time point) among the candidates in 2 of the 20 cases: "Adiac" ($n = 1$) and "FiftyWords" ($n = 7$). For all other cases, the amount of time series used for prediction proposed by AIC is larger than the suggested by MDL, that is, $n_{MDL} < n_{AIC}$. The difference (in absolute value) between n_{MDL} and n_{AIC} varies between 1 ("BirdChicken", "Computers", "SwedishLeaf", "SyntheticControl", "Wafer") and 226 ("BeetleFly"), but it is lower or equal than 10 in 16 of the 20 cases.

Concerning the 18 datasets with $n_{MDL} \neq n_{AIC}$, the classification accuracy results of AIC outperform the ones for MDL in all cases. This suggests that, based on these experiments, AIC surpasses MDL, with respect to accuracy. Excluding the cases where $n_{MDL} = n_{AIC}$, the difference in the percentage of correctly classified instances varies from 0.72% ("Earthquakes") to 20% ("BeetleFly"). Note that in the "Earthquakes" example, in spite of the different values of n, the accuracy is very close for both model selection criteria. In this case, MDL presents a better suggestion than AIC in terms of earliness, for similar classification outcomes. However, while the values from n_{MDL} obtained an accuracy greater or equal than 70% in 5 of the 20 cases, n_{AIC} achieved it in 9 of the 20 experiments. In all situations, the percentage of correctly classified instances for both criteria is lower than for the full-length data.

In addition, the results from Table 4.4 show that $n_{MDL} = 1$ for 12 of the 20 cases, and $n_{AIC} = 1$ for only 1 dataset ("Adiac"). As mentioned in Section 4.1, while AIC is known to select a model more readily, MDL's choice is considered more consistent. The large number of examples where MDL identifies the first time point as the best alternative for earliness may indicate that, given the information available, in all those 12 cases the criterion recognized that the increase in the knowledge obtained from the data did not justify the growth in the model complexity required for describing it. Conversely, the AIC results

demonstrate a more adventurous disposition in choosing the value for n, and, in these experiments, that seems to have produced relative success.

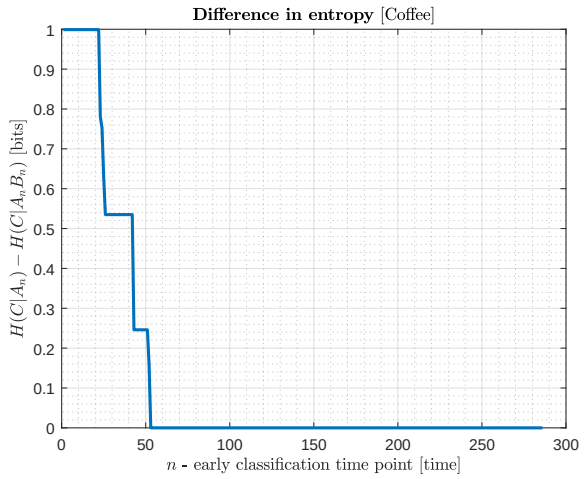
From the comparison between the three measures, $CH-70$ achieves higher classification accuracy in 18 of the 20 cases, AIC in 1 dataset ("BeetleFly"), and in 1 example ("ECG200") a draw is verified between the difference in entropy and AIC. MDL has a percentage of correctly classified instances always lower or equal than the other measures. Regarding the Earliness percentage, except for the events where $n_{MDL} = n_{AIC}$, MDL proposes always the lowest values for the early classification time point. Therefore, from the experimental tests described in Table 4.4, in general, $CH-70$ achieves better results, in terms of classification accuracy, and MDL demonstrates a superior earliness ability. AIC evidences the foremost competence in balancing these two targets. Nevertheless, the early classification capabilities of the MCEC algorithm are acknowledged, seeing that this context is based on the tradeoff between these two main objectives: accuracy and earliness.

The experimental results of the proposed method on the "Coffee" example are represented in Figure 4.5. For a more detailed analysis, one of the cases with maximum classification accuracy on the "Full" column was chosen. This data is obtained from food spectrographs, which are commonly used for classifying types of food, in safety and quality control applications. In this case, the classification problem corresponds to identify two types of coffee beans: Robusta and Arabica (2 classes).

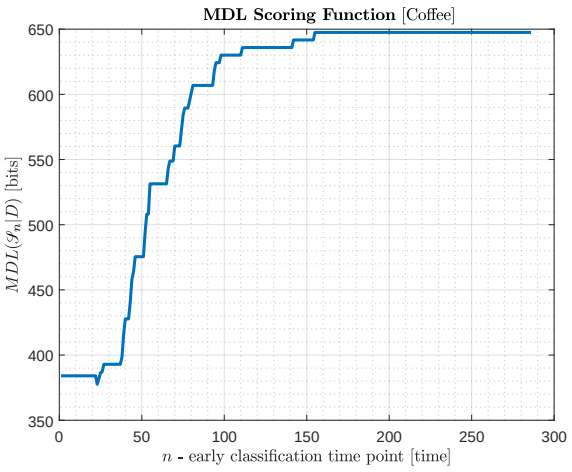
Figure 4.5(a) describes the behaviour of $H(C|A_n)-H(C|A_nB_n)$ while varying n from 1 to $L = 286$. As previously stated, this measure quantifies the lack of information caused by describing the class attribute in the data, using only the time series until time point n, instead of using all the information available in the dataset. Three decreasing jumps, followed by temporary stabilizations, are depicted in the graph: the first during $n \in \{22, ..., 26\}$, the second at $n \in \{42, 43\}$ and the third between $n \in \{51, ..., 53\}$. For $n \geq 53$, $H(C|A_n) - H(C|A_nB_n) = 0$, which in theory means that, from that time point forward, the information in the time series can be neglected, since it does not provide any more relevant knowledge about the classes. A decrease of 70% from the initial entropy value is obtained at $n = 43$, where the difference in entropy is lower than $0.3 \times 1 = 0.3$ bits.

Figures 4.5(b) and 4.5(c) represent the variation of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$, respectively, for $n \in \{1, ..., 286\}$. These measures take the complexity of the model into consideration when choosing the time point until which the information is considered to be effectively described. The value of both criteria is constant until $n = 22$. While for MDL, the minimum is reached at $n = 23$, for AIC, the lowest value is attained at $n = 26$. In both cases, this extreme is followed by an irregular growth until $n = 155$, where it stabilizes at a maximum value.

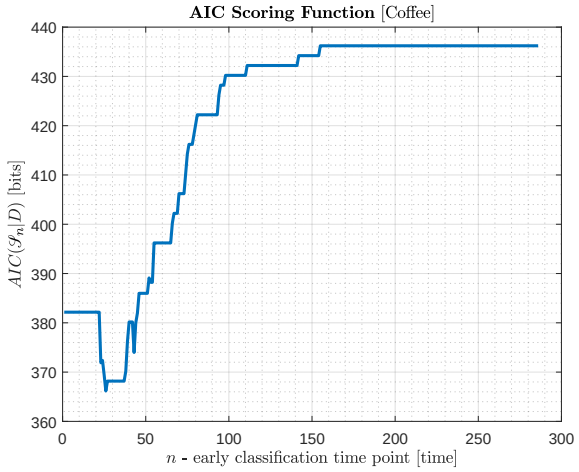
Figure 4.5(d) includes the classification accuracy of the "Coffee" dataset, considering the time series to have length L equal to $n \in \{1, ..., 286\}$. The graph only describes the percentage of correctly classified instances for the Random Forest classifier, which was the one with the best performance on the full-length data. Note that the accuracy is constant at 48.21% until $n = 22$, and it increases to 76.79% at $n = 23$. That is not only in the interval where the difference in entropy describes the first accentuated decrease, but also precisely the same instant where MDL reaches its minimum. The percentage decreases to 75% at $n = 24$, increases to 82.14% at $n = 25$, and decreases to 78.57% at $n = 26$. This last



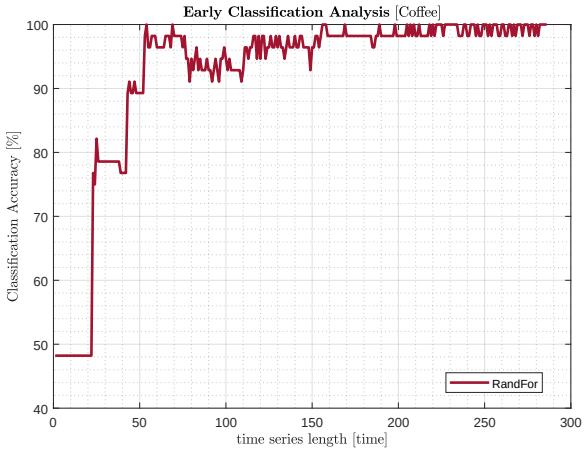
(a) Difference in entropy



(b) MDL



(c) AIC



(d) Classification results of the Random Forest classifer

Figure 4.5: Experimental results of the MCEC algorithm on the "Coffee" dataset. Parameters: $w = 56$, $L = 286$, 2 classes and numeric attributes.

time point corresponds to the instant where the minimum is attained by AIC. In fact, considering these accuracy results as a reference, the early classification time point at $n = 25$ seems to be a better alternative. The graph continues with an irregular behaviour, but there occurs another significant increase between $n \in \{42, 43\}$, from 76.79% to 89.29%. This period consists of the the exact interval where the difference in entropy reaches a 70% decrease from its initial value.

Figure 4.5(d) shows some time points for which Accuracy $= 100\%$, such as $n = 54$, $n = 69$, $n = 156$, among others. Although these correspond to instants with higher percentage of correctly classified instances, a correlation between the three measures from MCEC algorithm and the classification results is observable, given the relationship that the behaviour of these four graphs indicate. Therefore, the inferences about the early classification time point, obtained from the proposed method, seem to find meaning in the analysis of the information that a dataset of incomplete time series provide about the class attribute.

Comparison with related work

Table 4.4 includes the performance of the ECTS algorithm [79] on the selected datasets. As mentioned in Section 2.2, this is one of the benchmark methods in early classification and one of the first to formally define this data mining problem. The main difference between the MCEC algorithm in comparison with the majority of the proposed methods in the early prediction context (in particular the ECTS) is related with the classification approach. In the MCEC case, the goal consists of analysing an entire given dataset for discovering the information that each time point provides about the classes. As explained in Section 3.3, the ability to describe or predict the class attribute is examined as the instants used chronologically increase. Therefore, the proposed method is not intended to be a classifier, seeing that it does not include the training and test phases, and it does not assign a class label to a new given instance. Conversely, ECTS corresponds to an extension of the 1-nearest-neighbour classifier for early classification on univariate time series data. During the training phase, this method analyses the early classification opportunity, through the calculation of the Minimum Prediction Length (MPL), and for each new instance to be classified, the class label is assigned based on MPLs comparison.

The ECTS method was used through stratified cross-validation with 10 folds. Seeing that the early time point computed is different for each instance, in each fold, the Earliness percentage is calculated as:

$$\text{Earliness}[\%] = \frac{1}{w_{\text{test}}} \sum_{i=1}^{w_{\text{test}}} \frac{n_i}{L} \times 100, \tag{4.3}$$

where w_{test} consists of the number of instances in the test set and n_i corresponds to the computed early time point for instance i. The mean of the ten outcomes from Equation (4.3) gives the final value of Earliness, included in Table 4.4. Likewise, the Accuracy value is obtained as the mean of the percentages of correctly classified instances from all folds. Moreover, in this case, the preprocessing of the data did not include any discretization since the ECTS algorithm only deals with numeric attributes.

In 18 of the 20 cases, the MCEC algorithm shows a lower value of Earliness than the ECTS method. However, in 15 of the 20 datasets, the Accuracy percentage is higher for the ECTS approach, though always beneath the "Full" outcome, except for the "TwoPatterns" case. As previously mentioned, in general, the difference in entropy demonstrates greater results for the percentage of correctly classified instances than the other measures from the proposed method. Concerning the $CH-70$ and the ECTS outcomes, in 14 of the 20 cases, the latter obtains a superior Accuracy but a worst (higher) Earliness. The difference in the percentage of correctly classified instances between the two measures varies from 1.77% to 46.77%, and it is lower or equal than 20% in 8 of the 20 experiments. The difference in the earliness percentage between both approaches varies from 4.26% to 84.80%, and it is above 50% in 11 of the 20 datasets. This suggests that although the ECTS algorithm accomplishes more precise predictions, the difference in entropy measure demonstrates a greater anticipation ability. For instance, in the "Wafer" case, with considerably less information (minus more than 50%), the difference in the classification accuracy is beneath 5%. Moreover, for the "GunPoint" example, the same percentage of correctly classified instances is obtained for ECTS and $CH-70$ (92%), yet the latter uses less amount of time series (24% versus 57.39%).

"Earthquakes" is the only experiment where the three measures from the MCEC algorithm outperform (at the same time) the ECTS method, in terms of Earliness and Accuracy. In "Beef", "BeetleFly", "ChlorineConcentration", "Computers" and "GunPoint" some of the measures from the proposed approach obtain competitive results in comparison with the ECTS algorithm. Overall, the performance of the ECTS method surpasses the one from the MCEC algorithm. Nevertheless, it is important to note the distinction between them with regard to the classification approach. This means that, although the comparison between these two methods is relevant and informative, it has some limitations due to the dissemblances in the testing conditions. In addition, seeing that the ECTS algorithm only deals with univariate time series, it will not be used for the experiments on multivariate datasets, presented in the following section.

4.2.2 Multivariate Time Series

Seeing that the MCEC algorithm is applicable on time series with multiple features in each time point, experimental tests were also performed on multivariate data. As a supplement to a study on Multivariate Time Series Classification [7], a group of investigators gathered a collection of datasets,[1] useful for experiments on methods capable of dealing with this type of data. These examples were obtained from a variety of sources, such as archive repositories [49, 17] and other websites.[2] For analysing the performance of the proposed algorithm, six benchmark datasets were selected from the available resources, as an attempt to provide experimental results in an expansive set of conditions. Similarly to the univariate case, some preprocessing tasks were executed to these numeric multivariate time series datasets. The training and test sets were aggregated in one single database (data integration), and a supervised discretization by Fayyad & Irani's MDL method [24] was performed (data transformation). In addition, the cases which contained time series with different lengths (within the same dataset) were adjusted, that is, for each example, all instances were set to a value of L equal to the shortest sample length. This means that part of the information available was disregarded, in the interest of obtaining datasets with fixed L. In none of the cases was needed any imputation, since there were no missing values in the data. Table 4.5 lists the results of the experiments performed. The process used for the univariate tests was replicated in these six experiments. Note that "ECG" and "Wafer" in Table 4.5 have the same name as two examples in Table 4.4, but they consist of distinct datasets.

The accuracy results from column "Full" demonstrate that the selected classifiers are able to perform an acceptable classification, since in 4 of the 6 cases, the percentage of correctly classified instances is higher than 90%. The lowest value corresponds to 79.17% ("Libras") which is still quite satisfactory, given that the dataset contains 15 classes. Once more, these outcomes serve as reference values and they confirm both the competence of the data for classification purposes, and the classifiers capabilities in reliably assigning class labels.

