Petrônio Cândido de Lima e Silva

Scalable Models for Probabilistic Forecasting with Fuzzy Time Series

Belo Horizonte - Minas Gerais

July, 2019

Petrônio Cândido de Lima e Silva

Scalable Models for Probabilistic Forecasting with Fuzzy Time Series

Final thesis presented to the Graduate Program in Electrical Engineering of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

Federal University of Minas Gerais - UFMG

Graduate Program in Electrical Engineering - PPGEE

Machine Intelligence and Data Science Laboratory - MINDS

Supervisor: Frederico Gadelha Guimarães

Co-supervisor: Hossein Javedani Sadaei

Belo Horizonte - Minas Gerais

July, 2019

We all are the sum of many people - the result of their direct or indirect efforts for us. We have no merit alone neither did nothing in our life by ourselves. Because this I dedicate this work to the wide web of people that somehow helped me to become what I am. When I remember the people around me in my life, with varying degrees of closeness, I realize that it is almost impossible to separate what is my own and what is influence absorbed from others. Because of this, I am continuously meditating about all my ancestors (also my parents in law) and my friends. It's always on my mind their disposition to help and their sacrifices to offer better life conditions to their sons, actions that paved the road I'm walking now. I'm grateful to all of you!

Acknowledgements

I would like to thank all my familly, but at first place my parents Joaquim Cândido and Édina Lúcia that are the main responsible for this achievement. You have no idea the respect, admiration and love I have for you. You are my heros! My brothers André and Ivana, my best friends since ever and forever and people that always inspired me. My parents in law, Antônio Dourado and Arlete Fraga, precious gifts that life gave me and I love and keep inside my heart.

My grandfathers Olavo Cândido and Otávio Perpétuo and my grandmothers Maria da Conceição (Lia) and Maria Salomé (Nina), *in memoriam.*

My precious jewel of light, my wife Nikaelle Fraga, I'm really devoted to you! You suffer with me each step of this odyssey, each pain and each victory. You deserve more credits for this than me, for having supported me and encouraged me to follow my dreams. You give meaning to my favorite quote: *Ubi thesaurus tuus est, ibi cor tuum est.* I love you!

All my uncles, especially those from BH who were close to me during this journey: João Domingos, Maria Vita, Geralda Magela and Alencar Sanches (no, I didn't forget about those from VGP and SSA, I love you!). All my cousins, especially Bruno Araújo and Alexandra Ank, thank you for everything! My sister-in-law Ingrid and my godchildren João Rafael and Antônio Nicolau, thanks for the good vibes.

My advisor Prof. Dr. Frederico Gadelha Guimarães, the best advisor ever and really a gifted person. I'm really proud to be your student and it's a honor to have the opportunity to learn with you! Best wishes also for his wife Rúbia Pereira and his lovely daughter Laura.

My co-advisor Prof. Dr. Hossein Javedani Sadaei, a fantastic person which to a person who helped me and taught me a lot, also possibly the world's greatest specialist on Fuzzy Time Series. It's a privilege to count on you!

My lifelong best friends Prof. Dr. Gefter Thiago, Prof. Dr. Márcio Ramos (Bisa) and Prof. Rodrigo Carneiro Brandão. Even living so far and been so unplugged, my thoughts will always be with you! You are the best guys ever!

All professors and staff of the PPGEE - UFMG and my teammates on MINDS - Machine Intelligence and Data Science Laboratory, very special and careful people!

Marcos Alves, Carlos Severiano, Gustavo Vieira, Tamires Rezende, Rodrigo Pedrosa, Cristiano Leite, Ivan Reinaldo, Rúbia Reis, Leonardo Augusto, Giulia Zanon, Maria Victória, Fernando Galindres, Roozbeh, Kossar, Omid Orang, Babak and Bruno Alberto. Thank you very much!

All colleagues and students of IFNMG - Instituto Federal do Norte de Minas Gerais. I will not nominate to avoid any omission, except for these special guys: Paulo Vitor (PV), Felipe (Bolinha) and Patrícia Lucas, also my colleagues at PPGEE.

Finally, my kittens Perrengo, Yôda, Salém and Vingador, my companies during the long nights writing this work. Meow!

Glória in Excélsis Deo!

“Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better” (Edsger Wybe Dijkstra)

Abstract

No campo da previsão de séries temporais os métodos mais difundidos baseiam-se em predição por ponto. Esse tipo de previsão, no entanto, tem um sério inconveniente: ele não quantifica as incertezas inerentes aos processos naturais e sociais nem outras incertezas decorrentes da captura e processamento dos dados. Por isso nos últimos anos os métodos de previsão intervalar e probabilística têm ganhado a atenção dos pesquisadores, particularmente nas ciências climáticas e na econometria. Mas outro inconveniente vem do fato de grande parte dos métodos de previsão probabilística serem métodos de caixa preta e demandarem simulações estocásticas ou *ensembles* de métodos preditivos que são computacionalmente despendiosos.

Por outro lado, o volume (número de registros) e a dimensionalidade (número de variáveis) dos dados vêm alcançando magnitudes cada vez maiores, graças ao barateamento dos dispositivos computacionais de captura e armazenamento de dados, um fenômeno conhecido como *Big Data*. Tais fatores impactam diretamente no custo de treinamento e atualização dos modelos e, para séries temporais com essas características, a escalabilidade tornou-se um fator decisivo na escolha dos métodos preditivos.

Nesse contexto emergem os métodos de Séries Temporais Nebulosas, que vêm em crescente expansão nos últimos anos dado os seus resultados acurados, a facilidade de implementação dos métodos, o seu baixo custo computacional e a interpretabilidade de seus modelos. Os métodos de Séries Temporais Nebulosas têm sido utilizados em áreas como previsão de demanda energética, indicadores e ativos de mercado, turismo entre outras. Mas há lacunas na literatura de tais métodos referentes a escalabilidade para grandes volumes de dados e previsão probabilística e por intervalos.

A presente tese propõe novos métodos escaláveis de Séries Temporais Nebulosas e investiga a aplicação desses modelos na previsão por ponto, intervalar e probabilística, para uma ou mais variáveis e para mais de um passo à frente. Os parâmetros e hiperparâmetros dos métodos são discutidos e são apresentadas alternativas de ajuste fino dos modelos. Os métodos propostos são então comparados com as principais técnicas de Séries Temporais Nebulosas e outros modelos estatísticos utilizando dados ambientais e do mercado de ações. Os modelos propostos apresentaram resultados promissores tanto nas previsões por ponto quanto nas previsões por intervalo e probabilísticas e com baixo custo computacional, tornando-os úteis para um vasta gama de aplicações.

Palavras-chave: Séries Temporais Nebulosas, Previsão Probabilística, Escalabilidade, Previsão por Intervalo.

Abstract

In the field of time series forecasting, the most known methods are based on point forecasting. However, this kind of forecasting has a serious drawback: it does not quantify the uncertainties inherent to natural and social processes neither other uncertainties caused by the data gathering and processing. Because this in last years the interval and probabilistic forecasting methods have been gaining more attention of researches, specially on environmental and economical sciences. But these techniques also have their own issues due to the methods being black-boxes and requiring stochastic simulations and ensembles of multiple forecasting methods which are computationally expensive.

On the other hand, the data volume (number of instances) and dimensionality (number of variables) have reached magnitudes even greater, due to the commoditizing of the capturing and storing computational devices, in a phenomenon known as Big Data. Such factors impact directly on the model's training and updating costs, and for time series with Big Data characteristics, the scalability became a decisive factor in the choosing of predictive methods.

In this context the Fuzzy Time Series (FTS) methods emerge, which have been growing in recent years due to their accurate results, easiness of implementation, low computational cost and model explainability. The Fuzzy Time Series methods have been applied to forecast electric load, market assets, economical indicators, tourism demand etc. But there is a lack on FTS literature regarding interval and probabilistic forecasting.

This thesis proposes new scalable Fuzzy Time Series methods and discusses its application to point, interval and probabilistic forecasting of mono and multivariate time series, for one to many steps ahead. The parameters and hyper-parameters are discussed and fine tunning alternatives are presented. Finally the proposed methods are compared with the main Fuzzy Time Series techniques and other literature approaches using environmental and stock market data. The proposed methods obtained promising results on point, interval and probabilistic forecasting and presented low computational cost, making it useful for a wide range of applications.

Keywords: Fuzzy Time Series, Probabilistic Forecasting, Interval Forecasting, Scalable Models.

Preface

"From my part I know nothing with any certainty, but the sight of the stars makes me dream."
— Vincent Van Gogh

When Prof. Fred suggested me to study Fuzzy Time Series - I need to confess - I became excited. Because one of the most fascinating issues on scientific research is to deal with uncertainty. Uncertainty is pervasive, omnipresent and self propagated. Pliny the Elder, early on first century of Cristian Age, stated that "the only certainty is the uncertainty". The mankind expanded the boundaries of the knowledge and some uncertainties could be reduced or eliminated. Others, however, remain irreducible. And here we are!

I always felt uncomfortable with the mechanistic and deterministic view of the world. The advances of science have forced us to assume some limitations of our knowledge and accept the separation between our known-knowns, the known-unknowns and even of the unknown-unknowns. We know now that we live in a fuzzy and probabilistic world.

And until here we just talked about the present and the past. Things get even more interesting when we try to look ahead and predict the future. If we can't measure accurately some natural, social and economical processes, due to instrumentation limitations for example, and these processes are also intrinsically non-deterministic, these uncertainties combined make the forecasting task complex and barely precise.

The fog of uncertainty becomes yet more dense as the forecasting horizon goes away: the forecasting methods need to into take account all uncertainties on present to forecast ranges of possibilities on future. When we look more than one step in the future the forecasting method should consider all possible combinations in the range of variation of each past step - and this increases the complexity and the output uncertainty.

With this research problem in hand, many ideas in mind, and a lot of excitement, we expect to give some contributions to this field. We focused on non-deterministic processes and assume that all measurements are not completely accurate, every single value actually represents a fuzzy neighborhood. We propose to bring the fuzzy time series to the domain of probabilistic forecasting.

I hope you enjoy this work as I enjoyed dreaming and implementing it.

List of abbreviations and acronyms

ARIMA	Autoregressive Integrated Moving Average
BSTS	Bayesian Structural Time Series
CRPS	Continuous Ranked Probability Score
DEHO	Distributed Evolutionary Hyperparameter Optimization
FIG	Fuzzy Information Granule
FIG-FTS	Fuzzy Information Granule Fuzzy Time Series
FLR	Fuzzy Logical Relationship
FLRG	Fuzzy Logical Relationship Group
FTP	Fuzzy Temporal Pattern
FTPG	Fuzzy Temporal Pattern Group
FTS	Fuzzy Time Series
HOFTS	High Order Fuzzy Time Series
IFTS	Interval Fuzzy Time Series
LHS	Left Hand Side
MAE	Mean Absolute Error
MAPE	Mean Average Percent Error
MVFTS	Multivariate Fuzzy Time Series
PWFTP	Probabilistic Weighted Fuzzy Temporal Pattern
PWFTPG	Probabilistic Weighted Fuzzy Temporal Pattern Group
PWFTS	Probabilistic Weighted Fuzzy Time Series
QAR	Quantile Autoregression

RHS	Right Hand Side
RMSE	Root Mean Squared Error
UoD	Universe of Discourse
WHOFTS	Weighted High Order Fuzzy Time Series
WIFTS	Weighted Interval Fuzzy Time Series
WMVFTS	Weighted Multivariate Fuzzy Time Series

List of symbols

$Y \in \mathbb{R}^n$	the crisp time series data
$n = \|\mathcal{V}\|$	the number of variables of Y, univariate if $n = 1$ or multivariate if $n > 1$
$y(t) \in Y$	an individual instance of Y on time t
$F \in \tilde{A}$	Linguistic time series produced by Y fuzzyfication
$f(t) \in F$	an individual instance of F on time t, such that $f(t) = \{A_j \mid \mu_{A_j}(y(t)) \geq \alpha \; \forall A_j \in \tilde{A}\}$
$U = [\underline{l}, \overline{u}]$	the Universe of Discourse of a univariate Y, where the lower bound is $l = \min Y$ and the upper bound is $u = \max Y$.
$T \in \mathbb{N}^+$	the total length of Y
$t \in T$	the time index
$k \in \mathbb{N}^+$	the number of partitions of U
$\alpha \in [0, 1]$	the Alfa-Cut, the minimal membership grade to be considered in fuzzyfication process
$\Omega \in \mathbb{N}^+$	the Model order, the number of time series lags used by model
L	the lag indexes
$\mu_{A_j} : U \rightarrow [0, 1]$	the fuzzy membership function for fuzzy set A_j, $j = 1..k$
$\tilde{A}$	the linguistic variable for univariate Y
$A_j \in \tilde{A}$	the individual fuzzy sets in $\tilde{A}$, $j = 1..k$
$\mathcal{V}$	the set of variables of a multivariate Y
$\mathcal{V}_i \in \mathcal{V}$	an individual variable of Y, $i = 1..n$
$*\mathcal{V} \in \mathcal{V}$	the target variable (or endogenous variable) of Y
U_i	the Universe of Discourse of each $\mathcal{V}_i$, for multivariate Y , $i = 1..n$

k_i	Number of partitions of U_i for each $\mathcal{V}_i$, for multivariate Y, $i = 1..n$
α_i	Alpha-Cut for each $\mathcal{V}_i \in \mathcal{V}$, $i = 1..n$
$\widetilde{\mathcal{V}}_i$	the linguistic variable for each $\mathcal{V}_i \in \mathcal{V}$, the group of fuzzy sets, $i = 1..n$
$A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i$	the individuals fuzzy sets in $\widetilde{\mathcal{V}}_i$, $i = 1..n$ and $j = 1..k_i$
$\mathcal{M}$	the FTS model, including the linguistic variable $\tilde{A}$ and the knowledge model
$\|\mathcal{M}\|$	the model parsimony, the amount of parameters of the model
$H \in \mathbb{N}^+$	The forecasting horizon, i.e, the number of steps to predict ahead
$\hat{y}(t+1) \in \mathbb{R}^n$	a point forecast for time $t+1$
$\mathbb{I} = [\underline{l}, \overline{u}]$	a prediction interval for time $t+1$ with lower bound $l \in \mathbb{R}$ and upper bound $u \in \mathbb{R}$
$P : U \to [0,1]$	a probability distribution forecast for time $t+1$

Contents

Chapter 1

Introduction

"In the strict formulation of the law of causality - if we know the present, we can calculate the future - it is not the conclusion that is wrong, but the premise."
— Werner Heisenberg

A significant part of scientific applications demand the forecasting of natural, social and economical processes and there is an extensive literature on forecasting methods and models. Such methods are also preceded by many processes as instrumentation, measurement, storing, aggregation, etc. However, a great recurrent problem is how to deal with the uncertainty generated or captured in each step of this task, and measure how it spreads. Makridakis and Taleb [2009] stated that "Statistical models underestimate uncertainty, sometimes catastrophically", by assuming, for example, that events are independent, forecasting errors are tractable, the variance of forecasting errors is finite, known and constant.

In these natural and social processes the uncertainty can be intrinsic or extrinsic and is classified, by Georgescu [2014], in two categories: the epistemic uncertainty and the ontological uncertainty. The ontological uncertainty represents the intrinsic and irreducible uncertainty of a process defined basically as the non-deterministic behavior – randomness and stochasticity – that usually is modeled by the probability theory.

Contrarily, the epistemic uncertainty represents the extrinsic and reducible sources of uncertainty on a process like vagueness, lack of information and imprecision due to measurement errors, sensor calibration, rounding and limitations of numerical precision, among others. Another possibility is the conversion of continuous processes to discrete processes. This conversion is not lossless and some uncertainty is imputed on converted data. The epistemic uncertainty can be modeled by the fuzzy theory.

This is the case of data preprocessing tasks, for example. Very often time series datasets need to be aggregated by some time resolution (daily, hourly, etc) and this

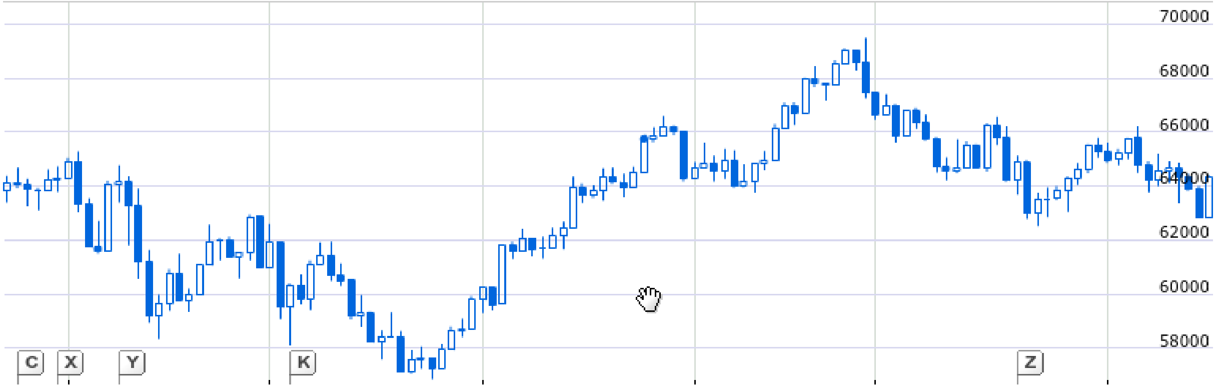


Figure 1 – Candlestick chart for IBOVESPA. Source: Google Finance[1]

aggregation also introduces the epistemic uncertainty on data. A good example are the financial time series that summarize all transactions of a whole day in four numbers: opening, minimum, maximum and closing prices. This method is an attempt to represent the volatility (e. g. the uncertainty) of the price value inside a certain time window and is also a tool for detecting patterns on data, the Candlestick Graph techniques, as shown in Figure 1. Sometimes the aggregation is even more aggressive and all the values are summarized into one, the average or median value, hiding all information about the volatility. When this data is used as input for fitting a forecasting model the extrinsic uncertainty is introduced.

The forecasting methods propagate the input uncertainties on their outputs and compromise the reliability of the forecast. Despite these points, the majority of forecasting methods are concerned with one step ahead point forecasting without output uncertainty measures. When the many steps ahead forecasting is considered, the uncertainty grows yet more and affects the accuracy and reliability of models. This effect becomes even worst as the forecasting horizon becomes wider.

This fact led to the development of methods for Probabilistic Forecasting Gneiting and Katzfuss [2014] and Interval Forecasting Chatfield [1993], to deal with forecasting uncertainty by estimating distributions of possible values instead of a unique point forecast. However, traditional methods of probabilistic forecasting require the use of parametric models with distribution assumptions, as in Bayesian Inference, or costly estimation techniques and Monte-Carlo simulations. Probabilistic forecasting has been used in areas such as weather forecasting (Fraley et al. [2011] and Leutbecher and Palmer [2008]) eletric load forecasting (Hong and Fan [2016], Hong et al. [2016] and Liu et al. [2015]), wind power generation prediction (Pinson et al. [2006] and Netto et al. [2016]) and hydrological forecasting (Laio and Tamea [2007]).

Side by side with the uncertainty representation in time series forecasts, some

[1] https://www.google.com/financeq=INDEXBVMF%3AIBOV&ei=zHz2WOmMKODrep_bgbAI. Access in 18/04/2017

other factors became prevailing in the evaluation of predictive methods, such as Big Data scalability and model explainability. The Big Data phenomenon emerged in the late 2000s decade Lynch [2008] calling attention to new demands on data analysis brought by the growth of data volume, dimensionality and variety. This revolution was possible due to the advances on the technologies of data capturing and storage, as well as its commoditizing, allowing the distributed management of amounts of data never seen before.

But traditional forecasting methods, and even some newer ones, were not designed to deal with such high volume of data. The most critical issues were the high dimensionality (dozens of hundreds of attributes) and volume (hundreds of millions or billions of samples) Qiu et al. [2016]. Such data volume cannot be grounded on a single machine memory and demands a distributed architecture of storage and processing. New technologies emerged to tackle these issues, for instance the Map Reduce based frameworks Dean and Ghemawat [2008], a divide-and-conquer approach which is the basis of Hadoop clusters White [2012], where the processing units also act as storage units of the data subsets using commodity and cheap hardware assets.

The Big Data issues do not stop on data volume and, indeed, the velocity and variability of the data must be seriously considered when developing forecasting models. Many sources of data have non-stationary behaviors, meaning that their statistical properties may change along the time. Some models are time-variant, they are specially designed to be self-adaptable and evolve as the data change. But the majority of the conventional methods are time-invariant and need to be retrained periodically depending on the variability of the data, what can be problematic given the computational cost of the training and adapting procedures. Yet more critical is the data with Concept Drifts [Gama et al., 2014], which have high volume with high velocity and need to be adaptable to new behaviors in feasible time.

Another traditionally neglected aspect of the forecasting methods also started to gain more attention in recent years: the explainability. With the expansion of the machine learning methods enabled to tackle Big Data, another issues came up to the light: black-box methods started to find legal barriers or adoption resistance due to its lack of transparency and auditability [Leslie, 2019, EC AI HLEG, 2019]. Diversely, the white-box methods can help in the knowledge extraction and simplification of complex temporal patterns, besides being easily auditable.

The exposed scenario favored the rising of the Fuzzy Time Series (FTS) methods Song and Chissom [1993b], which have been drawing more attention and relevance in recent years due to many studies reporting its good accuracy compared with other models Singh and Prabakaran [2008]. Fuzzy Time Series are soft-computing methods that produce data-driven, non-parametric, simple, computationally cheap and readable models for time series analysis and forecasting. FTS methods are also an approach to deal with the epistemic uncertainty, as on the time-aggregation case of the financial time series. The fuzzyfication of data gives a more flexible representation to the individual measures, embracing the range of possible value fluctuations not covered by the single values.

Despite the great improvements published in the FTS literature in the recent years, there are still some notable lacks. Interval and probabilistic forecasting, specially for many steps ahead, are not properly explored, besides the absence of scalable methods to tackle big multivariate time series. There is, indeed, a plethora of soft-computing forecasting methods. But very few of them are flexible enough to incorporate scalable point, interval, probabilistic and multivariate forecasting, with one to many steps ahead, for univariate and multivariate time series. This research opportunity is exploited in this work by the proposition of new FTS methods and their subsequent applications on several case studies, including financial, environmental and energy time series.

1.1 Objectives

The main goal of this thesis is to develop a scalable probabilistic forecasting approach based on the Fuzzy Time Series methods, providing a flexible computational framework for applications with uncertainties. Specific goals are divided in:

- Identify the strengths and weaknesses of the main FTS approaches presented in literature;
- Identify extension opportunities on known probabilistic forecasting methods;
- Introduce the Probabilistic Weighted Fuzzy Time Series, a new method family that exploits uncertainties on datasets to capture time series patterns and translate them into the rule-based knowledge system (the Probabilistic Weighted Fuzzy Logical Relationship Groups);
- Improve PWFTS scalability in order to enable it to deal with big time series by proposing a distributed processing design with the Map/Reduce paradigm;
- Extend the PWFTS method to enable multivariate time series using Fuzzy Information Granules.

1.2 Work structure

This thesis is organized as follows:

- **Chapter 2 - Fuzzy Time Series** presents a contextual background on Fuzzy Time Series methods and discusses the most relevant methods in literature;

- **Chapter 3 - Probabilistic Forecasting** introduces the probabilistic forecasting concepts and main techniques, reviewing the surrounding literature. In Section 3.3 the Interval Fuzzy Time Series method is proposed for binding the fuzzy uncertainty on forecasts, and in Section 3.5 the EnsembleFTS method is proposed to probabilistic forecasting;

- **Chapter 4 - Probabilistic Weighted Fuzzy Time Series** introduces the Probabilistic Fuzzy Time Series method for generating the Probabilistic Weighted Fuzzy Temporal Pattern rules and a method to generate one step ahead probabilistic forecasts; in Section 4.3.2 the previous method is extended to create prediction intervals and in Section 4.3.3 a simple heuristic to produce point forecasts is presented . Section 4.4 presents extensions for many steps ahead forecasting and high-order models. The characteristics and parameters of the methods are discussed in Section 4.5;

- **Chapter 5 - Scalability And Hyperparameter Optimization** proposes a distributed approach for PWFTS, based on Map/Reduce paradigm and computational clusters, enabling it to deal with big time series. In Section 5.3 the Distributed Evolutionary Hyperparameter Optimization (DEHO) is proposed, employing evolutionary algorithms with the distributed processing to improve the performance.

- **Chapter 6 - Multivariate Models** proposes an extension for multivariate time series using Fuzzy Information Granules (FIG) and an incremental universe of discourse partitioner. With this extension PWFTS can be used for multivariate forecasting in a Multiple Input/Multiple Output (MIMO) design or monovariate forecasting in a Multiple Input/Single Output(MISO) design.

- **Chapter 7 - Conclusion** the findings are summarized and overall conclusions are given, as well as the contributions, known limitations and future investigations.

1.3 Main contributions

This research presents contributions to the Forecasting and Fuzzy Time Series research fields, whose the most important are summarized below:

- development of the Probabilistic Weighted Fuzzy Time Series - PWFTS, which is a new non-parametric, data driven and highly accurate forecasting method, the first FTS method in the literature that integrates point, interval and probabilistic forecasting in the same model, for one to many steps ahead;

- a new representation method for fuzzy temporal rules, with weights on the precedent and the consequent of the rules, reflecting its *a priori* and *a posteriori* empirical probabilities and aiding with model explainability;
- new defuzzyfication methods capable to produce probability distributions, prediction intervals and point forecasts and multivariate output using Fuzzy Information Granules (FIG);
- the pyFTS library Silva et al. [2018][2] for Python programming language, an open and free framework to facilitate the development of new models and help on research reproducibility;
- application of the proposed methods in the forecasting of renewable energy and environmental processes;

[2] `http://pyfts.github.io/pyFTS/`

Chapter 2

Fuzzy Time Series

"There's nothing worst than a sharp image of a fuzzy concept"

— Ansel Adams

This chapter aims to introduce the Fuzzy Time Series methods and review the relevant literature, offering a soft background on the key concepts and models on this research field.

The name Fuzzy Time Series (FTS) can be used to refer to F, a time series composed by fuzzy linguistic terms, or the family of non-parametric forecasting methods introduced by Song and Chissom [1993b] based on Fuzzy Set theory Zadeh [1965]. These methods are easy to implement and very flexible, affording ways to deal with numeric and non-numeric data. FTS methods have been commonly employed in forecasting of university enrollments (Song and Chissom [1993b], Song and Chissorn [1994], Ismail and Efendi [2011]), stock markets (Sadaei et al. [2016b], Lee et al. [2013a], Chen [2014], Sun et al. [2015], Talarposhti et al. [2016], Efendi et al. [2013]), tourism (Lee and Javedani [2011]), electric load (Ismail et al. [2015], Sadaei et al. [2017]), seasonal time series [Song [1999], Chang [1997]] among many others. There are still some gaps in FTS methods (Sadaei [2013] and Georgescu [2010]) related with methodological problems but many of them have been approached in more recent studies Sadaei et al. [2016a]. There are several categories of FTS methods whose main features and its variations can be seen in Figure 2, and will be discussed in the remaining sections.

The most important categories of FTS methods are related with their the time behavior. The time invariant models are the ones used when the Universe of Discourse and data behavior does not change with time, as in stationary time series. Non stationary time series, in its turn, require time variant models as proposed in Song and Chissorn [1994] and Wong et al. [2010]. This research is focused on time invariant models and hereinafter all models discussed belongs to it. However, Time Variant models are not discarded for future investigations.

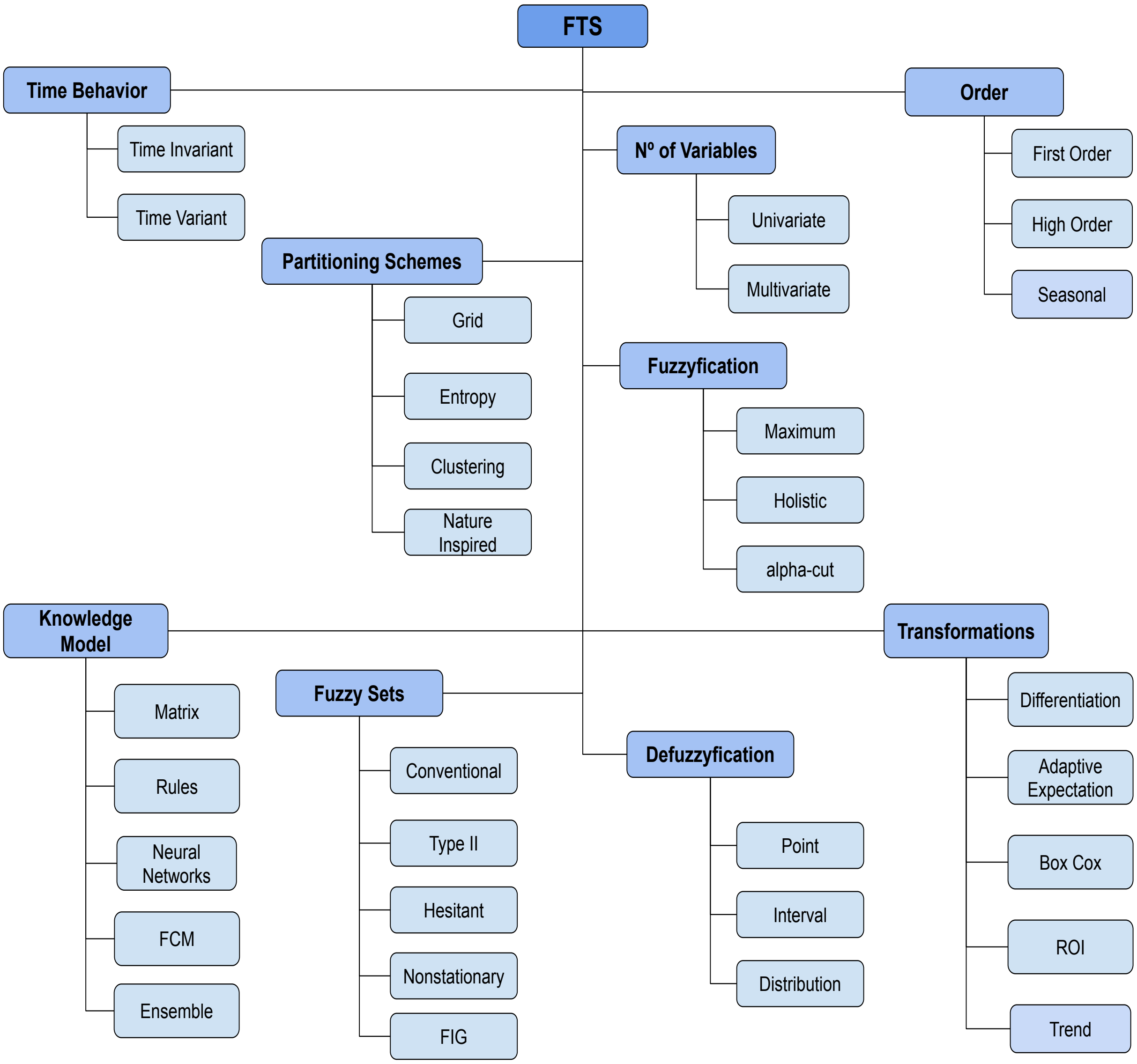


Figure 2 – A brief taxonomy of FTS methods

This chapter also focus only on monovariate and non Big Data time series. The multivariate and Big Data methods will be discussed in the following chapters. In the next section the main processes of the FTS are introduced and discussed.

2.1 Fuzzy Time Series common processes

The definition of Fuzzy Time Series, from Song and Chissom [1993b], starts with a univariate time series $Y \in \mathbb{R}^1$, for $t = 0, 1, ..., T$, where the Universe of Discourse U is delimited by the known bounds of Y, such that $U = [\min(Y), \max(Y)]$. Upon U, k fuzzy sets A_j, for $j = 1..k$, are defined and each one with its own membership function μ_{A_j}. F is called a Fuzzy Time Series over Y if $f(t) = \mu_{A_j}(y(t))$ is the collection of fuzzyfied values of Y for $j = 1..k$ and $t = 0, 1, ..., T$. The group of fuzzy sets A_j, for $j = 1..k$, can also be understood as a Linguistic Variable $\tilde{A}$, and each fuzzy set $A_j \in \tilde{A}$ is a linguistic value of the linguistic variable.

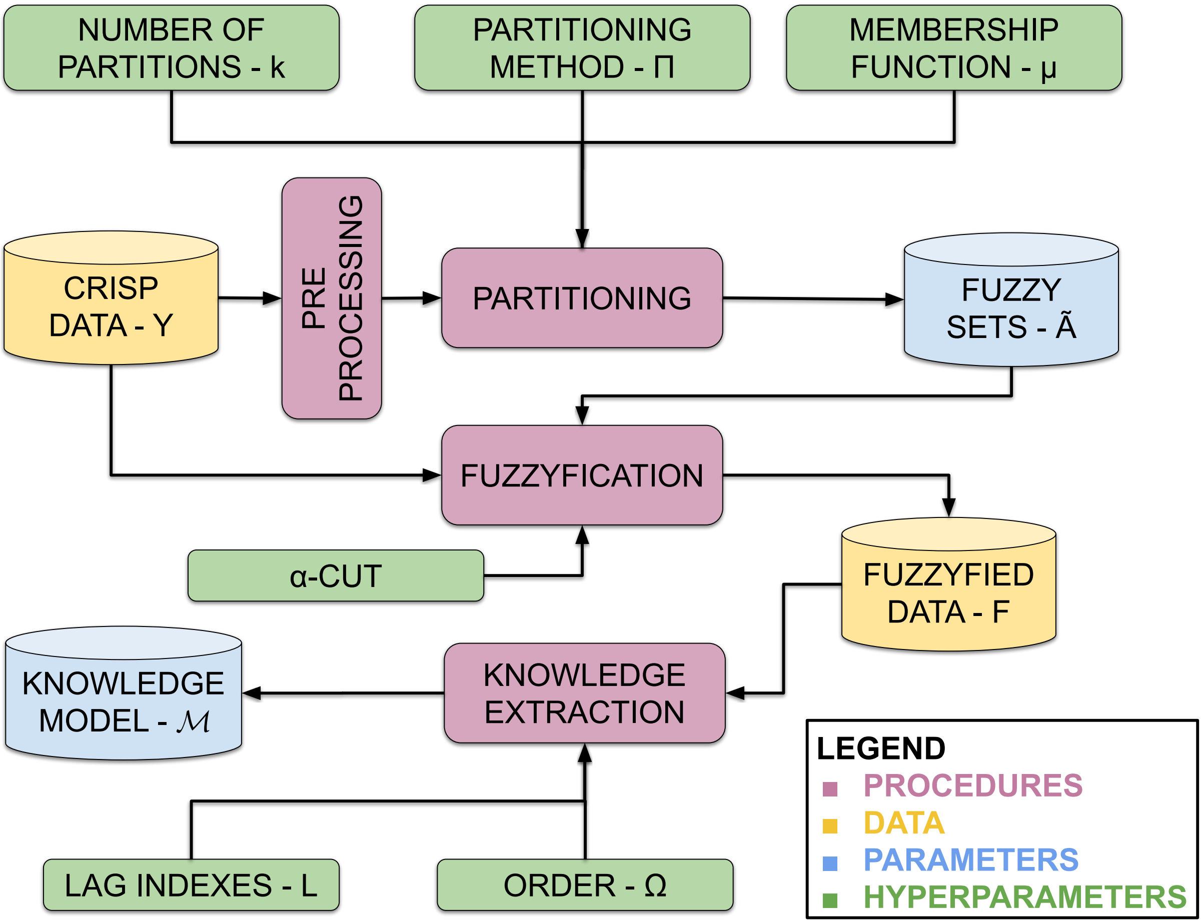


Figure 3 – Generic time invariant Fuzzy Time Series training procedure and its components

Song and Chissom [1993a] proposed the first FTS methodology and the following authors basically extended or modified some steps of the method. A generic method can be extracted from the wide range of variations of FTS methods by splitting the FTS approach in two main procedures, the training and forecasting methods. The training method, illustrated in Figure 3, has the basic objective to create the linguistic variable $\tilde{A}$ and a knowledge representation of the time series dynamics. These two objects compose the FTS model $\mathcal{M}$. The main components of this process are listed below:

Step 1 - **Pre-processing**: First, one or more pre-processing data transformations can be applied to input data Y, responsible to reduce noise, detrending, or de-seasonalize, or change the U, etc. Several methods contain these operators and their impact will be discussed in detail in Section 2.6.