Concerning the difference in entropy measure (column $CH - 70$), a reduction of 70% from the initial value is verified before the end of the time series, for all datasets (Earliness $< 100\%$). This per-

[1] http://www.mustafabaydogan.com/
[2] http://www.cs.cmu.edu/~bobski/

Dataset		MCEC algorithm			Full
		$CH-70$	MDL	AIC	
ECG ($N=2$)	n	13	1	3	–
2 classes	Earliness	33.33%	2.56%	7.69%	–
$L=39$	Accuracy	87.00%	77.00	80.00%	86.00%
$w=200$	Classifier	RandFor	SMO*	J48	SMO
JapaneseVowels ($N=12$)	n	2	1	1	–
9 classes	Earliness	28.57%	14.29%	14.29%	–
$L=7$	Accuracy	88.13%	85.47%	85.47%	94.53%
$w=640$	Classifier	RandFor	SMO	SMO	SMO
Libras ($N=2$)	n	14	1	1	–
15 classes	Earliness	31.11%	2.22%	2.22%	–
$L=45$	Accuracy	60.28%	30.28%	30.28%	79.17%
$w=360$	Classifier	kNN	RandFor	RandFor	RandFor
PenDigits ($N=2$)	n	3	1	1	–
10 classes	Earliness	37.50%	12.50%	12.50%	–
$L=8$	Accuracy	78.26%	47.95%	47.46%	98.45%
$w=10992$	Classifier	RandFor	RandFor	RandFor	SMO
RobotLP1 ($N=6$)	n	2	1	1	–
4 classes	Earliness	13.33%	6.67%	6.67%	–
$L=15$	Accuracy	89.77%	84.09%	82.96%	95.46%
$w=88$	Classifier	kNN	NB*	NB*	SMO
Wafer ($N=6$)	n	51	1	24	–
2 classes	Earliness	49.04%	0.96%	23.08%	–
$L=104$	Accuracy	95.31%	90.29%	93.55%	98.49%
$w=1194$	Classifier	SMO*	SMO*	RandFor	SMO

Table 4.5: Experimental results of the MCEC algorithm in 6 datasets containing numeric multivariate time series ($N \geq 2$). Three early classification time points (n) are identified. The first corresponds to the time point where a reduction of 70% from the initial value of $H(C|A) - H(C|AB)$ is verified ($CH-70$). The second and third consist of the time point associated with the minimum value of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$, respectively. The classification of the data is performed through stratified cross-validation with 10 folds, for a collection of classifiers, set with default parameters (Table 3.1). At each experiment, the one with the best accuracy for the given data is selected. While column "Full" contains the outcomes for the complete time series, the "MCEC algorithm" columns indicate the results for the incomplete time series, where L is defined according to the values of n. The symbol (*) means that more than one classifier achieved the best percentage of correctly classified instances.

centage is lower than 40% in 5 of the 6 cases, and under 30% in 2 examples ("JapaneseVowels" and "RobotLP1"). The classifiers achieve an accuracy higher than 80% in 4 experiments ("ECG", "JapaneseVowels", "RobotLP1" and "Wafer"), nevertheless, the value for the full-length data is outperformed only for the "ECG" example. When comparing the percentage of correctly classified instances for "Full" and for $CH-70$, the difference (in absolute value) is beneath 20% in 5 of the 6 cases (all except "PenDigits"). The lowest difference is found for "ECG", where using the information until $n=13$ (33.33% of the total time series length), a classification can be performed with Accuracy $=87\%$ ($+1\%$ in comparison with the full-length result). In this case, a reduction on the amount of information used, or in other words, a prediction earlier in time, is associated to a better classification performance (even though with a very subtle difference).

Regarding both model selection criteria (columns MDL and AIC), the value of n is lower than L for all datasets (Earliness $< 100\%$). The proposed early classification time points from both scoring functions are coincident in 4 of the 6 cases. In all these examples, where $n_{MDL} = n_{AIC}$, the criteria

suggestion consists of using only the first instant of the time series for classifying the data. In 2 of those 4 cases, the classification accuracy is above 80% ("JapaneseVowels" and "RobotLP1"), which corresponds to a considerably decent outcome. Table 4.5 shows $n_{AIC} = 1$ in 4 of the 6 cases, and $n_{MDL} = 1$ for all experiments. As previously mentioned, the tendency of MDL and AIC to display a minimum at $n = 1$ may be explained with the growth in the model complexity being so expressive that does not legitimate the increase on the amount of information used.

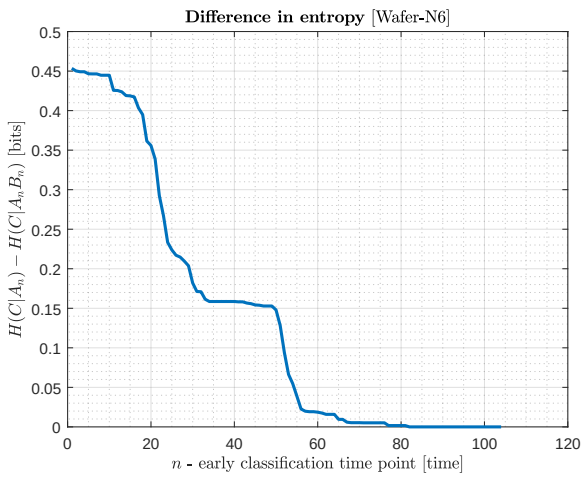
With regard to the examples where $n_{MDL} \neq n_{AIC}$ ("ECG" and "Wafer"), the amount of time series used for prediction, proposed by AIC, is always larger than the suggested by MDL ($n_{MDL} < n_{AIC}$). While for "ECG", the difference between the two early classification time points is 2, for "Wafer", it is 23. In both cases, n_{AIC} surpasses n_{MDL} in terms of classification accuracy results, yet with a value always beneath the reference ("Full" column). Similarly to what was verified in the univariate experiments, the daring disposition of AIC in choosing a value of n further in time proved successful with respect to the classification outcomes. However, seeing that the largest accuracy difference between MDL and AIC is 3.26% ("Wafer"), these results suggest that by giving priority to earliness, the percentage of correctly classified instances is not extensively affected. In fact, for both model selection criteria, the values from n obtained Accuracy $\geq 70\%$ in 4 of the 6 cases, which assigns some confidence to the MCEC algorithm in analysing the early classification opportunity.

When comparing the three measures, the difference in entropy achieves higher classification accuracy in all cases. Concerning the Earliness percentage, the model selection criteria obtain always the lowest values, and, particularly, MDL achieves the best results. In general, based on the experimental tests described in Table 4.5, the difference in entropy measure performs better with regard to classification accuracy and MDL manifests a higher disposition to earliness. However, the most efficient tradeoff between these two requirements seems to be found for AIC. These conclusions are in line with the inferences drawn from the experiments with univariate time series.

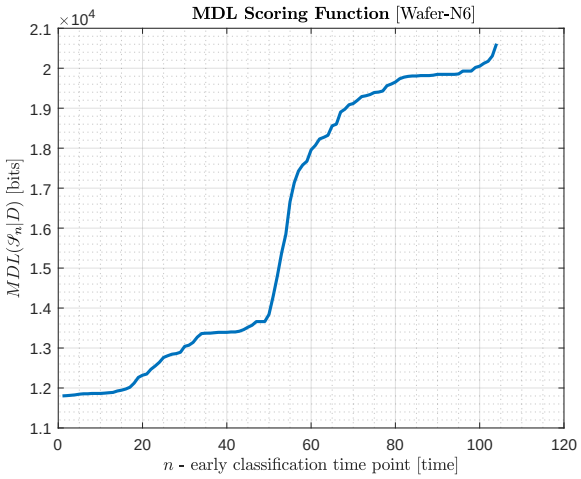
The experimental results of the MCEC algorithm on the "Wafer" dataset are depicted in Figure 4.6. With the aim of providing a more thorough analysis, the example with the greatest classification accuracy on the full-length time series is examined. This data consists of sequence measurements, from six sensors ($N = 6$), obtained during the manufacture of semiconductor microelectronics. The classification problem corresponds to distinguish the normal from the abnormal wafers (2 classes), which are caused by a number of complications in the process.

Figure 4.6(a) represents the variation of the difference in entropy for n from 1 to $L = 104$. Overall, two intervals with a significant decrease are observed in the graph: the first during $n \in \{11, ..., 34\}$, and the second between $n \in \{49, ..., 57\}$. For $n \geq 82$, $H(C|A_n) - H(C|A_nB_n) = 0$, which suggests that the last 22 time points do not contain any more useful information about the class labels. The decrease of 70% from the initial entropy value is reached at $n = 51$, where the lack of knowledge, caused by using incomplete data to describe the classes, is beneath $0.3 \times 0.4534 = 0.136$ bits.

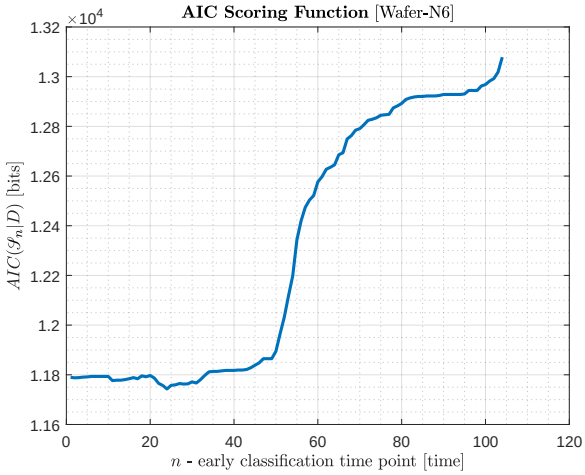
Figures 4.6(b) and 4.6(c) describe the behaviour of $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$, respectively, while varying n from $\{1, ..., 104\}$. The first graph shows an irregular growth, with intervals where the slope is steeper, and others with a lower variation. The minimum value is reached at $n = 1$, nonetheless, a small



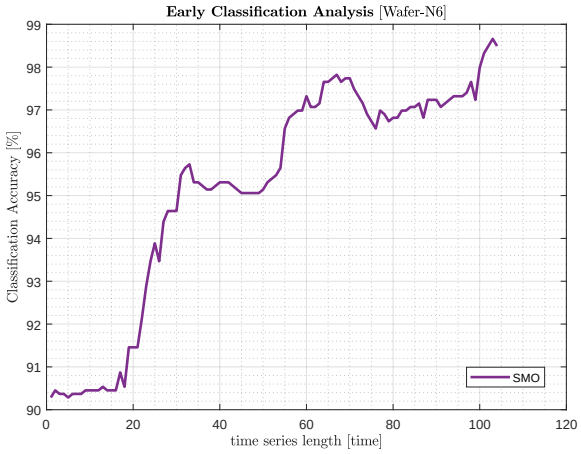
(a) Difference in entropy



(b) MDL



(c) AIC



(d) Classification results of the SMO classifer

Figure 4.6: Experimental results of the MCEC algorithm on the "Wafer" dataset. Parameters: $N = 6$, $w = 1194$, $L = 104$, 2 classes and numeric attributes.

variation in MDL is observed between $n \in \{1, ..., 17\}$, at $n \in \{34, ..., 44\}$ and during $n \in \{83, ..., 95\}$. With regard to the AIC case, its behaviour is similar to MDL from $n = 34$ forward. From the first until that time point, the variation of the graph is small, but the minimum is obtained during that period, for $n = 24$. Note that the temporary relative stabilization during $n \in \{34, ..., 44\}$ is common in both scoring functions, and in the entropy graph. In fact, one could say that a symmetric behaviour is verified between the $H(C|A_n) - H(C|A_nB_n)$ and the model selection criteria graphs. The increase on the amount of time series used, not only decreases the lack of information about the class attribute (Figure 4.6(a)), but also increases the complexity of the model that attempts to describe the data (Figures 4.6(b) and 4.6(c)).

Figure 4.6(d) includes the percentage of correctly classified instances by the SMO classifier, on the "Wafer" dataset, according to the length L equivalent to $n \in \{1, ..., 104\}$. Note that the percentage values are greater than 90% for the entire range of L, however, an irregular growth is observed, with two main jumps and two more stable periods. During $n \in \{1, ..., 16\}$, the first relatively constant interval is verified, which means that the classification accuracy at $n = 1$ is very close to the same measure at $n = 16$. This indicates that, by selecting the early classification time point as $n = 1$ (Accuracy $= 90.29\%$), MDL is

disregarding information that does not have a great impact on the accuracy, seizing the opportunity of a prediction in advance. The first increasing jump occurs between $n \in \{18, ..., 33\}$, the same interval where the minimum is attained for the other model selection criteria. Considering these accuracy results as a reference, $n = 33$ (Accuracy $= 95.73\%$) seems to be a more convenient early classification time point than the $n = 24$ (Accuracy $= 93.47\%$), proposed by AIC. This period is followed by a less changeable one at $n \in \{35, ..., 52\}$, and another accentuated growth during $n \in \{52, ..., 67\}$. The difference in entropy reaches 70% decrease from its initial value at $n = 51$, which corresponds to an accuracy of 95.31%. This early classification time point is inserted between the two main increasing jumps in the percentage value. This is closely related with one of the fundamental challenges in early classification: how much are we willing to neglect accuracy in exchange of earliness?