Step 2 - **Partitioning**: The most important process of the training is executed, the partitioning. This process is responsible to split the universe of discourse U into k fuzzy sets A_j, creating the linguistic variable $\tilde{A}$ used to describe Y. There are many ways that the partitioning can be performed, and the most important are discussed in Section 2.2.

Step 3 - **Fuzzyfication**: With the linguistic variable $\tilde{A}$ the crisp data Y can be transformed in a linguistic representation, the fuzzy time series F where each $f(t) \in F$ is a fuzzyfied version of $y(t) \in Y$. Details of the fuzzyfication are discussed in Section 2.3.

Step 4 - **Knowledge Extraction and Representation**: The second most important process is the knowledge extraction. This process is responsible to induce the knowledge model $\mathcal{M}$ by performing a pattern recognition over F and learning the temporal patterns between Ω lags, whose indexes are identified by L, and their consequent ones. The most important learning algorithms and knowledge models are discussed in Section 2.4.

Once the linguistic variable $\tilde{A}$ is defined and the FTS model $\mathcal{M}$ was learned, new samples $y(t) \in U$ can be presented to produce a forecast $\hat{y}(t+1)$. The generic forecasting procedure is illustrated in Figure 4, and its main components are listed below:

Step 1 - **Pre-processing**: First, one or more pre-processing and post-processing data transformations can be applied to input sample $y(t)$ (the data transformations are discussed in Section 2.6).

Step 2 - **Fuzzyfication**: The fuzzyfication procedure follows the same schema of the training procedure and it is discussed in Section 2.3.

Step 3 - **Inference**: The inference engine deeply depends on the knowledge model $\mathcal{M}$. Indeed, the learning algorithm, the knowledge model and the inference engine are intrinsically correlated, and they are discussed in Section 2.4. The aim of the inference process is to produce $f(t+1)$, candidate fuzzy sets (and other additional information, as weights) to represent the future crisp value $y(t+1)$.

Step 4 - **Deffuzyfication**: The deffuzyfication process aims to transform the $f(t+1)$ to a crisp numeric estimate $\hat{y}(t+1)$ for the real (but unknown) value $y(t+1)$. The deffuzyfication usually also depends on the inference engine but there are common methods discussed in Section 2.5. The present work extended the possibilities of deffuzyfication to beyond of point forecasting, proposing methods for prediction intervals $\mathbb{I}$ and probabilistic distributions P.

Step 5 - **Post-processing**: Finally, one or more post-processing data transformations can be applied to output forecast $\hat{y}(t+1)$ (the data transformations are discussed in Section 2.6).

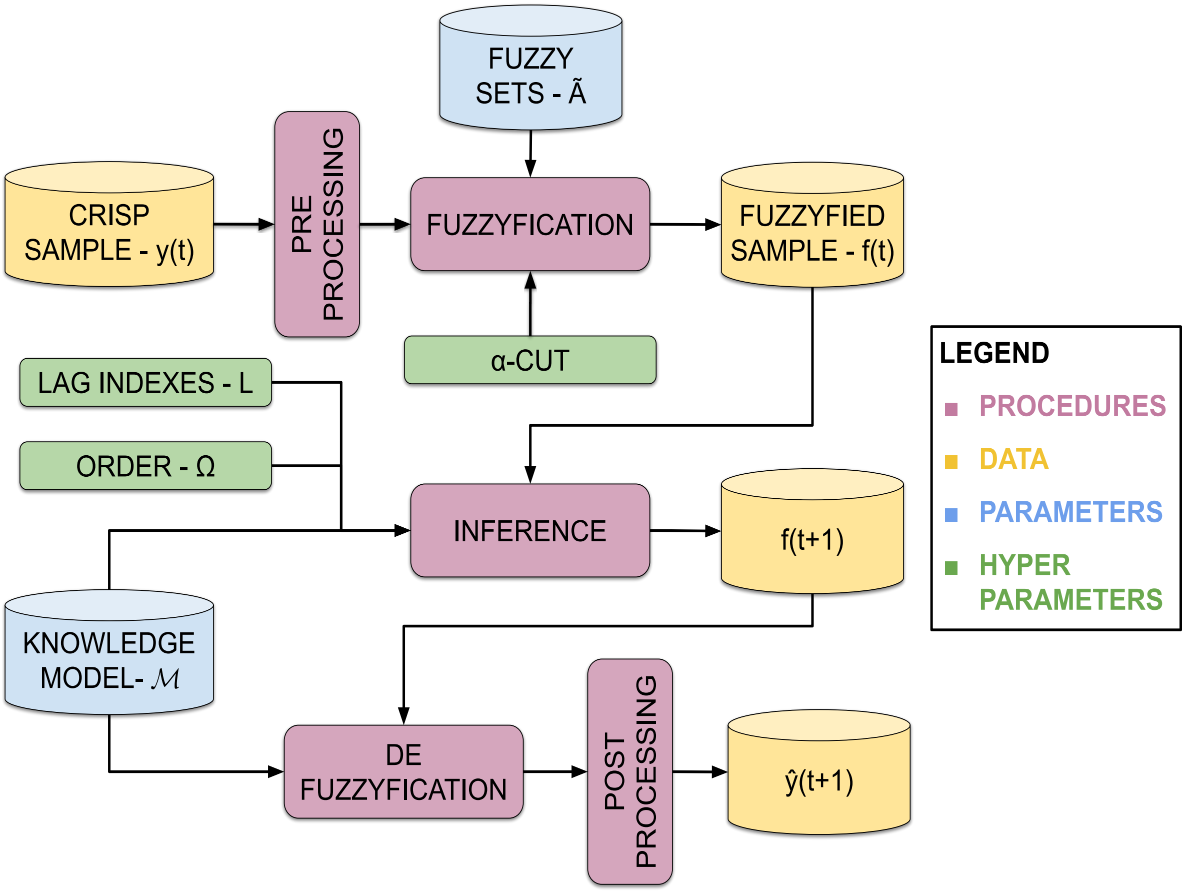


Figure 4 – Generic time invariant Fuzzy Time Series forecasting procedure and its components

The main hyperparameters which affect these processes are listed in Table 1. The selection of the values for these hyperparameters affects the training process and the parameters of the model, including its accuracy and parsimony (the length of the model). In the following sections each one of these processes are discussed in detail recurring to the most relevant works in the FTS literature, while its strengths and drawbacks are highlighted.

2.2 Universe of Discourse Partitioning

This process aims to split the Universe of Discourse U and to create the linguistic variable $\tilde{A}$, composed by the fuzzy sets A_j, $j = 1..k$. This is the most important process of the FTS approach and the following sections detail its main sub-processes and parameters.

2.2.1 Universe of Discourse U

The natural definition of the Universe of Discourse is $U = [\min(Y), \max(Y)]$, but it is common that the upper and lower bounds be exceeded by a confidence margin.

Alias	Parameter	Process
$k \in \mathbb{N}^+$	Number of partitions (fuzzy sets)	Universe of Discourse Partitioning
$\mu : U \rightarrow [0, 1]$	Membership function, measures the membership of a value $y \in U$ to a fuzzy set	Universe of Discourse Partitioning, Fuzzyfication
Π	Partitioning method	Universe of Discourse Partitioning
$\alpha \in [0, 1]$	the α-cut, the minimal membership grade to take account on fuzzyfication process	Fuzzyfication
$\Omega \in \mathbb{N}^+$	Order, the number of lags	Knowledge model
$L \in \Omega \times \mathbb{N}^-$	Time lag indexes	Knowledge model

Table 1 – Common hyperparameters for FTS methods

Then, it can be established as $U = [\underline{l}, \overline{u}]$, with the lower bound as $\underline{l} = \min(Y) + l_d$, where $l_d = \min(Y) \cdot 0.2$ (exceeding the original lower bound by 20%) and the upper bound as $\overline{u} = \max(Y) + u_d$, where $u_d = \max(Y) \cdot 0.2$ (exceeding the original upper bound by 20%).

Even considering just time invariant methods in this work and a stationary time series Y, it is natural that the testing U will be a little different from the training U, sometimes by a small fraction of the original values. Even on stationary processes the presence of outliers cannot be discarded. The objective of these exceeding margins l_d and u_d is to help in the fuzzyfication process of the forecasting procedure, in order to accommodate fluctuations in the bounds of the known U.

2.2.2 Membership function μ

Once U has been defined, three hyperparameters will determine the creation of $\tilde{A}$: the number of partitions k, the membership function μ and the partitioning scheme. The membership function $\mu : U \rightarrow [0, 1]$ defines how much a crisp value belongs to a fuzzy set, in terms of the membership grade $[0, 1]$. Some options of fuzzy membership functions are shown in Table 2, and simple partitioning using these functions are shown in Figure 5. The real impact of the membership function on accuracy is low, it will be demonstrated empirically later on this work, but the chosen of the correct μ can help in the readability and explainability of the model.

Many other kinds of fuzzy sets were presented in the literature and new FTS methods were developed using them, as instance Type 2 fuzzy sets [Huarng and Yu, 2005, Bajestani and Zare, 2011], Hesitant fuzzy sets [Bisht and Kumar, 2016], non-stationary fuzzy sets [Alves et al., 2018], etc. These fuzzy sets and the methods developed with them, however, are considered out of the scope of this research.

Name	Parameters	Definition
Singleton	c, the central value	$\mu(x,c) = \begin{cases} 1 & if \quad x = c \\ 0 & if \quad x \neq c \end{cases}$
Triangular	a: lower bound, b: midpoint, c: upper bound	$\mu(x,a,b,c) = \begin{cases} 0 & if \quad x \leq a \\ \frac{x-a}{b-a} & if \quad a \leq x \leq b \\ \frac{c-x}{c-b} & if \quad b \leq x \leq c \\ 0 & if \quad c \geq x \end{cases}$
Trapezoidal	a: lower bound, b: top-left, c: top-right, d upper bound	$\mu(x,a,b,c,d) = \begin{cases} 0 & if \quad x \leq a \\ \frac{x-a}{b-a} & if \quad a \leq x \leq b \\ 1 & if \quad b \leq x \leq c \\ \frac{d-x}{d-c} & if \quad c \leq x \leq d \\ 0 & if \quad d \geq x \end{cases}$
Gaussian	m: midpoint, d: spread	$\mu(x,m,d) = \exp\left(-\frac{(x-d)^2}{2m^2}\right)$

Table 2 – Most common fuzzy membership functions

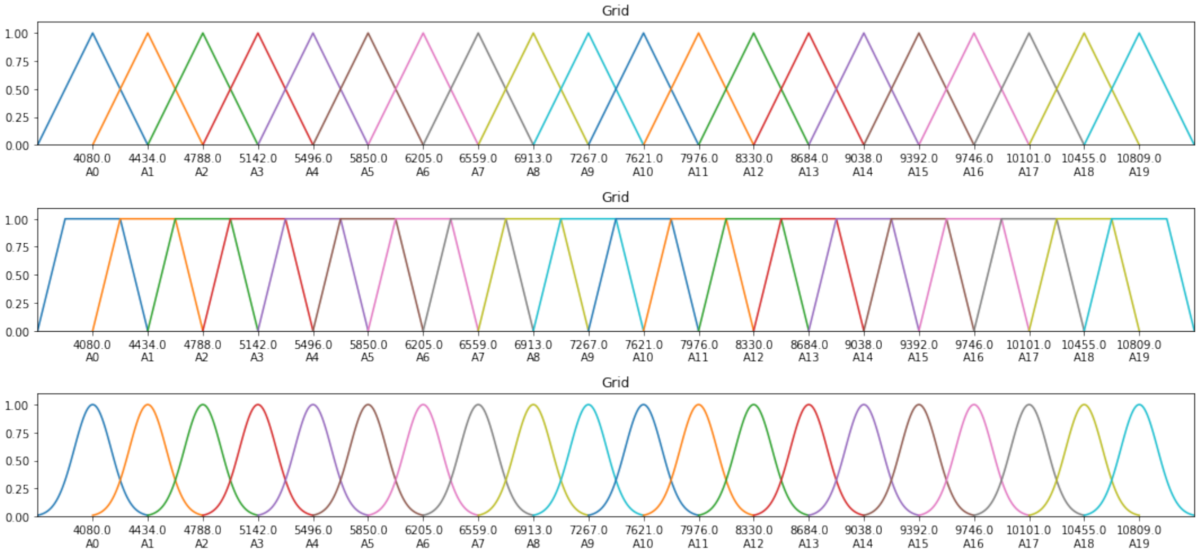


Figure 5 – UoD Partitioning using different membership functions

2.2.3 The number of partitions k

The selection of the hyperparameter k impact directly on the model accuracy and parsimony, as discussed in Duru and Yoshida [2012]. The number of partitions impact on the model parsimony directly and, for instance, given a rule model the maximum number of rules is the cartesian product between the fuzzy sets $A_j \in \tilde{A}$ for each order Ω.

There is a non-linear relationship between k and the model accuracy, a trade off between specific accuracy (bias) and overall generalization (variance). A small value of k will generate too few fuzzy sets to represent Y correctly, making the model $\mathcal{M}$ underfit by producing a gross generalization with simplistic patterns. A high value of k will generate too much fuzzy sets, exceeding the needed to represent Y and makeing the model $\mathcal{M}$ overfit by reproducing excessive specificities and small noisy fluctuations. The optimal number of k must be optimized for each problem, balancing the accuracy and the model parsimony, since this last value affects the computational performance as the number of parameters grows.

The impact of the U partitioning can also be seen by other perspectives: the model human readability and model explainability. In his seminal work, Miller [1956] stated that the human being, on average, can learn 7 ± 2 concepts. In other words, the linguistic variable $\tilde{A}$ to be reasonable for human understanding must have around this number of fuzzy sets. But this, depending on the range of U and the behavior of Y, may be a very small number of partitions. Otherwise, the explainability does not depend on how much humans can learn from $\mathcal{M}$ but how easily the forecastings produced can be explained, as a white box model. In this last case the type of the knowledge representation of $\mathcal{M}$ has greater impact than the number of fuzzy sets in $\tilde{A}$.

A study of the linguistic characterization of time series can be found in Novák [2016a]. using Fuzzy Transform (Perfilieva [2006]) to generate linguistic summaries of time series. A study on mining information from fuzzyfied linguistic time series can be found in Novák [2016b], where are presented the impacts of fuzzyfication on the knowledge extraction.

2.2.4 The partitioning method - Π

The partitioning method will determine, for each fuzzy set, their length, midpoints and bounds and also have impact on accuracy. The simplest partitioning scheme – the division of the data range in k equal length intervals – is called Grid Partitioning and was proposed in Song and Chissom [1993b]. In the Grid Partitioning, U is divided in $k+2$ even intervals $u_1, u_2, ..., u_k$ whose midpoints are $c_1, c_2, ..., c_k$. Then with these k intervals k overlapped fuzzy sets $A_1, A_2, ..., A_k$ are defined using triangular membership functions whose parameters are c_{j-1}, c_j, c_{j+1} for each $j = 2..k-1$.

Some works use simple heuristics to define k or even the lengths of the fuzzy sets. Huarng [2001] use a grid partitioning approach, but it proposes an empirical method to find the ideal number of partition lengths according to the magnitude of U, in a work that was the first to deeply discuss the impact of the partitioning on FTS forecast accuracy. Several other works used this or define other simple heuristic for U partitioning, see for instance Chang [1997], Huarng and Yu [2005], Rubio et al. [2016], Cheng and Chen [2018].

U partitionings where the fuzzy sets have unequal lengths are also present in the literature. Cheng et al. [2006] employ a fixed value of k but the entropy of data that defines the best midpoints for the fuzzy sets, which also use trapezoidal membership functions. This method is known by Entropy Partitioning and is also employed in Cheng et al. [2008b] and Chen et al. [2014]. The statistical approaches yet count with Ismail et al. [2015], which

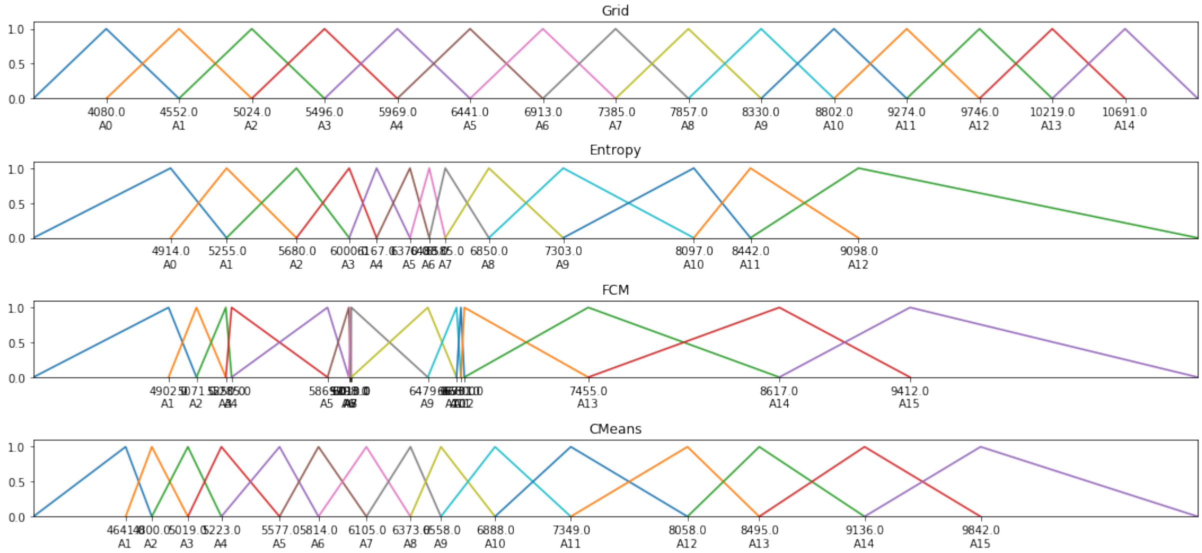


Figure 6 – Partitioning using different approaches within the same sample

determine the length of the fuzzy sets proposing a method based on data quantiles, and Yang et al. [2017a] which use a Chi-Square distribution do identify the fuzzy sets number and lengths.

Clustering techniques are used in several works, as Fuzzy C-Means in Li et al. [2008], Askari and Montazerin [2015], Bas et al. [2015], Sun et al. [2015], Yolcu and Lam [2017], Fuzzy K-Medoids in Dincer and Akkuş [2018], Self Organizing Maps in Bahrepour et al. [2011], and other methods as in Saberi et al. [2017] and Bose and Mali [2017].

The use of metaheuristics, especially of nature-inspired optimization approaches, is also spread in the literature. Particle Swarm Optimization is used by Davari et al. [2009], Kuo et al. [2009], Hsu et al. [2010], Huang et al. [2011], Zhang et al. [2018b], Genetic Algorithms in Chen and Chung [2006], Enayatifar et al. [2013], Zhang et al. [2018a], and other less known methods such as Harmony Search in Talarposhti et al. [2016], Jiang et al. [2017] and Imperialist Competitive in Sadaei et al. [2017].

A sample of different partitioning schemes on the same data can be seen in Figure 6. The partitioning method has influence on the model accuracy, parsimony and readability but its computational cost must be also considered. The Grid Partitioning gives a uniform distribution of the fuzzy sets over U but, if in one hand it is computationally cheaper, however it may not represent the importance of some data regions accordingly. It is probable that some specific regions of U have more variance than others, depending on Y behavior, and some regions may be better represented having more fuzzy sets than others. It also can not be denied that some approaches are computationally expensive, as the clustering and metaheuristics ones, and in Big Data scenarios this may be prohibitive. Despite the fact that the Grid Partitioning should be always the first approach to start with, due to its simplicity and small cost, the fine tuning of FTS models can not exclude more sophisticated methods.

Once the linguistic variable $\tilde{A}$ is created, the fuzzyfication process can be started. This process and its parameters are discussed in the next section.

2.3 The Fuzzyfication Process

This process aims to transform the crisp numerical time series Y into a linguistic time series F, also known as fuzzy time series. There are few, but important, variations of the fuzzyfication method.

The initial FTS methods, for instance Song and Chissom [1993b], Chen [1996] and Yu [2005], only considered one fuzzy set in fuzzyfication process, for instance $y(t) \in Y$, the one with the greatest membership grade. More specifically:

$$f(t) = A_j \mid \mu_{A_j}(y(t)) = \max\{\mu_{A_1}(y(t)), \ldots, \mu_{A_k}(y(t))\} \tag{2.1}$$

This method helps to control overfit, reducing the number of spurious patterns generated by low membership grades fuzzy sets. However, it can also contribute to underfit the learning by eliminating fuzzy sets which are very close to the maximum grade. It is possible to deduce that some relevant information can be lost when several minor membership grades are discarded.

A contrasting method is the holistic fuzzyfication, where all the membership grades, despite their magnitude, are considered. The fuzzyfied value $f(t)$ is the the vector of the $y(t)$ membership grades with respect to each $A_j \in \tilde{A}$:

$$f(t) = [\mu_{A_1}(y(t)), \ldots, \mu_{A_k}(y(t))] \tag{2.2}$$

The holistic fuzzyfication can help to the learning overfit, because even very small membership grades, which can be considered insignificant for that fuzzy set, are considered. An intermediate approach can be achieved by using the α-cut hyper-parameter. The α-cut represents the minimal value of the membership grade that will be accepted in fuzzyfication, while membership values below the α-cut will not be considered.

$$f(t) = A_j \mid \mu_{A_j}(y(t)) \geq \alpha \;\; \forall A_j \in \tilde{A} \tag{2.3}$$

The α-cut makes the sensibility of the fuzzyfication process adjustable and the user can control it, unlike the maximum membership and the holistic methods. The fuzzyfication method is significant in the search of the training best fit, controlling the accuracy and models parsimony. Particularly, the method of Carvalho Jr and Costa Jr

[2017] makes an explicit use of the α-cut parameter and conduct a comprehensive study of its impact on the method accuracy.

Once the crisp data Y is converted to the fuzzy time series F the process of knowledge extraction and representation is ready to start. This process and its variations are discussed in the next section.

2.4 Knowledge Extraction, Representation and Inference

This section aims to investigate the several approaches in the literature that were used to learn and represent temporal patterns found on the fuzzyfied data F. Looking back to Figures 3 and 4, the fuzzy sets and the fuzzyfication process may be interpreted as a feature extraction layer that precedes a pattern recognition and inference layer, that is finally succeeded by a reconstruction layer – the deffuzyfication process. Besides small variations, the fuzzyfication and deffuzyfication do not differ among the methods. But the way these methods learn and store the patterns suffer a strong variation among them.

By far, most of FTS methods make use of simple heuristics to learn the temporal patterns from the fuzzyfied data and store the learned patterns using rules or matrices. But it can not be denied that there are many other backends for the knowledge extraction and representation in FTS models: metaheuristics, neural networks, fuzzy cognitive models, hybrid approaches with traditional statistical models, etc. In the following sections the most relevant methods with their variations will be discussed.

2.4.1 The Order Ω, Lags L and Seasonality

First it is needed to consider the hyperparameters order Ω and the lag indexes L. These parameters also impact directly on the model accuracy and parsimony. The number of lags Ω indicate how much past information is available to the model $\mathcal{M}$ to recognize the possible temporal patterns and make a forecast. Very short-term memory or even memory-less processes will require just the last time lag, consequently produce a first order model ($\Omega = 1$). Processes with longer memories will require more lags and produce higher-order models ($\Omega > 1$).

Otherwise, the hyperparameter L indicates which past lags are taken into account during the forecast. Not always the most recent time lags contain the best information to predict the near future and this is particularly important for seasonal time series, where L will indicate the time lags which have periodically similar values. Initially the values of L can be extracted from the Autocorrelation Function (ACF), examining the most significant lags. However, this number can be optimized with fewer lags.

The *High Order Fuzzy Time Series* - HOFTS defines the High Order Fuzzy Logical Relationships - HOFLR as $LHS \rightarrow RHS$ form, where LHS is the set of $f(t - L(\Omega - 1)), ..., F(t - L(0))$ fuzzy sets, and the RHS is $f(t + 1)$, the group of consequent fuzzy sets. We can find these kind of models in Chen [2002], Chen and Chung [2006], Jilani and Burney [2008], Li et al. [2008], Egrioglu et al. [2010], Bahrepour et al. [2011], Enayatifar et al. [2013], Chen et al. [2014], Chen and Chen [2015b], Ye et al. [2016], Lee et al. [2017], Bose and Mali [2017], Sadaei et al. [2017], Guney et al. [2018], Cheng and Chen [2018], Yang et al. [2018], Zhang et al. [2018b].

Seasonal models try to represent cyclical behaviors, e. g, repeated values of the time series in regular periods. *Seasonal Fuzzy Time Series* - SFTS methods, make use of the L parameter to represent the seasonal periods as lag indexes. SFTS were first proposed in Song et al. [1997], basically by defining a seasonal index in L, such that $f(t + 1) = f(t - L)$. Chang [1997] proposed a method for capturing fuzzy trend and fuzzy seasonal indexes using Fuzzy Regression. Other seasonal methods include Tseng et al. [1999], Song [1999], Lee and Javedani [2011].

The hyperparameters Ω and L are used across several learning algorithms and knowledge representation models, which are the ways the patterns of F are extracted, stored and inferred. As stated before, the knowledge representation of $\mathcal{M}$ is important due to the human readability and explainability. White-box models have high explainability and human readability but suffer to represent high dimensional data and very complex dynamics of temporal patterns. In other hand the black-box models have low explainability and almost zero human readability (which are unfortunately subjective concepts) but are very efficient in representing high-dimensional spaces and complex temporal dynamics.

2.4.2 Matrix Models

The original work of Song and Chissom [1993b] used a Fuzzy Relationship Matrix to represent the temporal dynamics of the fuzzy time series F. In this method, each sequential pair $f(t - 1), f(t) \in F$ is grouped in *Fuzzy Logical Relationships* - FLR[1]. The FLR are fuzzy rules that describe a temporal pattern $f(t - 1) \rightarrow f(t)$, or $A_i \rightarrow A_j$ where the Left Hand Side - LHS of the rule (or the precedent) A_i is the fuzzyfied historical value at time $t - 1$ and the Right Hand Side - RHS of the rule (or the consequent) A_j the fuzzyfied value at time t. The $A_i \rightarrow A_j$ rule can be read as “IF $f(t - 1)$ is A_i THEN $f(t)$ is A_j”. The F dataset will generate $T - 1$ FLRs, as the fuzzyfication process of Song uses the maximum membership method.

[1] It should be noted that the nomenclature of FLR may be misunderstood. The word *relationship*, in the fuzzy sets research field, has a different meaning than that used by Song and Chissom. Fuzzy relationships are operations between fuzzy sets, e. g. projection and cylindrical extension, well discussed in Klir and Yuan [1995]. The intention of the authors was to nominate a temporal pattern between two fuzzy sets, a temporal succession relationship, not a logical fuzzy relationship. However this nomenclature is spread in the FTS literature and will be kept on this text.

Then, for each FLR $A_i \to A_j$ a matrix $R_t = A_i^T \times A_j = a_{ij}$ will be created, with dimensions $k \times k$ where $a_{ij} = \min\{\mu_{A_i}(t), \mu_{A_j}(t-1)\}$ for $i, j = 1, ..., k$ and $t = 1, ..., T$. This matrix contains the fuzzy membership of the FLR for all fuzzy sets. The Operation Matrix $R(t, t-1)$ is computed as the union of all relationship matrices R_t, such that $R(t, t-1) = \bigcup_{t=1}^{T} R_t$. The Operation Matrix contains the memberships of all FLR for all fuzzy sets.

The inference using Fuzzy Relational Matrices demands to find the membership of the relation $f(t-1) \to f(t)$ on $R(t, t-1)$, such that $f(t) = f(t-1) \circ R(t, t-1)$, where $\circ$ is the Max-Min fuzzy relational operator. The operation $f(t-1) \circ R(t, t-1) = \max_j\{\min_i\{\mu_{A_i}(f(t-1)), r_{ij}\}\}$ for $i, j = 1, ..., k$ and $r_{ij} \in R(t, t-1)$ produces a vector with the memberships of $f(t)$ for all A_j fuzzy sets.

Several other studies use this heuristic to extract Fuzzy Relational Matrices, for instance Song et al. [1997], Jeng-Ren Hwang et al. [1998], Song [1999], Chen and Hwang [2000], Chen and Chung [2006], Cheng et al. [2008b], Jilani and Burney [2008], Davari et al. [2009], Qiu et al. [2011], Cheng and Li [2012], Qiu et al. [2013] and Chuang et al. [2014].

2.4.3 Rule Models

A great improvement was given by Chen [1996] who proposes a simplification of Song and Chissom's method by creating the Fuzzy Logical Rule Groups (FLRG), making the forecasting process cheaper by avoiding the use of matrix manipulations. The FLRG represent the knowledge base (rule base) of the model and are human readable and easy to interpret.

Create the Fuzzy Logical Relationship Group - FLRG with the form $LHS \to RHS$, where all FLR's with the same LHS are grouped and the RHS is the set of possible fuzzy sets that can follow the LHS set. The $LHS \to RHS$ pattern can be read as "IF $F(t-1) = LHS$ THEN $\exists A_j\ \forall A_j \in RHS \mid F(t) = A_j$" . An example of rule set is demonstrated on (2.4), given $k = 6$.

$$\begin{aligned}
&A_0 \to A_1\\
&A_1 \to A_1, A_2\\
&A_2 \to A_4\\
&A_3 \to A_2, A_3, A_5\\
&A_4 \to A_3, A_4\\
&A_5 \to A_4
\end{aligned} \tag{2.4}$$

The inference using [Chen, 1996], produces a forecast for the one step ahead value $f(t+1)$, given a past lag $f(t) = A_i$. A search is performed on $\mathcal{M}$ to find the FLRG where the $LHS = A_i$. The RHS of the FLRG will average all the possible fuzzy sets that follow A_i when $f(t) = A_i$, i. e., the forecast $f(t+1)$ is the RHS set of the selected FLRG.

The Chen's FLRG models allowed a compact and human readable representation of the time series behavior using fuzzy rules, which could in principle be used by business experts and researchers in knowledge extraction, for instance Lee et al. [2006]. But there is also another good reason to prefer the Chen's model over the Song and Chissom, the performance. The relation matrix dimension grows as the number of UoD partitions grows and the curse of dimensionality tends to impact negatively on the computational time spent on forecasting large datasets.

Several other works use this heuristic to extract rule models, for instance Chen [2002], Huarng and Yu [2004], Lee et al. [2006], Li et al. [2008], Hsu et al. [2010], Bahrepour et al. [2011], Huang et al. [2011], Sun et al. [2015], Sadaei et al. [2016b], Lee et al. [2017], Yang et al. [2017a], Bose and Mali [2017], Carvalho Jr and Costa Jr [2017]. Other heuristics are also present in the literature, such as the use of the APriori algorithm in Cheng and Chen [2018].

2.4.4 Weighted Rule Models

The generation of FLRG from the fuzzyfied data in FTS model has, at least, two drawbacks: the losing of rule's recurrence and their chronological order. Thus at the forecasting process a very recurrent pattern of data has the same importance of a unique occurrence pattern. Moreover, newer and older patterns also have the same weight in the forecast.

To fix these drawbacks Yu [2005] proposed the *Weighted Fuzzy Time Series* (WFTS) model by including weights on FLRG's. These weights are monotonically increasing and have a smoothing effect, giving more importance to the most recent data in forecasting process. The *Weighted Fuzzy Logical Relationship Group* - WFLRG has the same structure as the FLRG but weights w_j are associated with each fuzzy set $A_j \in RHS$.

The works of Ismail and Efendi [2011] and Efendi et al. [2013] have presented the *Improved Weighted Fuzzy Time Series* (IWFTS) model and changed the way in which the weights are assigned to the RHS rules on Yu's model. The main difference is that the weights are calculated by the recurrence of each rule, discarding the chronological order. The *Exponentially Weighted Fuzzy Time Series* (EWFTS) method, proposed by Sadaei et al. [2014] and Talarposhti et al. [2016], replaces the linear weight growth of WFTS model by an exponential growth.

Lee et al. [2013b] proposed a broad generalization of the weighted methods with the *Polynomial Fuzzy Time Series*- PFTS. This method demands the coefficient fitting by optimization techniques but is capable of approximating WFTS, IWFTS and EWFTS methods.

Cheng et al. [2008b] and Cheng et al. [2009] proposed the *Trend Weighted Fuzzy Time Series* - TWFTS which separates the FLRG's in three trends - no change, up trend and down trend - and assigns a weight to them according to the recurrence of the trend on the FLRG. Another contribution of these works is the Adaptive Expectation step, after defuzzyfication the forecast value a transformation is employed such as $Adaptative\_Forecast(t) = F(t-1) + h \cdot [F(t) - F(t-1)]$, where $F(t)$ is the forecasted value, $F(t-1)$ is the true past value and h is weight parameter that smooth the transition between the actual value and the forecasted value. Table 3 presents a summary of the weighting methods in FTS.

Method	**Weights**
WFTS	$\frac{1}{\sum_{i=1}^{n} i}\ ,\ \frac{2}{\sum_{i=1}^{n} i}\ ,\ \cdots\ ,\ \frac{n}{\sum_{i=1}^{n} i}$
IWFTS	$\frac{f_1}{\sum_{i=1}^{n} f_i}\ ,\ \frac{f_2}{\sum_{i=1}^{n} f_i}\ ,\ \cdots\ ,\ \frac{f_n}{\sum_{i=1}^{n} f_i}$
EWFTS	$\frac{c^0}{\sum_{i=1}^{n} c^i}\ ,\ \frac{c^1}{\sum_{i=1}^{n} c^i}\ ,\ \cdots\ ,\ \frac{c_{n-1}}{\sum_{i=1}^{n} c^i}$
TWFTS	$\frac{f_1}{\sum_{i=1}^{n} f_i}\ ,\ \frac{f_2}{\sum_{i=1}^{n} f_i}\ ,\ \cdots\ ,\ \frac{f_n}{\sum_{i=1}^{n} f_i}$

Table 3 – Weighting schemes for Fuzzy Time Series

2.4.5 Neural Networks Models

Neural Networks are black-box methods known to be the state-of-the-art in several pattern recognition domains. Its ability to deal with high dimensional and complex domains makes it attractive for many FTS models, specially the ones that deal with many variables and time simultaneously, as the case of Egrioglu et al. [2009].

Simpler univariate methods can be found on Yolcu and Lam [2017] which used a single multiplicative neuron whose inputs are the fuzzyfied values of several time lags. Bas et al. [2015] and Bas et al. [2018] used a Pi-Sigma Network, a variation of the well known ANFIS network, trained with Particle Swarm Algorithm.