4.2.3 Wilcoxon signed-ranks sum test

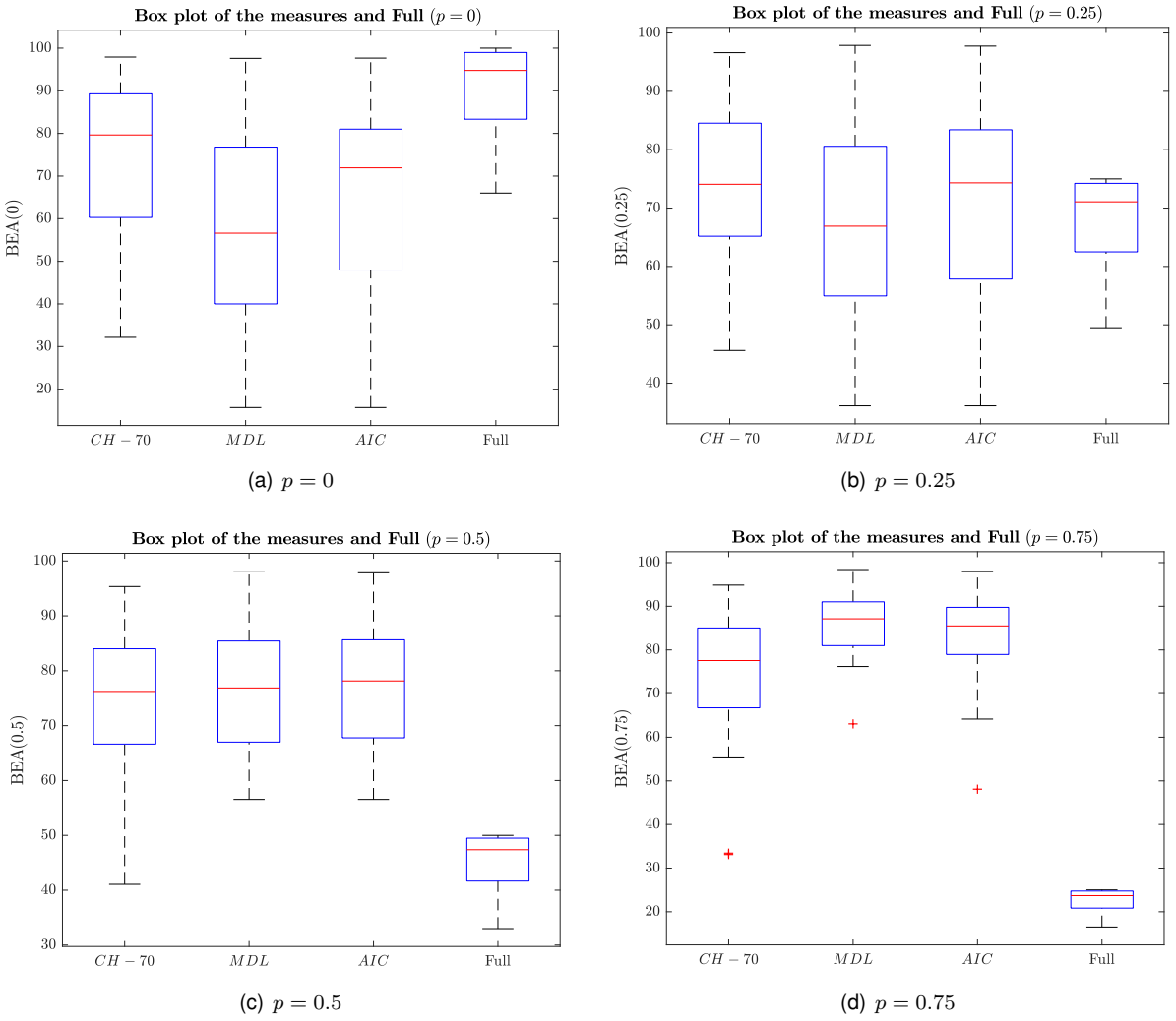
In the attempt to answer this question, the univariate and multivariate experimental results were compared with statistical significance tests in order to understand the benefit of the tradeoff between the two main goals in early classification: accuracy and earliness. Among the tested datasets, the MCEC algorithm provided a value of n, with an associated percentage (Earliness). For each situation, the group of classifiers determined the percentage of correctly classified instances (Accuracy). In addition, the classification of the full-data worked as a reference framework: no earliness and complete time series accuracy. Aiming for a representation of the balance between these two requirements, a mathematical expression can be defined as:

$$BEA(p) = p \times (100 - E) + (1 - p) \times A, \tag{4.4}$$

where E and A correspond to the Earliness and Accuracy percentages, respectively; and p consists of the weight that determines the relevance given to each variable. For instance, while $p = 0.5$ represents an equal degree of importance for E and A, $p > 0.5$ gives preference to E to the detriment of A, and $p < 0.5$ prioritizes A over E. As previously mentioned, a low value of Earliness corresponds to few time points used (earlier classification). On the other hand, the higher the Accuracy, the better the classifier is capable of making reliable predictions. Seeing that an accurate classification is desirable, as early as possible, Equation (4.4) describes the management of the two fundamental challenges of the early classification problem. Therefore, $BEA(p)$ denotes the Balance between E and A, according to the value of p.

Figure 4.7 includes the box plots of the experimental results, for the $CH - 70$, MDL, AIC and Full measures, converted through Equation (4.4), according to the weight $p \in \{0, 0.25, 0.5, 0.75\}$. The 26 datasets from Tables 4.4 and 4.5 were considered, as well as their respective values of $E =$ Earliness and $A =$ Accuracy, for each of the three measures that compose the MCEC algorithm, together with the reference framework. Note that all Full outcomes verify $E = 100\%$, since the entire time series are considered for classification. These diagrams describe the variation of statistical populations, based on their quartiles. The red line corresponds to the median value; the lower and upper limits of the blue

box indicate the 25th and 75th percentiles, respectively; the black extending lines (whiskers) denote the extreme values (maximum and minimum); and the red '+' symbols represent the data points that are considered outliers. A given data point is treated as an outlier if it is higher than $q_3 + \alpha \cdot (q_3 – q_1)$ or lower than $q_1 – \alpha \cdot (q_3 – q_1)$, where α is the maximum whisker length, and q_1 and q_3 are the 25th and 75th percentiles, respectively.[1]



(a) $p = 0$ (b) $p = 0.25$

(c) $p = 0.5$ (d) $p = 0.75$

Figure 4.7: Box plot of the experimental results, for the three measures that compose the MCEC algorithm and the reference full-data results, according to the weight $p \in \{0, 0.25, 0.5, 0.75\}$. The red line indicates the median; the lower and upper limits of the blue box correspond to the 25th and 75th percentiles, respectively; the black extending lines (whiskers) represent the extreme values (maximum and minimum); and the red '+' symbols denote the data points considered outliers. A given data point is treated as an outlier if it is higher than $q_3 + \alpha \cdot (q_3 – q_1)$ or lower than $q_1 – \alpha \cdot (q_3 – q_1)$, where α is the maximum whisker length, and q_1 and q_3 are the 25th and 75th percentiles, respectively.

For $p = 0$ (Figure 4.7(a)), the Earliness portion is completely disregarded, and the data distribution reflects the case where the Accuracy of the classification is entirely prioritized. The Full measure has the greatest median (94.77%), followed by $CH - 70$ (79.63%), AIC (71.93%) and finally, MDL (56.6%). Concerning the interquartile range (IQR), which corresponds to the difference between the upper and lower quartiles, MDL has the widest spread (36.79%), followed by AIC (33.05%), $CH - 70$ (28.98%) and

[1] `https://www.mathworks.com/help/stats/boxplot.html`

lastly, Full (15.65%). With regard to the data points range (difference between maximum and minimum values), AIC has the largest interval (81.97%), followed by MDL (81.91%), $CH-70$ (65.75%) and Full (34%). Based on these observations, one could say that Full demonstrates the best results over all other measures. Seeing that, in general, the classification accuracy is higher for the full-length data, this conclusion is in line with what would be expected. Although $E = 100\%$, the Earliness side of $BEA(0)$ is cancelled by $p = 0$. Moreover, the difference in entropy shows a superior performance in comparison with both model selection criteria, and the AIC box plot suggests that this scoring function obtains better outcomes than MDL.

Figure 4.7(b) describes the distributions of the measures, considering $p = 0.25$. In this case, more relevance is given to A, but E is not completely neglected. The AIC measure has the highest median (74.32%), followed by $CH-70$ (74.08%), Full (71.07%) and finally, MDL (66.9%). With respect to the IQR, Full has the narrowest spread (11.74%), followed by $CH-70$ (19.37%), AIC (25.58%) and lastly, MDL (25.63%). Regarding the data points range, Full has the lowest interval (25.5%), followed by $CH-70$ (51.02%), AIC (61.63%) and MDL (61.75%). According to this analysis, the difference in entropy seems to surpass all other measures. In spite of $CH-70$ having a median lower than AIC, their difference is lower than 1% and both the IQR and the data points range are larger for the model selection criterion. Additionally, the direct count of the number of times $CH-70$ has a higher $BEA(0.25)$ score than AIC corresponds to 18 in 26 cases, which corroborates the drawn conclusion. Furthermore, Figure 4.7(b) indicates that AIC has better results than both MDL and Full, but between these last two, the full-length data appears to outperform the scoring function.

For $p = 0.5$ (Figure 4.7(c)), both Earliness and Accuracy are considered equally important. The AIC measure has the greatest median (78.14%), followed by MDL (76.86%), $CH-70$ (76.05%) and finally, Full (47.38%). Concerning the IQR, MDL has the widest spread (18.42%), followed by AIC (17.86%), $CH-70$ (17.36%) and lastly, Full (7.82%). With regard to the data points range, $CH-70$ has the largest interval (54.27%), followed by MDL (41.59%), AIC (41.3%) and Full (17%). Based on the observations from the box plot, the measure Full distinctly demonstrates the worst results over all others. In this case, the E portion represents 50% of the $BEA(0.5)$ score. Since, $E = 100\%$ for all full-length data experiments, though the classification accuracy is higher, the contribution in terms of Earliness is always null. Regarding the other three measures, the comparison is considerably dubious, since the medians and the IQR values between $CH-70$, MDL and AIC are fairly close. Seeing that the data points range is wider for the difference in entropy than for the scoring functions, one could say that $CH-70$ is outperformed by both model selection criteria. Furthermore, AIC seems to obtain a superior performance in comparison with MDL because the latter has a lower median and greater values of both IQR and data points range. In addition, the direct count of the number of times AIC has a higher $BEA(0.5)$ score than MDL is 13 among the 20 cases where they have distinct outcomes.

Figure 4.7(d) depicts the experimental data distribution box plot, under the tradeoff with $p = 0.75$. In this case, more relevance is given to E, however, A is not entirely disregarded. The MDL measure has the highest median (87.13%), followed by AIC (85.5%), $CH-70$ (77.57%) and finally, Full (23.69%). With respect to the IQR, Full has the narrowest spread (3.92%), followed by MDL (10.03%), AIC (10.79%) and

lastly, $CH - 70$ (18.29%). Regarding the data points range, Full has the lowest interval (8.5%), followed by MDL (35.35%), AIC (49.84%) and $CH - 70$ (61.72%). Similarly than for $p = 0.5$, the box plot shows that Full is surpassed by all measures. The greater the relevance given to the E portion ($p \to 1$), the lower the $BEA(p)$ score for the full-length data. Concerning the other three measures, the difference in entropy seems to perform worst than MDL and AIC, since its median is lower and the values of both IQR and data points range are larger in comparison with both model selection criteria. Moreover, Figure 4.7(d) suggests that MDL outperforms AIC. Additionally, the direct count of the number of times MDL has a higher $BEA(0.75)$ score than AIC corresponds to 11 among the 20 cases where they have distinct outcomes.

Overall, the box plots from Figure 4.7 provide a general perspective on the comparison between the measures in analysis, under four distinct scenarios of relevance. The decay of the Full measure with the growing importance given to the E portion ($p \to 1$) is visible, because of its inefficiency in terms of Earliness. Conversely, the progressive prioritization of E is followed by a rise of the MDL performance, due to having the earliest outcomes. In addition, $CH - 70$ is the most stable measure to the variation of p, with a median always between 70% and 80%. The case $p = 1$ was not considered, on behalf of the inconsistency in completely disregarding accuracy at a classification problem. Table 4.6 includes the results of the Wilcoxon signed-rank sum test [76], for comparing the performance of the MCEC algorithm measures. This statistical hypothesis test is known as a non-parametric alternative to the paired t-test, since it deals with independent groups of data, but it does not assume normal distributions nor homogeneity of variance. Demšar [23] recommends using Wilcoxon applied to classification evaluation measures, model sizes and computation times. In particular, he considers this test convenient for comparing machine learning algorithms across multiple datasets, as a result of its robustness.

On the one hand, the box plots from Figure 4.7 provide a graphical comparison between the performance of $CH - 70$, MDL, AIC and Full. On the other hand, Wilcoxon signed-rank sum tests[1] examine the relation between these measures in pairs in order to verify if there is enough evidence to claim that the differences are significant, for a significance level of $\alpha = 0.05$. The Null Hypothesis (H_0) can be rejected when W-value $\leq W_c$, p-value < 0.05 and z-value < -1.96 [23]. The critical value (W_c) is tabulated and it depends on α and on the number of samples (datasets) considered. The arrow in Table 4.6 points towards the measure with better performance, according to the value of $p \in \{0, 0.25, 0.5, 0.75\}$. Double arrow means there is enough evidence to claim that the difference is significant, at the $\alpha = 0.05$ significance level. Rejecting H_0 means the result is significant, that is, at the p-value < 0.05 level, the difference between measures is statistically significant.

Table 4.6 demonstrates that, for $p = 0$, there is enough evidence to claim that Full surpasses all other measures. Furthermore, between $CH - 70$ and the model selection criteria, the difference in entropy outperforms both scoring functions, and AIC shows better results than MDL. All these differences are statistically significant at p-value < 0.05 level. With an increase on the relevance given to Earliness ($p = 0.25$), $CH - 70$ has the best performance in comparison with all the remaining. The AIC measure seems to achieve significantly superior results than MDL, however, there is not enough evidence to

[1] `https://www.mathworks.com/help/stats/signrank.html`

Comparison of measures		$p = 0$	$p = 0.25$	$p = 0.5$	$p = 0.75$
$CH - 70 \Leftrightarrow MDL$	size (W_c)	26 (98)	26 (98)	26 (98)	26 (98)
	W-value	0	34	166	14
	z-value	-4.46	-3.59	-0.24	-4.10
	p-value	< 0.01	< 0.01	0.81	< 0.01
	better	$\Leftarrow$	$\Leftarrow$	$\rightarrow$	$\Rightarrow$
$CH - 70 \Leftrightarrow AIC$	size (W_c)	25 (89)	26 (98)	26 (98)	26 (98)
	W-value	4.5	65	138	8
	z-value	-4.25	-2.81	-0.95	-4.25
	p-value	< 0.01	0.01	0.34	< 0.01
	better	$\Leftarrow$	$\Leftarrow$	$\rightarrow$	$\Rightarrow$
$CH - 70 \Leftrightarrow$ Full	size (W_c)	26 (98)	26 (98)	26 (98)	26 (98)
	W-value	3	56	0	0
	z-value	-4.38	-3.04	-4.46	-4.46
	p-value	< 0.01	< 0.01	< 0.01	< 0.01
	better	$\Rightarrow$	$\Leftarrow$	$\Leftarrow$	$\Leftarrow$
$MDL \Leftrightarrow AIC$	size (W_c)	20 (52)	20 (52)	20 (52)	20 (52)
	W-value	0	10	73	81
	z-value	-3.92	-3.55	-1.19	-0.90
	p-value	< 0.01	< 0.01	0.23	0.37
	better	$\Rightarrow$	$\Rightarrow$	$\rightarrow$	$\leftarrow$
$MDL \Leftrightarrow$ Full	size (W_c)	26 (98)	26 (98)	26 (98)	26 (98)
	W-value	0	160	0	0
	z-value	-4.46	-0.39	-4.46	-4.46
	p-value	< 0.01	0.70	< 0.01	< 0.01
	better	$\Rightarrow$	$\rightarrow$	$\Leftarrow$	$\Leftarrow$
$AIC \Leftrightarrow$ Full	size (W_c)	26 (98)	26 (98)	26 (98)	26 (98)
	W-value	0	140	0	0
	z-value	-4.46	-0.90	-4.46	-4.46
	p-value	< 0.01	0.37	< 0.01	< 0.01
	better	$\Rightarrow$	$\leftarrow$	$\Leftarrow$	$\Leftarrow$

Table 4.6: Comparison of the MCEC algorithm measures and Full against each other, using the Wilcoxon signed-rank sum test applied to the tradeoff experimental data, scored according to $BEA(p)$. Size represents the number of samples (datasets) considered; W_c corresponds to the critical value (tabulated); and W-value consists of the test statistic (smaller of the sums). For a confidence level of $\alpha = 0.05$, the difference between the measures is significant if W-value $\leq W_c$, p-value < 0.05 or z-value < -1.96. The arrow points towards the measure with better performance, according to the value of $p \in \{0, 0.25, 0.5, 0.75\}$. Double arrow means there is enough evidence to claim that the difference is significant, at the $\alpha = 0.05$ significance level.

claim that AIC outperforms Full, nor that the latter surpasses MDL. For an equal balance between E and A ($p = 0.5$), the only assurance consists of Full performing the worst. Among $CH - 70$, MDL and AIC, the differences between them are not statistically significant at the $\alpha = 0.05$ significance level. Lastly, at $p = 0.75$, the Full measure continues to be surpassed by all the others, as well as the difference in entropy in comparison with both model selection criteria. Nevertheless, between MDL and AIC, there is not enough evidence to confirm which one performs the best.