Another hybrid FTS architecture is proposed by Bas et al. [2015], the *Fuzzy Time Series Network* - FTS-N which proposes a new topology for high-order FTS with a network layout, somehow similar to an ANFIS network. The partitioning of UoD and the fuzzyfication of the data use FCM clustering and the overall network is trained with PSO, combined yet with an autoregressive layer.

More recently, the new Deep Learning models begin to interact with the FTS field. Starting with Tran et al. [2018], which proposed a method that uses Long-Short Term Memory networks as knowledge model, trained with Backpropagation Through The Time algorithm. Sadaei et al. [2019] proposed the Image FTS, where the fuzzyfied data of several past lags are stacked to compose a binary image, which in turn is processed by a Convolutional Neural Network model trained by the backpropagtion method.

Fuzzy Cognitive Maps (FCM), developed by Kosko [1986], is a different kind of neural architecture inspired in the Mind Map tools, which is simpler than the Multilayered Neural Networks but also very powerful to represent nonlinear and causal behaviors. FCM are used as backend for FTS on Homenda et al. [2014], Homenda and Jastrzebska [2017], Yang and Liu [2018].

2.4.6 Metaheuristics

It was already seen in Section 2.2 that metaheuristics are widely used to determine the best partitioning scheme. However metaheuristics also can be used to extract or optimize the knowledge model representation from the fuzzyfied data.

The already cited Kuo et al. [2009] also use Particle Swarm Optimization (PSO) to build optimal rule sets on the *Hybrid Particle Swarm FTS* - HPSO-FTS. The PSO metaheuristic is also used in other works to train neural models, as in Bas et al. [2015], Yolcu and Lam [2017], Bas et al. [2018]. Genetic Algorithms to learn a matrix of weighted rules are employed in Ye et al. [2016].

The optimization of Ω and L are the focus of Enayatifar et al. [2013], which proposes the *Refined High-order Weighted FTS with Imperialist Competitive Algorithm* - RHWFTS–ICA, using evolutionary computing to optimize the number of lags for the high order seasonal FTS and the weights for adaptive expectation. The Adaptive Sine-Cosine Human Learning Optimization (ASCHLO) was used in Yang et al. [2018] for rule and weight induction.

2.4.7 Hybrid Approaches

Autoregressive and polynomial models were adopted in Chang [1997], Tseng et al. [1999], Askari and Montazerin [2015], Talarposhti et al. [2016]. These methods used classic optimization approaches to fit regression coefficients mixed with fuzzy terms.

Sadaei et al. [2016a] propose the ARFI–FTS, a hybrid approach that combines statistical method ARFIMA with FTS for forecasting of long-memory time series. Also Bas et al. [2015] contains a hybrid approach, combining its network model with an autoregressive layer.

2.5 The Deffuzyfication Process

The result of the inference is a set of $f(t+1)$ possibilities, or rules involving it, to be converted in a crisp numerical value $\hat{y}(t+1)$ that estimates the unknown value of

$y(t+1)$. The deffuzyfication method aims to deliver a $\hat{y}(t+1) \in U$ that meets the expected value, or the expected mean of the several patterns contained in $f(t+1)$ forecast.

In Song and Chissom [1993b], the defuzzyfication process converts the membership vector $f(t+1)$ into a scalar value on the universe of discourse. Taken the maximum membership values of $f(t)$, with the following method:

1. If there is only one maximum, $\hat{y}(t+1)$ will be the midpoint of the maximum membership fuzzy set;
2. If there are more than one consecutive maxima, $\hat{y}(t+1)$ will be the mean of the midpoints;
3. Otherwise, $\hat{y}(t+1)$ will be the weighted mean of the fuzzy sets midpoints with the memberships, such that $\hat{y}(t+1) = \sum_{j \in f(t)} \mu_j \cdot c_j$, where μ_j is the membership degree and c_j is the midpoint of the fuzzy set $A_j \in \tilde{A}$

In the method of Chen [1996], the deffuzyfication is adapted to the following steps, given the $f(t+1) = RHS$ of the selected FLRG:

1. If the RHS contains only one fuzzy set, $\hat{y}(t+1)$ will be the midpoint of the set;
2. If the RHS contains more than one fuzzy set, $\hat{y}(t+1)$ will be the mean of the midpoints of these sets.

The above methods are considered the Simple Mean methods. For weighted rule models as Sadaei et al. [2014], for each rule i, the expected mean point $\mathbb{E}_i$ of the rule is the weighted mean of the midpoints mp_{A_j} of their RHS consequents by the weights w_{ij}.

$$\mathbb{E}_i = \sum_{A_j \in RHS}^{k} w_{ij} \cdot mp_{A_j} \tag{2.5}$$

When more than one pattern (in the case of rules) was found in the inference step the expected values of each pattern must be mixed. The simplest way is performing a Simple Mean, where i is each active rule of the model $\mathcal{M}$, $\mathbb{E}_i$ is the expected mean point of each rule i and $|\mathcal{M}|$ is the number of active rules in model $\mathcal{M}$.

$$\hat{y}(t+1) = |\mathcal{M}|^{-1} \sum_{i \,\in\, \mathcal{M}} \mathbb{E}_i \tag{2.6}$$

The drawback of this method is to give the same importance for all patterns. In the Weighted Sum each pattern is weighted by its activation, where i is each active rule of the model $\mathcal{M}$, μ_i is the fuzzy membership of the rule (or its activation) and $\mathbb{E}_i$ is the expected mean point of each rule i.

$$\hat{y}(t+1) = \frac{\sum_{i \,\in\, \mathcal{M}} \mu_i \cdot \mathbb{E}_i}{\sum_{i \,\in\, \mathcal{M}} \mu_i} \tag{2.7}$$

The output of the deffuzyfication is the crisp number $\hat{y}(t+1)$, which can be yet post-processed by some data transformation. In the next section the post-processing data transformations will be discussed.

2.6 Data Transformations for Pre and Post Processing

Data transformations have several functions, such as changing the original U of Y, removing noise, de-trending, de-seasonalizing, normalizing or standardizing data, etc. Some of these operations transform multivariate data in monovariate (as the Fuzzy Information Granules - FIG) and others decompose a monovariate time series in several sub-signals (as the Empirical Mode Decomposition - EMD), transformations that will be studied in Chapter 6.

The most common transformation is the differentiation, defined as $\Delta y(t) = y(t-1) - y(t)$, and the inverse operation as $y(t) = y(t-1) + \Delta y(t)$. This operation changes the original U for a smaller and stationary space. This is relevant because all methods presented before are time invariant models, which assume that Y is stationary. Indeed, this is, according to Duru and Yoshida [2012], one of the greatest weakness of the FTS methods. The differentiation can be used to make Y stationary and can be employed as pre and post processing of almost all FTS methods, being in some cases explicitly part of the model, as in Cheng et al. [2011], Lee and Javedani [2011], Sadaei et al. [2016b]. In these cases the FTS model aims to forecast the change magnitude $\Delta y(t)$ instead of the time series level $y(t)$.

Not only the differentiation is used to transform U in a smaller interval and Y in a stationary time series. Box-Cox power transformations are employed in [Lee et al., 2013a], ROI in Sadaei and Lee [2014], Moyse and Lesot [2016] and normalization in Tran et al. [2018]. Other pre-processing transformations can help to improve overall FTS, as moving averages and exponential smoothing, but it was not commonly seen in the literature.

There are transformations only for post-processing as the Adpative Expectation, defined as $AE(t+1) = y(t) + h \cdot (\hat{y}(t+1) - y(t))$, where h is a weight that balances the impact of $\hat{y}(t+1)$ in the last known value $y(t)$. The Adpative Expectation is a conservative weighted persistence model, where the predicted value $\hat{y}(t+1)$ is used only to change last known value. This method is employed in Cheng et al. [2008b], Huang et al. [2011], Enayatifar et al. [2013], Sadaei et al. [2014], Singh [2015], Sadaei et al. [2016b], Ye et al. [2016], Yang et al. [2017a], Bose and Mali [2017].

2.7 A Conventional High Order Fuzzy Time Series Method - HOFTS

The main focus of this research is the rule-based Fuzzy Time Series, descendants of Chen [1996] method, that largely dominate the field, as can be seen in Table 8. Beyond its first-order original work, many extensions were proposed that modified several aspects of the method, changing the order, partitioning method, fuzzyfication, defuzzyfication, introducing transformations, etc.

This section proposes a consensus conventional FTS method, that aggregates the most common properties of the rule based methods in the literature. This method embodied all explored hyperparameters but their definition involves more complex optimization methods, which will be explored in Chapter 5. Indeed some default values are defined for hyperpameters, as shown in Table 4, but they can be overridden by the user. However, the two most impacting hyperparameters still must be determined by the user: k and Ω.

Parameter	Default Value
Ω	User defined
k	User defined
Π	Grid
μ	triangular
α-cut	0
L	$\{1,\ldots,\Omega\}$

Table 4 – HOFTS and WHOFTS hyperparameter default values

In Section 2.7.1 the training procedure follows the same steps shown in Figure 3 to produce a rule based knowledge model $\mathcal{M}$ with the linguistic variable $\tilde{A}$. These parameters are used by the forecasting procedure presented in Section 2.7.2, which follow the same steps presented in Figure 4, to produce point forecastings $\hat{y}(t+1)$. An extension is presented in Section 2.7.3, where the training and forecasting procedures are modified to incorporate weights on rules that aims to improve the performance by giving more importance to the more frequent fuzzy sets.

2.7.1 Training Procedure

The methods below take as input a training sample Y, the number of partitions k, the number of lags Ω (and the other default values presented in Table 4) and outputs the model $\mathcal{M}$.

Step 1 *Partitioning*:

a) *Defining U*: The UoD defines the sample space, i.e., the known bounds of time series Y, such that $U = [\min(Y) - D_1, \max(Y) + D_2]$, where $D_1 = \min(Y) \times 0.2$

and $D_2 = \max(Y) \times 0.2$ are used to extrapolate the known bounds as a safe margin.

b) *UoD Partitioning*: Split U in k intervals U_i with midpoints c_i, by invocation of the partitioning method Π, which will define the lengths of all intervals;

c) *Define the linguistic variable* $\tilde{A}$: For each interval U_i create an overlapping fuzzy set A_i, with the membership function μ_{A_i}. The midpoint of the fuzzy set A_i is going to be c_i, the lower bound $l_i = c_{i-1}$ and the upper bound $u_i = c_{i+1}$ $\forall\ i > 0$ and $i < k$, and $l_0 = \min U$, $l_k = \max U$. Each fuzzy set $A_i \in \tilde{A}$ is a linguistic term of the linguistic variable $\tilde{A}$;

Step 2 *Fuzzyfication*:

Transform the original numeric time series Y into a fuzzy time series F, where each data point $f(t) \in F$ is an $1 \times k$ array with the fuzzyfied values of $y(t) \in Y$ with respect to the linguistic terms $A_i \in \tilde{A}$, where the fuzzy membership is greater than the predefined α-cut, i.e., $f(t) = \{A_i \mid \mu_{A_i}(y(t)) \geq \alpha\ \forall A_i \in \tilde{A}\}$;

Step 3 *Rule Induction*:

a) *Generate the high order temporal patterns*: The fuzzy temporal patterns have format $A_{i0}, ..., A_{i\Omega} \rightarrow A_j$, where the precedent (or Left Hand Side) is $f(t - L(\Omega)) = A_{i0}$, $f(t - L(\Omega - 1)) = A_{i1}$, ..., $f(t - L(0)) = A_{i\Omega}$, and the consequent (or RHS) is $f(t + 1) = A_j$.

b) *Generate the rule base* $\mathcal{M}$: Select all temporal patterns with the same precedent and group their consequent sets creating a rule with the format $A_{i0}, ..., A_{i\Omega} \rightarrow A_k, A_j, ...$, where the LHS is $f(t - L(\Omega)) = A_{i0}$, $f(t - L(\Omega - 1)) = A_{i1}$, ..., $f(t - L(0)) = A_{i\Omega}$ and the RHS is $f(t + 1) \in \{A_k, A_j, ...\}$. Each rule can be understood as the weighted set of possibilities which may happen on time $t + 1$ (the consequent) when a certain precedent $A_{i0}, ..., A_{i\Omega}$ is identified in previous L lags (the precedent).

2.7.2 Forecasting Procedure

The method below take as input a test sample Y, the model $\mathcal{M}$ and the forecasting horizon H (whose default value is 1) to output a crisp point forecasts $\hat{y}_H$.

Step 1 *Fuzzyfication*: Compute the membership grade μ_{ti} for each $y(t) \in Y$ where $t \in L$ and each fuzzy set $A_i \in \tilde{A}$, such that $\mu_{ti} = \mu_{A_i}(y(t))$.

Step 2 *Rule matching*: Select the K rules where all fuzzy sets A_i on the LHS have $\mu_{ti} > \alpha$; The rule fuzzy membership grade is shown below, using the minimum function as

T-norm.

$$\mu_j = \bigcap_{t \in L\ i \in \tilde{A}} \mu_{ti} \tag{2.8}$$

Step 3 *Defuzzyfication*:

a) *Rule mean points*: For each selected rule j, compute the mean point mp_j as below, where c_i is the c parameter of the μ function from fuzzy set A_i:

$$mp_j = |RHS|^{-1} \sum_{i \in RHS} c_i \tag{2.9}$$

b) *Defuzzyfication*: Compute the forecast as the weighted sum of the rule mid-points mp_j by their membership grades μ_j for each selected rule j:

$$\hat{y}(t+1) = \frac{\sum_{j \in K} \mu_j \cdot mp_j}{\sum_{j \in K} \mu_j} \tag{2.10}$$

Step 4 *Many steps ahead forecast*:If the forecasting horizon is $H > 1$, define $\hat{y}_H = \{\hat{y}(t+1)\}$ as the set of forecasts and repeat the steps below for each $h = 2..H$, otherwise return $\hat{y}(t+1)$.

a) Call recursively the forecasting method using $\hat{y}(t+h-1)$ as input to produce $\hat{y}(t+h)$;

b) Append $\hat{y}(t+h)$ to $\hat{y}_H$ and if $h = H$ then return $\hat{y}_H$.

2.7.3 The Weighted Extension - WHOFTS

As pointed in Section 2.4.4, a common drawback of rule-based models is that all fuzzy sets in the RHS of the rules have the same importance. To fix this it is common to add weights to the RHS fuzzy sets which indicate its relevance on deffuzyfication phase. To extend the HOFTS method to a weighted version it is needed to change the Step 3.b of the training procedure presented in Section 2.7.1 to the below:

Step 3.b) *Generate the rule base*: Select all temporal patterns with the same precedent and group their consequent sets creating a rule with the format $A_{i0}, ..., A_{i\Omega} \rightarrow w_k \cdot A_k, w_j \cdot A_j, ...$, where the LHS is $f(t - L(\Omega)) = A_{i0}$, $f(t - L(\Omega - 1)) = A_{i1}$, ..., $f(t - L(0)) = A_{i\Omega}$ and the RHS is $f(t+1) \in \{A_k, A_j, ...\}$ and the weights $w_j, w_k, ...$ are the normalized frequencies of each temporal pattern such that:

$$w_i = \frac{\#A_i}{\#RHS}\ \forall A_i \in RHS \tag{2.11}$$

where $\#A_i$ is the number of occurrences of A_i on temporal patterns with the same precedent LHS and $\#RHS$ is the total number of temporal patterns with the same precedent LHS. Each rule can be understood as the weighted set of possibilities which may happen on time $t+1$ (the consequent) when a certain precedent $A_{i0}, ..., A_{i\Omega}$ is identified on previous L lags (the precedent).

Naturally the weights w_i will fit the condition $\sum_{i=1}^{k} w_i = 1$. These weights are exploited in forecasting procedure presented in Section 2.7.2, which also need to be changed in the Step 3.a by the method below:

Step 3.a) *Rule mean points*: For each selected rule j, compute the mean point mp_j as below, where c_i is the c parameter of the μ function from fuzzy set A_i:

$$mp_j = \sum_{i \in RHS} w_i \cdot c_i \tag{2.12}$$

In the next sections these methods will be evaluated in relation to their main parameters

2.8 Computational Experiments

In this section experiments were performed in order to evaluate the impact of the two main hyperparameters - Ω and k - over the methods HOFTS and WHOFTS. As these methods generalize a wide spectrum of proposed methods in the literature, specially the rule based ones which are focus of this work, the computational experiments illustrate the general performance of FTS methods.

First, in Section 2.8.1 common point forecasting measures and statistical tests are discussed. In Section 2.8.2 the results of a Grid Search optimization of the hyperparamters are presented and in Section 2.8.3 a residual analysis of the best models is employed.

In order to contribute with the replication of all the results in the research, all data and source codes employed in this chapter are available at the URL: `http://bit.ly/scalable_probabilistic_fts_chap2`

2.8.1 Evaluation Measures for Point Forecasts

The accuracy metrics usually employed to evaluate point forecasting models are the Symmetrical Mean Average Percent Error (SMAPE), described in Equation (2.13), Root Mean Squared Error (RMSE), described in Equation (2.14) and Theil's U Statistic, described in Equation (2.15), where y means the real data and $\hat{y}$ the forecasted values. The U Statistic measures how much the forecaster is better than the Naïve method, with

$U = 1$ meaning both methods are equal, $U > 1$ the proposed method is worse than Naïve and $U < 1$ is better.

$$SMAPE = \frac{1}{T}\sum_{t=1}^{T}\frac{|y(t) - \hat{y}(t)|}{|\hat{y}(t)| + |y(t)|} \tag{2.13}$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y(t) - \hat{y}(t))^2} \tag{2.14}$$

$$U = \sqrt{\frac{\sum_{t=1}^{T-1}\left(\frac{\hat{y}(t+1)-y(t-1)}{y(t)}\right)^2}{\sum_{t=1}^{T-1}\left(\frac{y(t+1)-y(t-1)}{y(t)}\right)^2}} \tag{2.15}$$

It is also practice to perform a residual analysis in order to check the white-noise assumption, such that $\epsilon \sim \mathcal{N}(0, 1)$. Some statistical tests were proposed in the literature to assert this condition, as the Box - Pierce Test, proposed by Box and Pierce [1970], and its improved version, the Ljung - Box Test, found in Ljung and Box [1978].

The Ljung-Box Test checks, for each lag of the autocorrelation function of the residuals, the hypotheses H_0 - the residuals are i.i.d (independent and identically distributed) and H_1 - the residuals are not i.i.d. To reject H_0, the test statistic Q must satisfy the condition $Q > \chi^2_{1-\alpha,df}$ where α is the confidence level and df is the number of the lag.

The next section discusses the results for one or H-steps ahead that try to quantify the uncertainties of point forecasts.

2.8.2 Hyperparameter Grid Search

To assess the impact of the order and the number of partitions on HOFTS and WHOFTS methods a Grid Search was employed, using the search spaces presented in Table 5. The results can be seen in Figure 7 which details the sensitivity of the methods to the hyperparameters. A sample of these responses can also be seen in Figures 8 and 9. The non-stationary behavior of benchmark datasets make them predictable accurately just for very short terms, and in the previous figures the considered forecasting horizon is $H = 1$.

The number of partitions and order have different effects in HOFTS and WHOFTS, where WHOFTS performs better in general. For $k > 65$ and $\Omega > 1$, HOFTS and WHOFTS have similar performances.

When considering the combinations of number of partitions and orders, the results show that $k = 35$ and $\Omega = 1$ are the best combination of hyperparameters, mixing good RMSE accuracy with a parsimonious model. The results also shown that when the partitioning increases to $k \geq 65$ and $\Omega \geq 2$ the models overfit, and below $k \leq 25$ the models underfit.

Hyperparameter	Search space
k	$\{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95\}$
Ω	$\{1, 2, 3\}$

Table 5 – Hyperparameter search spaces for HOFTS and WHOFTS grid search

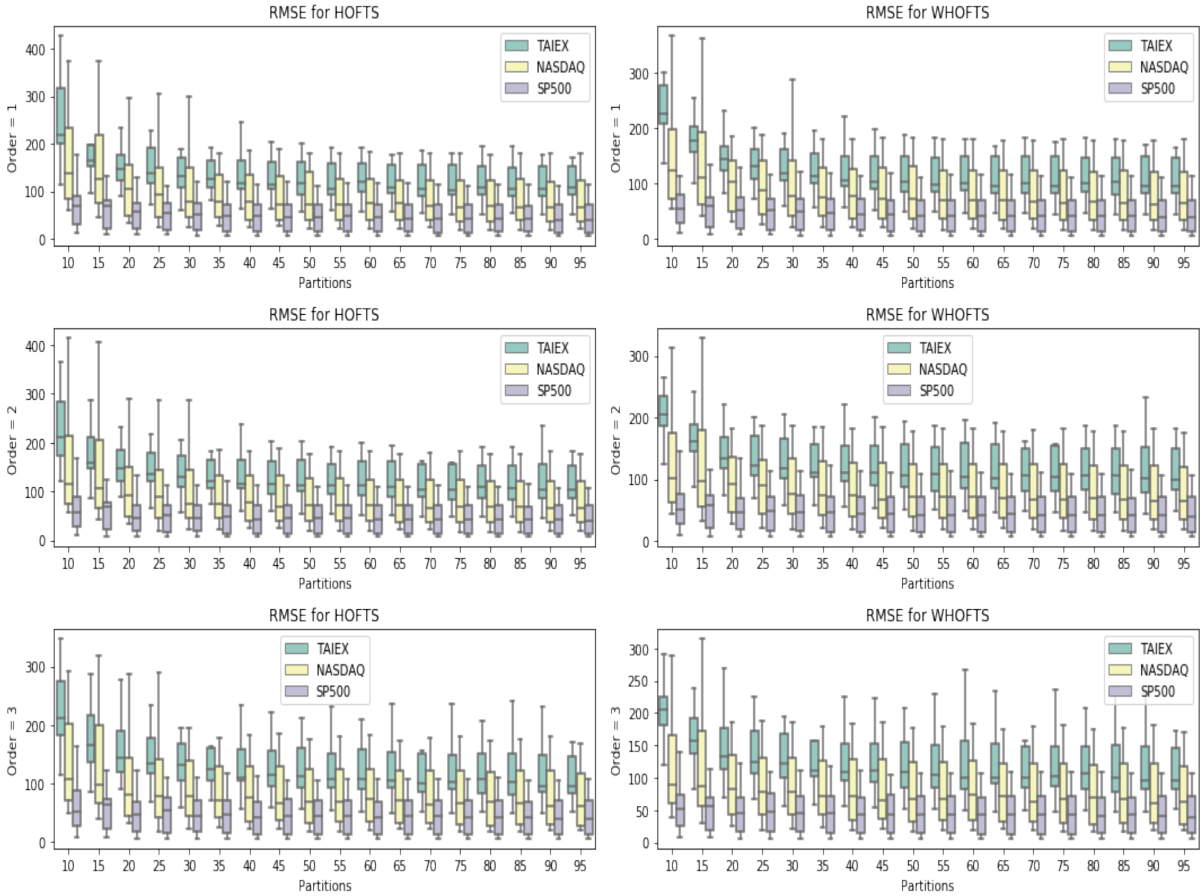


Figure 7 – HOFTS and WHOFTS grid search over hyperparameters k and Ω

2.8.3 Residual Analysis

The residuals of the models are presented in Figures 10 and 11 and the Ljung-Box tests for the 3 first lags are presented in Tables 6 and 7, which show the good fit of model. However, high correlated residuals were detected in some non-stationary sub samples of the datasets, what was also expected since the models are time-invariant. Best performance is expected for time-variant models, capable to adjust its behavior due to changes in data.

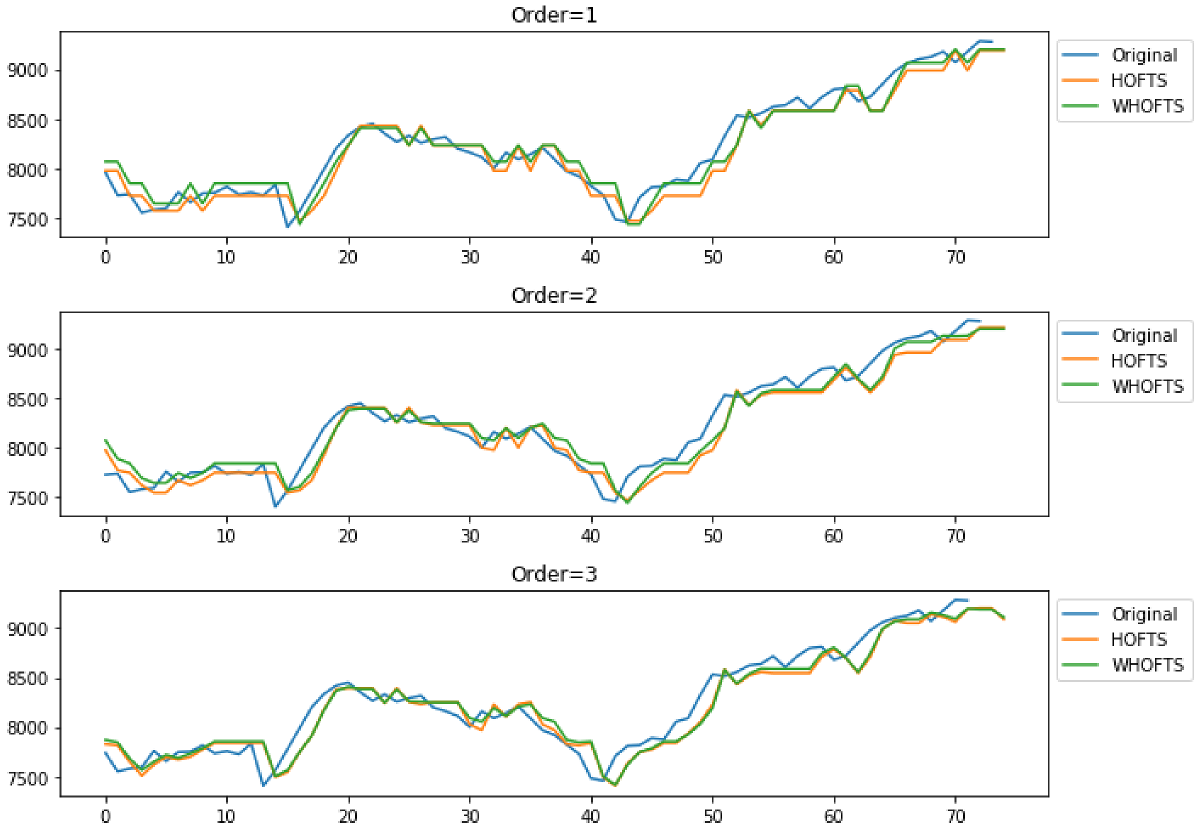


Figure 8 – The impact of order in forecasting

Lag	Statistic	p-Value	Critical Value	Result
1	341.295085	0.0	3.841459	H0 accepted
2	412.500903	0.0	5.991465	H0 accepted
3	441.435962	0.0	7.814728	H0 accepted

Table 6 – Ljung-Box Test for HOFTS residuals

Lag	Statistic	p-Value	Critical Value	Result
1	336.838522	0.0	3.841459	H0 accepted
2	394.472179	0.0	5.991465	H0 accepted
3	411.754344	0.0	7.814728	H0 accepted

Table 7 – Ljung-Box Test for WHOFTS residuals

2.9 Conclusion

This chapter provided a brief overview related to the Fuzzy Time Series models. A literature review and some state-of-the-art works related to FTS were presented and summarized in Table 8.

The presented methods have some common drawbacks. The matrix-based methods have scalability issues, suffering from the curse of dimensionality. With the rule-based methods, in the forecasting step, just one rule is chosen for computing the result, based on

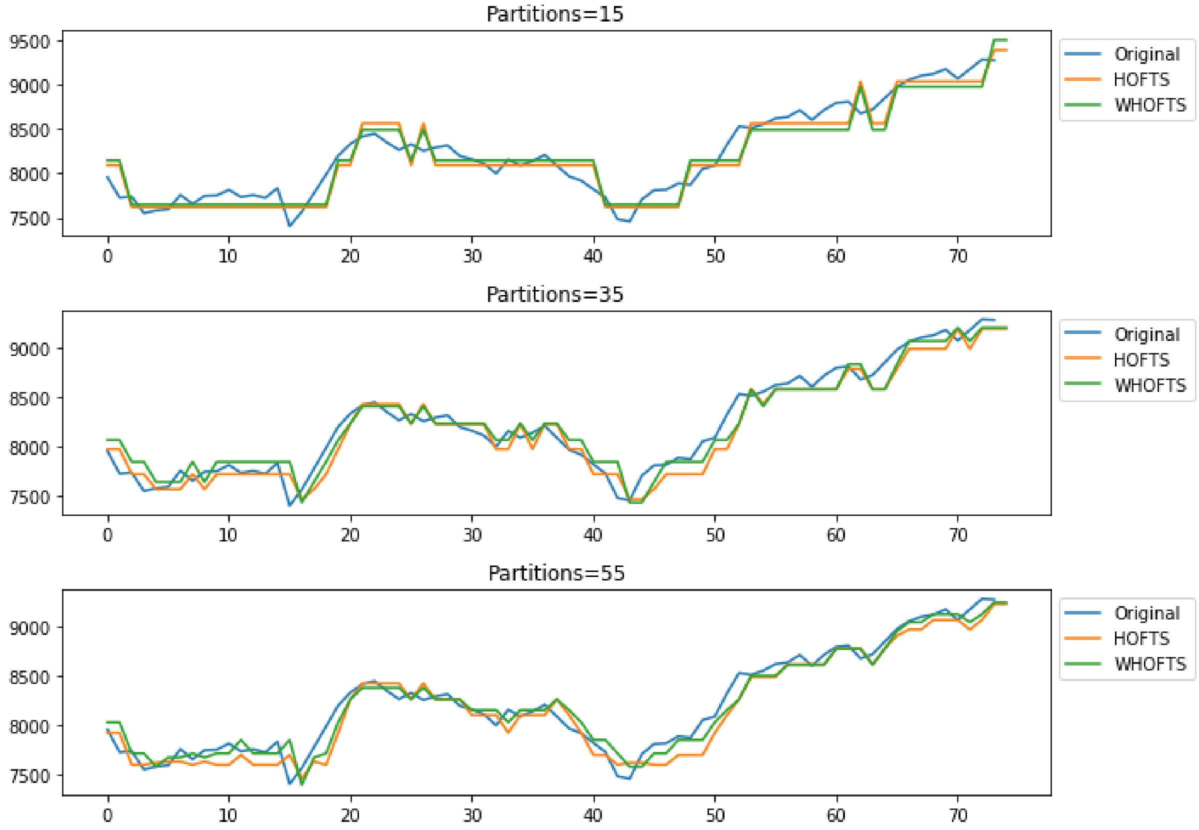


Figure 9 – The impact of partitioning in forecasting

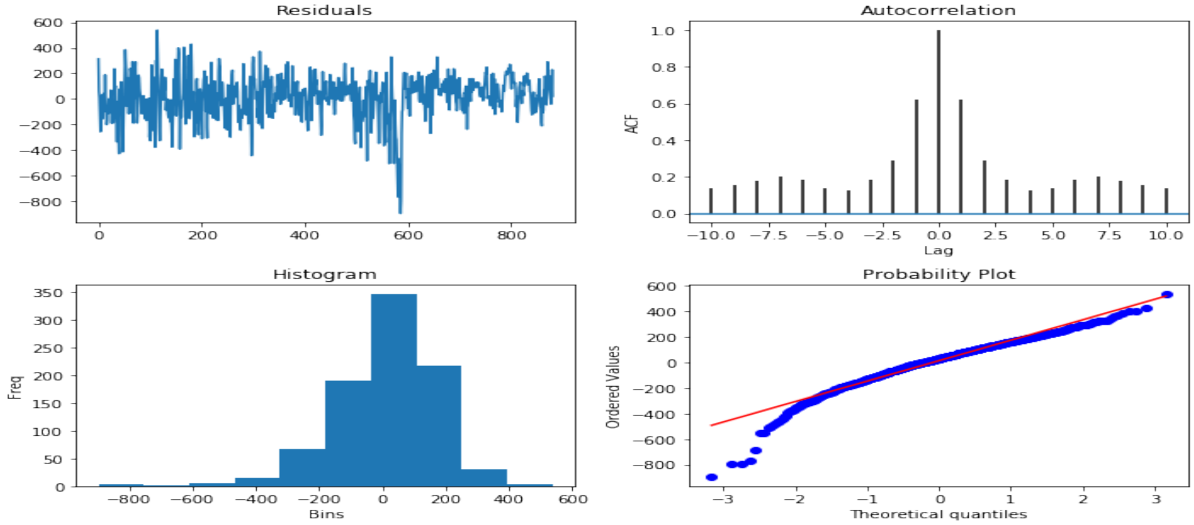


Figure 10 – HOFTS residuals

the maximum membership between the input value and all the rules. This causes the loss of “smoothing” effect of fuzzy methods, which demands mixing many sets according to their fuzzy membership values. Lastly, these models are point-based forecasters and give no uncertainty measures about their results. Otherwise, black-box knowledge models eliminate the readability and auditability of the model, and in some cases are not parsimonious.

To enforce the focus of this research on rule-based FTS methods, the High Order

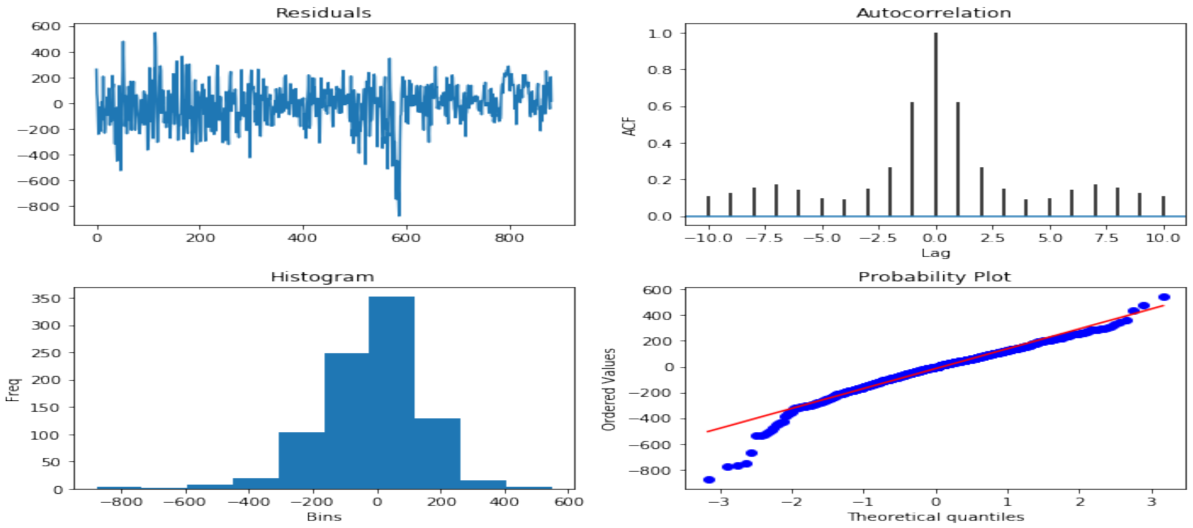


Figure 11 – WHOFTS residuals

FTS (HOFTS) and the Weighted High Order FTS (WHOFTS) methods were developed, following a consensus construction from the several approaches present in literature. Computational experiments were employed to assess the point forecasting performance of the methods using financial datasets.

It is necessary to highlight the absence of probabilistic forecasting methods in the Fuzzy Time Series literature. These methods will be discussed in next chapter, where their main features are pointed out and a new method for interval-forecasting with FTS is proposed.