4.3 Rheumatoid Arthritis data

In this section, the MCEC algorithm is applied to a clinical dataset with information about patients suffering from Rheumatoid Arthritis (RA). This is a systemic inflammatory disease, which is primarily char-

acterized by progressive symmetric joint destruction. RA is known as an autoimmune disorder since it occurs when the immune system attacks the body tissues by mistake. Inflammation, pain and loss of function are most common in the wrists and hands, however, RA can also affect other joints and many other organs, such as skin, eyes, lungs, heart and blood vessels. Its cause remains under study, nevertheless, treatments attempt to minimize symptoms (reduce pain, decrease inflammation and prevent bone deformity), and to improve the patient's body functioning. The diagnosis is mainly based on the signs and symptoms manifested by the individuals. Given that there is no cure for RA, Disease-Modifying Antirheumatic Drugs (DMARDs) and biologic agents are some of the primary treatments used for slowing the disease progression. In RA patients, an early identification, together with an aggressive intervention, is crucial for minimizing the irreversible physical disabilities caused by this disease. [22]

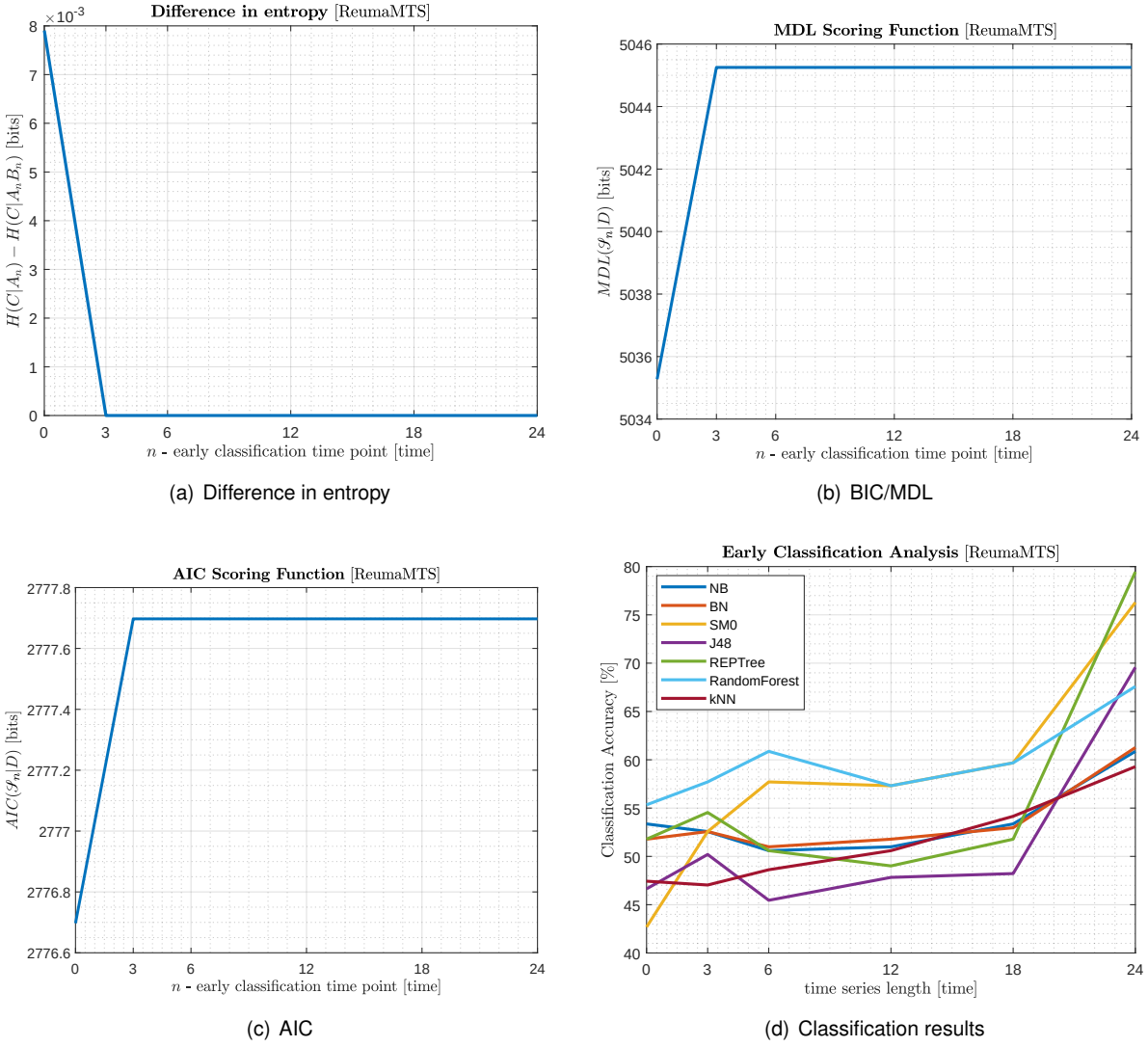
The goal of this experiment is to analyse the effectiveness of the proposed method for predicting the treatment outcome of patients with RA. For this purpose, the MCEC algorithm was used to examine the early classification opportunity in the response of individuals to certain treatments, based on their chronological health condition observations.

Since June 2008, the Portuguese Society of Rheumatology (SPR) has been developing Reuma.pt, the Rheumatic Diseases Portuguese Register [14]. This database contains information from RA patients, which are being treated with DMARDs and biological agents. The clinical data is useful for monitoring the disease evolution, for evaluating treatments efficacy and also for scientific research. The dataset used for this experiment was provided by Reuma.pt, and it includes observations from about 9305 medical appointments concerning 424 patients. Each register describes the individual (demographic and anthropometry data, life style habits, work status, and clinical history), the appointment (time and location), the disease activity (laboratory measurements, medical evaluation and functional assessment scores), and the treatment (previous and current therapies).

The leading preprocessing of the used dataset was performed in a recent thesis [10]. The raw data was converted into a static and dynamic panel format, and the dynamic section was organized into time series with multiple features at each time point. In the data cleaning phase, errors and incongruities were corrected, redundant attributes removed and missing values replaced. A longitudinal imputation of missing values was initially accomplished in the interest of taking the sequential character of the data into consideration. Then, the remaining missing values were replaced with the mean (for numeric attributes) or mode (for categorical attributes), representing a vertical imputation. The missing values in the static data were also dealt through descriptive statistics of the field. At the end of the combination between data cleaning, data transformation and data reduction, the dataset had no missing values and it was composed of two sections: the static (time-invariant) and the dynamic (time-variant) attributes [10]. The preprocessed data included a total of 253 instances ($w = 253$), each with 38 static features and 290 dynamic attributes ($N = 290$). The dynamic section was organized in a time series format, where each time point corresponds to the medical appointment date: month 0, month 3, month 6, month 12, month 18 and month 24 ($L = 6$). Each instance has a class label associated, which describes the patient's response to their treatment at month 24: No Response ($C0$), Moderate Response ($C1$) and Good Response ($C2$). Regarding the numeric attributes, a supervised discretization by Fayyad

& Irani's MDL method [24] was performed (data transformation), using the filter from the Weka Data Mining Software in Java. Note that, similarly to what was done for the datasets from repository, the data classification was performed through stratified cross-validation with 10 folds, using seven distinct classifiers, set with default parameters (Section 3.3).

Figure 4.8 depicts the experimental results of the MCEC algorithm on the RA dataset. Note that the static attributes are not included in this test, given that only the dynamic observations for each appointment are examined. The difference in entropy graph, in Figure 4.8(a), demonstrates an expressive



(a) Difference in entropy

(b) BIC/MDL

(c) AIC

(d) Classification results

Figure 4.8: Experimental results of the MCEC algorithm on the Rheumatoid Arthritis dataset. Parameters: $w = 253$, $L = 6$, $N = 290$, 3 classes. Includes both numeric and categorical dynamic attributes.

decrease in the amount of information required to describe the class attribute. This reduction occurs from $n = 0$ to $n = 3$, and $H(C|A_n) - H(C|A_nB_n) = 0$ for $n \geq 3$. This suggest that the data from the appointments after month 3 do not provide relevant knowledge to predict the treatment outcome of the patients at month 24. On the other hand, the model selection criteria results, represented in Figures 4.8(b) and 4.8(c), consider $n = 1$ to be the time point that best balances the complexity of the model with its effectiveness in fitting the data. Both graphs display a growth from $n = 0$ to $n = 3$, followed by a

stabilization from that point forward. Similarly to what was verified for the repository datasets, the scoring functions behaviour suggests that the growth in the amount of information used from the data do not compensate the increase on the complexity of the associated model. However, the absence of variation on both graphs for $n \geq 3$ may indicate a lack of available information for a proper selection of the best distribution. As previously mentioned, the parameters w, L and N consist of dimensional limitations on the convenient functioning of the MCEC algorithm, in particular, on MDL and AIC measures. The behaviour depicted in Figures 4.8(b) and 4.8(c) may imply the occurrence of overfitting, as a result of the large number of features per time point and the shortage of instances. As studied in Table 4.1, the higher the N, the greater the w required for an appropriate minimization of the model selection criteria. With such a wide number of dimensions, there must be a sufficient amount of samples in order to have a comprehensive and representative set of observations.

The classifiers' accuracy values, according to the time series length, are included in Figure 4.8(d). Although REPTree was the classifier with higher percentage of correctly classified instances for the full-length data, the performance of the seven classifiers is investigated, in the interest of a more thorough analysis. For $n = 0$, the accuracy values range from 42.69% to 55.34%, and RandomForest is the classifier with the highest result. An increase from $n = 0$ to $n = 3$ is verified in 5 of the 7 cases (RandomForest, SMO, REPTree, J48 and BN). In particular, for these three last classifiers, this growth is followed by a decrease from $n = 3$ to $n = 6$. This occurrence is in line with the difference in entropy measure outcome, which suggests month 3 as the early classification time point. In fact, the accuracy obtained at $n = 3$ is higher than the one attained both at $n = 6$ and $n = 12$ for REPTree, J48 and BN, and even greater than the percentage reached at $n = 18$ for REPTree and J48. The classification results indicate a relative stabilization between $n = 6$ and $n = 18$ for the majority of the tested classifiers. This fact supports the idea that the knowledge obtained during that interval does not have a relevant impact on the prediction of the classes. Furthermore, for $n = 24$, the accuracy values range from 59.29% to 79.45%. A significant growth on the percentage of correctly classified instances is verified from $n = 18$ to $n = 24$, which is expected since the treatment outcomes (class labels) concern the medical appointments at month 24.

4.3.1 Feature Selection

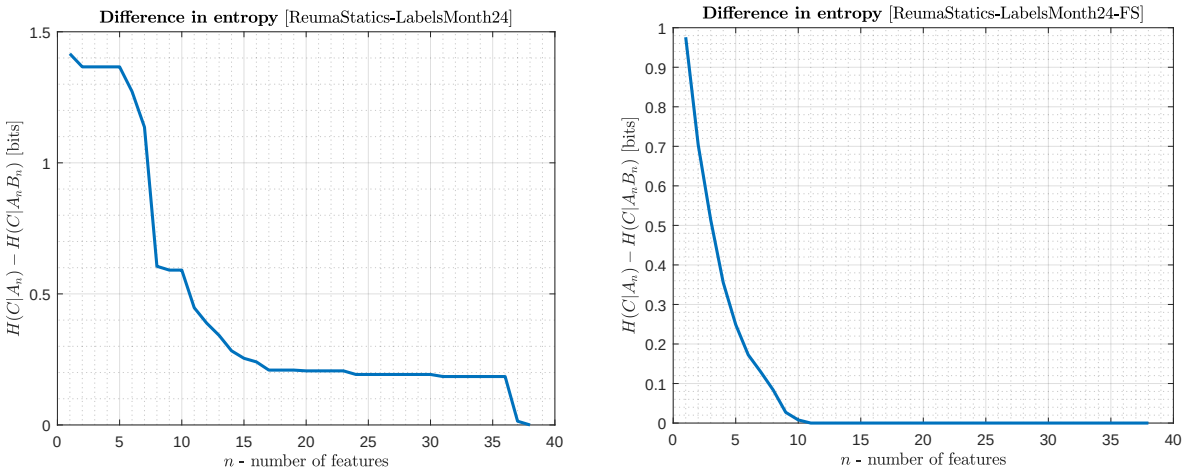
Due to the high dimensionality of the Rheumatoid Arthritis dataset, a feature selection procedure was considered useful. Feature selection is a data reduction method, where irrelevant or redundant attributes are removed in the interest of obtaining an efficient and accurate data mining performance [51]. The goal is to select a subset of features according to an optimality evaluation criterion. These processes are usually greedy, seeing that, while searching among the available attributes, the strategy is based on a locally optimal choice [38]. Literature describes multiple feature selection approaches, which include wrappers, filters and embedded methods [34]. At first, given that the MCEC algorithm examines the information contained in the data (in particular, the existing correlations with the class labels), the capability of the proposed method in selecting the attributes subset from the RA dataset was investigated. This approach

was applied to both static and dynamic features. Then, an informed feature selection was performed, based on the variables used for calculating an important Rheumatoid Arthritis disease activity measure.

Greedy Feature Selection

Considering the univariate time points $X_1, X_2, ..., X_L$ as being distinct features, associated to the class attribute (instead of one feature described over time), the presented algorithm was used in a greedy feature selection procedure. In this case, L represents the total number of attributes, whose correlations with the class labels are being examined. From all the measures, only the difference in entropy was considered for this purpose. The strategy consists of iteratively choosing the feature that most decreases $H(C|A_n) - H(C|A_nB_n)$, where n represents the attribute under analysis. For each iteration, the difference in entropy is computed considering n as each of the remaining features. Among all results, the attribute that obtains the lowest value of $H(C|A_n) - H(C|A_nB_n)$ is chosen, meaning that the combination of features that more information give about the class attribute is selected. At the end, the output consists of the collection of features ordered by relevance in terms of correlation with the classes. Thereupon, the subset of most relevant attributes can be found.

Static Features Seeing that, besides dynamic features, the RA dataset contains also static attributes, the analysis of the latter was also included in this procedure. Thus, the greedy feature selection approach was initially used for identifying the static attributes that more information give about the therapy response of the patients at month 24. Figure 4.9 describes the experimental results of the difference in entropy before and after applying this procedure. The graph from Figure 4.9(a) describes the variation of



(a) Before greedy feature selection.

(b) After greedy feature selection.

Figure 4.9: Variation of the entropy difference for all static attributes before and after performing the greedy feature selection.