Reference	n	Π	μ	**Fuzz.**	Ω	**Ind.**	$\mathcal{M}$	**Transf.**	**Defuzz.**	**Application**
Song and Chissom [1993b]	1	G	Tri	Max	1	H	M	-	SM	Enrollments
Chen [1996]	1	G	Tri	Max	1	H	R	-	SM	Enrollments
Chang [1997]	1	H	Tri	All	S	O	P	-	-	Sales
Song et al. [1997]	1	G	Tri	Max	1	H	M	-	-	-
Tseng et al. [1999]	1	G	Tri	Max	S	O	P	-	-	Industrial Production
Song [1999]	1	G	Tri	Max	S	H	M	-	SM	-
Chen and Hwang [2000]	2	G	Tri	Max	1	H	M	-	WS	Temperature
Huarng [2001]	1	H	Tri	Max	1	H	R	-	SM	Stock Price
Chen [2002]	1	G	Tri	Max	5	H	R	-	SM	Enrollments
Huarng and Yu [2004]	1	H	Tri	Max	1	H	R	-	SM	Stock Price
Yu [2005]	1	G	Tri	Max	1	H	WR	-	WS	Stock Price
Huarng and Yu [2006]	1	G	Tri	Max	1	BP	NN	-	SM	Stock Price
Chen and Chung [2006]	1	MH	Tri	Max	3	H	M	-	SM	Enrollments
Lee et al. [2006]	4	G	-	Max	1	H	R	-	SM	Stock Price
Cheng et al. [2006]	1	E	Trap	Max	1	H	M	-	SM	Project Cost
Cheng et al. [2008b]	1	E	Trap	Max	1	H	M	-	WS	Outpatient visits
Cheng et al. [2008a]	1	G	-	-	1	H	WR	D, AE	WS	Stock Price
Jilani and Burney [2008]	6	G	Tri	Max	4	H	M	-	WS	-
Li et al. [2008]	2	C	FCM	All	2	H	R	-	SM	Temperature

Reference	n	Π	μ	**Fuzz.**	Ω	**Ind.**	$\mathcal{M}$	**Transf.**	**Defuzz.**	**Application**
Davari et al. [2009]	2	MH	Tri	Max	1	H	M	-	WS	Enrollments
Kuo et al. [2009]	1	MH	-	-	?	MH	WR	-	?	Enrollments
Egrioglu et al. [2009]	2	G	Trap	All	3	BP	NN	-	SM	Accident
Hsu et al. [2010]	2	MH	-	Max	1	H	R	-	SM	Temperature
Chen and Chen [2011]	2	-	-	-	?	H	WR	D	-	Stock Price
Huang et al. [2011]	1	MH	Tri	Max	3	H	R	AE	WS	Enrollments
Lee and Javedani [2011]	1	G	-	-	S	H	WR	D	-	Stock Price
Ismail and Efendi [2011]	1	G	-	Max	1	H	WR	-	WS	Enrollments
Cheng and Li [2012]	1	-	-	-	?	H	WR	?	-	-
Enayatifar et al. [2013]	1	MH	-	-	3	MH	WR	AE	-	Energy Load
Lee et al. [2013a]	1	G	-	-	1	H	WR	BC	WS	Stock Price
Chen et al. [2014]	1	E	-	-	2	H	WR	?	-	Stock Price
Sadaei and Lee [2014]	1	-	-	Max	1	H	WR	ROI, AE	WS	Stock Price
Askari and Montazerin [2015]	3	C	FCM	All	1	H	P	-	WS	Stock Price
Bas et al. [2015]	1	C	FCM	All	1	MH	NN	-	WS	Stock Price
Cai et al. [2015]	1	MHs	trmf	All	1	H	WR	-	WS	Stock Price
Chen and Chen [2015b]	2	G	Tri	Max	2	H	R	-	WS	Stock Price
Ismail et al. [2015]	1	Q	-	-	1	H	M	-	SM	Energy Load
Sun et al. [2015]	3	C	FCM	All	1	H	R	-	WS	Stock Price

Reference	n	Π	μ	Fuzz.	Ω	Ind.	$\mathcal{M}$	Transf.	Defuzz.	Application
Singh [2015]	3	G	-	Max	1	BP	NN	AE	WS	Stock price
Rubio et al. [2016]	1	H	Trap	All	1	H	M	-	WS	Portfolio returns
Sadaei et al. [2016b]	1	G	Tri	Max	1	H	R	D, AE	SM	Stock Price
Talarposhti et al. [2016]	1	MH	Tri	Max	1	H	P	-	WS	Stock Price
Ye et al. [2016]	1	G	Tri	AC	3	MH	M	ROC, AE	WS	Sock price
Lee et al. [2017]	1	G	Tri	Max	3	H	R	-	WS	Enrollments
Yang et al. [2017a]	3	CS	Tri	Max	1	H	R	EMD, AE	SM	Wind Speed
Yolcu and Lam [2017]	1	C	FCM	All	1	MH	MLP	-	WS	Sock price
Bose and Mali [2017]	1	C	Tri	-	3	H	R	AE	WS	-
Carvalho Jr and Costa Jr [2017]	1	G	Tri	AC	1	H	R	-	SM	Stock price
Jiang et al. [2017]	1	MH	Tri	Max	3	H	WR	-	WS	Stock price
Saberi et al. [2017]	1	C	fefts	All	1	H	M	-	WS	-
Sadaei et al. [2017]	1	MH	Tri	Max	*	O	P	-	?	Energy Load
Severiano et al. [2017]	1	G	Tri	All	3	H	R	-	SM	Energy Load
Bas et al. [2018]	1	C	FCM	All	1	P	NN	-	WS	Stock price
Guney et al. [2018]	1	G	Tri	Max	2	H	MC	-	WS	-
Cheng and Chen [2018]	1	H	Trap	Max	3	A	R	-	WS	Stock price
Dincer and Akkuş [2018]	1	C	-	Max	1	H	M	-	SM	Air pollution
Yang et al. [2018]	5	MH	Trap	Max		MH	P	EMD	WS	Stock price

Reference	n	Π	μ	**Fuzz.**	Ω	**Ind.**	$\mathcal{M}$	**Transf.**	**Defuzz.**	**Application**
Yang et al. [2018]	5	C	-	-	2	RG	FCM	WV	WS	-
Tran et al. [2018]	3	C	-	All	1	BP	NN	N	WS	-
Zhang et al. [2018a]	4	H	FCM	-	1	BP	NN	-	WS	Stock price
Zhang et al. [2018b]	2	MH	Tri	Max	2	H	R	-	SM	Stock price
Chen et al. [2019]	1	G	Trap	Max	1	H	M	-	WS	Flood
Sadaei et al. [2019]	1	G	Tri	Max	*	BP	NN	-	WS	Energy Load

n - Number of variables

Π - Partitioning method: G - Grid, H - Heuristic, MH - Metaheuristic, CS - Chi-Square, E - Entropy, Q - Quartile

μ - Membership function: Tri - Triangular, Trap - Trapezoidal, FCM - Fuzzy C-Means

Fuzz - Fuzzyfication method: Max - Maximum membership, All - All memberships, AC - alpha-cut

Ω - Order

Ind. - Knowledge induction method: H - Heuristic, MH - Metaheuristic, BP - Backpropagation, O - Optimization, RG - Regression analysis, AP - Apriori

$\mathcal{M}$ - Knowledge model: M - Matrix, R - Rules, WR - Weighted Rules, NN - Neural Network, FCM - Fuzzy Cognitive Map, P - Polynomial, MC - Markov Chain

Transf - Transformations: A - Adaptive expectation; B - Box-Cox; D - Differentiation; R - ROI, N - Normalization

Table 8 – Summary of the most relevant FTS methods

Chapter 3

Probabilistic Forecasting

"Uncertainty is an uncomfortable position. But certainty is an absurd one."

— Voltaire

This chapter briefly discusses the uncertainty of point forecasts to introduce the interval and probabilistic forecasting and review the related literature, pointing the most representative methods of each type. To fill the gaps in the FTS literature, two new FTS methods are proposed for representing the fuzzy uncertainty, the Interval FTS and the Ensemble FTS.

Previously on this work it was presented the two main kinds of uncertainties , the epistemic and the ontological, one which impose limits to predictability of forecasting methods. The origin of these limitations, according to Krzysztofowicz [2001] are "theoretical, technological and budgetary". In his seminal work, Lorenz [1963] proved that even some deterministic systems decay to chaotic behavior due to minimal fluctuations on their start conditions. This effect puts limits on the predictability of these systems and the probabilistic forecasts are the ideal way to deal with these limitations.

Also Makridakis et al. [2010] poses that "statistical regularity does not equal predictability" on his study about the common sources of unreliability on forecasts. In Makridakis and Bakas [2016] the authors split the forecasting uncertainty in four categories: Known Knowns (normal and usual conditions), Unknown Knowns (uncertainty known but not covered by models), Known Unknowns (rare, unusual and special conditions) and Unknown Unknowns (unexpected and unpredictable conditions, also called by black swans). This survey also discuss the specific sources of uncertainty at several forecasting areas as natural (weather, earthquakes, volcanoes, tsunamis, floods) and social (economical and demographical) events.

In this context the probabilistic forecasting approaches emerged, defined by Gneiting and Katzfuss [2014] as "the form of a predictive probability distribution over future quantities or events of interest". This definition enclose two main forecasting types:

intervals and probability distributions. Their importance appears as we analyze the impact of the intrinsic uncertainty on the point forecasts, and how this uncertainty grows as the forecasting horizon increases.

In the following section a short review of uncertainties in forecasting models is presented, starting with the limitations of point forecasts in Section 3.1, and its accuracy measures. In Section 3.2, the most known interval forecasting methods and its evaluation measures are presented. In Section 3.3 new FTS methods are proposed to produce prediction intervals representing the fuzzy uncertainty. In Section 3.4 the major approaches for probabilistic forecasting and its accuracy measures are discussed. In Section 3.5 a new FTS method is proposed to produce interval and probabilistic forecastings. In Section 3.6, computational experiments are performed to assess the accuracy and computational performance of the proposed methods and finally, in Section 3.7, the conclusions are given.

3.1 The Point Forecast Limitations

Point forecasts, are usually defined by the conditional expectation $\mathbb{E}[y(t{+}1)|y(t), y(t-1), ...]$ which in turn minimizes a cost function that represents the accuracy error, as the mean squared error in Equation (2.14). In statistics textbooks, for instance Steven M. Kay [2006], this conditional expectation is known to be the best linear and non-linear estimator for $y(t+1)$ given the lagged values $y(t), y(t-1), ...$ But for the layman this optimality may be misunderstood as the absence of error and not the normality of error terms ϵ, defined as the white noise $\epsilon \sim \mathcal{N}(0,1)$. These deterministic forecasts, according to Krzysztofowicz [2001], create an “illusion of certainty in a user’s mind”.

It is expected that the conditional variance $Var[y(t+1)|y(t), y(t-1), ...]$ be presented with the conditional expectation to represent the uncertainty around this result, but this is not really usual as it needs to be. Even this common statistical approach is not enough to capture all uncertainty of an estimate and Makridakis and Taleb [2009] point out that error variance may not be known, constant or finite.

When dealing with the many steps ahead forecasts, where $H \in \mathbb{N}^+$ is the forecasting horizon, it is necessary to consider the propagation of errors. Leutbecher and Palmer [2008] address this problem in the context of weather forecasting, where the major source of uncertainty is inaccuracy on initial parameters estimation. Also Smith [2003] states that “the question of prediction then turns to how to best quantify the dynamics of uncertainty” of propagating errors.

Many steps ahead forecasts can be calculated in several ways, for instance by fitting a specific model for each $h = 1 \ldots H$ step, as $\mathbb{E}[y(t{+}h)|y(t), y(t{-}1), \ldots]$, or iterating the model. If the time series is stationary, long runs $(h \to \infty)$ of the conditional mean will fatally fall on unconditional mean.

3.2 Interval Forecasts

The simplest evolution of point forecasts are the interval forecasts, that represent and incorporate uncertainty Hansen [2006]. At first sight interval forecasts may be confused with confidence intervals because they share the same structure, but they are slightly different things. Both are defined with respect to an unknown value $y(t+1)$ as an interval $\mathbb{I} = [\underline{l}, \overline{u}]$ with α confidence level to contain the real value of $y(t+1)$. The probability of $\mathbb{I}$ to contain $y(t+1)$ is given by $P(\underline{l} \leq y(t+1) \leq \overline{u}) = 1 - \alpha$.

Confidence intervals deal with fixed (but unknown) estimates. Prediction intervals instead, as proposed by Chatfield [1993], are an "estimate of an (unknown) future value that can be regarded as a random variable at the time the forecast is made. This involves a different sort of probability statement to a confidence interval as discussed". Traditional approaches for this kind of forecasting include the parametric methods as studied in Chatfield [1993]. These methods use strong statistical assumptions about the data that can make it less useful where data is not conforming.

The confidence level $\alpha \in (0,1)$ is then a way to determine a symmetric interquantile interval $[\alpha, 1-\alpha]$ for some forecasted value of interest. If a cumulative probability distribution $F : U \to [0,1]$ finds the probability $F(x) = P(X \leq x)$, the quantile function $Q : (0,1) \to U$ performs the opposite process: $Q(\tau) = \min_x\{x \in U \mid \tau < F(x)\}$ where $\tau \in (0,1)$ is a quantile.

Chatfield [2001] proposed a simple method for creating α-level prediction intervals for generic forecasting models, the called mean-variance model. From the point forecast $\mu = \mathbb{E}[Y_{t+1}|Y_t, Y_{t-1}, ...]$ with the variance of the residuals $\sigma_\epsilon = \sqrt{VAR[\epsilon]}$ by assuming that these residuals as $\epsilon \sim \mathcal{N}(0,1)$. The prediction interval is calculated by $I = [\mu - z_{\alpha/2}\sigma_\epsilon \ , \ \mu + z_{\alpha/2}\sigma_\epsilon]$ and $z_{\alpha/2} = \Phi((1-\alpha)/2)$ is the standard normal distribution function. In Figure 12a an example of ARIMA(2,0,0) process is shown, where the prediction intervals were calculated with the previous model, for $\alpha \in \{0.05, 0.25\}$.

For H-steps ahead, the variance σ_ϵ^h can be estimated from the 1-step ahead variance σ_ϵ^1 through exponential smoothing by $\sigma_\epsilon^h = (1 + h\beta)\sigma_\epsilon^1$, for some smoothing value $\beta \in (0,1)$. Despite its simplicity, the main drawback of this method is the parametric and homoskedastic assumption over the residuals distribution. In Figure 12a an example of ARMA(2,0,0) process is represented, where the prediction intervals for 7 steps ahead were calculated with the exponential smoothing, for $\alpha \in \{0.05, 0.25\}$ and $\beta = 0.5$. But Chatfield [2001] warns that mean-variance model is a generic approximation and does not replace prediction interval models specifically developed from the statistical methods and their error distribution assumptions.

The main probabilistic approach for interval forecasting is the Quantile Auto Regression - QAR proposed by Koenker and Xiao [2006] based on the Quantile Regression

Koenker and Hallock [2001]. The QAR estimates a conditional quantile function in Equation (3.1), where $\hat{y(t)}$ is the estimated quantile value, τ is the quantile level, θ are the fitted coefficients for the $y(i)$ lagged values and $\rho_\tau(u)$ is the Pinball Loss Function, defined on Equation (3.2), where $\mathbf{1}(x) = \{1 \text{ if } x \geq 0 \text{ or } 0 \text{ if } x < 0\}$. Quantile Regression approaches have been used at many application fields, for instance energy load forecasting [Liu et al. [2015], Hong and Fan [2016], Hong et al. [2016]] and wind forecasting Pinson et al. [2006].

$$Q_{y(t)}(\tau|y(t-1),\ldots) = \min_{\theta} \sum_{i=1}^{n} \rho_\tau(y(t) - y(i)\theta) \tag{3.1}$$

$$\rho_\tau(u) = u(\tau - \mathbf{1}(u < 0)) \tag{3.2}$$

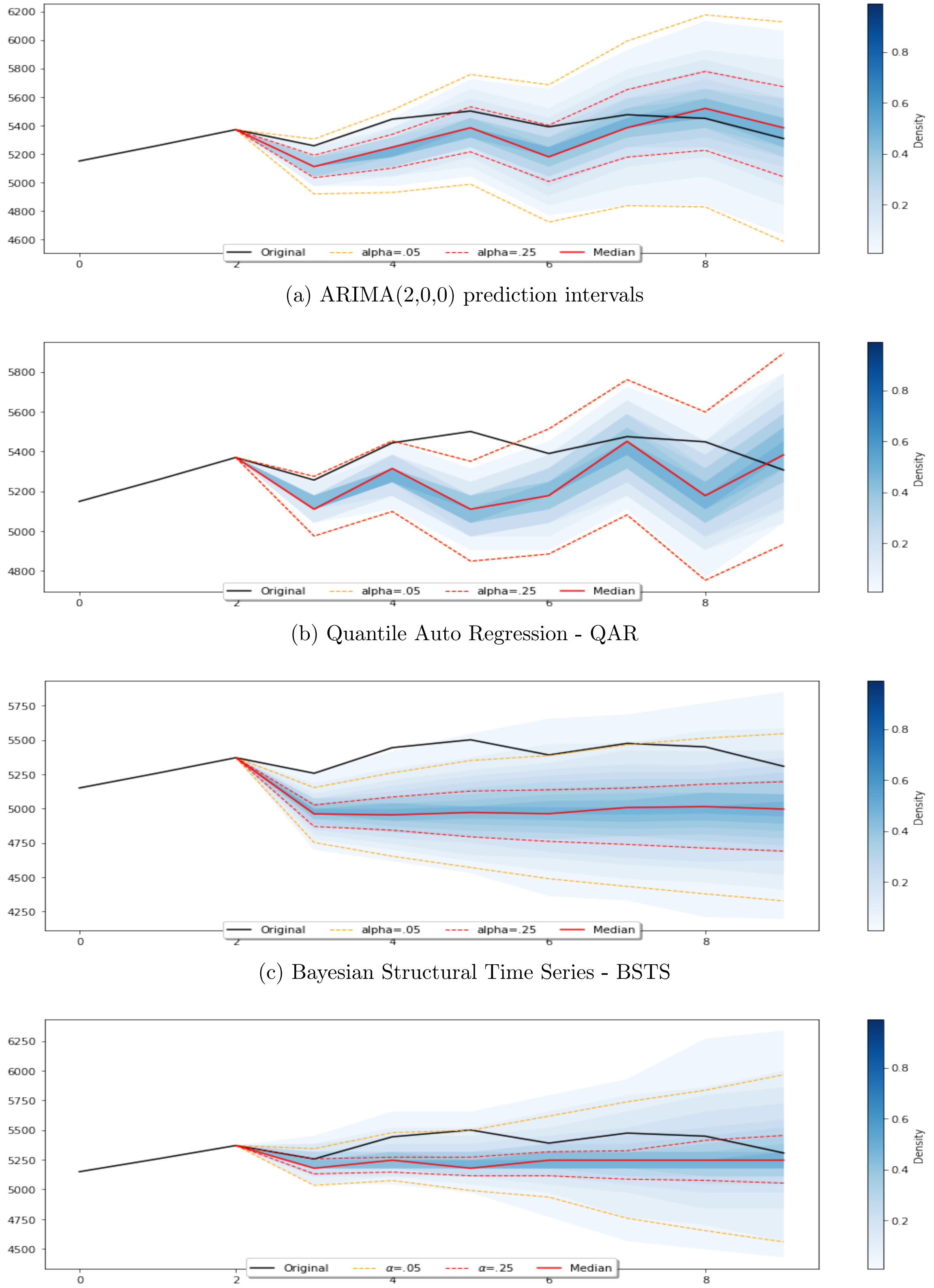
Each QAR model is fitted for a specific τ, so for a certain α two QAR models are necessary. The independence of quantiles also allows to create asymmetric inter quantile intervals, if needed. In Figure 12b an example of QAR(2) for $\tau \in \{0.05, 0.25, 0.75, 0.95\}$ is represented, equivalent for $\alpha \in \{0.05, 0.25\}$. The same principle is applied for H-steps ahead forecasts, it is needed to fit an specific model for each step ahead. In this case, for instance given $\alpha \in \{0.05, 0.25\}$ and $H = 10$, an specific QAR model will be estimated for each value of τ and each value of $h = 1..H$, resulting in 40 models. This approach is not very flexible and complicate its adoption by final users.

The Pinball Loss Function $\rho_\tau(u)$ is a general measurement to quantile approximation and Steinwart and Christmann [2011] also use it with Support Vector Machines to perform quantile regressions. Wan et al. [2014] use it with Extreme Learning Machines to fit quantile regression models. Other approaches are available in Takeuchi et al. [2006] for non parametric quantile estimation, Taylor [2007] proposes a Exponentially Weighted Quantile Regression and Hansen [2006], which proposes a semi-parametric k-step ahead approach for quantile estimation. Everette S. Gardner [1998] proposed a simple non parametric method for computing intervals based on the Chebyshev Inequality $P(|(Y - \mu)/\sigma| \geq \epsilon) \leq 1/\epsilon$ where μ is the mean, σ the standard deviation and ϵ will be estimated value, such as the forecasted interval is $[y(t+1) - \epsilon, y(t+1) + \epsilon]$.

3.2.1 Accuracy Measures for Interval Forecasts

In a wide sense the point forecasting accuracy measures can be used to assess the interval forecasts. This is possible by using the midpoint of the prediction interval, if the interval is based on α-levels or symmetric quantiles. But, by far, this is not the ideal way to measure the interval accuracy, once several aspects of prediction intervals are neglected by single point measures.

Some of the main aspects to be considered when evaluating prediction intervals are the *coverage rate*, *calibration* and *sharpness*, as proposed in Gneiting et al. [2007] and



(a) ARIMA(2,0,0) prediction intervals

(b) Quantile Auto Regression - QAR

(c) Bayesian Structural Time Series - BSTS

(d) k-Nearest Neighbors with Kernel Density Estimation - kNN/KDE

Figure 12 – Prediction Intervals

Pinson et al. [2006]. The *coverage* refers to the statistical consistency between the forecasts and the observations, and measures which proportion of the observations are inside the interval. This can be done by an Indicator Function, developed by Christoffersen [1998], as shown in Equation (3.3). Given a forecasting interval $\mathbb{I} = [\underline{l}, \overline{u}], u, l \in U$ and the real value $y \in Y$, the value of an indicator function $\mathbf{1}(\mathbb{I}(t), y(t))$ verifies if $y(t)$ is covered by $\mathbb{I}(t)$ or not.

$$\mathbf{1}(\mathbb{I}(t), y(t)) = \begin{cases} 1 & \text{if } y(t) \in \mathbb{I}(t) \\ 0 & \text{if } y(t) \ni \mathbb{I}(t) \end{cases} \tag{3.3}$$

The *coverage rate* is the average value of indicator function between forecasted intervals and the real values, in which the ideal value is 1. The coverage rate is shown at Equation (3.4) where $y(t) \in Y$ are the real values and $\mathbb{I}(t) \in \mathbb{I}$ are the predicted intervals for these values.

$$C(Y, \mathbb{I}) = T^{-1} \sum_{t=1}^{T} \mathbf{1}(\mathbb{I}(t), y(t)) \tag{3.4}$$

The property of *sharpness* and *resolution* refers to the concentration of the predictive distribution, or how wide and variable are the intervals and refers uniquely to the forecasts. *Sharpness*, presented in Equation (3.5), is the average size of the intervals and *resolution*, presented in the equation (3.6), is the variability of the intervals.

$$\overline{\delta(\mathbb{I})} = T^{-1} \sum_{t=1}^{T} \delta(\mathbb{I}(t)) = T^{-1} \sum_{t=1}^{T} \overline{u_t} - \underline{l_t} \tag{3.5}$$

$$\sigma(\mathbb{I}) = T^{-1} \sum_{t=1}^{T} |\delta(\mathbb{I}(t)) - \overline{\delta(\mathbb{I})}| \tag{3.6}$$

While small values of $\overline{\delta(\mathbb{I})}$ are desirable, meaning a compact interval, wide values of $\sigma(\mathbb{I})$ are best, meaning the capability of the model to adapt the length of interval with the increase of uncertainty. There are no absolute reference values for sharpness and resolution, which depend on the statistical properties of the data. Empirically, when the sharpness is reduced to make the intervals thinner and more precise, the risk of reducing the coverage increases, and that's why the resolution is important.

Steinwart and Christmann [2011] proposed the use of Pinball Loss Function - $\rho_\tau(u)$, defined in Equation (3.2) where $u = y(t) - \hat{y}(t)$, to indicate the proximity of a forecast $\hat{Y}$ with a certain τ quantile of the true value Y. As a loss function, the minor value of ρ_τ indicates the closest forecast to quantile τ. The Pinball Score ρ_τ^S is defined as the mean ρ_τ for a set true values $y(t)$ and forecasts $\hat{y}(t)$, listed in Equation (3.7). At this research the quantiles $\tau = \{0.05, 0.25, 0.75, 0.95\}$ were chosen for testing the intervals, where the lower quantiles were compared with the interval lower bound and the upper quantiles with the interval upper bounds.

$$\rho_\tau^S(Y,\hat{Y}) = \frac{1}{T}\sum_{t=1}^{T}\rho_\tau(y(t) - \hat{y}(t)) \tag{3.7}$$

However, using three separate metrics make the analysis of interval forecasters more complex. The most common option in these cases is the Winkler score Winkler [1972], which encompasses the three characteristics in only one measure. Given a target value y and a prediction interval $\mathbb{I} = [\underline{l}, \overline{u}]$ with nominal probability $(1-\alpha)$, the Winkler Score (WS) is defined by (3.8), where $\delta = \overline{u} - \underline{l}$. The score value is the interval width, but it increases when the target value is not covered by the interval and the penalty is proportional to the error given the nominal probability. Lower values therefore represent better prediction intervals. The mean score is defined by Equation (3.9), where T is the sample size.

$$WS(\alpha, y(t), \mathbb{I}(t)) = \begin{cases} \delta & if \quad \underline{l} \leq y \leq \overline{u} \\ \delta + 2(\underline{l} - y)/\alpha & if \quad y < \underline{l} \\ \delta + 2(y - \overline{u})/\alpha & if \quad \overline{u} < y \end{cases} \tag{3.8}$$

$$\overline{WS(\alpha, Y, \mathbb{I})} = T^{-1}\sum_{t=1}^{T} S(\alpha, y(t), \mathbb{I}(t)) \tag{3.9}$$

All revised interval forecasting methods are based on non-FTS methods and this is a gap in the FTS literature. In the next section a method for quantifying the bounds of fuzzy uncertainty is proposed.

3.3 The Interval Fuzzy Time Series - [$\mathbb{I}$]FTS

It was already discussed in Section 2.2.3 the impact of the number of partitions k on the accuracy of a FTS model. Also, in Section 2.5 it can be seen that in the deffuzification process only the midpoint of each fuzzy set is taken into account. This leads to the following question: what is the impact of the fuzzy set uncertainty (due to the overlapped bounds of the fuzzy set) on the final forecasting uncertainty? Since the fuzzy sets represent the empirical uncertainty of Y, how this uncertainty is propagated to the point forecasts?

The objective of this section is to propose a simple, fast and effective method to deal with fuzzy empirical uncertainty, combining the flexibility of the FTS models with the properties of Interval Forecasts without the need to resort to parametric methods or optimization techniques as in quantile estimation methods. The main advantage of this feature is to keep the model fast and scalable, for instance when the method is used in a high volume data or on a fast stream with concept drifts, which demands the model to be frequently updated. The scalability of FTS models will be discussed with more details in Chapter 5.

The Interval Fuzzy Time Series Model ($[\mathbb{I}]FTS$) aims to produce a prediction interval that represents the empirical uncertainty caused by the number of partitions k and the fuzzy sets bounds, but without any probabilistic meaning. The method $[\mathbb{I}]FTS$ is a time invariant, rule based and high-order method that just introduces a new deffuzyfication type on forecasting procedure, without modifying the training method and, because of this, it can be applied to every conventional FTS method.

The model training procedure is the same of HOFTS presented in Section 2.7.1 and aims to construct the FLRG rule base $\mathcal{M}$. The prediction intervals are based on the mean interval of the RHS fuzzy sets on each FLRG weighted by their fuzzy membership in relation to input value.

In the following section the forecasting method of $[\mathbb{I}]FTS$ will be presented, it extends the HOFTS forecasting method, presented in Section 2.7.2, changing its output from the crisp value $\hat{y}(t+1)$ to the prediction interval $\mathbb{I}(t+1)$.

3.3.1 Forecasting Procedure

Step 1 *Fuzzyfication*: Compute the membership grade μ_{ti} for each $y(t) \in Y$ where $t \in L$ and each fuzzy set $A_i \in \tilde{A}$, such that $\mu_{ti} = \mu_{A_i}(y(t))$.

Step 2 *Rule matching*: Select the K rules where all fuzzy sets A_i on the LHS have $\mu_{ti} > \alpha$; The rule fuzzy membership grade is shown below, using the minimum function as T-norm.

$$\mu_j = \bigcap_{t \in L \; i \in \tilde{A}} \mu_{ti} \tag{3.10}$$

Step 3 *Interval Defuzzyfication*:

a) *Rule intervals*: Each chosen rule j will generate an interval $\mathbb{I}^j = [\underline{\mathbb{I}^j_{min}}, \overline{\mathbb{I}^j_{max}}]$ where $\mathbb{I}^j_{min}$ is the minimum lower bound of all RHS fuzzy sets of the rule j and $\mathbb{I}^j_{max}$ is the maximum upper bound of RHS fuzzy sets of rule j;

$$\begin{aligned} \mathbb{I}^i_{min} &= \min(A_1, ..., A_k) \\ \mathbb{I}^i_{max} &= \max(A_1, ..., A_k) \\ & A_1, ..., A_k \in RHS \end{aligned} \tag{3.11}$$

b) *Final Prediction Interval*: The final forecast interval $\mathbb{I}(t+1)$ is calculated as the sum of the rules intervals weighted by the membership value of each rule, as shown in Equation (3.12)

$$\mathbb{I}(t+1) = \frac{\sum_{j\in A} \mu_i \mathbb{I}^j}{\sum_{j\in A} \mu_j} = \frac{\sum_{j\in A} [\mu_j \underline{\mathbb{I}^j_{min}}, \mu_j \overline{\mathbb{I}^j_{max}}]}{\sum_{j\in A} \mu_j} \tag{3.12}$$

Step 5 *Many steps ahead forecast*:If the forecasting horizon is $H > 1$, define $\mathbb{I}_H = \{\mathbb{I}(t+1)\}$ as the set of intervals and repeat the steps below for each $h = 2..H$, otherwise return $\mathbb{I}(t+1)$.

a) Given $\mathbb{I}(t+h-1) = [\underline{l}, \overline{u}]$, call recursively the forecasting method, such that $\mathbb{I}_l = forecast(\underline{l})$ and $\mathbb{I}_u = forecast(\overline{u})$. The interval $\mathbb{I}(t+h)$ is given by:

$$\mathbb{I}(t+h) = [\underline{min(\mathbb{I}_l)}, \overline{max(\mathbb{I}_u)}] \tag{3.13}$$

b) Append $\mathbb{I}(t+h)$ to $\mathbb{I}_H$ and if $h = H$ then return $\mathbb{I}_H$.

The generated interval $\mathbb{I}(t+1)$ is bounded by a composition of the fuzzy sets bounds on the FLRG's which have some membership with the input value $y(t)$ and is expected to contain the true value $\hat{y}(t+1)$. A sample of the method performance, for one and many steps ahead, can be seen in Figures 14 and 15. In the next section a weighted version of $[\mathbb{I}]FTS$ is presented.

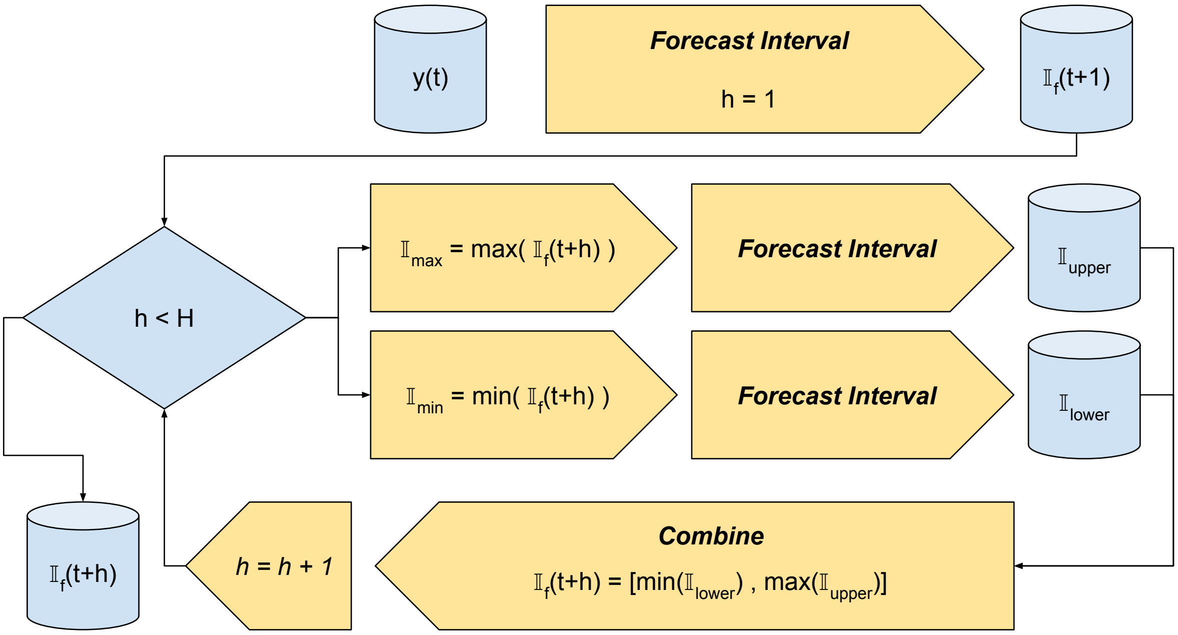


Figure 13 – $[\mathbb{I}]FTS$ many steps ahead interval forecasting procedure

3.3.2 Weighted $[\mathbb{I}]FTS$

A weighted $[\mathbb{I}]FTS$ extension uses the same model building procedure of WHOFTS method, presented in Section 2.7.3 and aims to construct the weighted FLRG rule base

$\mathcal{M}$. The prediction intervals are based on the weighted interval of the RHS fuzzy sets on each FLRG weighted by their fuzzy membership in relation to input value.