$H(C|A_n) - H(C|A_nB_n)$ for the initial given order of attributes. In spite of the irregular behaviour and the accentuated decreasing jumps, all the 38 features are required in order to have no lack of information for predicting the class attribute. Conversely, after performing the greedy feature selection based on the

difference in entropy measure from the MCEC algorithm, the results are considerably distinct, as demonstrated in the graph from Figure 4.9(b). The steep slope of the curve, caused by expressive decreases as attributes are added, confirms that the variables which most reduce the uncertainty about the class are selected. After organizing the features in relevance order, that is, in terms of amount of knowledge about the class labels provided, the subset of static attributes is identified. According to the proposed method, $H(C|A_n) - H(C|A_nB_n) = 0$ for $n \geq 11$, which suggests that there is enough information to predict the treatment outcome using only 11 static categorical features.

With regard to the classification results, using the 38 features for predicting the class labels obtains the following outcomes: NB/SMO – 47.83%, BN – 47.04%, J48/REPTree – 51.38%, RandomForest – 43.87% and kNN – 37.95%. After the greedy feature selection, which means based on the 11 selected features, the accuracy values are: NB – 44.66%, BN/RandomForest – 43.87%, SMO – 45.45%, J48 – 51.78%, REPTree – 51.38% and kNN – 41.11%. Overall, 3 cases have worse results (NB, BN and SMO), 2 cases maintain (REPTree and RandomForest) and 2 cases improve (J48 and kNN). Although data contains three classes, the classification outcomes are not satisfactory, since the best percentage of correctly classified instances, attained by the set of classifiers, is 51.78%. Moreover, these results do not confirm the inferences drawn by the greedy feature selection method, since the subset of attributes, in general, gets worse classification accuracies. This may be caused by the insufficient number of instances, which does not allow a representative and comprehensive sample of the real data. In fact, there are 79 distinct categories for the attribute `cod_profissao` (occupation code), which is plainly adverse among a dataset with only 253 instances. For that reason, the greedy feature selection procedure was repeated, yet disregarding this over-categorized feature. In this case, according to the proposed method, $H(C|A_n) - H(C|A_nB_n) = 0$ for $n \geq 15$, suggesting that the following 15 static categorical features provide the relevant knowledge about the treatment outcome at month 24:

1. `cod_naturalidade` – nationality code. Categories: `AGO`, `BRA`, `CHN`, `CPV`, `DEU`, `FRA`, `GIN`, `GNB`, `MOZ`, `PRT` or `STP`.
2. `DESC_BIO_ACTIVO` – name of the biologic agent therapy. Categories: `Abatacept`, `Adalimumab`, `Anacinra`, `Etanercept`, `Golimumab`, `Infliximab`, `Rituximab` or `Tocilizumab`.
3. `COD_GRAU_ACADEMICO` – educational level code. Categories associated to no scholling, 1st Cycle, 2nd Cycle, 3rd Cycle, Secondary Education, Bachelors degree, among others.
4. `COD_SIT_LABORAL_ACT` – current employment status code. Categories associated to unemployed, full time, retired, among others.
5. `COD_TABAGISMO` – smoking habits code. Categories associated to smoker, ex-smoker, non-smoker, among others.
6. `manif_ea` – indicator of patient with extra-articular manifestations. Categories: `yes` or `no`.
7. `COD_SIT_LABORAL_ANTES_DOENCA` – employment status before disease code. Categories associated to unemployed, full time, retired, among others.
8. `ANTI_CCP` – anti-cyclic citrullinated peptide indicator (categorical attribute). Categories: `yes` or `no`.

9. I_REFORMADO_POR_DOENCA – retirement due to disease indicator. Categories: yes or no.
10. COD_ALCOOL – alcohol consumption habits code. Categories associated to never consumed, current consumer, among others.
11. FACTOR_REUMATOIDE – rheumatoid factor indicator. Categories: yes or no.
12. osteoporose – osteoporosis disease indicator. Categories: yes or no.
13. uveite – uveitis disease indicator. Categories: yes or no.
14. COD_RACA – race code. Categories associated to asian, white, melanesian, black, among others.
15. linfadenopatia – lymphadenopathy disease indicator. Categories: yes or no.

Based on these 15 features, the accuracy outcomes obtained by the classifiers are: NB/REPTree – 51.78%, BN/J48 – 50.99%, SMO – 50.59%, RandomForest – 47.04% and kNN – 44.66%. In comparison with the classification results for the complete set of attributes, except for J48, all the remaining classifiers (6 in 7 cases) attain a better performance for this subset of features. Nevertheless, even though an improvement in the classifiers accuracy is verified, the best percentage of correctly classified instances obtained is still 51.78%. This may suggest that the predictive quality of these attributes is not significant for identifying the treatment outcome of a certain individual at month 24. Seeing that the time series (dynamic features) carry important information, this corresponds to an expected conclusion.

Dynamic Features After examining the static features, the data from each time point was analysed progressively together, having the treatment response of the patients at the last month always as class attribute. The greedy feature selection procedure was performed on the dynamic attributes from every time point, taking the previous selected features into consideration. Gradually, the attributes that provide more information about the classes, at each time point, were identified, without neglecting the prior findings. Note that these variables are chosen based on the difference in entropy measure, examining the correlations between the available features and the class attribute.

Table 4.7 includes the greedy feature selection results on the dynamic attributes from the RA dataset, at each available month. The prefixes T0, T3, T6, T12, T18 and T24 in the name of the features identify the time point to which they belong. A description of the variables can be found in Table 4.8. In addition, Figure 4.10 depicts the comparison of the classification results before and after applying the greedy feature selection method.

Among the 290 dynamic attributes available at month 0, 18 features were selected. Six questions from the Health Assessment Questionnaire (HAQ) were considered relevant (numbers 4, 6, 7, 13, 14 and 16), as well as five joints (numbers 33, 44, 50 - hands; 40 - right knee; and 64 - right wrist). Furthermore, four biologic agent (therapy) indicators were chosen (Prednisona, Infliximab, Etanercept and Deflazacorte). The Disease Activity Score (DAS) measure, based on 44 joints (DAS44_CALC), was identified as the variable that more information provides about the class attribute, from all the 290 features at the first time point. Concerning the classification results (Figure 4.10), except for RandomForest, all the other classifiers (6 in 7 cases) obtain a higher accuracy for the given subset of features. The best

Time Point	Nr. Features	Features
Month 0	18	T0_DAS44_CALC, T0_haq04, T0_haq06, T0_haq14, T0_haq07, T0_haq16, T0_haq13, T0_eva_doente, T0_tum44, T0_Prednisona_i_terap, T0_dol50, T0_dol64, T0_Infliximab_i_terap, T0_dol40, T0_Etanercept_i_terap, T0_Deflazacorte_i_terap, T0_dol33, T0_na44
Month 3	16	T3_DELTA_DAS, T3_SDAI, T0_haq07, T3_haq03, T0_haq14, T0_haq13, T0_haq06, T3_iidDAS, T0_eva_doente, T3_Prednisolona_i_terap, T0_Prednisona_i_terap, T3_dol46, T0_Infliximab_i_terap, T0_Etanercept_i_terap, T0_tum44, T0_dol64
Month 6	13	T3_DELTA_DAS, T3_SDAI, T6_haq02, T0_haq13, T6_haq05, T0_haq06, T6_iidDAS, T0_haq14, T0_eva_doente, T6_Etanercept_i_terap, T6_Prednisolona_i_terap, T0_tum44, T3_dol46
Month 12	15	T3_DELTA_DAS, T3_SDAI, T6_haq02, T0_haq13, T6_haq05, T12_haq06, T6_iidDAS, T12_iitDAS, T0_tum44, T12_haq18, T12_dol40, T12_Adalimumab_i_terap, T0_haq06, T12_eva_doente, T0_haq14
Month 18	13	T3_DELTA_DAS, T18_SDAI, T18_haq03, T0_haq13, T6_haq05, T18_PCR, T3_SDAI, T0_haq14, T12_haq18, T18_I_ENV_PUNHO, T6_iidDAS, T0_haq06, T12_eva_doente
Month 24	9	T24_DAS28_3V, T24_DELTA_DAS, T0_haq14, T0_haq06, T24_haq13, T24_tum45, T24_DAS44_CALC, T24_VS, T6_iidDAS

Table 4.7: Greedy feature selection results on the dynamic attributes from the RA dataset. The procedure is performed progressively for each time point, taking the previously selected features into consideration, at each step. The features are ordered by relevance in terms of correlation with the class attribute. The characterization of the features is described in Table 4.8. Note that the prefixes T0, T3, T6, T12, T18 and T24 denote the time point to which the feature belongs.

percentage of correctly classified instances at month 0 is obtained by J48 (57.71%), after the greedy feature selection procedure.

From the 290 dynamic features at month 3, together with the subset of previously selected ones from month 0 (290 + 18 = 308 features), 16 attributes were considered necessary for providing information about the classes, according to the proposed method. Among these variables, 10 attributes are maintained from the previously selected at month 0, and 6 are obtained from the collection of features at month 3. Some of the HAQ questions from the first time point are repeatedly selected, and in particular numbers 6 and 14 (T0_haq06 and T0_haq14) are preserved until the last month. The swollen indicator of joint 44 (right hand) at month 0 (T0_tum44) is included in the chosen subsets of attributes until the results from month 12. The variables T3_DELTA_DAS and T3_SDAI belong to the selected features up to time point 18. In fact, the first one is considered the most relevant attribute in four months in a row (from months 3 to 18). This suggests that T3_DELTA_DAS contains meaningful knowledge about the treatment outcome at month 24. The same conclusion can be drawn to T0_haq06, T0_haq14 and T6_iidDAS, seeing that once detected, these variables are always included in the subsets of features.

Except for month 12, the number of features selected by the proposed algorithm demonstrates a decreasing tendency. Since the treatment outcomes are associated to the data from the last time point,

Feature Name	Description
DAS28_3V	Disease Activity Score-28 based on 3 variables (numeric attribute): number of swollen joints, number of painful joints (both from a collection of 28 alternatives), and erythrocyte sedimentation rate (VS).
DAS44_CALC	Disease Activity Score measuring 44 joints (numeric attribute).
DELTA_DAS	Variation of DAS28_4V between month 0 and the current month (numeric attribute).
dol$X$	Joint X painful indicator (categorical attribute). Categories: yes or no.
eva_doente	Visual Analogue Scale (VAS or EVA, in Portuguese) according to the patient's opinion (numeric attribute).
haq$X$	Score of question X (categorical attribute) from Health Assessment Questionnaire (HAQ). Categories: 0, 1, 2 or 3.
I_ENV_PUNHO	Disease involving the wrist indicator (categorical attribute). Categories: yes or no.
iidDAS	Non-existence of painful joints (DAS28) indicator.
iitDAS	Non-existence of swollen joints (DAS28) indicator.
na$X$	Joint X non-evaluable indicator (categorical attribute). Categories: yes or no.
PCR	C-reactive protein (CRP or PCR, in Portuguese) test (numeric attribute).
SDAI	Simple Disease Activity Index (SDAI) value (numeric attribute).
tum$X$	Joint X swollen indicator (categorical attribute). Categories: yes or no.
VS	Erythrocyte sedimentation rate (ESR or VS, in Portuguese) value (numeric attribute).
X_i_terap	Therapy X activity indicator (categorical attribute). Categories: yes or no.

Table 4.8: Description of the dynamic attributes from the RA dataset, selected from the greedy feature selection procedure (Table 4.7).

this behaviour suggests that the closer we get to the end of the time series, the more relevant attributes we found (i.e. more information the features provide about the class attribute). Overall, the attributes that are chosen the most correspond to: DELTA_DAS, eva_doente, haq06, haq13, haq14, iidDAS, SDAI and tum44.

Regarding the classification results, 6 in 7 classifiers attain greater accuracies for the subset of features described in Table 4.7, at every time series length $(0, 3, 6, 12, 18, 24)$. In the case of RandomForest, the percentage of correctly classified instances is higher using all available features until month 6. However, from month 12 forward, the subsets verify a better performance, also for this classifier. According to these results, one could say that the greedy feature selection method is capable of identifying a subset of dynamic features with predictive qualities. From the early classification perspective, the dimensionality reduction has a positive impact on the treatment outcome prediction. In fact, at $n = 6$, an accuracy greater than 63% is attained for 3 of the 7 classifiers (NB, BN and SMO). Conversely, based on the complete set of available features, the highest percentage of correctly classified instances achieved at month 6 is 60.87%, for RandomForest.

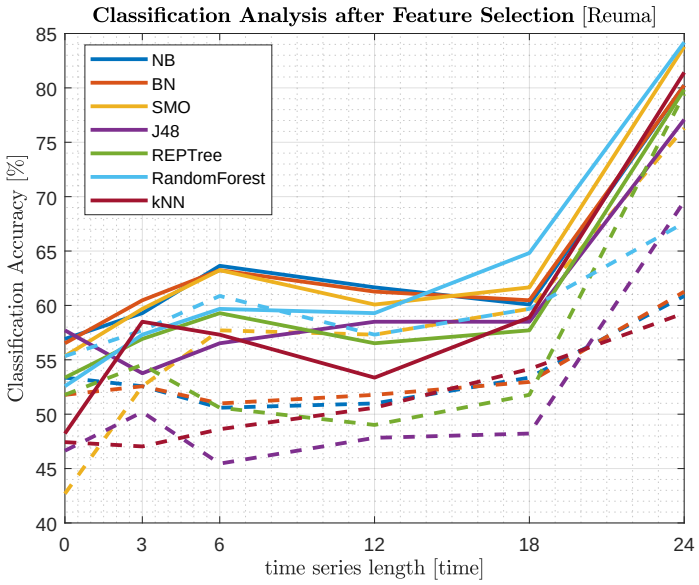


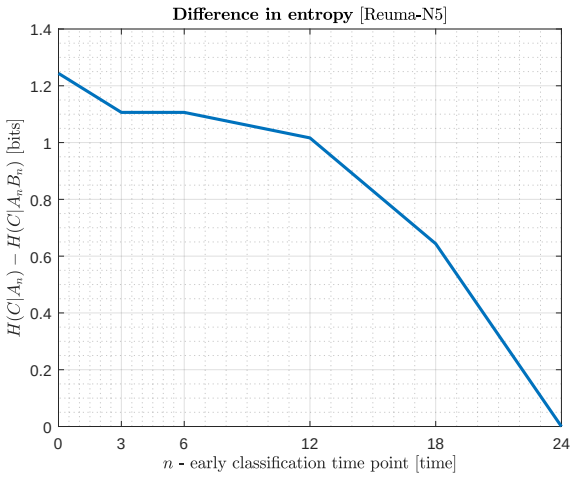
Figure 4.10: Comparison of the classification accuracy results before and after the greedy feature selection procedure applied to the dynamic attributes from the RA dataset. The outcomes from the seven classifiers are included. The dashed lines represent the classification results using all available features ($N = 290$, at each time point) and the solid ones denote the classifiers' accuracy outcomes for the data subsets obtained from the greedy feature selection procedure (Table 4.7).