$W[\mathbb{I}]FTS$ extension also changes the Steps 3.a of the Forecasting Procedure, replacing the Equation (3.11) by the Equation (3.14), where $\underline{A_j}$ and $\overline{A_j}$ represents respectively the lower and upper bounds of each fuzzy set $A_j \in RHS$:

$$\begin{aligned} \mathbb{I}_{min} &= \sum_{j \in RHS} w_j \cdot \underline{A_j} \\ \mathbb{I}_{max} &= \sum_{j \in RHS} w_j \cdot \overline{A_j} \end{aligned} \tag{3.14}$$

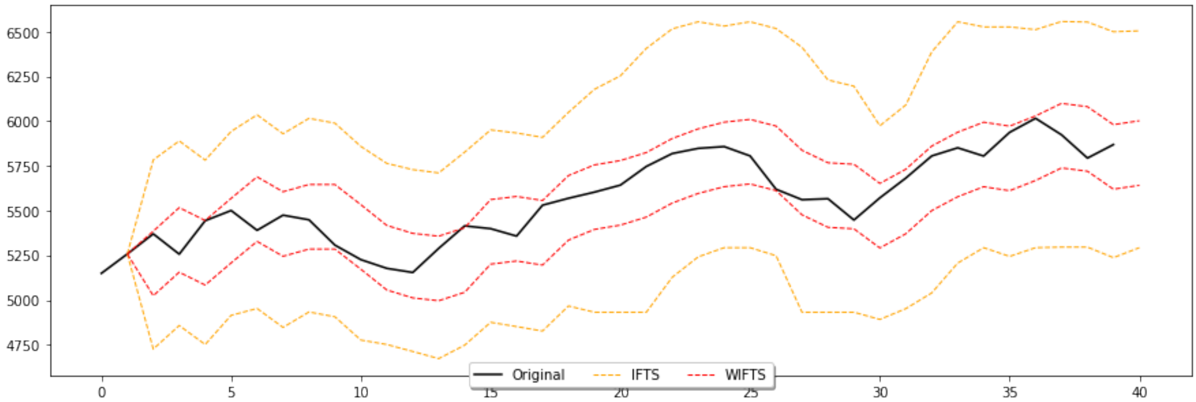


Figure 14 – Sample of $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods for one step ahead

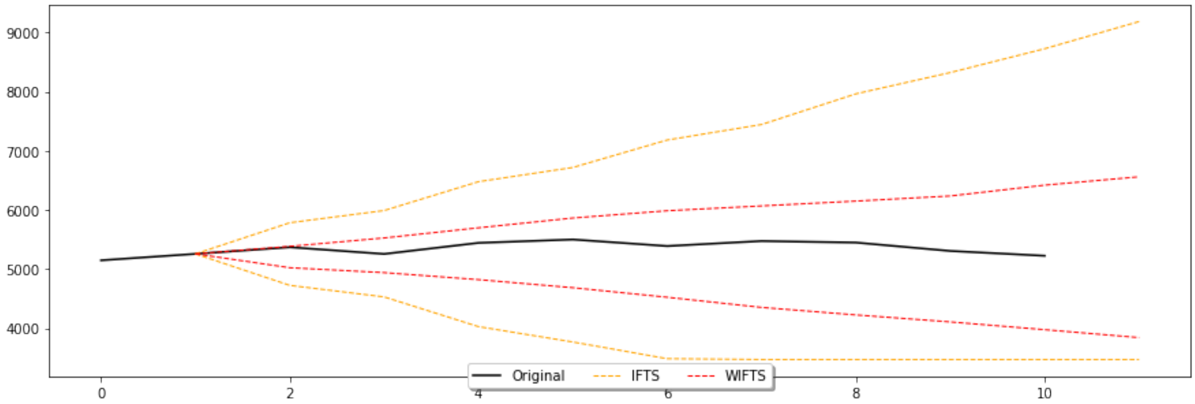


Figure 15 – Sample of $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods for many steps ahead

Different from $[\mathbb{I}]FTS$, which considers the extremum of the fuzzy sets, the generated interval $\mathbb{I}(t+1)$ of $W[\mathbb{I}]FTS$ is bounded by a weighted composition of the fuzzy sets bounds, making it more sharper. A sample of the method performance, for one and many steps ahead, can be seen in Figures 14 and 15.

Interval forecasts help to embrace the notion of forecast uncertainty, but did not offer a complete landscape of the uncertainty. In the next sections, the probabilistic forecasting is presented as a more embracing way to represent the uncertainty.

3.4 Probability Distribution Forecasts

If the interval forecast represents a range of uncertainty, according to Krzysztofowicz [2001] the probabilistic distribution forecast "should quantify the total uncertainty that remains about the predictand, conditional on all information utilized on the forecasting process".

The probability distribution forecasts cover the complete range of the U and can be continuous, using a probability density function, or discrete using both a discrete probability mass function or an empirical probability distribution, as histograms. In this last case, U is split in n equal length sub-intervals b_i - called bins -, which are associated to a probability p_i of occurrence, such as $\sum_{i=1}^{n} p_i = 1$. The length of each bin is the unit of discretization also referenced as the resolution of the distribution.

Probability distribution forecasts can be represented also as the empirical cumulative probability distribution function $F(x) = n^{-1} \sum_{i \in Y(t)} \mathbf{1}(i < x)$ where $\mathbf{1}(c) = \{1$ if $c = True;\ 0$, otherwise $\}$. Using α-level prediction intervals, described in Section 3.2, for $\alpha \in [0.05, ..., 0.5]$, it is easy to construct the empirical distribution F by iteration over the bounds of intervals. This approach was used to produce the distributions in Figure 12a, from 7-steps ahead mean-variance prediction intervals over an ARIMA(2,0,0) model. The same approach can be used to construct probabilistic forecastings using QAR method, as shown in Figure 12b.

The Gaussian Process Regression (GPR), as discussed in Rasmussen and Williams [2006] and Roberts et al. [2013], is an instance based parametric approach which interpolates the instances $y(t) \in Y$ and also produces an extrapolation for $y(t+1)$ in the form of a Gaussian Distribution. A Gaussian Process $\mathcal{GP}(m, \kappa)$ is defined by a mean function $m(Y)$ and covariance kernel $\kappa(y(i), y(j))$ which produce the covariance matrix Σ, such that $y(t+1) \sim \mathcal{N}(m(Y), \Sigma)$. The mean function $m(Y)$ is defined as the unconditional expectation of Y, such that $m(Y) = \mathbb{E}[Y]$. The covariance matrix Σ assigns the similarity $\sigma_{ij} \in \Sigma$ between all pairs of instances of the time series, and it is defined by the covariance function $\sigma_{ij} = \kappa(y(i), y(j))$ for all $i, j = 1..T$. The covariance function is defined as $\kappa : U, U \rightarrow \mathbb{R}^+$ and measures the similarity between the two instances.

The covariance function κ is the most important parameter of $\mathcal{GP}$ model, and it is responsible to measure how the instances relate with themselves and with time. κ is usually defined by a set of parameters or hyperparameters θ, and because this κ is also often written as $\kappa(y(i), y(j)|\theta)$ to make the dependence on θ explicit.

The most notable drawbacks of the GPR approach are the parametric assumption, the non-sparsity, i.e., it uses the whole set of samples to perform one prediction which is undesirable for Big Data scenarios. Even with the fast and direct aproaches developed by Ambikasaran et al. [2014], that method still loses efficiency as the number of variables grows.

The Bayesian Structural Time Series (BSTS), discussed in Scott and Varian [2014] and Barber et al. [2011], mix the well known State-Space models with the Bayesian Statistics approach for parameter estimation. A structural time series is a state-space model which associates the observed value $y(t)$ with an unobserved latent state s_t. The structure is defined by a system of equations where the observation equation (also known as measurement equation) is defined in Equation (3.15) and the transition equation is defined in Equation (3.16), and this system of equations can also be referred as a Linear Gaussian Model.

$$y(t) = Z_t s_t + \epsilon_t \qquad \epsilon_t \sim \mathcal{N}(0, H_t) \tag{3.15}$$

$$s_t = T_t s_{t-1} + R_t \eta_t \qquad \eta_t \sim \mathcal{N}(0, Q_t) \tag{3.16}$$

An observed value $y(t)$ is understood as a signal with noise, where the signal is the product of the unobserved state s_t by the regressor parameter Z_t, and the noise (the extrinsic uncertainty) represented by the Gaussian error term ϵ_t whose variance is controlled by the parameter H_t. The unobserved latent state s_t is represented by a vector with the several components of the time series, as trend, seasonality, level, etc, and is recursively defined by the transition matrix T_t and the Gaussian error term η_t (the intrinsic uncertainty), which in turn is controlled by the vector R_t and the covariance matrix Q_t. The error terms ϵ_t and η_t are mutually independent. This state-space representation can unify several approaches of time series forecasting and the parameters Z_t, T_t, H_t, R_t and Q_t define the structure of the model, hereafter called as the Θ or the parameter space. For instance, for an ARMA approach the regressors are represented by Z_t and the coefficients by T_t.

Once the state space model is defined, it is necessary to infer the values of Θ parameters from the training data Y, keeping in mind that Y is a sample and it is composed with several sources of uncertainty, so it will be also Θ. The Bayesian framework is employed in this task, which represents all uncertainties as probability distributions, from the learning process, passing through the parameter space Θ, to the prediction space U. In such approach, the model parameters are probability distributions $P(\Theta|Y)$, reflecting the uncertainty around the real (but unknown) values of each parameter in $\theta \in \Theta$ given the learning sample Y. The forecast of $\hat{y}(t+1)$ is a probability distribution $P(y(t+1)|\Theta, Y)$ that reflects the intrinsic uncertainty inherent of the time series Y, and the extrinsic uncertainty of the unknown real parameter values $P(\Theta|Y)$.

The learning of the parameter space Θ is guided by the Bayes Rule. It states that, given a set of known evidences $d \in \mathcal{D}$ and a set of possible hypothesis $h \in \mathcal{H}$, the posterior distribution $P(h|\mathcal{D})$ is given by the Equation (3.17), where $P(\mathcal{H})$ is the prior distribution, the $P(\mathcal{D}|h)$ is the likelihood, and $P(\mathcal{D})$ is the normalizing term, such that

$P(\mathcal{D}) = \sum_{h \in \mathcal{H}} P(\mathcal{D}|h)P(h)$ according to the Law of Total Probability.

$$P(h|\mathcal{D}) = \frac{P(\mathcal{D}|h)P(h)}{P(\mathcal{D})} = \frac{P(\mathcal{D}|h)P(h)}{\sum_{h \in \mathcal{H}} P(\mathcal{D}|h)P(h)} \tag{3.17}$$

The prior distribution $P(\mathcal{H})$ assigns the knowledge about the chances of each $h \in \mathcal{H}$ be the real value. The likelihood function $\mathcal{L}(h|\mathcal{D})$ assembles the plausibility of the evidences $d \in \mathcal{D}$ to have been generated by the parameter h, and it is equals to $P(\mathcal{D}|h)$.

Some methods are available to estimate the best hypothesis h in the search space $\mathcal{H}$. The Maximum A Posteriori (MAP) principle poses the best hypothesis $h \in \mathcal{H}$ is that one for which $h_{MAP} = \arg\max_{\mathcal{H}} P(h|\mathcal{D})$. Given that $P(\mathcal{D})$ is a constant and it is hard to compute, it is eliminated from the calculation, considering just $P(h|\mathcal{D}) \propto P(\mathcal{D}|h)P(h)$. The h_{MAP} is used to update $P(\mathcal{H})$ and improve the aproximation of $P(\mathcal{H}|\mathcal{D})$ as new data is acquired. The Maximum Likelihood Estimation (MLE) method uses the average log-likelihood function $\hat{\mathcal{L}}(h|\mathcal{D}) = |\mathcal{D}|^{-1} \sum_{i=0}^{|\mathcal{D}|} lnP(d_i|h)$ as a cost function, such that $h_{MLE} = \arg\max_{\mathcal{H}} \hat{\mathcal{L}}(h|\mathcal{D})$ is the best estimate parameter $h \in \mathcal{H}$. The drawback of these estimators is to estimate a unique point value without representing the uncertainty around the best hypothesis in $\mathcal{H}$.

However, the great strength of the Bayesian Methods is its ability to represent the uncertainties contained both on data and model parameters with probability distributions. This is also the great drawback of Bayesian Methods: their expensive computational cost. It is mainly because not all parts of its equation are always available – like the likelihood function $P(\mathcal{D}|h)$ – and those values needs to be simulated using Monte Carlo methods.

The Monte Carlo (MC) methods, initially proposed in Metropolis and Ulam [1949], are techniques to solve complex integration problems using random sampling. They aim to generate a set of samples $x_1, \ldots, x_n$ from a target distribution $\pi(x)$ in order to estimate some hard-to-compute feature $\phi(x)$ using the expected value $\mathbb{E}[\phi(x)] = n^{-1} \sum_i^n \phi(x_i)$ which converges to the unobserved real value of ϕ. Given the estimated value as $\overline{\phi} = \mathbb{E}[\phi(x)]$, and its variance as $\phi_\sigma = Var[\phi(x)]$, some statistical concepts support the convergence of the MC methods. The Law of Large Numbers (LLN) asserts that, for a large enough number of samples n, the difference between the estimated value $\overline{\phi}$ and the true value ϕ decays to zero, or $P(\lim_{n \to \infty} |\overline{\phi} - \phi| = 0) = 1$. The Central Limit Theorem (CLT) states that, for a large enough number of samples n, the $\overline{\phi}$ is normally distributed as $\overline{\phi} \sim \mathcal{N}(\phi, \phi_\sigma/n)$.

Markov Chain Monte Carlo (MCMC) methods improve the basic MC approach aiming to, instead of sampling $\pi(x)$ directly, sample from a Markov Chain with a transition matrix K, where $K_{i,j} = P(x_t = i|x_{t-1} = j)$, such that the next sample x_t be conditionally dependent on the previous x_{t-1}. The Markov Chain K needs to approximate the real $\pi(x)$ distribution, but estimating K is very often an intractable problem. An approximation is provided by the Metropolis-Hastings algorithm generalized in Hastings [1970], which provide a simple and efficient way to simulate K and generate samples.

Estimating $P(\Theta|Y)$ using Bayesian methods is an optimization task which employs MCMC in order to sample from $P(Y|\Theta)P(\Theta)$ distribution, while refining the parameters of $P(\Theta)$. The method demands the choice of an appropriate a priori distribution $P(\Theta)$ that will rule the search space of each parameter $\theta \in \Theta$. The likelihood $P(Y|\Theta)$ estimates the fit of each parameter $\theta \in \Theta$ when generating samples of $y(t) \in Y$. This likelihood is itself another challenge once the samples $y(t) \in Y$ may be identically distributed but are not independent, there is a temporal dependence between $y(t)$ and it's past lags $y(t-1), ...$ that must be respected. This problem demands the use of advanced MCMC methods as Sequential Monte Carlo, Sequential Importance Sampling and Particle Filters, deeply discussed in Smith [2013].

Once Θ values were estimated and represented by probability distributions $P(\Theta|Y)$, the estimation of $\hat{y}(t+1)$ will be represented by a probability distribution $P(y(t+1)|\Theta, Y)$, defined in Equation (3.18), which is also expensive to calculate and, again, needs to resort to MCMC methods.

$$P(y(t+1)|\Theta, Y) = \int_U \int_\Theta P(y|\theta, Y)P(Y|\theta)P(\theta)dyd\theta \tag{3.18}$$

A small sample of the BSTS method is shown in Figure 12c, for 7 steps ahead interval and probabilistic forecasting. If in one hand the Bayesian Structural Time Series are well succeeded in representing the intrinsic and extrinsic uncertainties, on the other hand it is complex to implement and computationally expensive to run, making it not applicable for a variety of scenarios where the time performance is mandatory.

There are other approaches to embody the uncertainties of model parameters. Monte Carlo methods itself evoke the idea of forecasting combination and Ensemble Methods, as posed in Smith [2003], “In practice, ensemble forecasting is a Monte Carlo approach to estimating the probability density function (PDF) of future model states given uncertain initial conditions”. Forecast combination is not a new concept, see [Clemen, 1989], and start from the idea to mix different sources to improve forecasting. This is sightly close to the concept of Ensemble Methods defined by Gneiting [2008] as “an ensemble prediction system consists of multiple runs of numerical weather prediction models, which differ in the initial conditions”. Also Leutbecher and Palmer [2008] states that “The ultimate goal of ensemble forecasting is to predict quantitatively the probability density of the state of the atmosphere at a future time”.

Initially Ensemble Learning methods were developed to produce point forecasts as combination of the individual model's forecasts by a weighted average or more complex methods as Bayesian Model Averaging, for instance Raftery et al. [2005]. Soon after, these methods were adapted for probabilistic forecasting as in Gneiting et al. [2005], Leutbecher and Palmer [2008] and Fraley et al. [2013]. Xie and Hong [2016] proposed a methodology for electric load probabilistic forecasting in three steps: *pre-processing step* consisting of data cleaning; *forecasting step* using point-forecasting methods, forecast combination and scenario-based probabilistic forecasting; *post-processing step* performing a simulation on the residuals of the selected point forecasting models in order to improve the probabilistic forecast.

These ensembles can be homogeneous (same method with different parameters) or hybrid (different methods with different parameters). This set of models $\mathcal{M}$ receive a set of parameters Θ to produce a set of forecasts $\hat{y}(t+1)$, such as $\hat{y}(t+1)^i = m_i(\theta_i)$, $\forall m_i \in \mathcal{M}$ and the $\theta_i \in \Theta$ values are drawn of a prior probability distribution $P(\Theta)$. The methods can be executed several times and the larger the sample is, the better approximations are made. After n runs, the empirical distribution $P(y(t+1))$ of the outputs is available.

Ensemble Learning is a variation of the Ensemble Forecasting on Machine Learning field, defined by Brown [2010] as "the procedures employed to train multiple learning machines and combine their outputs with individual predictions combined appropriately, should have better overall accuracy, on average, than any individual committee member". Ensemble Learning can be used in classification in regression tasks, also time series forecasting as in Chen and Zhang [2005], Bai et al. [2010] and Grmanová et al. [2016].

This concept is exploited in Mohammed et al. [2015] and Mohammed and Aung [2016], which proposed an ensemble learning approach for solar power probabilistic forecasting based on k-Nearest Neighbors, Regression Trees, Random Forests and regression methods. Given an ensemble with k models and taken the ordered set of the k individual forecasts, the probabilistic forecast is constructed as an empirical cumulative distribution F, calculated with the percentiles of the individual forecasted values. F can be made with three approaches: *quantile linear interpolation*, *normal distribution* and *normal distribution with initial different conditions*. The linear interpolation approach calculates the τ quantile position r_τ on the individual forecasts as $r_\tau = \frac{k\tau}{100} + 0.5$. The normal distribution approach is similar to mean-variance model of section 3.2. With the set of individual forecasts the mean μ and the variance σ are calculated, and the τ quantile is given by $\tau = \mu + z_\tau \cdot \sigma$. The third approach is specific for the application domain of solar power.

The advantage of Mohammed et al. [2015] approach is the flexibility and ease of implementation, since the individual models can be replaced (or added) for any other point forecaster, for instance, any FTS method. As more models are added to the ensemble, better will be the probabilistic distributions but also become more computationally expensive.

Other distributions generating techniques for ensemble forecasts exist as Kernel Density Estimation [Hong et al., 2016] and Kernel Dressing [[Pinson and Madsen, 2009] and [Bröcker and Smith, 2008]] and can be easily combined with instance-based methods as k-nearest neighbors. Both approaches smooth the discrete values in a continuous function

Kernel	**Definition**
Triangular	$K(u) = 1 - \|u\|$
Tophat	$K(u) = \frac{1}{2}\mathbb{I}(\|u\| < 1)$
Epanechnikov	$K(u) = 3/4(1-u^2)$
Gaussian	$K(u) = \frac{1}{\sqrt{2\pi}}e^{-1/2u^2}$

Table 9 – KDE Kernels

that approximates the empirical distribution of data, as exposed in Equation (3.19), where Y is the set of individual forecasts, K is the kernel function and h is a smoothing parameter also known as bandwidth.:

$$P(x) = (nh)^{-1} \sum_{i \in Y} K\left(\frac{x-i}{h}\right) \tag{3.19}$$

A kernel function K have to be a non-negative, real-valued, symmetric, integrable and normalized, such that $\int_{-\infty}^{+\infty} K(u)du = 1$. A review of density estimation methods can be found in Silverman [1986] and a specific study on estimation of h parameter can be found in Sheather and Jones [1991]. A small sample of the kNN with KDE approach is shown in Figure 12d, for 7 steps ahead interval and probabilistic forecasting.

The several methods discussed in this section are spread in the literature. In the next section accuracy measures for probabilistic forecasting are discussed.

3.4.1 Accuracy Measures for Probabilistic Forecasts

As the probabilistic forecast provides the landscape of uncertainty for the whole U, it is also possible to use the accuracy measures presented in Sections 2.8.1 and 3.2.1 to assess its accuracy. A probability distribution can be reduced to a point using its expected value $\mathbb{E}$ or it's median $m = F(.5)$. In both cases, the point forecasting accuracy values can be used to assess its accuracy.

A probability distribution P can also be expressed in terms of intervals as well, by using α-levels and their respective quantiles. In this case, the prediction interval accuracy measures can be used to evaluate P accuracy in several different α, in the many different aspects discussed in Section 3.2.1.

But pure probabilistic accuracy measures intend to assess how well the probabilities of P are spread over U when we know the true value $y(t)$, and how close P were able to predict the uncertainty around $y(t)$. The most simple probabilistic forecasting measure is the Logarithm Score (LS), proposed in Good [1952] and defined in Equation (3.20), which indicates how strong was the probability distribution P to predict the real value $y(t)$. The Logarithm Score presents some limitations as, for instance, when $P(y(t)) = 0$

then $LS(P, y(t)) = \infty$.

$$LS(P, Y) = T^{-1} \sum_{t=1}^{T} -log(P(y(t))) \tag{3.20}$$

The Brier Score (BS), proposed in Brier [1950] and defined in Equation (3.21), was originally defined for R categorical events but it is possible to extend it for numeric values by splitting the Universe of Discourse in R bins, or to consider the quantiles. This score can be interpreted as the Mean Squared Error (MSE) from the predicted probabilities for each bin r, represented by $P(r)$, to the real observed events represented by $\mathbf{1}\{y(t) \in r\}$.

$$BS(P, Y) = T^{-1} \sum_{t=1}^{T} \sum_{r=1}^{R} (P(r) - \mathbf{1}\{y(t) \in r\})^2 \tag{3.21}$$

The metric chosen to assess the distributions is the Continuous Ranked Probability Score (CRPS). CRPS is a proper measure for probabilistic forecasts, defined by Gneiting and Raftery [2007] as Equation (3.22) for one forecast and by Gneiting and Raftery [2007] as Equation (3.23) for more than one forecasts. CRPS provides a direct way to benchmark probabilistic forecast since it is expressed in the same unit as the observed variable and is a generalization of the Mean Absolute Error (MAE). Therefore, the perfect score for CRPS, as in MAE, is 0.

$$CRPS(F, x) = \int_{-\infty}^{+\infty} (F(y) - \mathbf{1}\{y \geq x\})^2 dy \tag{3.22}$$

$$CRPS(F, x) = \frac{1}{N} \sum_{t=1}^{N} \int_{-\infty}^{+\infty} (F_t(y) - \mathbf{1}\{y \geq x_t\})^2 dy \tag{3.23}$$

where F is the cumulative distribution function (CDF) of the forecasted distribution, x is the true value and $\mathbf{1}\{y \geq x\}$ is the Heavyside function representing the CDF of this punctual value.

3.4.2 Fuzzy Time Series Methods With Probabilistic Background

The first studies to make the relationship of probabilities with fuzzy sets came from Prof. Zadeh, Zadeh [1968], Zadeh [1984], which defines the fuzzy set probability as the expectation of the membership function. Also, Klement, Schwyhla and Lowen Klement et al. [1981] and Dubois and Prade [1989] explore the relationships between the fuzzy membership functions and the probability measures based on Measure Theory.

These theoretical works form the basis of the Fuzzy Stochastic Fuzzy Time Series (FSFTS) of Song, Leland and Chissom Song et al. [1997], where three models were presented but there is no empirical analysis of their results. In their work the probability space $[0, 1]$ is also described by a linguistic variable $\tilde{P}$ with few fuzzy sets that describe the probability in linguistic terms like "low","medium","high". The rules of the model are composed by tuples (p_j, A_j) where $p_j \in \tilde{P}$ and $A_j \in \tilde{A}$ and the deffuzyfication weights the fuzzy sets by their fuzzy probability.

Gangwar and Kumar [2014] proposed the *Probabilistic and Intuitionistic Fuzzy Time Series* - PIFTS method, strongly based on data normality and explicit Gaussian Process assumption. Cheng and Li [2012] and Chuang et al. [2014] use the Song and Chissom relation matrix and a Hidden Markov Chains for generating simulations for forecasting step, the *Probabilistic Smoothing Hidden Markov Model FTS* - psHMM-FTS, which has high computational cost.

3.5 The Ensemble FTS Method

The $[\mathbb{I}]FTS$ represented the forecasting uncertainty using the mean of the bounds of the fuzzy sets, or the empirical uncertainty, without any probabilistic sense. However, the FTS methods use of several ways to represent the time series uncertainties, as the number of partitions k, order Ω, lag indexes L, rule weights. There are uncertainties surrounding these values due to the ontological uncertainty of the data, uncertainties that will be reduced – but not removed – after the hyperparameter optimization proposed in Chapter 5.

A way to encompass these uncertainties is to generate a meta model $\mathcal{M}$, composed with several FTS models $m \in \mathcal{M}$, such that each one of these individual models is trained with different values of FTS hyperparameters, which aim to represent the hyperparameter uncertainty and its effects. The aggregation of the individual forecasts $\hat{y}(t+1)_m$ produced by each $m \in \mathcal{M}$ can represent the overall probabilistic uncertainty P over the possible values of $y(t+1) \in U$ using kernel density estimation seen in Section 3.4.

Several approaches can be adopted to represent the uncertainty of the hyperparameter, from varying the number of partitions and order, passing through varying the partitioning methods, until varying the FTS method itself. However, the present approach adopts a conventional rule-based high order FTS method with a Grid partitioning method, varying only the number of partitions k and order Ω.

On EnsembleFTS the hyperparameters k and Ω are intervals and not scalar values. During the training procedure, explained in Section 3.5.1 an FTS model will be trained for each combination of individual values in the Cartesian Product of k and Ω. In the forecasting procedure detailed in Section 3.5.2, the input sample is presented for all internal models and their outputs are aggregated using a KDE, producing a probability distribution $P : U \to [0, 1]$. Besides k and Ω ranges, the kernel function K and its bandwidth parameter h are also parameters of the model.

3.5.1 Training Procedure

The aim of the training procedure is to build an ensemble $\mathcal{M}$ with $k \times \Omega$ individual models m_i, given a crisp training set Y. The overall training procedure is shown in Figure 16 and it is composed of the following steps:

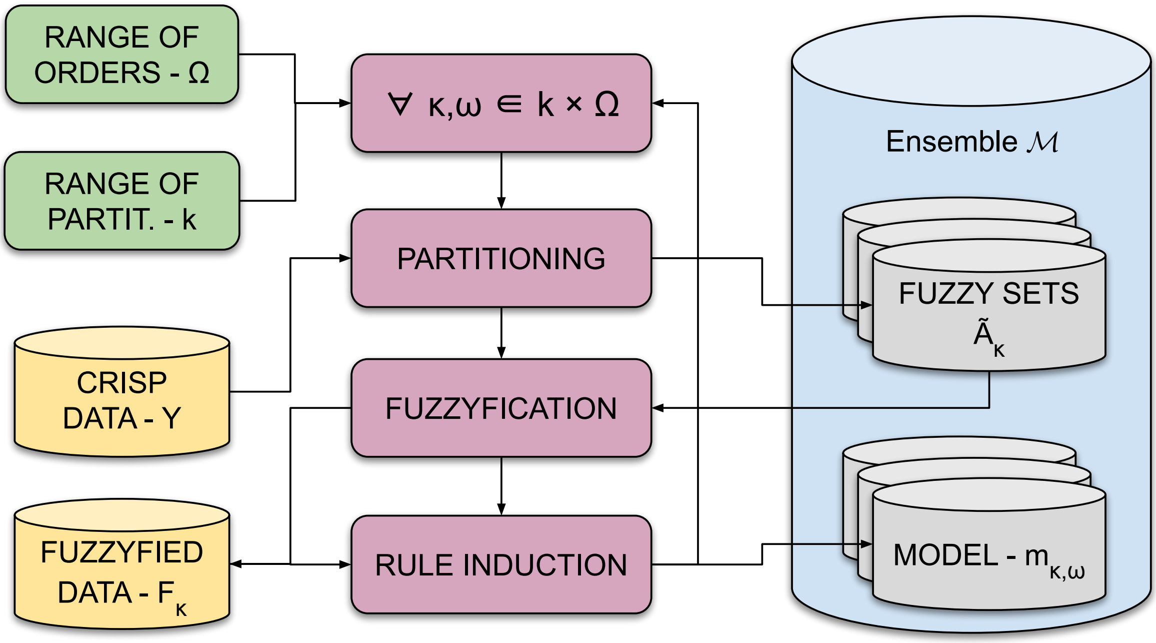


Figure 16 – Ensemble FTS training procedure

Step 1 *Main Loop*: For each pair (κ, ω) created by the cartesian product of each $\kappa \in k$ with each $\omega \in \Omega$, repeat Steps 2 to 5;

Step 2 *Partitioning*: Create a linguistic variable $\tilde{A}_\kappa$ over U with κ fuzzy sets using the Grid partitioning method and triangular μ;

Step 3 *Fuzzyfication*: Fuzzyfy the crisp time series Y using the linguistic variable $\tilde{A}_\kappa$, creating the fuzzyfied time series data F_κ;

Step 4 *ω-order model*: With F_κ, use the high order weighted rule knowledge model to infer ω-order fuzzy rules and compose the model $m_{\kappa,\omega}$;

Step 5) *Ensemble*: Append the model $m_{\kappa,\omega}$ on $\mathcal{M}$;

3.5.2 Forecasting Procedure

With the ensemble $\mathcal{M}$ built as in previous section, and given an input test sample $y(t)$, it is desired to forecast a full probability distribution $P(y(t+1)|y(t))$. The overall process is described in Figure 17 and is composed of: a) the forecasting of individual models; b) forecast selection and c) distribution smoothing with the KDE. The complete procedure is detailed below:

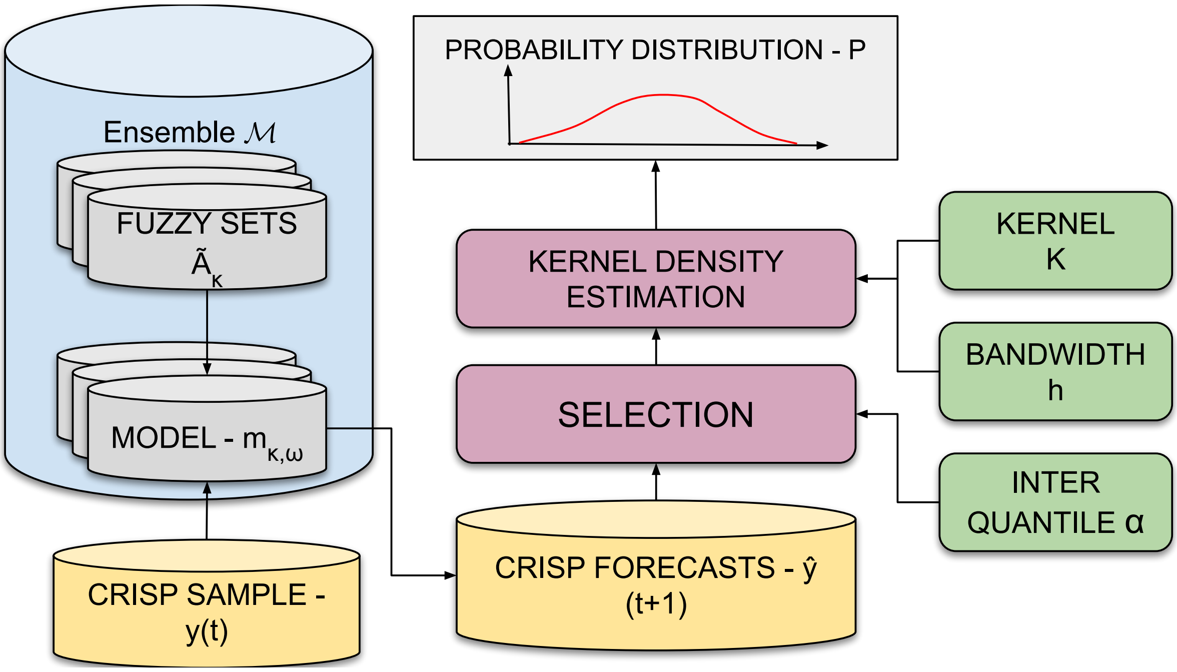

Ensemble $\mathcal{M}$
FUZZY SETS $\tilde{A}_\kappa$
MODEL - $m_{\kappa,\omega}$
CRISP SAMPLE - y(t)
PROBABILITY DISTRIBUTION - P
KERNEL DENSITY ESTIMATION
SELECTION
CRISP FORECASTS - ŷ (t+1)
KERNEL K
BANDWIDTH h
INTER QUANTILE α

Figure 17 – Ensemble FTS forecasting procedure

Step 1 *Individual forecasts*: The input sample $y \in Y$ is presented to each internal model $m_j \in \mathcal{M}$, which in turn will produce an individual forecast $\hat{y}_j(t+1)$. The set of crisp forecasts is hereafter called $\hat{y}(t+1)$.

Step 2 *Forecast selection*: In order to control the total forecast variance and eliminate the effect of possible outliers the forecasted output is limited by an inter quantile interval $(\alpha, 1-\alpha)$ where $\alpha \in (0,1)$ is the confidence level. By varying α parameter it is possible to fine tune the final distribution accuracy by eliminating forecasts that are too distant from the mean.

Step 3 *Kernel density estimation*: The set of crisp forecasts $\hat{y}(t+1)$ is used with a kernel density estimator K to estimate the probability distribution $P_{t+1} : U \to [0,1]$. Two parameters are necessary on this step: the type of kernel and the bandwidth h. Both parameters are domain specific and need to be empirically evaluated for each application.

Step 4 *Many steps ahead forecast*:If the forecasting horizon is $H > 1$, define $P_H = \{P_{t+1}\}$ as the set of intervals and repeat the steps below for each $h = 2..H$, otherwise return P_{t+1}.

a) Find the quantiles $Q = \{.1, .2, .3., .4, .5, .6, .7, .8, .9\}$ of the last $\max(\Omega)$ forecasted sets $\hat{y}(t-\omega)$, $\forall \omega \in \Omega$, such that $\hat{y}_Q(t-\omega) = \{Q_\tau(\hat{y}(t-\omega) | \tau \in Q)$ where Q is the Quantile Function;

b) Apply a Cartesian Product between the quantiles of the last $\max(\Omega)$ forecasted sets, such that $\hat{Y} = \prod_{\omega\in\Omega} \hat{y}_Q(t-\omega)$;

c) Use each sample $\hat{y}(t) \in \hat{Y}$ as input to Step 1 and 2 and aggregate all the results on the set $\hat{y}(t+1)$

d) Use $\hat{y}(t+1)$ with Step 3 to produce P_{t+h} and include it on P_H. If $h = H$ then return P_H.

The probabilistic forecast $P(y(t+1)|y(t))$ aims to represent $\hat{y}(t+1) \in U$ uncertainties of the model $\mathcal{M}$ in relation to k and Ω. A sample of the EnsembleFTS for one step and many steps ahead forecasts can be seen in Figures 18 and 19.

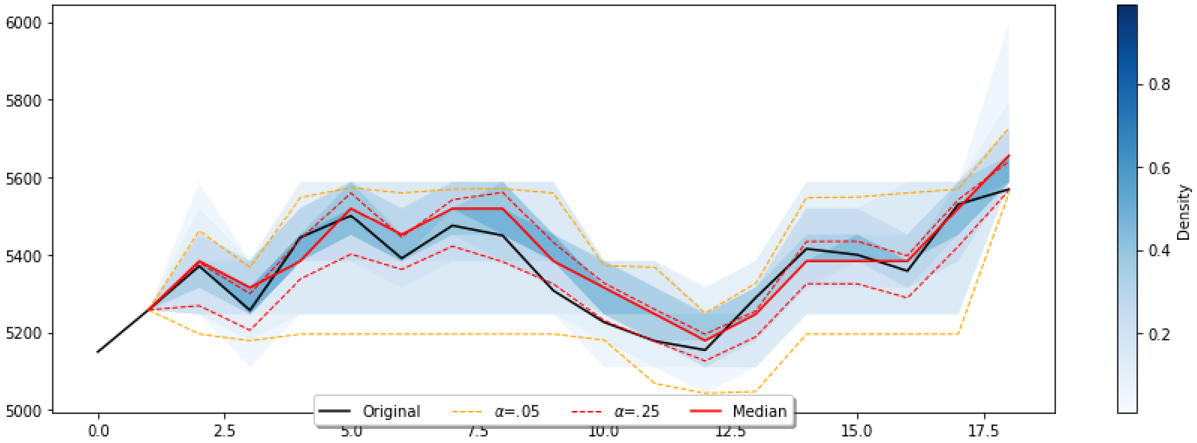


Figure 18 – Sample of EnsembleFTS performance for one step ahead

3.6 Computational Experiments

In this section an empirical study of the performance of the proposed methods is presented using three economic time series. Initially, the datasets, design of experiments and statistical tests employed are discussed. In Section 3.6.1, the accuracy sensitivity regarding to the hyperparameters of the proposed methods are analyzed using a grid search. In Section 3.6.2 the results of the interval forecasting experiments are presented and discussed and then, in Section 3.6.3, the probabilistic forecasting experiments are analyzed.

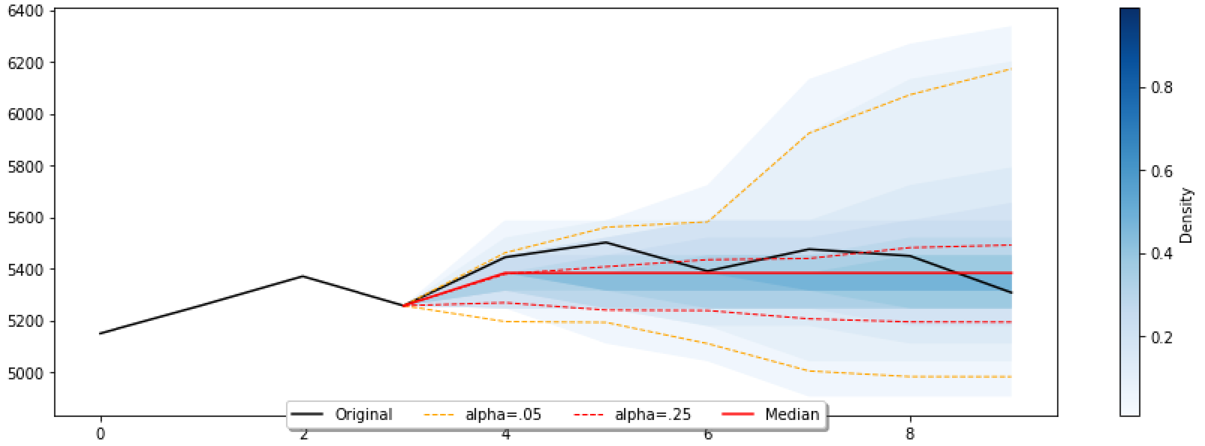


Figure 19 – Sample of EnsembleFTS performance for many steps ahead

To measure the performance of the proposed models, ARIMA, QAR, kNN/KDE and, BSTS were chosen as competitor models due to its possibility to perform interval and probabilistic forecasting for many steps ahead. The hyperparameters of each method were individually investigated and only the best model is considered in the validation of the results.

For these experiments three well known financial time series data (the TAIEX, S&P 500 and NASDAQ data sets) were selected, each of them with 5000 instances, whose descriptions and properties can be found at Appendix A. A rolling window cross-validation methodology Tashman [2000] was applied, using a working set of 1000 instances, 800 for training (80%) and 200 for testing (20%) and a sliding increment of 200 instances, totaling 23 experiments, and all measurements were performed out of sample.

Once the model fine tuning was performed for each method and the experiments were executed for each dataset, statistical tests were employed in order to compare the performance of the models. The hypothesis testing procedures adopted best practices discussed in García et al. [2010], Derrac et al. [2011], Trawiński et al. [2012]. The Friedman Aligned Ranks test Hodges and Lehmann [1962] non parametric procedure was adopted to test the equality of the means, where the null hypothesis H_0 stands for the equality of all means and the inability to distinguish between the methods and the alternative hypothesis H_1 stands for the difference of the means and the distinguishability among the models. The paired *post hoc* procedure adopted was the Finner test Finner [1993], in a one-versus-all design where the proposed methods are taken as control methods. In Finner test the null hypothesis H_0 stands for the equality between the control and the test methods and the alternative hypothesis H_1 stands for the significant difference between the control and test methods. All the tests adopted the significance level $\alpha = .05$ and were performed on STAC framework Rodríguez-Fdez et al. [2015], and all FTS methods were tested with the pyFTS library[1] Silva et al. [2018].

In order to contribute to the replication of all the results in the research, all data and source codes employed in this chapter are available at the URL: `http://bit.ly/scalable_probabilistic_fts_chap3`

3.6.1 Hyperparameter Grid Search

In order to assess the impact of the $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods hyperparameters on the accuracy, a Search Grid was performed for each benchmark dataset, using the search spaces contained in Table 10. The Winkler Score accuracy metric, where $\alpha \in \{.05, .25\}$, for each method, order, dataset and partitions can be observed in Figures 20 and 21. It is notable that $[\mathbb{I}]FTS$ has more sensitivity to the number of partitions than $W[\mathbb{I}]FTS$. This occurs because the length of the partitions is the mechanism used by $[\mathbb{I}]FTS$ method to adjust the importance of each fuzzy set. Otherwise, $W[\mathbb{I}]FTS$ uses the rule weights to balance the importance of each fuzzy set on deffuzyfication, diminishing the impact of the partition length.

Given that several numbers of partitions and order values achieved very close accuracy values, the Principle of Parsimony (or Occam's Razor) was adopted to choose the set of hyperparameters that leads to smallest number of rules $|\mathcal{M}|$, keeping the same accuracy. The chosen hyperparameters were $k = 45$ and $\Omega = 1$ and a sample of the best models performance can be seen in Figures 14 and 15.

Hyperparameter	Search space
k	$\{10, 15, 20, 25, 30, 35, 40, 45, 50\}$
Ω	$\{1, 2, 3\}$

Table 10 – Hyperparameter search spaces for IFTS and WIFTS grid search

The EnsembleFTS method has a different hyperparameter set than $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods. As a meta-model, an internal FTS model should be chosen, in addition to the k range and the Ω range. To reduce the complexity of this search space a set of four models were predefined and detailed in Table 11. The accuracy of EnsembleFTS was analyzed using both interval and probabilistic perspectives, the first one using the Winkler Score interval accuracy metric, for $\alpha \in \{.05, .25\}$, and second one using CRPS probabilistic metric. The Winkler Score for each method and dataset is shown in Figures 22, where can be observed that the Model 4 is the most stable model. The CRPS results are shown in Figure 23, where it can be observed that Model 4 again is the most stable model. The immediate conclusion is that the higher diversity of models help KDE to build a more precise probability distribution, with improved sharpness and resolution. A sample of the

[1] `https://pyfts.github.io/pyFTS/`. Access in 01/07/2018

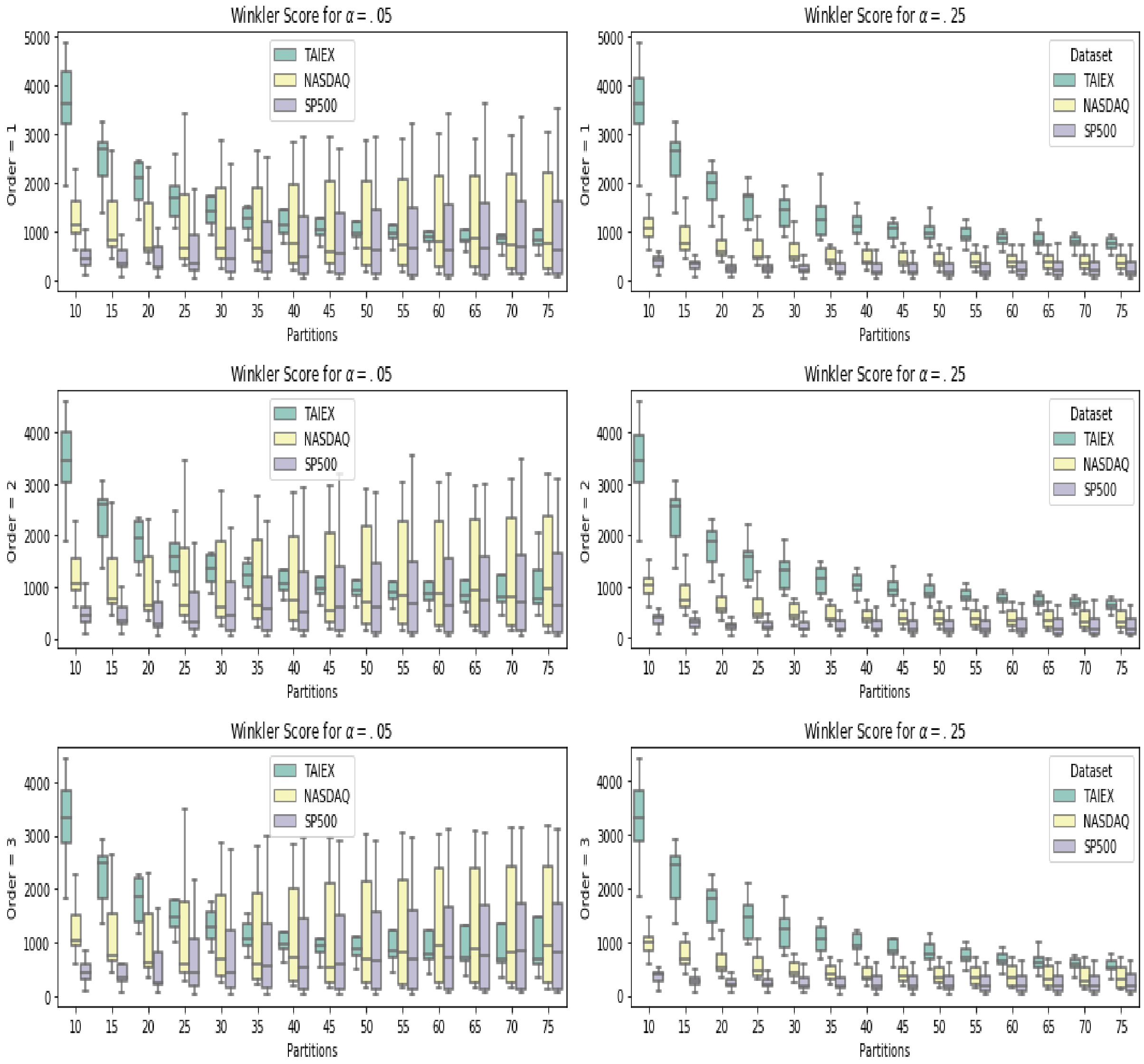


Figure 20 – IFTS Winkler Scores for $\alpha \in \{.05, .25\}$ by dataset, order and partitions

Model 4 performance can be seen in Figures 18 and 19, with $\alpha \in \{.05, .25\}$ prediction intervals and probabilistic forecasting for 7 steps ahead.

Name	Internal Model	k range	Ω range
EnsembleFTS Model 1	HOFTS	$\{10, 20, 30, 40, 50\}$	$\{1, 2, 3\}$
EnsembleFTS Model 2	HOFTS	$\{10, 15, 20, 25, 30, 35, 40, 45, 50\}$	$\{1, 2, 3\}$
EnsembleFTS Model 3	WHOFTS	$\{10, 20, 30, 40, 50\}$	$\{1, 2, 3\}$
EnsembleFTS Model 4	WHOFTS	$\{10, 15, 20, 25, 30, 35, 40, 45, 50\}$	$\{1, 2, 3\}$

Table 11 – Search spaces for Ensemble FTS grid search

3.6.2 Interval Forecasting Benchmarks

The Winkler Score Mean results for each method and dataset are presented in Table 12. The Friedman Aligned Ranks of the methods are presented in Table 13

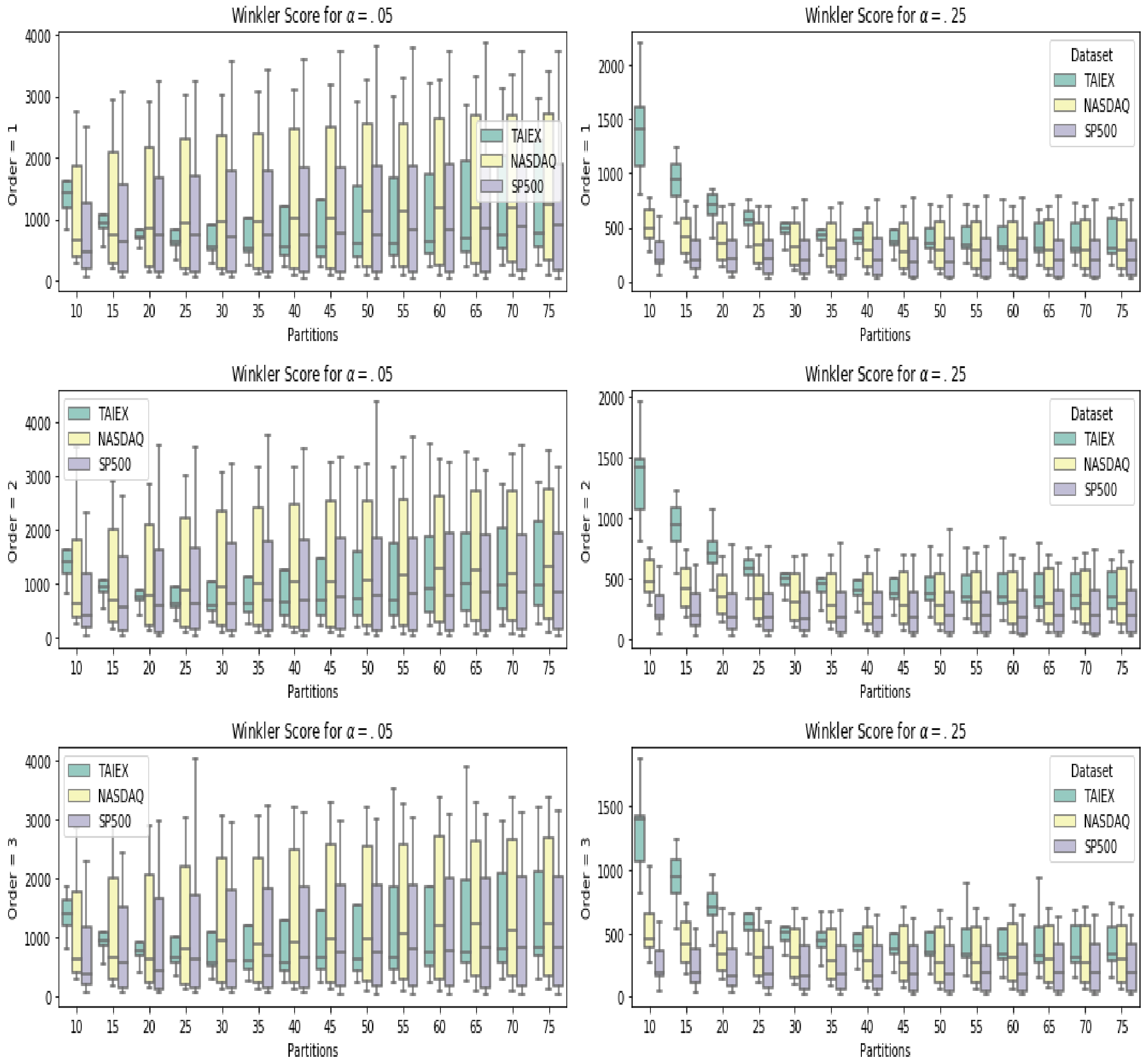


Figure 21 – WIFTS Winkler Scores for $\alpha \in \{.05, .25\}$ by dataset, order and partitions

and the test statistic for these results is $Q = 12.891861761426975$, where the p-Value is $P(\chi^2_{df} < Q) = 0.04478569463323567$, with $df = 5$ degrees of freedom. For this statistic the H_0 is rejected at the $\alpha = .05$ confidence level, indicating that there is difference between the means of the competitor models.

The *post-hoc* tests were employed using $[\mathbb{I}]FTS$, $W[\mathbb{I}]FTS$ and EnsembleFTS methods as control methods and their results are presented in Tables 14, 15 and 16, showing there is no prevalence of the methods except of $W[\mathbb{I}]FTS$ over BSTS. These results showed that $[\mathbb{I}]FTS$, $W[\mathbb{I}]FTS$ and EnsembleFTS interval forecasting methods perform satisfactorily when compared with the standard methods in the literature.

The statistical tests were employed on the one step ahead forecasts. Figure 24 shows, for each method and dataset, the impact of the forecasting horizon on the Winkler Score accuracy.

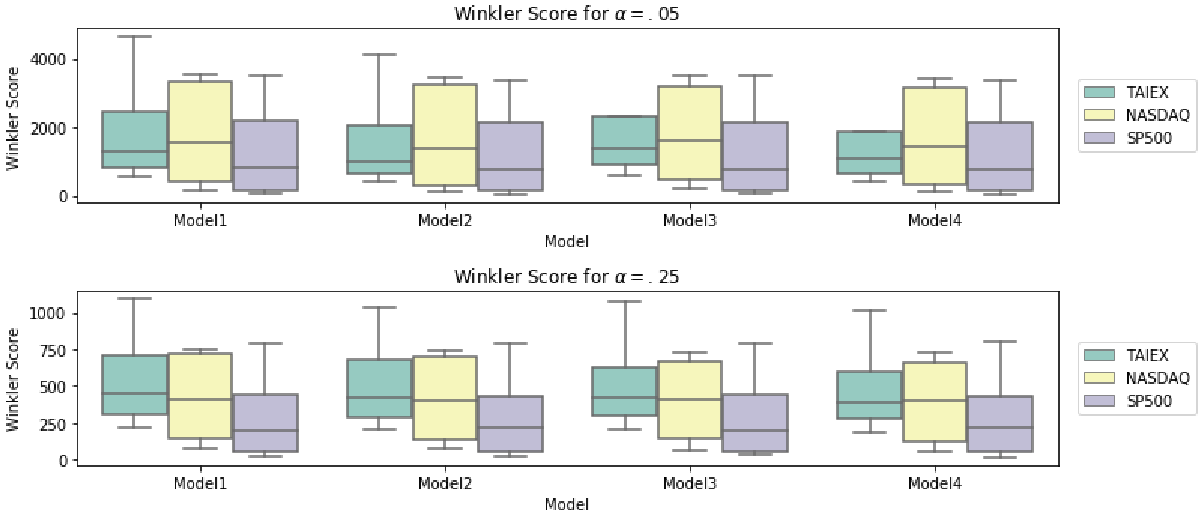


Figure 22 – Sample of IFTS and WIFTS for 7 steps ahead

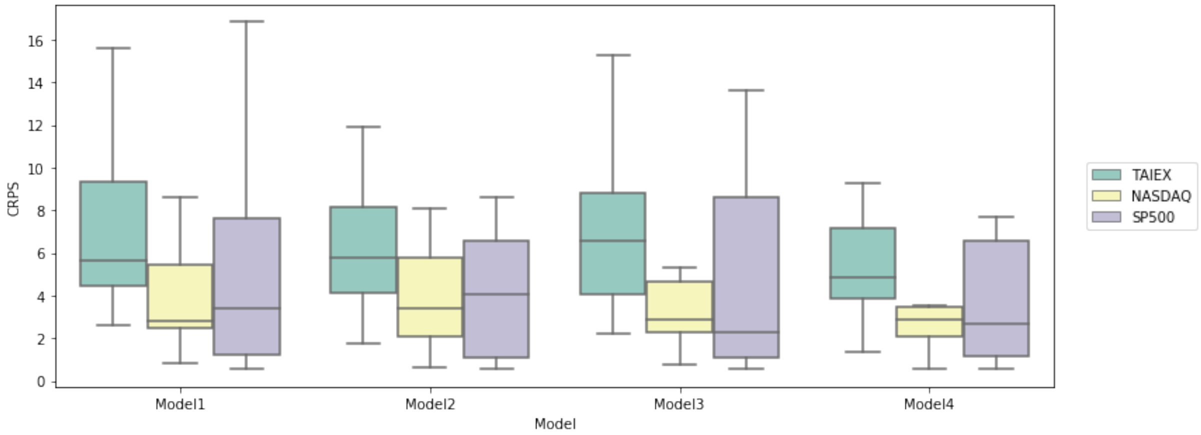


Figure 23 – CRPS response of EnsembleFTS

Dataset	ARIMA	QAR	WIFTS	IFTS	kNN	EnsembleFTS	BSTS
S&P 500	72.712 ± 135.871	121.694 ± 319.305	111.705 ± 156.013	113.516 ± 91.627	131.394 ± 166.31	268.567 ± 318.259	292.415 ± 384.499
NASDAQ	233.261 ± 486.735	106.416 ± 56.248	123.35 ± 141.251	284.692 ± 147.24	170.709 ± 156.097	603.881 ± 638.297	652.036 ± 963.624
TAIEX	858.124 ± 1337.139	340 ± 269.34	480.581 ± 561.826	917.879 ± 243.737	428.484 ± 269.459	898.531 ± 1175.107	1280.67 ± 1472.031

Table 12 – Average Winkler Score with $\alpha = .05$ for one step ahead interval forecasts

3.6.3 Probabilistic Forecasting Benchmarks

The CRPS Mean results for each method and dataset are presented in Table 17. The Friedman Aligned Ranks of the methods are presented in Table 18 and the test statistic for these results is $Q = 7.264833574529668$, where the p-Value is $P(\chi^2_{df} < Q) = 0.12253751253946543$, with $df = 4$ degrees of freedom. For this statistic the H_0 is accepted at the $\alpha = .05$ confidence level, indicating that there is no difference between the means of the competitor models. This result discards the need to employ *post-hoc* tests and shows

METHOD	RANK
QAR	5.333333
WIFTS	5.666667
kNN	6.666667
ARIMA	10.000000
IFTS	13.666667
EnsembleFTS	16.666667
BSTS	19.000000

Table 13 – Friedman aligned ranks

	COMPARISON	Z-VALUE	P-VALUE	ADJUSTED P-VALUE	Result
0	IFTS vs QAR	1.644879	0.099995	0.468540	H0 Accepted
1	IFTS vs WIFTS	1.579084	0.114317	0.468540	H0 Accepted
2	IFTS vs kNN	1.381699	0.167064	0.468540	H0 Accepted
3	IFTS vs BSTS	1.052723	0.292468	0.468540	H0 Accepted
4	IFTS vs ARIMA	0.723747	0.469221	0.532377	H0 Accepted
5	IFTS vs EnsembleFTS	0.592157	0.553746	0.553746	H0 Accepted

Table 14 – Post-hoc tests using IFTS as control method

	COMPARISON	Z-VALUE	P-VALUE	ADJUSTED P-VALUE	Result
0	WIFTS vs BSTS	2.631807	0.008493	0.049889	H0 Rejected
1	WIFTS vs EnsembleFTS	2.171241	0.029913	0.087081	H0 Accepted
2	WIFTS vs IFTS	1.579084	0.114317	0.215565	H0 Accepted
3	WIFTS vs ARIMA	0.855337	0.392364	0.526342	H0 Accepted
4	WIFTS vs kNN	0.197386	0.843526	0.892023	H0 Accepted
5	WIFTS vs QAR	0.065795	0.947541	0.947541	H0 Accepted

Table 15 – Post-hoc tests using WIFTS as control method

that there is no prevalence of one method over others. The EnsembleFTS probabilistic forecasting method performed satisfactorily when compared with the standard methods in the literature.

The statistical tests were employed on the one step ahead forecasts. Figure 25 shows, for each method and dataset, the impact of the forecasting horizon on the CRPS accuracy.

	COMPARISON	Z-VALUE	P-VALUE	ADJUSTED P-VALUE	Result
0	EnsembleFTS vs QAR	2.237036	0.025284	0.142432	H0 Accepted
1	EnsembleFTS vs WIFTS	2.171241	0.029913	0.142432	H0 Accepted
2	EnsembleFTS vs kNN	1.973855	0.048398	0.142432	H0 Accepted
3	EnsembleFTS vs ARIMA	1.315903	0.188206	0.268577	H0 Accepted
4	EnsembleFTS vs IFTS	0.592157	0.553746	0.620249	H0 Accepted
5	EnsembleFTS vs BSTS	0.460566	0.645110	0.645110	H0 Accepted

Table 16 – Post-hoc tests using Ensemble FTS as control method

Dataset	**QAR**	**kNN**	**ARIMA**	**EnsembleFTS**	**BSTS**
NASDAQ	1.028 ± 0.748	1.158 ± 0.477	1.444 ± 1.303	1.923 ± 1.416	3.208 ± 3.983
TAIEX	1.135 ± 0.613	1.229 ± 0.693	1.691 ± 1.239	1.301 ± 1.118	4.081 ± 5.306
S&P 500	1.557 ± 1.74	4.403 ± 3.261	1.216 ± 1.166	1.995 ± 2.255	3.278 ± 3.16

Table 17 – Average CRPS for one step ahead probabilistic forecasts

METHOD	RANK
QAR	3.000000
ARIMA	6.666667
kNN	8.333333
EnsembleFTS	8.666667
BSTS	13.333333

Table 18 – Friedman Test aligned ranks

3.7 Conclusion

This chapter provided a brief introduction about point forecasting uncertainties and the main kinds of probabilistic forecasting, reviewing the related literature and proposed new FTS methods for forecasting intervals and probability distributions, which were empirically assessed.

It is remarkable that point forecasts induce to overconfidence and, without uncertainty measures, point forecasts can be compared to lottery games. It is well known that all forecasting models have an irreducible uncertainty term, besides other not-known or not managed uncertainties, and sometimes this information is critical for decision makers.

The Prediction Interval forecasts allow users to assess the uncertainty, delimiting their expected bounds. Probabilistic forecasting methods assist users to know the uncertainty associated with the entire Universe of Discourse. However, these probabilistic forecasting methods can be computationally expensive and time consuming tasks. Also, the cited models were not adapted to deal with fuzzy numbers as input. The available methods in the FTS literature, at this point, are not capable to forecast prediction intervals or probability distributions.

To exploit this gap it was proposed the Interval FTS - $[\mathbb{I}]FTS$, the Weighted Interval FTS - $W[\mathbb{I}]FTS$, two new FTS approaches to bind the fuzzy uncertainty of the FTS models, and the Ensemble FTS, the first FTS approach capable of to producing probability distributions. In $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods, the prediction interval $\mathbb{I} = [\underline{l}, \overline{u}]$ contains the lower and upper bounds of all fuzzy sets involved on forecasting step, and the length of this interval measures the fuzzy uncertainty.

If $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods deal only with the fuzzy uncertainty, the Ensemble FTS method tries to represent the partitioning and ordering uncertainty to produce probabilistic forecasts. The performed computational experiments showed that the accuracy of the intervals and probability distributions provided by the proposed methods do not differ from the standard methods of the literature, showing its reliability.

3.7.1 Methods limitations

The main strength of these methods is their flexibility. These approaches can be used to extend all FTS methods to interval and probabilistic forecasting easily. However, some drawbacks still persist. $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ provide intervals, but not probability distributions, and its intervals do not carry a probabilistic uncertainty. Moreover, it is not parsimonious and is computationally expensive when compared to single FTS methods. An integrated method for point, interval and probabilistic forecasting is yet demanded.

To fix these lacks, in the next chapter a new Fuzzy Time Series method with the ability to represent epistemic and ontological uncertainty is proposed and its use for point, interval, and probabilistic forecasting is examined.

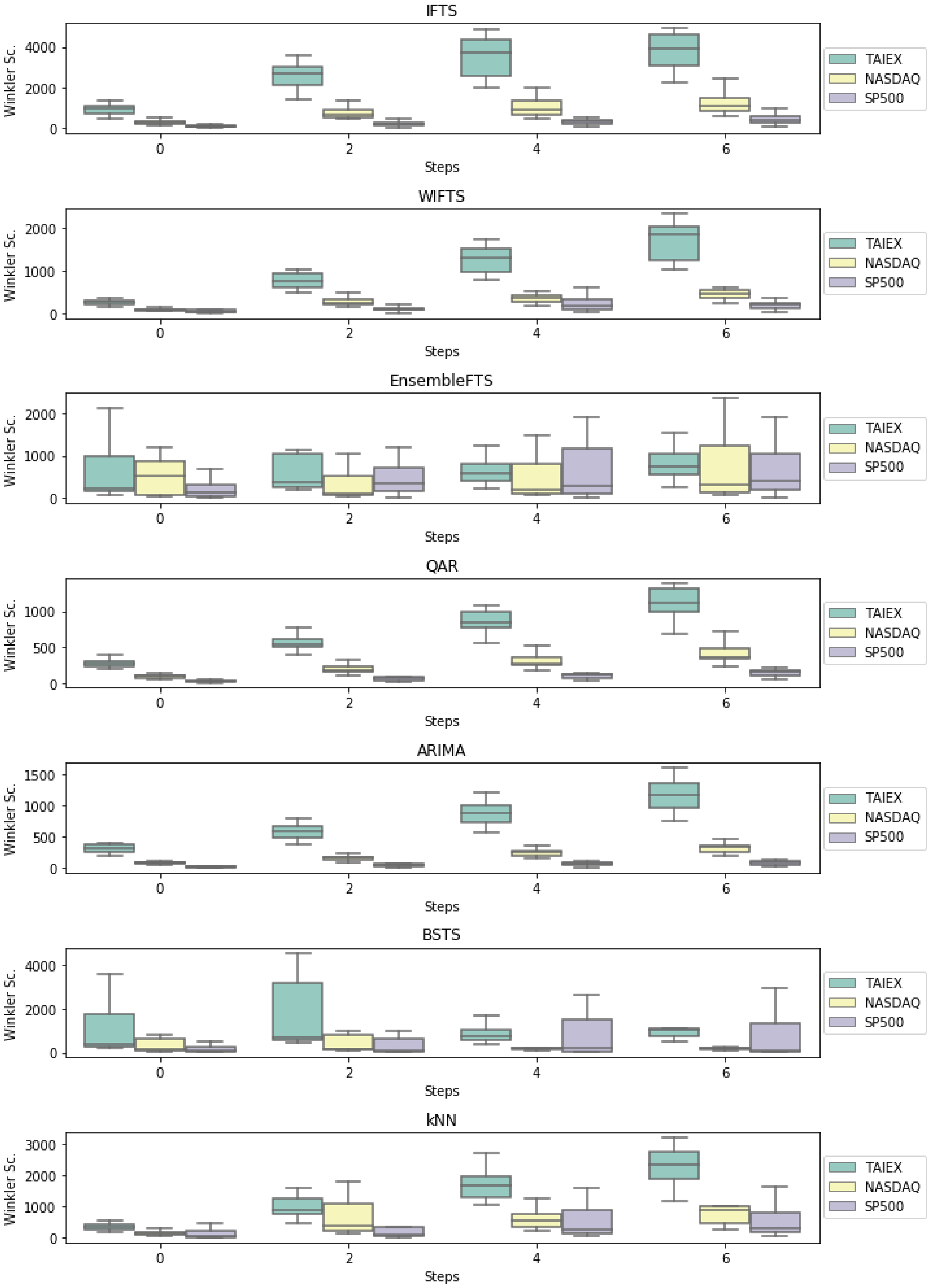


Figure 24 – Many steps ahead Winkler Score (with $\alpha = .05$) accuracy for each method

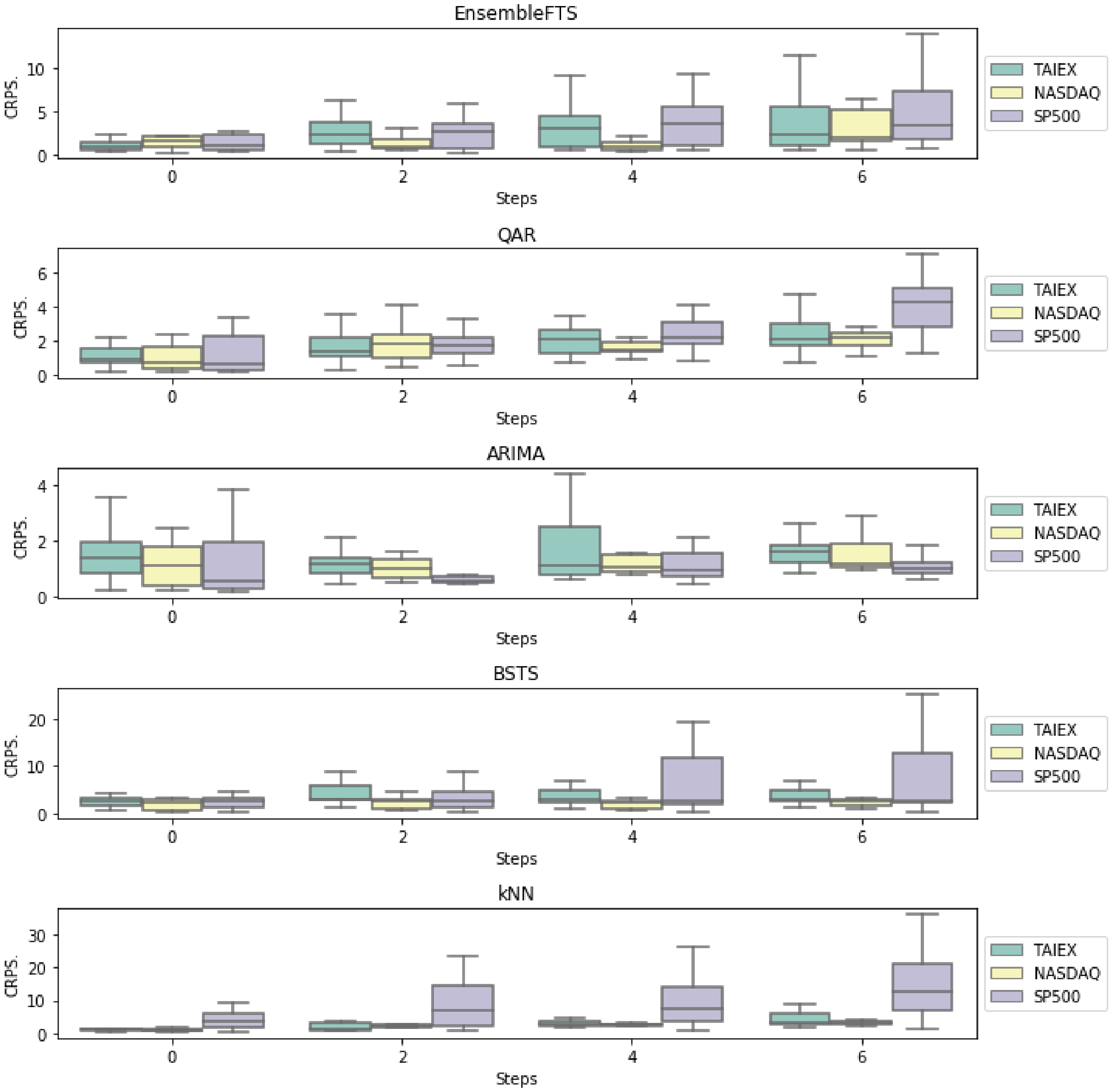


Figure 25 – Many steps ahead CRPS accuracy for each method

Chapter 4

Probabilistic Weighted Fuzzy Time Series

"Water is the softest thing, yet it can penetrate mountains and earth. This shows clearly the principle of softness overcoming hardness."
— Lao Tsu

This chapter proposes the Probabilistic Weighted Fuzzy Time Series (PWFTS) method, a new FTS method for point, interval and probabilistic forecasting for one to many steps ahead. The PWFTS method aims to produce forecasts by dealing with two sources of uncertainty: fuzzy measurements and stochastic behavior. The fuzziness is induced for a data reduction purpose, in a process that reminds a simple bin discretization. The stochastic behavior is deduced by the frequentist approach over the previous fuzzyfication.