Informed Feature Selection

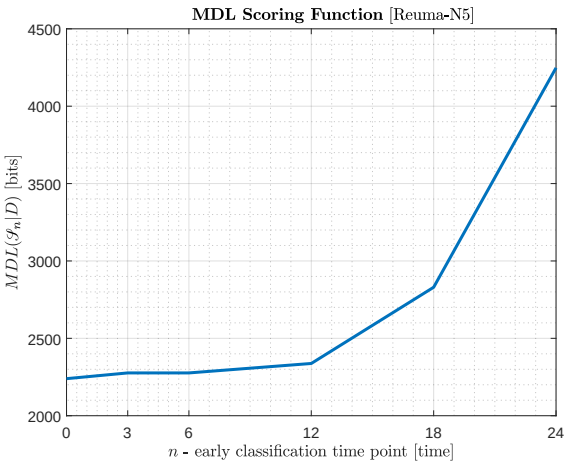
The Disease Activity Score (DAS) measures the disease activity in Rheumatoid Arthritis patients [26]. It is used in clinical trials and in daily clinical practice as a way to evaluate the treatment response of the individuals. While DAS uses the information from 44 joints, DAS28 consists of a simplified measurement, only based on 28 joints. These indicators quantify how active RA is at a certain moment, measuring the improvement or the response of the patient to a specific therapy. In fact, DAS28 is associated to the European League Against Rheumatism (EULAR) response criteria [27], which corresponds to the calculation procedure of the class attribute from the dataset under analysis. Decisions regarding RA treatments are commonly taken based on the comparison of DAS28 values and changes over time.

In general, five parameters are used to compute this measure: the number of swollen joints among the 28 ones (`ntDAS`), the number of painful joints among the 28 ones (`ndDAS`), the Erythrocyte sedimentation rate (`VS`), the C-reactive protein (`PCR`) and the Visual Analogue Scale according to the opinion of the patient (`eva_doente`). In the interest of studying the early classification opportunity based on the knowledge contained in the DAS28 calculation process, an informed feature selection was performed, before applying the MCEC algorithm. The goal was to examine the multivariate time series data, containing merely the information about the five attributes associated with the Disease Activity Score-28 (`ntDAS`, `ndDAS`, `VS`, `PCR` and `eva_doente`). The experimental results are represented in Figure 4.11. The dataset maintains the number of instances ($w = 253$), the time series length ($L = 6$) and the number of classes (3 class labels); only the number of features per time point is different ($N = 5$).

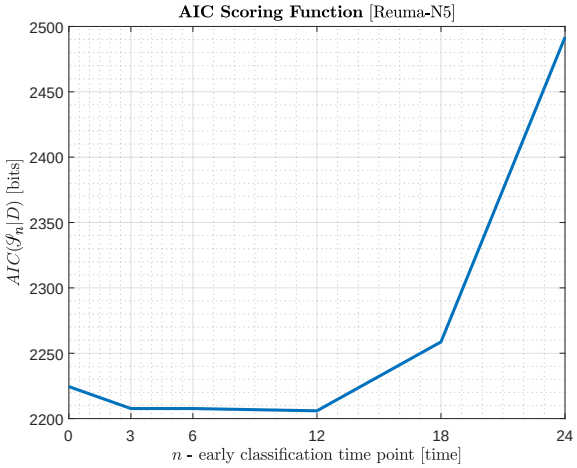
The difference in entropy curve, in Figure 4.11(a), shows a reduction from $n = 0$ to $n = 3$, followed by a lower variation period during $n \in \{3, ..., 12\}$, and, finally, an expressive decrease until the last



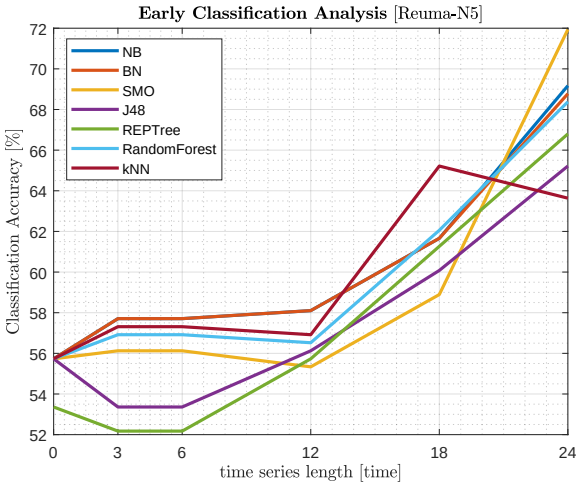
(a) Difference in entropy



(b) BIC/MDL



(c) AIC



(d) Classification results

Figure 4.11: Experimental results of the MCEC algorithm on the Rheumatoid Arthritis dataset, using an informed subset of dynamic features. The variables selected are used to calculate some important measurements in RA treatment: Disease Activity Score (DAS and DAS28). Parameters: $w = 253$, $L = 6$, $N = 5$, 3 classes. All attributes are numeric.

time point. A reduction of at least 70% from the initial value of entropy is verified only for $n = 24$, which corresponds to 0% of earliness. In this case, not much knowledge can be obtained from the $H(C|A_n) - H(C|A_nB_n)$ measure, with regard to the early prediction of the treatment outcome, based on the DAS28 calculation variables. Concerning the model selection criteria, they both demonstrate a similar behaviour. However, while MDL has a minimum for $n = 0$, the lowest value of AIC is found for $n = 12$. In the case of the latter, from $n = 3$ until $n = 6$, the variation is very subtle, appearing to be constant during that period (Figure 4.11(c)). From both scoring functions, the early classification time point may be located in the interval $n \in \{3, ..., 12\}$, since from month 12 forward the complexity of the model displays a more expressive growth. The classification results from Figure 4.11(d) corroborate these conclusions, seeing that 5 in 7 classifiers (NB, BN, SMO, RandomForest and kNN) increase in accuracy from $n = 0$ to $n = 3$, and then demonstrate a relatively constant behaviour until $n = 12$. Note that kNN has a higher percentage of correctly classified instances for $n = 18$, in comparison with the

full-length time series. According to this classifier, a more reliable prediction of the treatment outcome is achieved at month 18. Overall, based on the features used for computing the Disease Activity Score-28, the proposed method suggests there is relevant information that allows an early prediction of the RA patients' therapy response, in the observations between months 3 and 12.

4.3.2 Discussion

Overall, the obtained experimental results suggest that the clinical data from months 3 to 18 do not provide that much crucial information about the treatment outcome. This means that a relatively satisfactory prediction of the RA patients therapy response at month 24 is achievable from the third month. Moreover, from the feature selection approach, a collection of attributes were identified for their predictive qualities: `DELTA_DAS`, `eva_doente`, `haq06`, `haq13`, `haq14`, `iidDAS`, `SDAI` and `tum44`. Conversely, the dimensionality of the dataset was considered a limitation to the performance of the proposed algorithm. The large number of features per time point together with the insufficient number of instances hampered a proper early classification opportunity analysis.

Previous researches have worked on the same (or at least on similar) data, in the interest of investigating the evolution of the RA disease and the effectiveness of the available treatments. In his thesis [5], João Freitas identified RA as one of the leading causes of disability and of life expectancy decrease. He measured the effects of different biologic therapies as well as other factors (such as weight and age) on the disease activity. Based on the performed study, `tocilizumab` was identified as the most effective treatment, and both `adalimumab` and `rituximab` were considered ineffective in comparison with `etanercept`. In addition, no remission was verified with `anakinra`, and factors such as disease duration, age, weight and `eva_doente` were associated to the disease progression. This last variable was also one of the most chosen attributes in the greedy feature selection procedure based on the MCEC algorithm (Table 4.7). Moreover, the DBN structure learning algorithm, proposed by José Monteiro in his thesis [54], was also used for learning from medical data describing RA patients, in the interest of forecasting the disease evolution. He attempted to predict the DAS class from one medical appointment to the next, and obtained accuracies around 75%. Relations between the DAS and some attributes were consistently identified, in particular, with `VS`, `ndDAS` and `eva_doente`. Finally, in her thesis [10], Cátia Botas reached fairly similar conclusions to those obtained by the MCEC algorithm. In fact, she identified the interval between months 3 and 6 as the time period for which the treatment outcome at month 24 can be predicted with some modest confidence. She describes a thorough and intensive preprocessing required for a convenient utilization and manipulation of the dataset, as a result of the noisy and incomplete data. In addition, the `tocilizumab` indicator feature was identified as highly related with the response code, which is in line with the conclusions from the work of João Freitas.

Chapter 5

Conclusions

5.1 Achievements

An information-theoretic algorithm for examining the early classification opportunity in a dataset, containing a collection of time series together with their respective class labels, has been proposed. The MCEC algorithm is capable of dealing with univariate and multivariate data, as long as the time series length and the number of variables per time point are fixed and uniform for all instances. In addition, the experimental results on benchmark data were compared with statistical significance tests in the interest of studying the benefit of the tradeoff between the two fundamental challenges in the early classification context: accuracy and earliness. Furthermore, the software implementation of the algorithm was made freely available, and an article is submitted to an international journal.

Concerning the performance of the MCEC algorithm, the data dimensionality impact analysis and the computation time assay not only identified some limitations on the proper functioning of the proposed method, but also provided insights regarding the effectiveness of the model selection criteria. The results suggest that the number of instances and the number of attributes per time point significantly influence the minimization of both scoring functions. Moreover, AIC was considered less conditioned by the dimensionality than MDL, but the latter demonstrates more consistency in selecting the true model, provided that the true model is in the set of candidate models. In terms of computation time, the algorithm is fairly robust with regard to the number of instances, but not so much concerning the time series length. While the dataset scanning considers the complete set of samples en bloc, an increase on the number of time points denotes an enlargement on the observation window size.

The assessment of the proposed algorithm was accomplished through experimental tests in synthetic, simulated and real-world data. The achieved outcomes confirm the ability of the MCEC method to examine the early classification opportunity within a dataset. This means that, in general, the three main measures (difference in entropy, MDL and AIC) are capable of choosing an early time point based on which the time series classification is plausible. Overall, the first measure obtains better accuracy results, MDL demonstrates a superior tendency for earliness, and AIC attains the most competent balance between both aims. Regarding the study on the tradeoff between accuracy and earliness, al-

though, for an equal balance, AIC seems to surpass MDL, and the latter appears to outperform the difference in entropy, these inferences are not statistically significant at the $\alpha = 0.05$ significance level. On the other hand, the difference in entropy is surpassed by both model selection criteria, when priority is given to earliness over accuracy. However, in this case, there is not enough statistical evidence to claim that MDL outperforms AIC, in spite of the empirical outcomes. Conversely, when accuracy is the main goal, the entropy measure surpasses both scoring functions, and AIC obtains better results than MDL. Herein, these comparisons are statistically significant at p-value < 0.05 level.

5.2 Future Work

The MCEC algorithm can be extended to deal with datasets where the time series length and the number of attributes per time point vary among all instances. The requirement of fixed and uniform data size may be an impediment in some real-world applications. For instance, within a collection of clinical observations concerning a set of patients, two individuals may have gone to a distinct number of medical appointments, or their health records may include a different number of parameters. Therefore, the flexibility in handling dimensional variations can be of great benefit, in particular when the dataset contains missing values.

In addition, a classification method can be developed based on the capabilities of this information-theoretic approach. Seeing that the majority of the state-of-the-art methodologies in the early prediction context suggest self-sufficient classifiers, a procedure with a learning stage followed by a classification step could be proposed. In this case, instead of investigating the early classification opportunity within the entire collection of samples, the algorithm would be able to assign a class label to a new single incomplete time series.

Finally, the feature selection potentialities of the MCEC method can be exploited, as briefly introduced in Section 4.3. Since the presented approach analyses the correlations between the information contained in the data and the class labels, the procedure can be used for selecting a subset with the most relevant attributes, among a group of alternatives. In particular, a greedy feature selection could be performed based, not only on the difference in entropy measure, but also on the model selection criteria.

Bibliography

[1] H. D.-G. Acquah. Comparison of Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) in selection of an asymmetric price relationship. *Journal of Development and Agricultural Economics*, 2(1):001–006, 2010.

[2] D. Aha and D. Kibler. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[3] F. Aiolli and A. Sperduti. Multiclass classification with multi-prototype support vector machines. *Journal of Machine Learning Research*, 6:817–850, 2005.

[4] H. Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[5] J. ao Pedro Bento Machado Marques de Freitas. Analysis of electronic medical records of rheumatoid arthritis patients on biological therapies: a reuma.pt study. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2015.

[6] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, Online First, 2016.

[7] M. G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 29(2):400–422, Mar 2015.

[8] I. Ben-Gal. Bayesian networks. *Encyclopedia of statistics in quality and reliability*, 2008.

[9] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *Computer Communication Review*, 36(2):23–26, 2006.

[10] C. S. T. Botas. Feature analysis to predict treatment outcome in rheumatoid arthritis. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2017.

[11] A. Bregón, M. A. S. Hurtado, J. J. Rodríguez, C. J. Alonso, B. P. Junquera, and Q. I. Moro. Early fault classification in dynamic systems using case-based reasoning. In *Current Topics in Artificial Intelligence, 11th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2005, Santiago de Compostela, Spain, November 16-18, 2005, Revised Selected Papers*, pages 211–220, 2005.

[12] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[13] K. P. Burnham and D. R. Anderson. Multimodel inference: understanding AIC and BIC in model selection. *Sociological methods & research*, 33(2):261–304, 2004.

[14] H. Canhao, A. Faustino, F. Martins, and J. E. Fonseca. Reuma.pt - the rheumatic diseases portuguese register. *Acta reumatologica portuguesa*, 36(1):45–56, 2010.

[15] A. M. Carvalho. Scoring functions for learning Bayesian networks. *INESC-ID Tec. Rep*, 2009.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: synthetic minority oversampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002.

[17] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. The UCR time series classification archive, July 2015. `www.cs.ucr.edu/~eamonn/time_series_data/`.

[18] T. M. Cover and J. A. Thomas. *Elements of information theory (2. ed.)*. Wiley, 2006.

[19] T. M. Cover and J. A. Thomas. *Elements of information theory, 2nd edition*. Wiley, 2006.

[20] I. Csiszar and J. Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011.

[21] P. Cunningham and S. J. Delany. k-nearest neighbour classifiers. *Multiple Classifier Systems*, 34:1–17, 2007.

[22] J. E. C. da Fonseca, H. Canhão, and M. V. de Queiroz. *Reumatologia Fundamental*. Lidel, 2013.

[23] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, Dec. 2006.

[24] U. M. Fayyad and K. B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence. Chambéry, France, August 28 - September 3, 1993*, pages 1022–1029, 1993.

[25] M. A. T. Figueiredo. Elementos de teoria da informaçao. 2007.

[26] J. Fransen, G. Stucki, and P. L. C. M. van Riel. Rheumatoid arthritis measures: Disease activity score (das), disease activity score-28 (das28), rapid assessment of disease activity in rheumatology (radar), and rheumatoid arthritis disease activity index (radai). *Arthritis Care & Research*, 49(S5):S214–S224, 2003.

[27] J. Fransen, P. Van Riel, et al. The disease activity score and the eular response criteria. *Clinical and experimental rheumatology*, 23(5):S93, 2005.

[28] T. Fu. A review on time series data mining. *Eng. Appl. of AI*, 24(1):164–181, 2011.

[29] M. F. Ghalwash and Z. Obradovic. Early classification of multivariate temporal observations by extraction of interpretable shapelets. *BMC Bioinformatics*, 13:195, 2012.

[30] M. F. Ghalwash, V. Radosavljevic, and Z. Obradovic. Extraction of interpretable multivariate patterns for early diagnostics. In *2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, December 7-10, 2013*, pages 201–210, 2013.