Regarding to fuzzy time series architectural design, discussed in Chapter 2, the PWFTS method is a time invariant and heuristic method to produce probabilistic weighted rules model $\mathcal{M}$. This method embodied all explored hyperparameters but their definition involves more complex optimization methods, which will be explored in Chapter 5. Default values were defined for hyperparameters, except k and Ω which must be determined by the user, as shown in Table 19. These values, however, can be overridden by user.

Parameter	**Default Value**
Ω	User defined
k	User defined
Π	Grid
μ	triangular
α-cut	0
L	$\{1, \ldots, \Omega\}$

Table 19 – PWFTS hyperparameter default values

The option for a weighted rule knowledge model has the objective to help in the human readability and model explainability, also other knowledge extraction tasks. The weighted rule model will also help, as it will be seen in next chapters, in the distributed processing of the method and on its multivariate extension.

The model rules (the Probabilistic Weighted FLRG's) describe the most probable future behavior (the RHS - right hand side of the rule) given some past behavior (the LHS - left hand side of the rule). For a given data sample there will be many applicable rules with different activations (the membership weights) and all of them are taken into account.

The conception of this method combines the statistical approach of time series forecasting with the FTS techniques. Given some stochastic process Y, their best predictor is given by the conditional expectation $\mathbb{E}[y(t+1) \mid y(t), ...]$. The FTS methods, especially those ones based on Chen [1996], try to represent the behavior of Y process by splitting their UoD in overlapping fuzzy sets, fuzzyfying the crisp data Y to transform it on the fuzzy time series F and identifying the sequential patterns. The fuzzy sets are used to define zones, or fuzzy states, at the universe of discourse which have a common set of rules. That's what FLRGs really are: rules that describe sequential patterns.

For a given FLRG with the form $LHS \to RHS$, where $F(t-1) = LHS$ and $F(t) \in RHS$ our best predictor can be rewritten from $\mathbb{E}[F(t+1) \mid F(t), ...]$ to $\mathbb{E}[RHS|LHS]$. The weights assigned to these rules are the frequentist probabilities of the fuzzy sets, measured during the training phase.

Regarding the concepts introduced in Chapter 3, the PWFTPG can be seen as a representation of a discrete empirical probability distribution. The RHS weights represent the conditional probability $P(A_i \mid LHS)$, $\forall A_i \in RHS$ and the LHS weights represent the unconditional *a priori* probabilities of the fuzzy sets. With the midpoints of each fuzzy set and their probabilities it is possible then to compute the conditional expectation as a forecast for $F(t+1)$.

In the next sections this mechanism is detailed, starting in Section 4.1 which discusses the basics of the fuzzy empirical probabilities. In Section 4.2 the training procedure for first order model is presented, and in Section 4.3 the one step ahead method for probabilistic, interval and point forecasting is presented. Section 4.4 presents extensions for high-order models and many steps ahead forecasting. In Section 4.5 computational experiments are performed to assess the performance of the model and finally, in Section 4.6, the main features of the proposed method are summarized.

4.1 Fuzzy Empirical Probabilities

The core concept of PWFTS is the fuzzy empirical probability, used to compute the weights of the model, whose intuition is discussed in this section. The initial Zadeh's

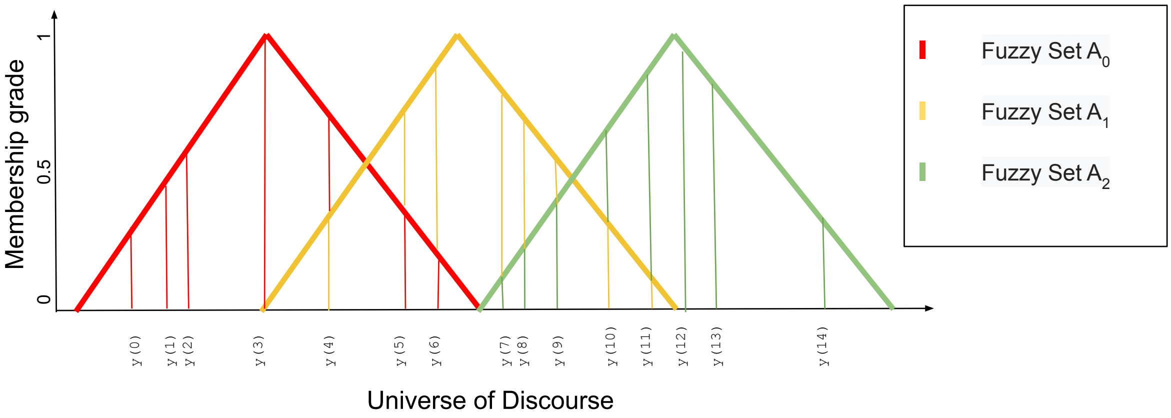


Figure 26 – Fuzzy frequencies of each $y(t) \in Y$ used to approximate the fuzzy empirical probabilities $P(A_i)$ for each fuzzy set A_i

proposition of fuzzy probability, $P(A) = E[\mu_A]$, proposed in Zadeh [1968], demands the previous knowledge of the probability distribution over the universe of discourse. Since this distribution for Y is unknown, an empirical distribution must take place. The simplest definition of empirical probability is the relative frequency of a discrete value or of a range of continuous values. Fuzzy Theory provides a different look at traditional Probability Theory because it affects the way the events are counted.

On fuzzy sets the notion of event is more complex because the same value can belong to several sets with different degrees of membership, as shown in Figure 26. In that case, instead of accounting the integral (i.e. unary) occurrence of the event, their partial occurrence is accounted using the membership value. This method is known by fuzzy frequency, and was firstly developed in Luo and Bridges [2000]. A related formulation can be found in Perfilieva [2006], with the concept of F-Transform, which decomposes the original domain of the time series into fuzzy frequencies over the fuzzy sets. This decomposition can also recreate the time series using the inverse transform.

Given the sample space U and the fuzzy sets $\tilde{A}$ over U, the partition function Z_{A_j}, $\forall A_j \in \tilde{A}$, is the integral of the membership function μ_{A_j} over the sample space U, such that $Z_{A_j} = \int_U \mu_{A_j}(y)dy$ or the discrete approximation $Z_{A_j} = \sum_{y \in U} \mu_{A_j}(y)$. With Z_{A_j} it is possible to approximate the empirical probability of a fuzzy set $A_j \in \tilde{A}$ as the sum of its memberships $\mu_{A_j}(y)$, $\forall y \in U$ divided by the sum of the partition functions Z_{A_j} of all fuzzy sets $A_j \in \tilde{A}$, as presented in Equation (4.1).

$$P(A_j) = \frac{\sum_{y \in U} \mu_{A_j}(y)}{\sum_{A_j \in \tilde{A}} Z_{A_j}} \tag{4.1}$$

The intuition behind this equation is that the empirical probability $P(A_j)$ is evenly spread over the shape of the fuzzy membership function μ_{A_j}, and the point y is a slice of this shape whose area is equal to the value $\mu_{A_j}(y)$ (as shown in Figure 26), and the area of μ_{A_j} is Z_{A_j}. $P(A_i)$ is measured from a sample of Y, and this empirical value is an approximation of the true (but unknown) probability. This approximation is used in (4.2) to approximate the conditional probability of a value $y \in U$, given a fuzzy set $A_j \in \tilde{A}$:

$$P(y|A_j) = P(A_j) \cdot \frac{\mu_{A_j}(y)}{Z_{A_j}} \tag{4.2}$$

Using (4.1) and (4.2) and the Law of Total Probability the empirical probability $P(y)$ can be approximated using the Equation (4.3).

$$P(y, \tilde{A}) = \sum_{A_j \in \tilde{A}} P(y|A_j) \cdot P(A_j) \tag{4.3}$$

The advantage of this approach is the convenience to obtain $P(A_j)$ from a sample of the time series dataset Y. The accuracy of $P(A_j)$ is determined mainly by k, the number of partitions of the universe of discourse U.

4.2 PWFTS Training procedure

The training procedure is a seven step method to learn the temporal dynamics of the time series training data Y and represent it on a fuzzy-probabilistic model, namely the Probabilistic Weighted Fuzzy Temporal Pattern Group - PWFTPG. The steps of the method are listed below :

Step 1 *Define the universe of discourse*: Define U as the sample space of in-sample training data Y, such that $U = [\min(Y) - D_1, \max(Y) + D_2]$, where $[\min(Y), \max(Y)]$ is the range of in-sample data and D_1 and D_2 are numbers used to extrapolate this range, for instance $D_1 = 0.1 \cdot \min(Y)$ and $D_2 = 0.1 \cdot \max(Y)$;

Step 2 *Partitioning*: split U in k even length intervals u_i, for $i = 1, \ldots, k$, with midpoints mp_i;

Step 3 *Define the linguistic variable* $\tilde{A}$: Create k overlapping fuzzy sets A_j, with membership functions μ_{A_j}, related to an interval u_j, and midpoints mp_j. Each fuzzy set $A_j \in \tilde{A}$ is a linguistic term of the linguistic variable $\tilde{A}$;

Step 4 *Fuzzyfication*: Transform the original numeric time series Y into the FTS F, whose each data point $f(t) \in F$ is a k-tuple with the membership value of $y(t)$ with respect to each linguistic term $A_j \in \tilde{A}$, such that:

$$f(t) = [\mu_{A_1}(y(t)),\ \mu_{A_2}(y(t)),\ \ldots,\ \mu_{A_k}(y(t))] \tag{4.4}$$

Step 5 *Generate the FTP set*: The Fuzzy Temporal Pattern - FTP[1] is a fuzzy rule with format $A_i \rightarrow A_k$ that indicates a temporal succession where the precedent (or the Left Hand Side - LHS) is $A_i \in f(t)$ and the consequent (or the Right Hand Side - RHS) is $A_k \in f(t{+}1)$, for each possible pair of $A_i \times A_k$ of membership values greater than zero, i.e., $\{A_i \rightarrow A_k\}$ $\forall A_i \in f(t) \mid \mu_{A_i}(y(t)) > 0$ and $\forall A_k \in f(t+1) \mid \mu_{A_k}(y(t+1)) > 0$. . Therefore, $A_i \rightarrow A_k$ can be read as "IF $y(t)$ is A_i THEN $y(t+1)$ is A_k". As each $f(t) \in F$ is a sparse k-vector of membership values, there will be many possible fuzzy sets combinations of two sequential vectors $f(t)$ and $f(t+1)$. Then for each sequential pair on F possibly more than one FTP will be generated;

Step 6 *Generate the FTPG set*: A Fuzzy Temporal Pattern Group - FTPG[2] represents the set of all FTPs with the same LHS and the union of their RHS, with the format $A_i \rightarrow A_k, A_j, ...$, where the LHS is $f(t) = A_i$ and the RHS is $f(t+1) \in \{A_k, A_j, ...\}$. Each FTPG can be understood as the set of possibilities which may happen at time $t+1$ (the consequent) when a certain set A_i is identified at time t (the precedent).

Step 7 *Calculate empirical probabilities*: The Probabilistic Weighted FTPG - PWFTPG adds weights on the LHS and the RHS that measure their fuzzy empirical probabilities. Each PWFTPG has the format $\pi_j \cdot A_j \rightarrow w_{j0} \cdot A_0, ..., w_{jk} \cdot A_k$ for $j = 1, \ldots, k$. The set of all PWFTPG, shown in Equation (4.5), form the model $\mathcal{M}$. Its size depends on the number of partitions k, and it could be represented in matrix form but the weights w_{ij} form a very sparse matrix, which justifies using optimized data structures for its representation.

$$\begin{array}{ccc} \pi_1 \cdot A_1 & \rightarrow & w_{11} \cdot A_1, ..., w_{1k} \cdot A_k \\ \ldots & \ldots & \ldots \\ \pi_k \cdot A_k & \rightarrow & w_{k1} \cdot A_1, ..., w_{kk} \cdot A_k \end{array} \tag{4.5}$$

Each weight π_j is associated with the fuzzy set in the LHS of the rule, and it is the normalized sum of all LHS memberships of all FTPs where the LHS is the fuzzy set A_j, as in Equation (4.1). π_j can be understood as the empirical *a priori* probability of the fuzzy set A_j independent of time, or $P(A_j)$, such that the condition of Equation (4.6) must be satisfied for the PWFTPG set in Equation (4.5).

$$\sum_{j \in \tilde{A}} \pi_j = 1 \tag{4.6}$$

Each weight w_{ji} is associated with a fuzzy set A_i on the RHS of the FTP whose the LHS is A_j, and it is the normalized sum of all RHS memberships of all FTPs

[1] This nomenclature is adopted in replacement of Fuzzy Logical Relationships (FLR) used in Song and Chissom [1993b], to avoid misunderstandings with the terms "logical" and "relationship" with their classical meanings in fuzzy theory literature.

[2] In replacement of Fuzzy Logical Relationship Group - FLRG used in Chen et al. [2006].

where $LHS = A_j$ and $RHS = A_i$. Therefore, the weight w_{ji} can be understood as the empirical conditional probability of the fuzzy set A_i on time $t+1$ when the fuzzy set A_j is identified on time t, or $P(A_i^{t+1} \mid A_j^t)$, such that the condition of Equation (4.7) must be satisfied for each A_j in LHS.

$$\sum_{i \in \tilde{A}} w_{ji} = 1 \quad \forall A_j \in \tilde{A} \tag{4.7}$$

The outcome of the Training Procedure is the PWFTPG set, whose simple example can be seen in Figure 27, and it represents the temporal dynamics of the original data. It is an empirical probability distribution of the linguistic variable A over the time series Y with sample space U, where each rule contains the unconditional probability $P(A_j) = \pi_j$ and conditional probabilities $P(A_i|A_j) = w_{ji}$, for $A_i, A_j \in \tilde{A}$, as illustred in Figure 28.

$$\begin{array}{rcrrr}
0.005 \cdot A0 & \rightarrow & 0.4 \cdot A0, & 0.6 \cdot A1 & \\
0.05 \cdot A1 & \rightarrow & 0.05 \cdot A0, & 0.6 \cdot A1, & 0.35 \cdot A2 \\
0.11 \cdot A2 & \rightarrow & 0.1 \cdot A1 & , 0.6 \cdot A2, & 0.3 \cdot A3 \\
0.14 \cdot A3 & \rightarrow & 0.15 \cdot A2, & 0.6 \cdot A3, & 0.25 \cdot A4 \\
0.15 \cdot A4 & \rightarrow & 0.2 \cdot A3, & 0.55 \cdot A4, & 0.25 \cdot A5 \\
0.1 \cdot A5 & \rightarrow & 0.2 \cdot A4, & 0.55 \cdot A5, & 0.25 \cdot A6 \\
0.12 \cdot A6 & \rightarrow & 0.2 \cdot A5, & 0.6 \cdot A6, & 0.2 \cdot A7 \\
0.09 \cdot A7 & \rightarrow & 0.25 \cdot A6, & 0.55 \cdot A7, & 0.2 \cdot A8 \\
0.06 \cdot A8 & \rightarrow & 0.25 \cdot A7, & 0.6 \cdot A8, & 0.15 \cdot A9 \\
0.02 \cdot A9 & \rightarrow & 0.6 \cdot A8, & 0.4 \cdot A9 &
\end{array}$$

Figure 27 – Example of PWFTPG model generated with $k = 10$ and a random Y dataset

4.3 Forecasting Procedure

The forecasting procedure is a four step procedure listed in this section, which takes as input the forecasting type, a sample $y(t) \in U$ and uses the PWFTPG model $\mathcal{M}$ learned in the previous section to generate the output, which depends on the type of forecasting (probabilistic, interval or point forecasting). The complete forecasting procedure is presented below:

Step 1 *Fuzzyfication*: For a given input value $y(t) \in Y$, find the fuzzyfied values $f(t) = \{A_j \mid \mu_{A_j}(y(t)) > \alpha\}$;

Step 2 *Pattern matching*: Locate all the PWFTPG's whose the LHS is $f(t)$.

Step 3 *Forecast*: The distribution of $f(t+1)$ is given by the RHS sets of each PWFTPG matched;

Step 4 *Defuzzyfication*:

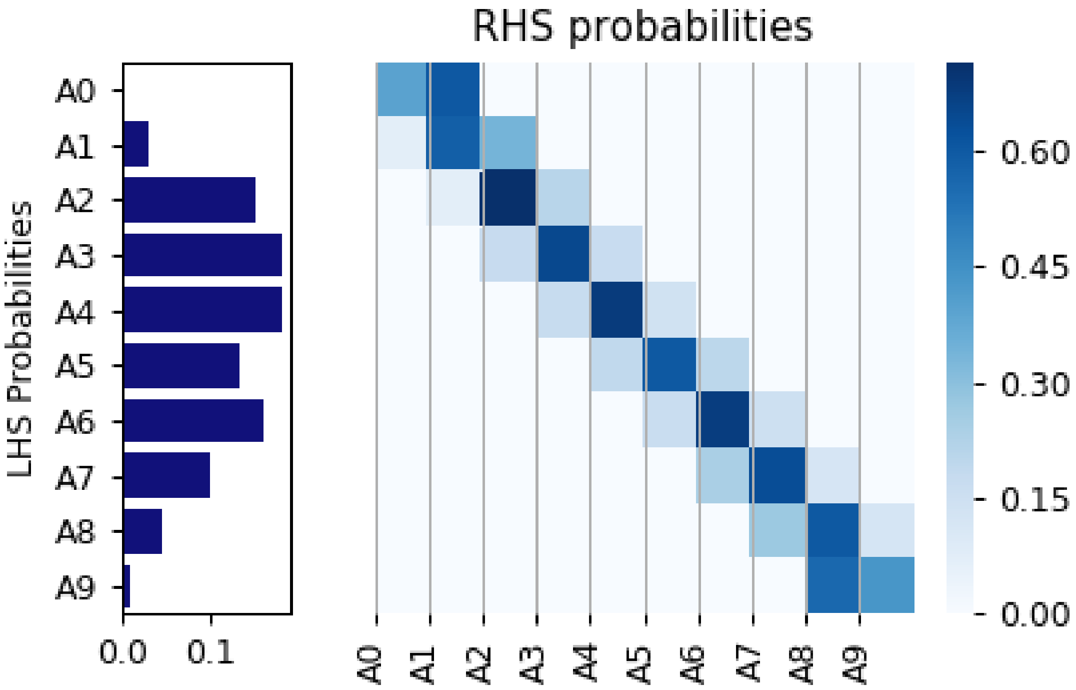


Figure 28 – Probability distributions representation of a PWFTPG model generated with $k = 10$ and a random Y dataset, where π_j weights represent the LHS empirical probabilities and w_{ji} weights represent the RHS empirical probabilities.

a) If the forecasting type is Probabilistic, then build the probability distribution $P(y(t+1)|y(t))$, $\forall y(t+1) \in U$ applying Equation (4.9) presented in Section 4.3.1;

b) If the forecasting type is Interval, then build the prediction interval $\mathbb{I}(t+1)$ applying Equation (4.12) presented in Section 4.3.2;

c) If the forecasting type is Point, then build the crisp estimate $\hat{y}(t+1)$ applying Equation (4.14) presented in Section 4.3.3;

4.3.1 Probabilistic Forecasting Procedure

A probability distribution $P(y(t+1)|y(t))$, for all $y(t+1) \in U$ can be computed using a Mixture Distribution approach to transform each PWFTPG probability into a continuous distribution, as described in Equation (4.9).

A mixture distribution is defined as $P(y) = \sum \omega_j \cdot \pi_j(y)$ where $\pi_j : U \to [0,1]$ are specific PDFs and ω_j is a weight associated to each PDF, such that $\sum \omega_j = 1$. Given an input value $y(t) \in Y$ and the PWFTPG set, the probability distribution for each $y(t+1) \in U$ is given by (4.9), where ω_j is replaced by the probability $P(y(t)|A_i)$, the LHS probability given the input value, and the distribution π_j is replaced by the probability $P(y(t+1)|A_j, A_i)$, $\forall A_j \in RHS$.

Looking back to Equation (4.2), it is clear that $\sum_{A_j \in \tilde{A}} P(y(t)|A_j) < 1$, once $P(A_i)$ is the probability of the whole fuzzy set and $y(t)$ is just a small slice of it, thus it does not comply with the $\sum \omega_j = 1$ restriction of the mixture distribution. To work around this issue $P(y(t)|A_j)$ is re-scaled using Equation (4.8).

$$\frac{P(y(t)|A_j)}{\sum_{A_j \in \tilde{A}} P(y(t)|A_j)} \tag{4.8}$$

Therefore, the final conditional probability $P(y(t+1)|y(t))$, given the linguistic variable $\tilde{A}$ and a PWFTPG set $\mathcal{M}$, is defined in Equation (4.9) and illustrated in Figure 29. A sample of the PWFTS probabilistic forecasting for one step ahead can be seen in Figures 30 and 31.

$$\begin{aligned} P(y(t+1)|y(t)) &= \sum_{A_j \in \tilde{A}} \frac{P(y(t)|A_j) \left(\displaystyle\sum_{i=1}^{k} P(y(t+1)|A_i, A_j) \right)}{\displaystyle\sum_{i=1}^{k} P(y(t)|A_i)} \\ &= \sum_{A_j \in \tilde{A}} \frac{\pi_j \dfrac{\mu_{A_j}(y(t))}{Z_{A_j}} \left(\displaystyle\sum_{i=1}^{k} w_{ji} \dfrac{\mu_{A_i}(y(t+1))}{Z_{A_i}} \right)}{\displaystyle\sum_{i=1}^{k} \pi_i \dfrac{\mu_{A_i}(y(t))}{Z_{A_i}}} \end{aligned} \tag{4.9}$$

4.3.2 Interval forecasting procedure

A forecasting interval $\mathbb{I}(t+1)$ can be produced from $P(y(t+1)|y(t))$, given that it is possible to build a cumulative density function $F(y(t+1)|y(t))$ and use it to construct the quantile function $Q(\tau) : [0,1] \to U$, as shown in Equation (4.10) where $\tau \in [0,1]$ is the desired quantile. Then, for a certain confidence level $\alpha \in [0,1]$, it is possible to compute an inter quantile interval $\mathbb{I}_f = [\underline{Q(\alpha)}, \overline{Q(1-\alpha)}]$.

$$Q(\tau) = \min\{x \in U \mid F(x|y(t)) = \tau\} \tag{4.10}$$

However, the above method demands the previous computation of the whole probability density function $P(y(t+1)|y(t))$, which is computationally expensive for larger input samples. A simpler and faster heuristic for generating prediction intervals extends the method $W[\mathbb{I}]FTS$, proposed in Section 3.3.2, to exploit the structure of the PWFTPG weights. Each PWFTPG will be represented by an interval $\mathbb{I}$ whose bounds are the expectation of the bounds of its RHS fuzzy sets, such that $\underline{A_j}$ and $\overline{A_j}$ represent the

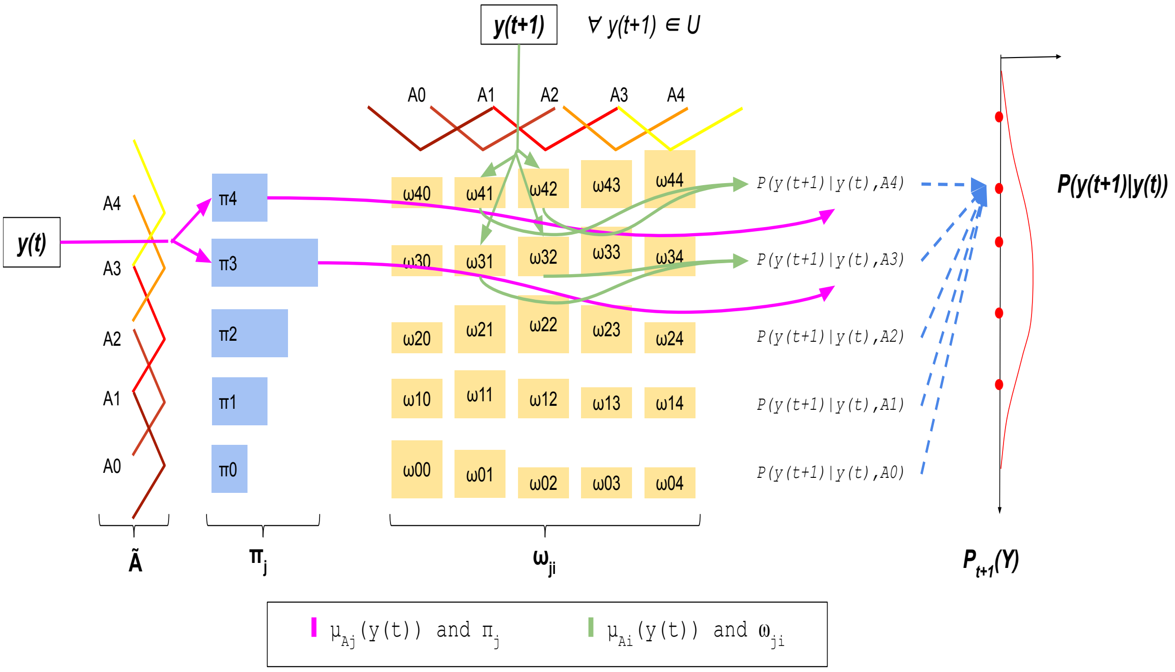


Figure 29 – Representation of PWFTS probabilistic forecasting procedure, where the length of blue boxes represents the magnitude of π_j weights and the height of yellow boxes represents the magnitude of ω_{ji} weights.

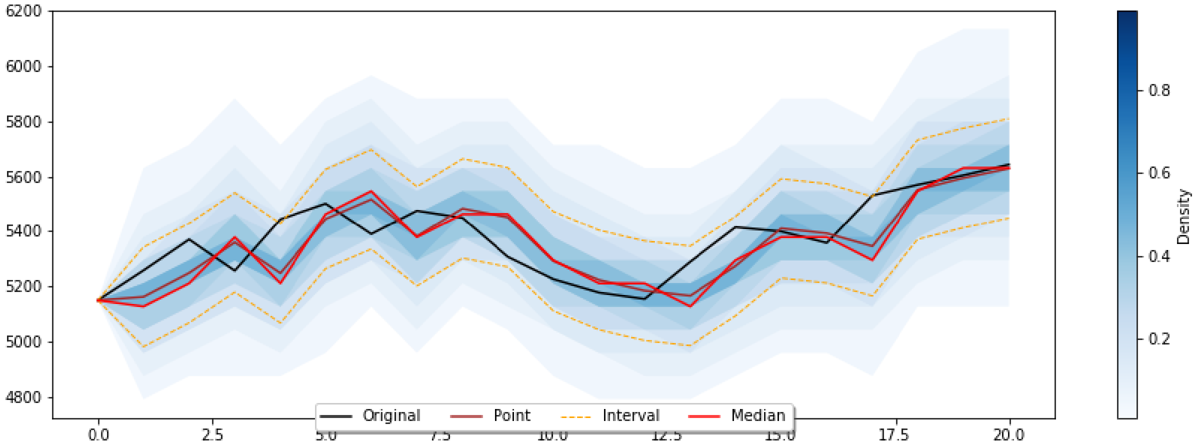


Figure 30 – Sample of PWFTS for one step ahead forecasting

lower and upper bounds of the fuzzy set A_j, and $\mathbb{E}[A_j]$ is the expectation of the PWFTPG where the LHS is A_j. The forecasting interval $\mathbb{I}(t+1)$ then is the sum of these expectations weighted by the $P(y(t)|A_j)$ probabilities, as presented in Equation (4.12). A sample of the PWFTS interval forecasting for one step ahead can be seen in Figure 30.

$$
\begin{aligned}
\mathbb{I}_j &= [\underline{\mathbb{E}[A_j]}\ ,\ \overline{\mathbb{E}[A_j]}] \\
\underline{\mathbb{E}[A_j]} &= \textstyle\sum_{A_i \in A_j^{RHS}} w_{ji} \cdot \underline{A_i} \\
\overline{\mathbb{E}[A_j]} &= \textstyle\sum_{A_i \in A_j^{RHS}} w_{ji} \cdot \overline{A_i}
\end{aligned}
\tag{4.11}
$$

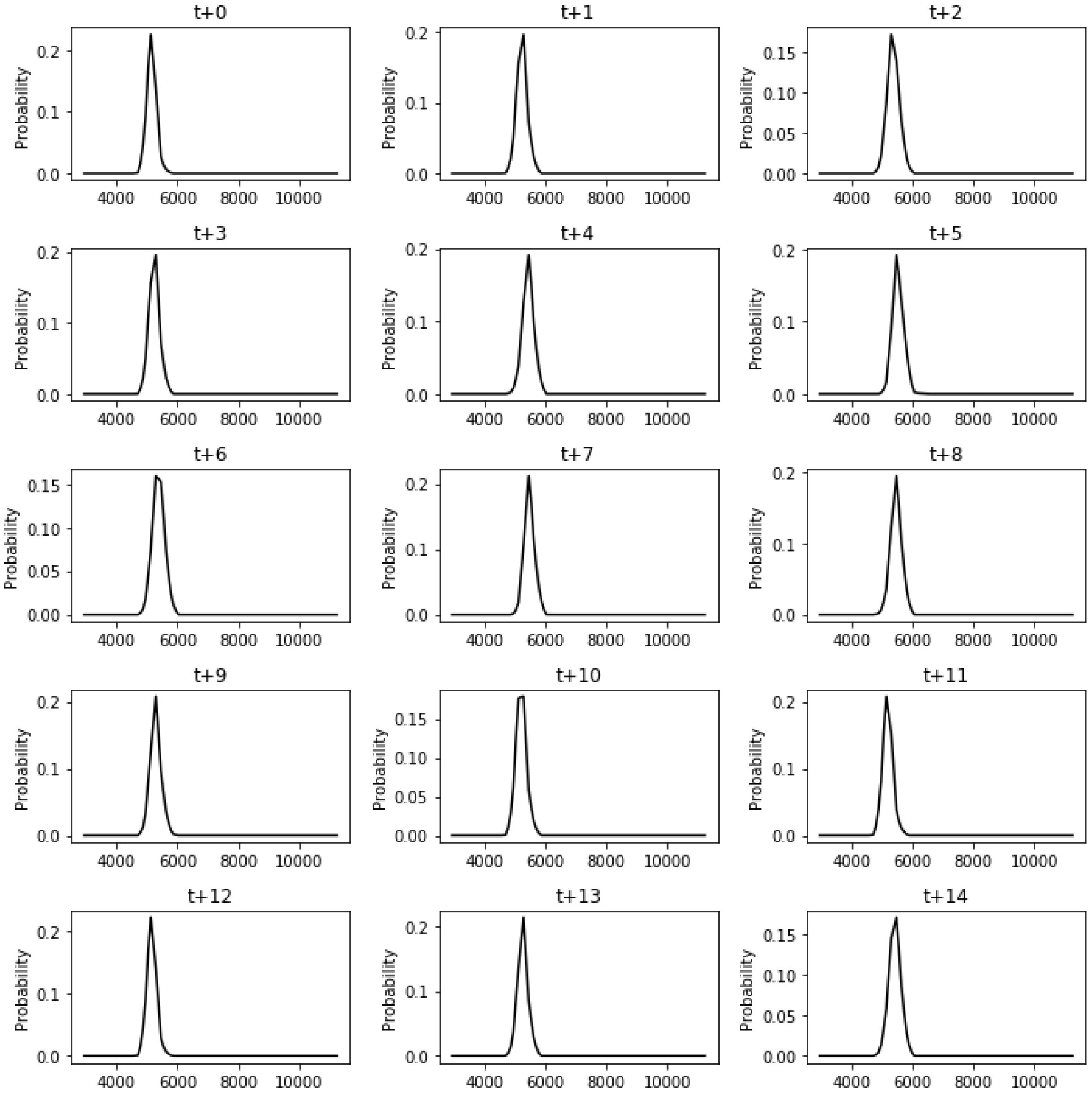


Figure 31 – Shapes of PWFTS probability distributions for one step ahead forecasting

$$
\begin{aligned}
\mathbb{I}(t+1) &= [\underline{\mathbb{E}[\tilde{A}|y(t)]}\ ,\ \overline{\mathbb{E}[\tilde{A}|y(t)]}] \\
&= \left[\frac{\sum_{A_j\in\tilde{A}} P(y(t)|A_j)\cdot\underline{\mathbb{I}_j}}{\sum_{A_j\in\tilde{A}} P(y(t)|A_j)}, \frac{\sum_{A_j\in\tilde{A}} P(y(t)|A_j)\cdot\overline{\mathbb{I}_j}}{\sum_{A_j\in\tilde{A}} P(y(t)|A_j)}\right]
\end{aligned}
\tag{4.12}
$$

4.3.3 Point forecasting procedure

To produce point forecasts $\hat{y}(t+1)$ from the existing distribution $P(\cdot|y(t))$ it is only needed to apply the expectation operator, such that $\hat{y}(t+1) = \mathbb{E}[P(y(t+1)|y(t))]$. This is also computationally expensive due to the computation of $P(y(t+1)|y(t))$. A simple heuristic for producing point forecasts is to compute the expectation $\mathbb{E}[A_j]$ of each

PWFTPG, as presented in Equation (4.13), where mp_i is the midpoint of each fuzzy set $A_i \in RHS$. The expectation $\mathbb{E}[A_j]$ for each PWFTPG A_j is constant and can be pre-computed. The final forecast $\hat{y}(t+1)$ then is the sum of these expectations weighted by $P(y(t)|A_j)$ probability, as shown in Equation (4.14). A sample of the PWFTS point forecasting for one step ahead can be seen in Figure 30.

$$\mathbb{E}[A_j] = \sum_{i \in A_j^{RHS}} w_{ji} \cdot mp_i \tag{4.13}$$

$$\hat{y}(t+1) = \mathbb{E}[\tilde{A}|y(t)] = \sum_{A_j \in \tilde{A}} \frac{P(y(t)|A_j) \cdot \mathbb{E}[A_j]}{\sum_{A_j \in \tilde{A}} P(y(t)|A_j)} \tag{4.14}$$

4.4 PWFTS extensions

In the next subsections the basic first-order and one-step-ahead method is extended to higher orders and wider forecasting horizons in order to increase PWFTS method flexibility and versatility.

4.4.1 Many steps ahead forecasting

The forecasting procedures listed in Section 4.3 are one step ahead methods. To extend the forecasting procedures to many steps ahead forecasting, an iterative approach is adopted, in which the $t+1$ step is computed with the previously presented methods and its output is fed back as input to the next H steps. From the step $t+2$ on, let $h \in [t+2, t+H]$ be the new time indexer. The simpler approach is to perform the point forecast of $y(h+1)$ with the input $y(h)$.

The interval procedure requires a few more modifications. Given the input $\mathbb{I}(h)$ the same interval forecasting procedure will be executed with inputs $\underline{\mathbb{I}(h)}$ and $\overline{\mathbb{I}(h)}$ producing two new intervals $\underline{\mathbb{I}(h+1)}$ and $\overline{\mathbb{I}(h+1)}$. Then the final forecasting interval will be $\mathbb{I}(h+1) = [\min\{\underline{\mathbb{I}(h+1)}\}, \max\{\overline{\mathbb{I}(h+1)}\}]$.