[31] M. F. Ghalwash, V. Radosavljevic, and Z. Obradovic. Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 402–411, 2014.

[32] M. F. Ghalwash, D. Ramljak, and Z. Obradovic. Early classification of multivariate time series using a hybrid HMM/SVM model. In *2012 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2012, Philadelphia, PA, USA, October 4-7, 2012*, pages 1–6, 2012.

[33] M. P. Griffin and J. R. Moorman. Toward the early diagnosis of neonatal sepsis and sepsis-like illness using novel heart rate analysis. *Pediatrics*, 107(1):97–104, 2001.

[34] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, Mar. 2003.

[35] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220 – 239, 2017.

[36] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[37] J. Hamilton. *Time series analysis*. Princeton Univ. Press, Princeton, NJ, 1994.

[38] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques, 3rd edition*. Morgan Kaufmann, 2011.

[39] N. Hatami and C. Chira. Classifiers with a reject option for early time-series classification. *CoRR*, abs/1312.3989, 2013.

[40] G. He, Y. Duan, R. Peng, X. Jing, T. Qian, and L. Wang. Early classification on multivariate time series. *Neurocomputing*, 149:777–787, 2015.

[41] G. He, Y. Duan, T. Qian, and X. Chen. Early prediction on imbalanced multivariate time series. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*, pages 1889–1892, 2013.

[42] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[43] S. Kalmegh. Analysis of weka data mining algorithm reptree, simple cart and randomtree for classification of indian news. *Int. J. Innov. Sci. Eng. Technol*, 2(2):438–446, 2015.

[44] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[45] D. T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley, 2005.

[46] S. Laxman and P. S. Sastry. A survey of temporal data mining. *Sadhana*, 31(2):173–198, Apr 2006.

[47] M. Lemus, J. Beirão, A. Carvalho, P. Mateus, and N. Paunković. Multivariate correlations for early classification. *In preparation*, 2018.

[48] K. Li, S. Li, and Y. Fu. Early classification of ongoing observation. In *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, pages 310–319, 2014.

[49] M. Lichman. UCI machine learning repository, 2013. `http://archive.ics.uci.edu/ml/`.

[50] Y. Lin, H. Chen, V. S. Tseng, and J. Pei. Reliable early classification on multivariate time series with numerical and categorical attributes. In *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD 2015, Ho Chi Minh City, Vietnam, May 19-22, 2015, Proceedings, Part I*, pages 199–211, 2015.

[51] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[52] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.

[53] T. Mitsa. *Temporal Data Mining*. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. CRC Press, 2010.

[54] J. M. P. S. L. Monteiro. Learning from short multivariate time series. Master's thesis, Instituto Superior Técnico, Universidade de Lisboa, 2015.

[55] U. Mori, A. Mendiburu, E. J. Keogh, and J. A. Lozano. Reliable early classification of time series based on discriminating the classes over time. *Data Min. Knowl. Discov.*, 31(1):233–263, 2017.

[56] A. Mueen, E. J. Keogh, and N. E. Young. Logical-shapelets: an expressive primitive for time series classification. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 1154–1162, 2011.

[57] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[58] J. Naisbitt. *Megatrends: Ten New Directions Transforming Our Lives*. Warner Books, 1984.

[59] H. T. Ng. Exemplar-based word sense disambiguation: Some recent improvements. *CoRR*, cmp-lg/9706010, 1997.

[60] N. Parrish, H. S. Anderson, M. R. Gupta, and D. Hsiao. Classifying with confidence from incomplete information. *Journal of Machine Learning Research*, 14(1):3561–3589, 2013.

[61] J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[62] J. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar 1986.

[63] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[64] R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[65] F. D. Ridder, R. Pintelon, J. Schoukens, and D. P. Gillikin. Modified AIC and MDL model selection criteria for short data records. *IEEE Trans. Instrumentation and Measurement*, 54(1):144–150, 2005.

[66] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46. IBM, 2001.

[67] J. J. Rodrıguez and C. J. Alonso. Boosting interval-based literals: Variable length and early classification. *Knowledge Discovery from Temporal and Spatial Data (W12)*, 2002.

[68] S. R. Safavian and D. A. Landgrebe. A survey of decision tree classifier methodology. *IEEE Trans. Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[69] T. Santos and R. Kern. A literature survey of early time series classification and deep learning. In *Proceedings of the 1st International Workshop on Science, Application and Methods in Industry 4.0 co-located with (i-KNOW 2016), Graz, Austria, October 19, 2016.*, 2016.

[70] J. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classifier for data mining. In *Proc. 1996 Int. Conf. Very Large Data Bases*, pages 544–555. Citeseer, 1996.

[71] D. Steinberg and P. Colla. Cart: classification and regression trees. *The top ten algorithms in data mining*, 9:179, 2009.

[72] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[73] S. Theodoridis and K. Koutroumbas. Pattern recognition and neural networks. In *Machine Learning and Its Applications*, pages 169–195. Springer, 2001.

[74] S. I. Vrieze. Model selection and psychological theory: a discussion of the differences between the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). *Psychological methods*, 17(2):228, 2012.

[75] W. Wang, C. Chen, W. Wang, P. Rai, and L. Carin. Earliness-aware deep convolutional networks for early time series classification. *CoRR*, abs/1611.04578, 2016.

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

[77] Z. Xing, J. Pei, G. Dong, and P. S. Yu. Mining sequence classifiers for early prediction. In *Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA*, pages 644–655, 2008.

[78] Z. Xing, J. Pei, and E. J. Keogh. A brief survey on sequence classification. *SIGKDD Explorations*, 12(1):40–48, 2010.

[79] Z. Xing, J. Pei, and P. S. Yu. Early classification on time series. *Knowl. Inf. Syst.*, 31(1):105–127, 2012.

[80] Z. Xing, J. Pei, P. S. Yu, and K. Wang. Extracting interpretable features for early classification on time series. In *Proceedings of the Eleventh SIAM International Conference on Data Mining, SDM 2011, April 28-30, 2011, Mesa, Arizona, USA*, pages 247–258, 2011.

[81] Y. Yang. Can the strengths of AIC and BIC be shared? *BIOMETRICA*, 92:2003, 2003.

[82] L. Ye and E. J. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, pages 947–956, 2009.

Appendix A

Synthetic example of the proposed method

This appendix describes the implementation details of the proposed method through the example of a synthetically generated dataset. As previously mentioned, for a given multivariate time series $T_i = (X_1, X_2, ..., X_L)$, where L corresponds to its time length, each component $X_k = (X_{k_1}, X_{k_2}, ..., X_{k_N})$ consists of a set of N features measured at time point k. A class label c_i is associated to each T_i through the relation $Class(T_i) = c_i$, and the dataset D to be analysed consists of a collection of pairs $(T_i, c_i) : i \in \{1, ..., w\}$, where w corresponds to the number of instances.

Consider the example represented in Table A.1. This dataset consists of ten instances $(w = 10)$ of univariate time series $(N = 1)$ with seven time points each $(L = 7)$. The feature described over time is of type boolean $(X_k \in \{0, 1\})$, and the class label attribute includes two alternatives $(c_i \in \{\text{C}0, \text{C}1\})$. The first two time points $(X_1$ and $X_2)$ are randomly generated and all the others correspond to the

Table A.1: Synthetic dataset example.

X_1	X_2	X_3	X_4	X_5	X_6	X_7	c_i
0	1	1	0	1	1	0	C1
1	1	0	1	1	0	1	C1
0	1	1	0	1	1	0	C1
1	1	0	1	1	0	1	C1
1	0	1	1	0	1	1	C0
1	0	1	1	0	1	1	C0
1	1	0	1	1	0	1	C1
0	1	1	0	1	1	0	C1
1	0	1	1	0	1	1	C0
0	1	1	0	1	1	0	C1

XOR (Table A.2) of the two previous time points. For example, in the first instance, X_3 corresponds to $X_1 \oplus X_2 = 1$, and X_4 is equal to $X_2 \oplus X_3 = 0$. The class label c_i is the result of the exclusive disjunction between X_6 and X_7, concatenated with the letter "C".

Table A.2: Exclusive disjunction (XOR).

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

According to the proposed approach, the multivariate time series are decomposed in

$$T_i = (X_1, X_2, ..., X_n, X_{n+1}, ..., X_{L-1}, X_L)\,, \tag{A.1}$$

and the dataset is organized in three groups:

$$\begin{aligned} A_n &= \{X_1, X_2, ..., X_n\}\,, \\ B_n &= \{X_{n+1}, X_{n+2}, ..., X_L\}\,, \\ C &= \{c_i\}\,. \end{aligned} \tag{A.2}$$

A.1 Difference in entropy

Once aiming for early classification, we are interested in predicting the class label of a time series as early as possible, provided that the classification accuracy is close to the one using the complete data. For this purpose, in the first stage, the conditional entropy of the dataset is the subject of study. The goal consists of analysing the difference in entropy:

$$H(C|A_n) - H(C|A_nB_n), \tag{A.3}$$

while varying the early classification time point n from $\{1, ..., L\}$.

Considering the three groups described in Equation (A.2) as well as the definition from Equation (3.3), the calculation of the conditional entropies is performed through:

$$\begin{aligned} H(C|A_n) &= \sum_{a,c} p(A_n = a, C = c) \log_2 \left[\frac{p(A_n = a)}{p(A_n = a, C = c)} \right], \\ H(C|A_nB_n) &= \sum_{a,b,c} p(A_n = a, B_n = b, C = c) \log_2 \left[\frac{p(A_n = a, B_n = b)}{p(A_n = a, B_n = b, C = c)} \right]. \end{aligned} \tag{A.4}$$

The conditional entropy $H(C|A_nB_n)$ quantifies the amount of information needed to describe the outcome of the class label c_i, based on the knowledge of the entire time series. This value is constant when varying n from $\{1, ..., L\}$, since it represents the lowest possible uncertainty in the outcome of C. On the other hand, $H(C|A_n)$ quantifies the amount of information needed to predict c_i, provided that there is only information until time point n. This value is expected to decrease with the increase of n, since the growth on the amount of available information is expected to reduce the uncertainty of the prediction.

The statistical parameters included in Equation (A.4) are estimated as the quotient between the

number of occurrences of each specific case and the total number of instances in the dataset:

$$\begin{aligned} p(A_n = a) &= \frac{\text{number of occurrences of } \{a\}}{w}, \\ p(A_n = a, B_n = b) &= \frac{\text{number of occurrences of } \{a, b\}}{w}, \\ p(A_n = a, C = c) &= \frac{\text{number of occurrences of } \{a, c\}}{w}, \\ p(A_n = a, B_n = b, C = c) &= \frac{\text{number of occurrences of } \{a, b, c\}}{w}, \end{aligned} \tag{A.5}$$

for which $\{a\}$, $\{a, b\}$, $\{a, c\}$ and $\{a, b, c\}$ represent the existing cases included in the respective group described in Equation (A.2).

For $n = 1$, the organization of the time series in three groups corresponds to:

$$A_1 = \{X_1\}, \quad B_1 = \{X_2, X_3, X_4, X_5, X_6, X_7\}, \quad C = \{c_i\}, \tag{A.6}$$

and the information from the dataset can be structured in lists such as:

$$\begin{aligned} \mathcal{A}_1 &= \begin{bmatrix} (\{0\}, 4) \\ (\{1\}, 6) \end{bmatrix}, \quad \mathcal{AC}_1 = \begin{bmatrix} (\{1\,\mathsf{C0}\}, 3) \\ (\{1\,\mathsf{C1}\}, 3) \\ (\{0\,\mathsf{C1}\}, 4) \end{bmatrix}, \\ \mathcal{AB} &= \begin{bmatrix} (\{1011011\}, 3) \\ (\{1101101\}, 3) \\ (\{0110110\}, 4) \end{bmatrix}, \quad \mathcal{ABC} = \begin{bmatrix} (\{1011011\,\mathsf{C0}\}, 3) \\ (\{0110110\,\mathsf{C1}\}, 4) \\ (\{1101101\,\mathsf{C1}\}, 3) \end{bmatrix}. \end{aligned} \tag{A.7}$$

where the format consists of $\mathcal{G} = [(\{g\}, \text{number of occurrences})]$. Considering the list $\mathcal{A}_1$, the observation $\{0\}$ for X_1 occurs 4 times, and $X_1 = 1$ is verified in 6 of the 10 instances from the data. From these lists, the parameters described in Equation (A.5) are calculated as:

$$\begin{aligned} p(A_1 = \{0\}) = \frac{4}{10} = 0.4 \text{ bits}, \quad p(A_1 = \{1\}) = \frac{6}{10} = 0.6 \text{ bits}, \\ p(A_1C = \{1\,\mathsf{C0}\}) = p(A_1C = \{1\,\mathsf{C1}\}) = 0.3 \text{ bits}, \quad p(A_1C = \{0\,\mathsf{C1}\}) = 0.4 \text{ bits}, \\ p(A_1B_1 = \{1011011\}) = p(A_1B_1 = \{1101101\}) = \frac{3}{10} = 0.3 \text{ bits}, \\ p(A_1B_1 = \{0110110\}) = 0.4 \text{ bits}, \\ p(A_1B_1C = \{1011011\,\mathsf{C0}\}) = p(A_1B_1C = \{1101101\,\mathsf{C1}\}) = 0.3 \text{ bits}, \\ p(A_1B_1C = \{0110110\,\mathsf{C1}\}) = 0.4 \text{ bits}, \end{aligned} \tag{A.8}$$

and the conditional entropies described in Equation (A.4) are computed through:

$$\begin{aligned} H(C|A_1) &= \sum_{a,c} p(A_1C = \{a, c\}) \log_2 \left[\frac{p(A_1 = \{a\})}{p(A_1C = \{a, c\})} \right] = \\ &= 0.3 \log_2 \left[\frac{0.6}{0.3} \right] + 0.3 \log_2 \left[\frac{0.6}{0.3} \right] + 0.4 \log \left[\frac{0.4}{0.4} \right] = 0.6 \text{ bits}; \end{aligned} \tag{A.9}$$

$$H(C|A_1B_1) = \sum_{a,b,c} p(A_1B_1C = \{a,b,c\}) \log_2 \left[\frac{p(A_1B_1 = \{a,b\})}{p(A_1B_1C = \{a,b,c\})} \right] = \\ = 2 \left(0.3 \log_2 \left[\frac{0.3}{0.3} \right] \right) + 0.4 \log_2 \left[\frac{0.4}{0.4} \right] = 0. \tag{A.10}$$

On the one hand, $H(C|A_1) = 0.6$ bits represents the amount of information needed to predict the classes of the time series, given that X_1 is known. On the other hand, $H(C|A_1B_1) = 0$ indicates that the complete time series provide enough information for describing the group C. The difference in entropy, equal to $H(C|A_1) - H(C|A_1B_1) = 0.6$ bits, denotes that with only the first time point of the time series there is still a lack of information for predicting the class labels.

For $n = 2$, the organization of the time series in three groups becomes:

$$A_2 = \{X_1, X_2\}, \quad B_2 = \{X_3, X_4, X_5, X_6, X_7\}, \quad C = \{c_i\}, \tag{A.11}$$

and the data structured in lists correspond to:

$$\mathcal{A}_2 = \begin{bmatrix} (\{10\}, 3) \\ (\{11\}, 3) \\ (\{01\}, 4) \end{bmatrix}, \quad \mathcal{AC}_2 = \begin{bmatrix} (\{11\,\mathsf{C1}\}, 3) \\ (\{10\,\mathsf{C0}\}, 3) \\ (\{01\,\mathsf{C1}\}, 4) \end{bmatrix}. \tag{A.12}$$

Note that the lists $\mathcal{AB}$ and $\mathcal{ABC}$ are the same as in Equation (A.7) since they do not change with the variation of n. In this case, the computation of the difference in entropy is equal to:

$$H(C|A_2) - H(C|A_2B_2) = 0 - 0 = 0. \tag{A.13}$$

This result denotes that with the first two time points of the time series (X_1 and X_2) there is enough information for predicting the class labels.

Figure A.1 describes the evolution of the difference in entropy from Equation (A.3) for $n \in \{1, ..., 10\}$. Since $H(C|A_n) - H(C|A_nB_n) = 0$ for $n \geq 2$, the correlations between the early states of T_i and the classes c_i are completely represented using only the first two time points. It is possible to infer that the information given by the time series after X_2 does not provide any useful knowledge about the class label attribute.

A.2 Complexity of the model

In the second stage, based on two Bayesian network scoring functions, the complexity of the model is examined in the interest of choosing the early time point, which is able not only to achieve an early classification, but also to consider the simplicity of the choice. The goal consists of analysing the function:

$$\phi(D|\mathscr{S}_n) = \alpha \cdot |\mathscr{S}_n| - LL(D|\mathscr{S}_n), \tag{A.14}$$

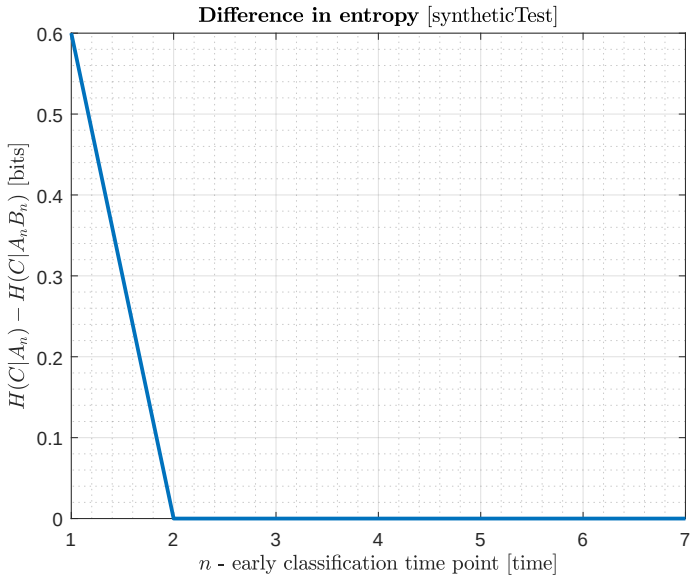


Figure A.1: Variation of the entropy difference while $n \in \{1, ..., L\}$, for the data represented in Table A.1.

while varying the early classification time point n from $\{1, ..., L\}$, where $|\mathscr{S}_n|$ is described in Equation (3.22) and $LL(D|\mathscr{S}_n)$ is defined as:

$$\begin{aligned} LL(D|\mathscr{S}_n) &= \sum_{i=1}^{w} \log_2 \left[p(C = c|A_n = a)p(B_n = b|A_n = a)p(A_n = a)\right] = \\ &= \sum_{i=1}^{w} \log_2 \left[\frac{p(A_n = a, B_n = b)p(A_n = a, C = c)}{p(A_n = a)}\right], \end{aligned} \tag{A.15}$$

according to the Bayes' theorem. The value of α is independent from the early time point, and while for the MDL score, $\alpha = \frac{1}{2}\log_2 w$, for the AIC, $\alpha = 1$. The statistical parameters for computing the log-likelihood of the model given the data are estimated as described in Equation (A.5).

Regarding the number of independent parameters, $||C||$ denotes the number of distinct cases in group C, i.e. the number of classes in the dataset. This value is constant while varying n, seeing that the variation of the early time point does not affect group C. Similarly, $||A_n||$ corresponds to the number of different existing cases in group A_n. This value is expected to increase with n, as a greater amount of analysed instants from the time series leads to a higher number of possible cases. At some point, $||A_n||$ is expected to stabilize, possibly when the information added is redundant, and consequently, unnecessary for the prediction.

For $n = 1$, the groups are organized as represented in Equation (A.6) and the information contained in the dataset can be structured in lists such as the ones denoted in Equation (A.7). From the statistical

parameters calculated in Equation (A.8), the log-likelihood can be computed as:

$$\begin{aligned} -LL(\mathscr{S}_1|D) &= -\sum_{i=1}^{10} \log_2 \left[\frac{p(A_1B_1 = \{a,b\}) \cdot p(A_1C = \{a,c\})}{p(A_1 = \{a\})} \right] = \\ &= 3\log_2 \left[\frac{0.3 \cdot 0.3}{0.6} \right] + 4\log_2 \left[\frac{0.4 \cdot 0.4}{0.4} \right] + 3\log_2 \left[\frac{0.3 \cdot 0.3}{0.6} \right] = \\ &= 21.7095 \text{ bits.} \end{aligned} \tag{A.16}$$

Note that this value corresponds to the amount of information required to represent the dataset D using the model $\mathscr{S}_1$, and the sum comprises all the instances included in D (in this case, $w = 10$). Since $||A_1||$ consists of the number of distinct cases in group $\mathcal{A}_1$, and $||C||$ denotes the number of classes in the dataset, from Equation (3.22), the number of independent parameters in the model is equal to:

$$|\mathscr{S}_1| = ||A_1|| \cdot ||C|| - 1 = 2 \cdot 2 - 1 = 3 \text{ bits.} \tag{A.17}$$

This value quantifies the amount of information needed to encode the model $\mathscr{S}_1$, as well as the data D given the model. It can be viewed as a measure of the complexity associated to using the model $\mathscr{S}_1$ to represent the dataset from Table A.1.

Concerning the AIC score, seeing that $\alpha = 1$ and according to Equation (3.21), its computation corresponds to:

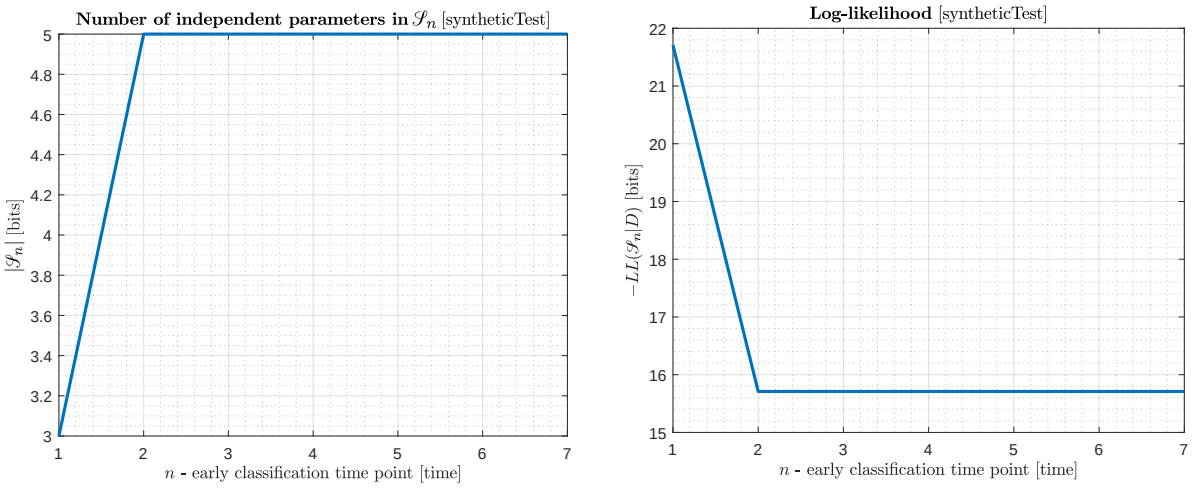
$$AIC(\mathscr{S}_1|D) = |\mathscr{S}_1| - LL(\mathscr{S}_1|D) = 3 + 21.7095 = 24.7095 \text{ bits.} \tag{A.18}$$

With regard to the MDL scoring function, the penalization factor is $\alpha = \frac{1}{2}\log_2 10 = 1.661$ bits, and through Equation (3.20) this score is calculated as:

$$MDL(\mathscr{S}_1|D) = \alpha \cdot |\mathscr{S}_1| - LL(\mathscr{S}_1|D) = 1.661 \cdot 3 + 21.7095 = 26.6925 \text{ bits.} \tag{A.19}$$

In these sort of model selection, the idea is to find the $\mathscr{S}_n$ that is good enough to capture the information in the data D, but not so complex that it makes the choice infeasible. The multiple models represent the variation of n from $\{1, ..., L\}$, which means that there are as many $\mathscr{S}_n$ as the number of time points (L). By minimizing the general function $\phi(D|\mathscr{S}_n)$ from Equation (A.14), we are trying to find a balance between the complexity of the model and its ability to fit to the data. The goal is to find the early time point for which both $MDL(D|\mathscr{S}_n)$ and $AIC(D|\mathscr{S}_n)$ are as low as possible, meaning that the information contained in the time series until n is enough to represent the dataset in an effective but simple manner.

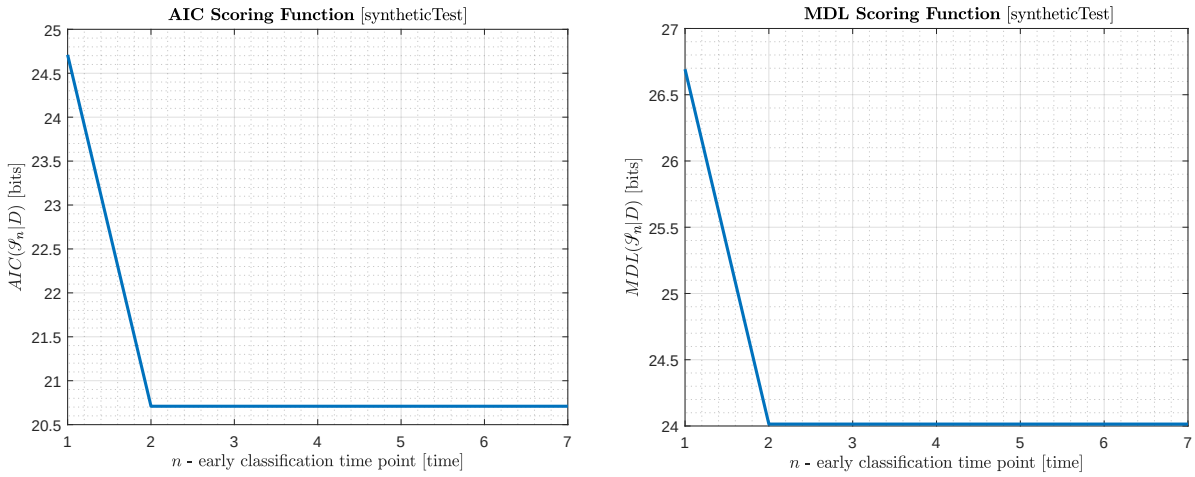
Aiming for a more detailed analysis of the terms that compose $\phi(D|\mathscr{S}_n)$, Figure A.2(a) denotes the variation of the number of independent parameters in $\mathscr{S}_n$ with n, and Figure A.2(b) represents the graph of the log-likelihood term for all time points. As depicted in Figure A.2(a), the complexity of the model increases from $n \in \{1, 2\}$, i.e. the more instants from the time series analysed, the higher the amount of information needed to encode $\mathscr{S}_n$. Notice that since $|\mathscr{S}_n|$ is constant for $n \geq 2$, the information added to the model in this interval does not affect its complexity. From Figure A.2(b), the significant decrease of the log-likelihood at $n = 2$ followed by a stabilization from that time point on indicates that the dataset



(a) $|\mathscr{S}_n|$ (b) $-LL(D|\mathscr{S}_n)$

Figure A.2: Variation of the terms from $\phi(D|\mathscr{S}_n)$ while $n \in \{1, ..., L\}$, for the dataset from Table A.1.

D is effectively described by $\mathscr{S}_2$, i.e. using only the observations from the two initial instants of the time series. Figure A.3 represents the values for the AIC and the MDL scoring functions. Seeing that



(a) $AIC(D|\mathscr{S}_n)$ (b) $MDL(D|\mathscr{S}_n)$

Figure A.3: Variation of the scoring functions while $n \in \{1, ..., L\}$, for the dataset from Table A.1.

the difference between the scores is mainly related with the penalization factor on the complexity term, the variation is very similar in both cases, achieving a minimum value at $n \geq 2$. This denotes that a satisfactory tradeoff between the complexity of the model and its ability to represent the data is found at $n = 2$.

The same conclusion is reached for the proposed approaches. Not only from the model complexity analysis but also from the study of the difference in entropy, the results demonstrate that we are expected to be able to accurately classify the time series from the synthetic dataset, based only on the first two time points. A closer look to the two initial columns from Table A.1 (X_1 and X_2) corroborates this inference, seeing that whenever $A_2 = \{11\}$ or $A_2 = \{01\}$, the class label is C1; and whenever $A_2 = \{10\}$, the class label is C0; i.e. the knowledge of A_2 is enough to describe C with no uncertainty.

A.3 Early classification analysis

The analysis of the percentage of correctly classified instances (Equation 2.1) for the synthetic dataset is represented in Figure A.4. Except for the REPTree classifier, all the others accomplish 100% accuracy for $n \geq 2$, which matches with the conclusions obtained from the proposed method. The utility of the classification accuracy investigation is further developed in Chapter 4.

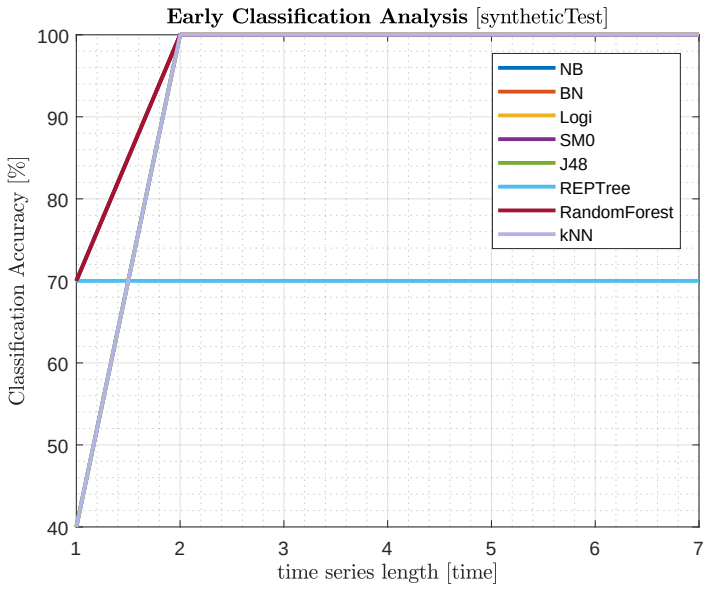


Figure A.4: Multiple classifiers performance accuracy on the data represented in Table A.1, for every time series lengths.