Finally, the probabilistic forecasting for $P(y(h+1)|y(h))$ given the input will change to Equation (4.15), instead of Equation (4.9). This equation replaces $P(y(h)|A_j)$ for the previous probability distribution $P(y(h)|y(h-1))$, as illustrated in Figure 32. A sample of the PWFTS many steps ahead forecasting can be seen in Figure 33.

$$P(y(h+1)|y(h)) = \sum_{A_j \in \tilde{A}} \frac{P(y(h)|y(h-1), A_j)}{\sum_{i=1}^{k} P(y(h)|y(h-1), A_i)} \times \left(\sum_{z=1}^{k} P(y(h+1)|A_z, A_j) \right) \tag{4.15}$$

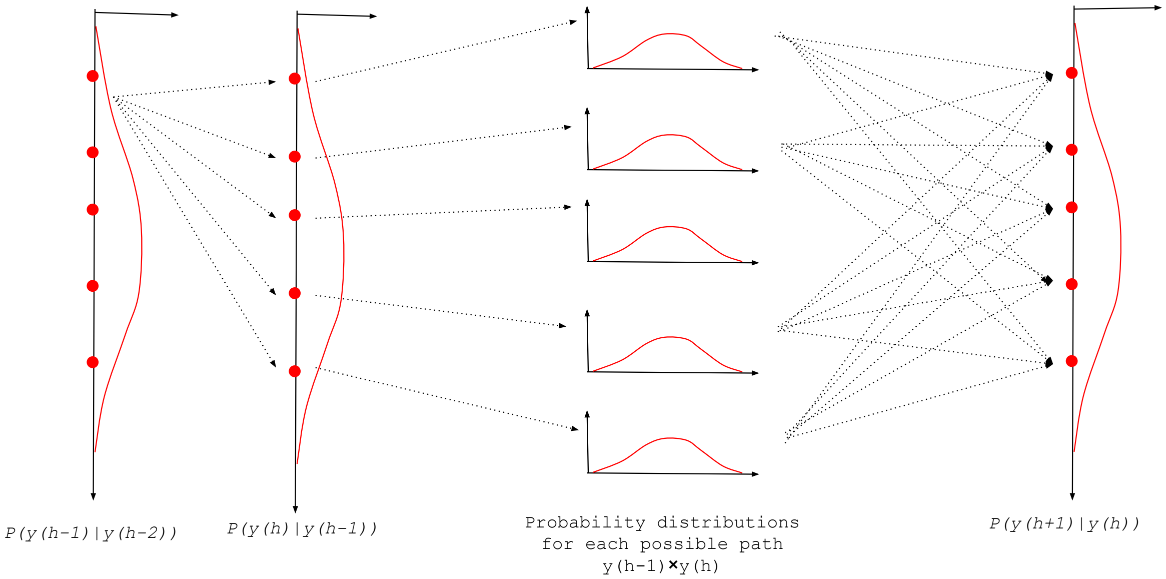


Figure 32 – Many steps ahead probabilistic forecasting process

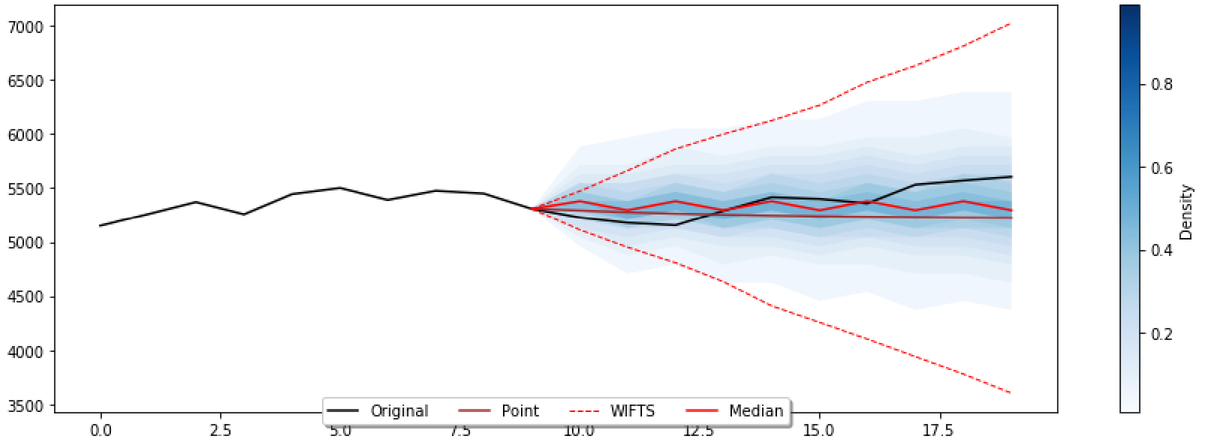


Figure 33 – Sample of PWFTS for 7 step ahead forecasting

4.4.2 High order models

The PWFTS method described in Section 4.2 is a first order method, i.e., it just needs $y(t)$ to forecast $\hat{y}(t+1)$, while high order models use Ω time lags, whose indexes are stored on vector L. To extend the standard approach to high order modification in Step 5 of the Training procedure is needed to adapt the FTPs and FTPGs to store Ω fuzzy sets on their LHS.

Once the fuzzyfied value $f(t)$ has multiple fuzzy sets (with different membership values greater than α-cut), a set of fuzzyfied values $f(t-L(\Omega)), ..., f(t-L(0))$ must be represented with all possible combinations between the fuzzy sets of each lag, such as $f(t-L(\Omega)) \times f(t-L(\Omega-1)) \times \ldots \times f(t-L(0))$, where × represents the Cartesian Product operator.

In Step 5 the FTPs will have the format $A_j^{L(\Omega)}, A_j^{L(\Omega-1)}, \ldots, A_j^{L(0)} \rightarrow A_i$, which

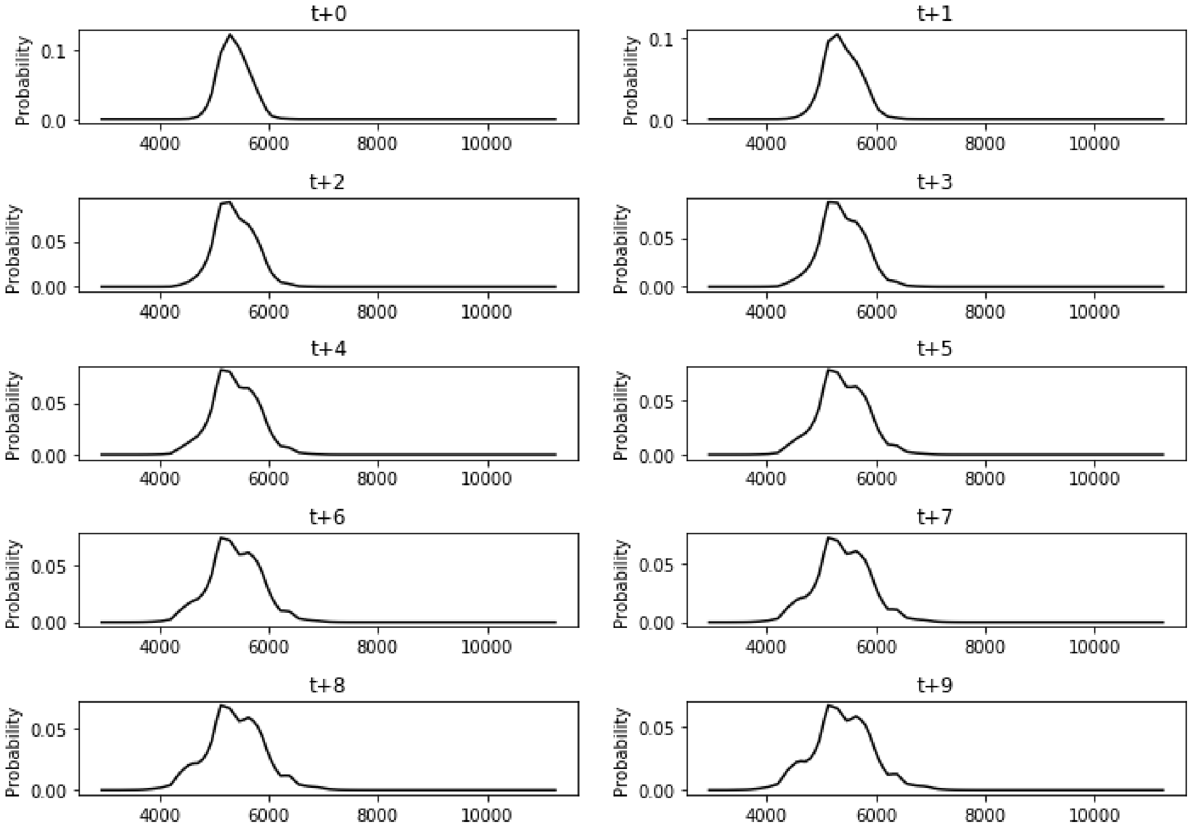


Figure 34 – Shapes of PWFTS probability distributions for many steps ahead forecasting

can be read as "IF $f(t - L(\Omega))$ is $A_j^{L(\Omega)}$ AND $f(t - L(\Omega - 1))$ is $A_j^{L(\Omega-1)}$ AND ... AND $f(t - L(0))$ is $A_j^{L(0)}$ THEN $f(t + 1)$ is A_i". In Step 6, the high order FTPGs gather all high order FTPs with the same LHS.

In Step 7 the π_j weight is replaced by π_{LHS} that aggregates the μ_{LHS} memberships of each FTPG for the samples. Given a sample $y(t - \Omega), \ldots, y(t) \in Y$ with Ω lags, their membership grades with an FTPG is the product T-norm between all memberships of the LHS:

$$\mu_{LHS}(y(t - L(\Omega)), \ldots, y(t - L(0)) = \bigcap_{i=\Omega}^{0} \mu_{A_j}(y(t - L(i))) \tag{4.16}$$

In the forecasting procedure, the Step 1 requires a sample with Ω lags that will generate Ω fuzzyfied values. In Step 2, all combinations between the fuzzy sets of each fuzzyfied lag will be the LHS of the affected PWFTPGs. In Step 3, in Equations (4.9), (4.12) and (4.14), the empirical conditional probability $P(y(t)|A_i)$ will be replaced by $P(y(t - m), ..., y(t)|LHS)$, the empirical conditional probability of the sample $y(t -$

$m), ..., y(t)$ given the LHS of the PWFTPG.

$$P(y(t-L(\Omega))\dots y(t-L(0)|LHS) = \pi_{LHS}\frac{\mu_{LHS}(y(t-L(\Omega))\dots,y(t-L(0))}{\sum_{A_j \in LHS} Z_{A_j}} \tag{4.17}$$

4.5 Computational Experiments

This section presents an empirical study of the PWFTS performance using the same datasets, design of experiments and statistical tests employed in Sections 2.8 and 3.6. PWFTS method can forecast points, intervals, and probability distributions, then the following sections will present all these compared results.

To measure the performance of the proposed models, ARIMA, QAR, kNN/KDE, and BSTS were chosen as competitor models due to its ability to perform point, interval and probabilistic forecasting for many steps ahead. The hyperparameters of each method were individually investigated and only the best model is considered in the validation of the results. The HOFTS and WHOFTS methods were also used to compare the point forecasts, using the best models determined in Section 2.8. The $[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ methods were also used to compare the interval forecasts, and EnsembleFTS was also used to compare probabilistic forecasts, using the best models determined in Section 3.6.

In Section 4.5.1, the accuracy sensitivity regarding to the hyperparameters of the proposed methods are analyzed using a grid search. The result of the experiments are presented in Sections 4.5.2 for point forecasting, 4.5.3 for interval forecasting and 4.5.4, for probabilistic forecasting.

In order to contribute with the replication of all the results in the research, all data and source codes employed in this chapter are available at the URL: http://bit.ly/scalable_probabilistic_fts_chap4

4.5.1 Hyperparameter Grid Search

In order to assess the impact of the hyperparameters on PWFTS accuracy, a Grid Search was performed for each benchmark dataset, using the same search spaces contained in Table 10 of Section 3.6.1. But, different from the previous experiments this grid search was performed for point, interval and probabilistic forecasting, in order to chose the hyperparameter values that best fit all cases.

The RMSE accuracy is shown in Figure 35, by order, number of partitions and dataset. The Winkler Score accuracy, where $\alpha \in \{.05, .25\}$, can be observed in Figure 36 and the CRPS accuracy in Figure 37.

Since several numbers of partitions and order values achieved very close accuracy values, the Principle of Parsimony (or Occam's Razor) was adopted to choose the set of hyperparameters that lead to the smallest number of rules $|\mathcal{M}|$, keeping the same accuracy. The chosen hyperparameters were $k = 45$ and $\Omega = 1$ and a sample of the best models performance can be seen in Figures 30 (for one step ahead) and 33 (for many steps ahead).

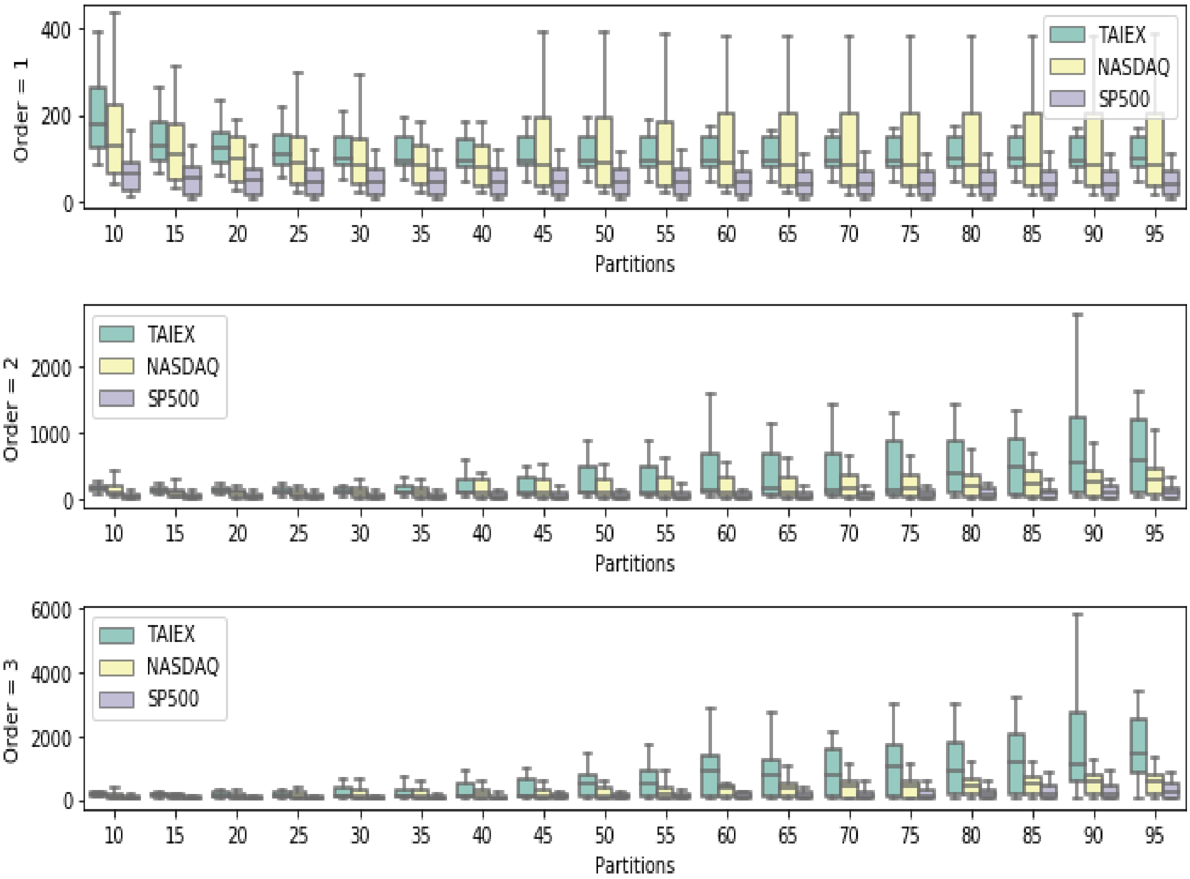


Figure 35 – RMSE accuracy for order, partitions and dataset

4.5.2 Point Forecasting Benchmarks

The RMSE results for each method and dataset are presented in Table 20. The Friedman Aligned Ranks of the methods are presented in Table 21 and the test statistic for these results is $Q = 13.903114186851207$, where the p-value is $P(\chi^2_{df} < Q) = 0.030737356514312197$, with $df = 7$ degrees of freedom. For this statistic the H_0 is rejected at the $\alpha = .05$ confidence level, indicating that there is difference between the means of the competitor models.

The *post-hoc* tests were employed using PWFTS as control methods and their results are presented in Table 22, showing that there is no prevalence of PWFTS method over all others. The mean difference detected by the Friedman Test occurred between QAR and BSTS, where QAR prevails over BSTS with p-value of 0.006984, rejecting H_0

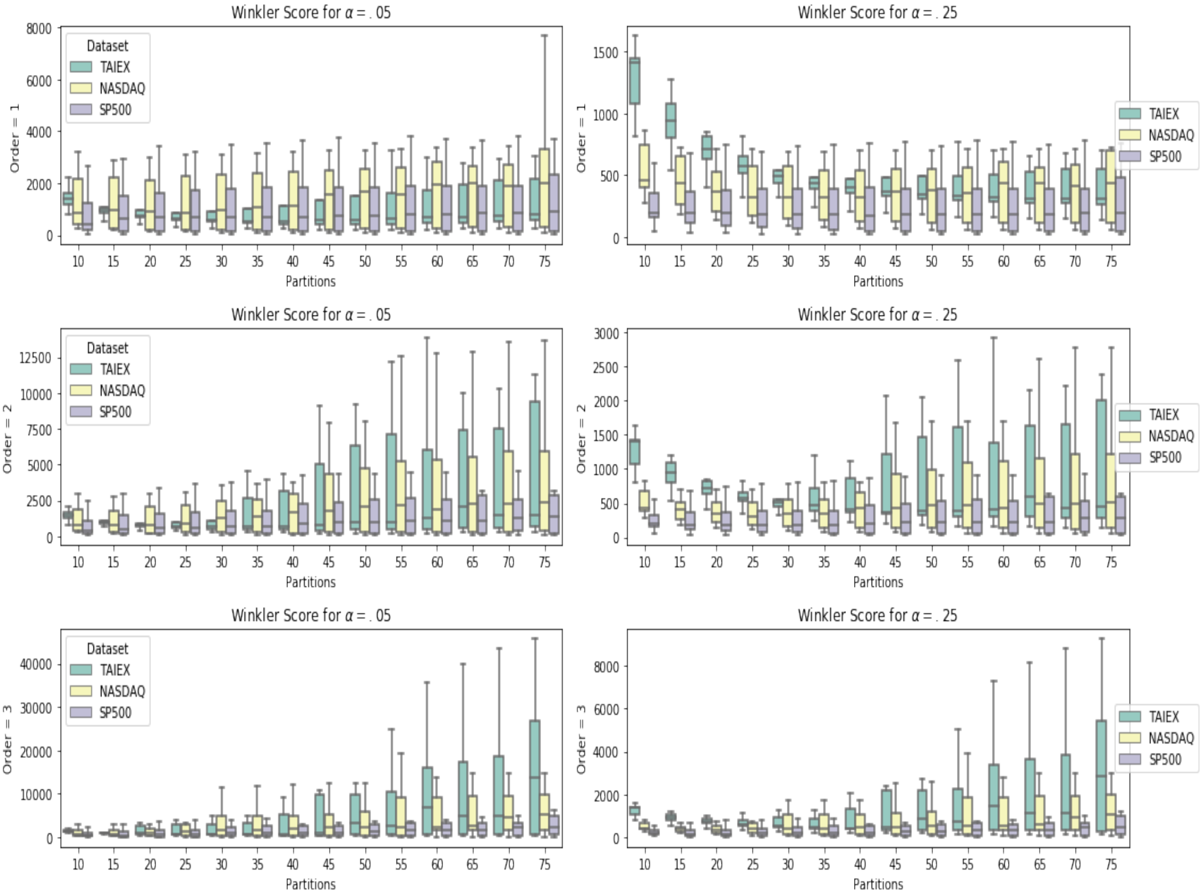


Figure 36 – Mean Winkler Score for $\alpha \in \{.05, .25\}$ by order, partitions and dataset

of *post-hoc* tests. These results showed that PWFTS point forecasting method performs satisfactorily when compared with the standard methods in the literature.

The statistical tests were employed to one step ahead forecasts. Figure 38 shows, for each method and dataset, the impact of the forecasting horizon on the RMSE accuracy.

Dataset	ARIMA	QAR	PWFTS	WHOFTS	HOFTS	kNN	BSTS
S&P 500	6.091 ±7.452	8.177 ±11.366	10.541 ±10.19	12.822 ±11.336	13.605 ±12.392	19.242 ±24.97	380.466 ±947.809
NASDAQ	22.592 ±24.991	17.951 ±11.965	24.839 ±18.198	27.154 ±15.05	29.713 ±12.875	34.742 ±25.096	413.494 ±837.281
TAIEX	91.311 ±63.249	66.9 ±44.369	75.558 ±56.739	90.433 ±58.93	100.787 ±62.932	80.213 ±56.494	271.66 ±250.078

Table 20 – RMSE for one step ahead point forecasts

4.5.2.1 Residual Analysis

The residuals of the models are presented in Figure 39 and the Ljung-Box tests for the 3 first lags are presented in Table 23 , showing the good fit of the model.

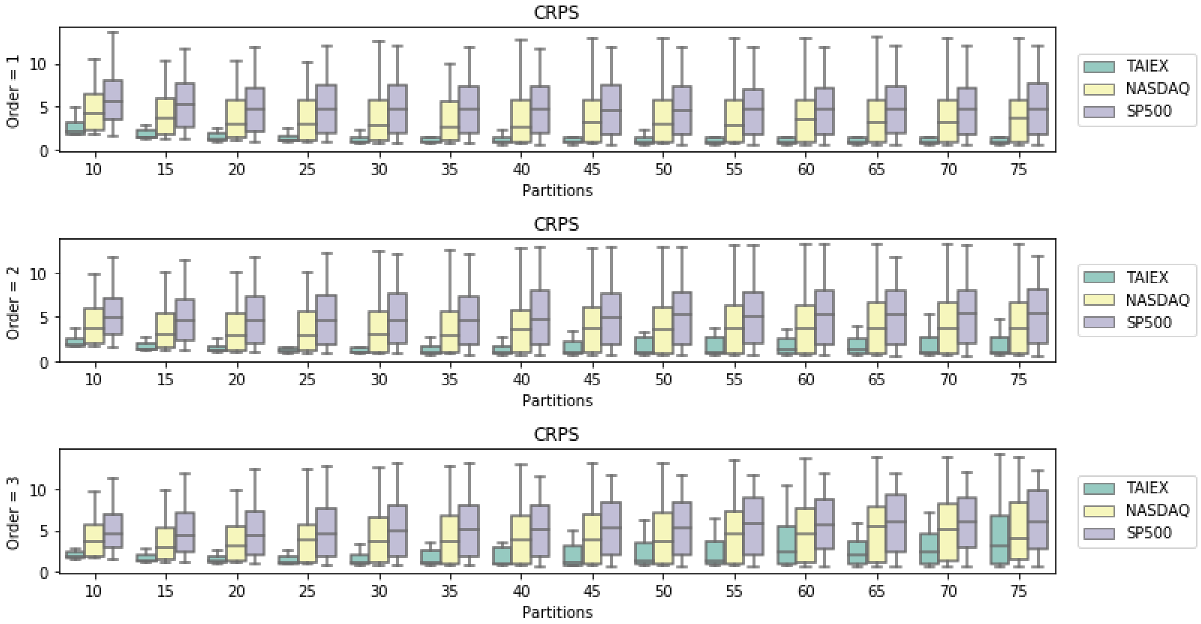


Figure 37 – CRPS accuracy by order, partitions and dataset

METHOD	RANK
QAR	6.333333
ARIMA	7.333333
PWFTS	8.333333
WHOFTS	10.333333
HOFTS	12.000000
kNN	12.666667
BSTS	20.000000

Table 21 – Friedman aligned ranks for point forecasts

	COMPARISON	Z-VALUE	P-VALUE	ADJUSTED P-VALUE	Result
0	PWFTS vs BSTS	2.302831	0.021288	0.121122	H0 Accepted
1	PWFTS vs kNN	0.855337	0.392364	0.775648	H0 Accepted
2	PWFTS vs HOFTS	0.723747	0.469221	0.775648	H0 Accepted
3	PWFTS vs QAR	0.394771	0.693012	0.829909	H0 Accepted
4	PWFTS vs WHOFTS	0.394771	0.693012	0.829909	H0 Accepted
5	PWFTS vs ARIMA	0.197386	0.843526	0.843526	H0 Accepted

Table 22 – Post-hoc tests using PWFTS as control method

4.5.3 Interval Forecasting Benchmarks

The Winkler Score Mean results for each method and dataset are presented in Table 24. The Friedman Aligned Ranks of the methods are presented in Table 25 and the test statistic for these results is $Q = 14.812664$, where the p-Value is $P(\chi^2_{df} < Q) = 0.038477$, with $df = 7$ degrees of freedom. For this statistic, the H_0 is rejected at the $\alpha = .05$ confidence level, indicating that there is a difference between the means of the competitor

Lag	Statistic	p-Value	Critical Value	Result
1	34.846143	3.568157e-09	3.841459	H0 accepted
2	35.497255	1.958254e-08	5.991465	H0 accepted
3	35.871542	7.971605e-08	7.814728	H0 accepted

Table 23 – Ljung-Box Test for the 3 first lags

models.

The *post-hoc* tests were employed using PWFTS as control method and their results are presented in Table 26, showing the prevalence of PWFTS over BSTS. These results showed that PWFTS interval forecasting methods perform satisfactorily when compared with other standard methods in the literature.

The statistical tests were employed on the one-step ahead forecasts. Figure 40 shows, for each dataset, the impact of the forecasting horizon on the Winkler Score accuracy of PWFTS.

Dataset	**ARIMA**	**PWFTS**	**QAR**	**WIFTS**	**IFTS**	**kNN**	**EnsembleFTS**	**BSTS**
S&P 500	72.712 ±135.871	73.505 ±99.09	121.694 ±319.305	111.705 ±156.013	113.516 ±91.627	131.394 ±166.31	268.567 ±318.259	292.415 ±384.499
NASDAQ	233.261 ±486.735	112.944 ±33.666	106.416 ±56.248	123.35 ±141.251	284.692 ±147.24	170.709 ±156.097	603.881 ±638.297	652.036 ±963.624
TAIEX	858.124 ±1337.139	348.647 ±82.036	340 ±269.34	480.581 ±561.826	917.879 ±243.737	428.484 ±269.459	898.531 ±1175.107	1280.67 ±1472.031

Table 24 – Average Winkler Score with $\alpha = .05$ for one step ahead interval forecasts

METHOD	RANK
PWFTS	6.000000
QAR	6.666667
WIFTS	7.666667
kNN	8.666667
ARIMA	13.000000
IFTS	16.666667
EnsembleFTS	19.666667
BSTS	21.666667

Table 25 – Friedman aligned ranks for interval forecasts

4.5.4 Probabilistic Forecasting Benchmarks

The CRPS Mean results for each method and dataset are presented in Table 27. The Friedman Aligned Ranks of the methods are presented in Table 28 and the test statistic for these results is $Q = 10.352711804324706$, where the p-Value is $P(\chi^2_{df} < Q) = 0.06583635032195168$, with $df = 5$ degrees of freedom. For this statistic the H_0 is accepted at the $\alpha = .05$ confidence level, indicating that there is no significant difference between

COMPARISON	Z-VALUE	P-VALUE	ADJUSTED P-VALUE	Result
PWFTS vs BSTS	2.713546	0.006657	0.045677	H0 Rejected
PWFTS vs EnsembleFTS	2.367136	0.017926	0.061349	H0 Accepted
PWFTS vs IFTS	1.847521	0.064672	0.144442	H0 Accepted
PWFTS vs ARIMA	1.212436	0.225346	0.360355	H0 Accepted
PWFTS vs kNN	0.461880	0.644167	0.764634	H0 Accepted
PWFTS vs WIFTS	0.288675	0.772830	0.822550	H0 Accepted
PWFTS vs QAR	0.115470	0.908073	0.908073	H0 Accepted

Table 26 – Post-hoc tests using PWFTS as control method

the means of the competitor models. This result discards the need to employ *post-hoc* tests and shows that there is no prevalence of one method over others, showing also that PWFTS probabilistic forecasting method performed satisfactorily when compared with the standard methods in the literature.

The statistical tests were employed on the one-step ahead forecasts. Figure 41 shows, for each dataset, the impact of the forecasting horizon on the CRPS accuracy of PWFTS method.

Dataset	**PWFTS**	**QAR**	**kNN**	**ARIMA**	**EnsembleFTS**	**BSTS**
NASDAQ	0.882 ±0.347	1.028 ±0.748	1.158 ±0.477	1.444 ±1.303	1.923 ±1.416	3.208 ±3.983
TAIEX	0.967 ±0.404	1.135 ±0.613	1.229 ±0.693	1.691 ±1.239	1.301 ±1.118	4.081 ±5.306
S&P 500	1.257 ±0.722	1.557 ±1.74	4.403 ±3.261	1.216 ±1.166	1.995 ±2.255	3.278 ±3.16

Table 27 – CRPS for one step ahead interval forecasts

METHOD	RANK
PWFTS	3.333333
QAR	5.666667
ARIMA	8.666667
kNN	11.333333
EnsembleFTS	11.666667
BSTS	16.333333

Table 28 – Friedman aligned ranks for probabilistic forecasts

4.6 Conclusion

This chapter proposed a new univariate and time invariant FTS method – the Probabilistic Weighted FTS (PWFTS) – a weighted rule-based FTS method which represents their temporal patterns with an empirical probability, based on the proposed concept of fuzzy frequency. The PWFTS rule model, the Probabilistic Weighted Fuzzy Temporal Pattern Groups (PWFTPG), describes fuzzy and stochastic behavior of time series and combines them to produce forecasts.

Among the methods already proposed for interval and probabilistic forecasting, none of them integrate all these capabilities. The strength of PWFTS lies is its flexibility and performance. This model is used to produce probability densities, prediction intervals and point forecasting, with high order models and multiple-step ahead forecasting.

Computational experiments were performed to evaluate the accuracy of the proposed model which showed equivalent or better performance than standard methods in the literature. Its computational cost is low when compared with BSTS and EnsembleFTS approaches and its interval accuracy is better than WIFTS and IFTS.

The proposed PWFTS method extends FTS methods to deal with interval and probabilistic forecasting applications, which is the major contribution of this research. Moreover, PWFTS improves on former FTS methods in the literature by considering the concept of fuzzy frequency and empirical probabilities in the generation of the rule knowledge base. The proposed method improves previous FTS methods by aggregating probabilistic and interval forecasting capabilities into a single model, being useful for a wide range of applications and user needs.

4.6.1 Method limitations

As in previous FTS methods, the PWFTS accuracy depends on the hyperparameter fine tunning. The method does not embody this optimization and it is advisable that this fine tunning be performed. Another issue about the model optimization is the parsimony: PWFTS weights may vanish as the number of rules increases. The weights precision is limited by the computational numerical precision.

In general all forecasting procedures are computationally cheap but the probabilistic forecasting for multiple-steps ahead is computationally expensive and the forecasting horizon H must be chosen carefully.

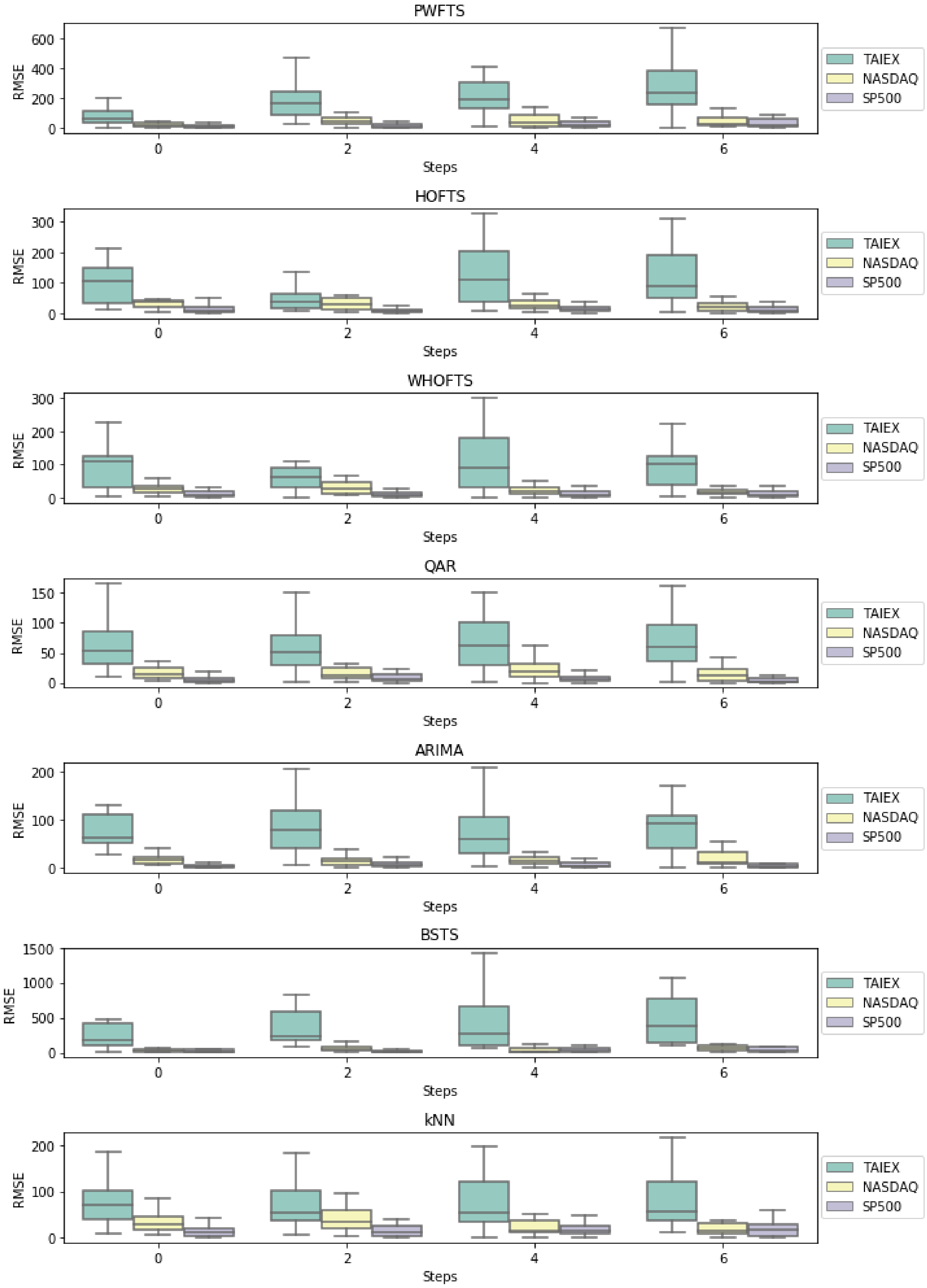


Figure 38 – Impact of the forecasting horizon on RMSE accuracy

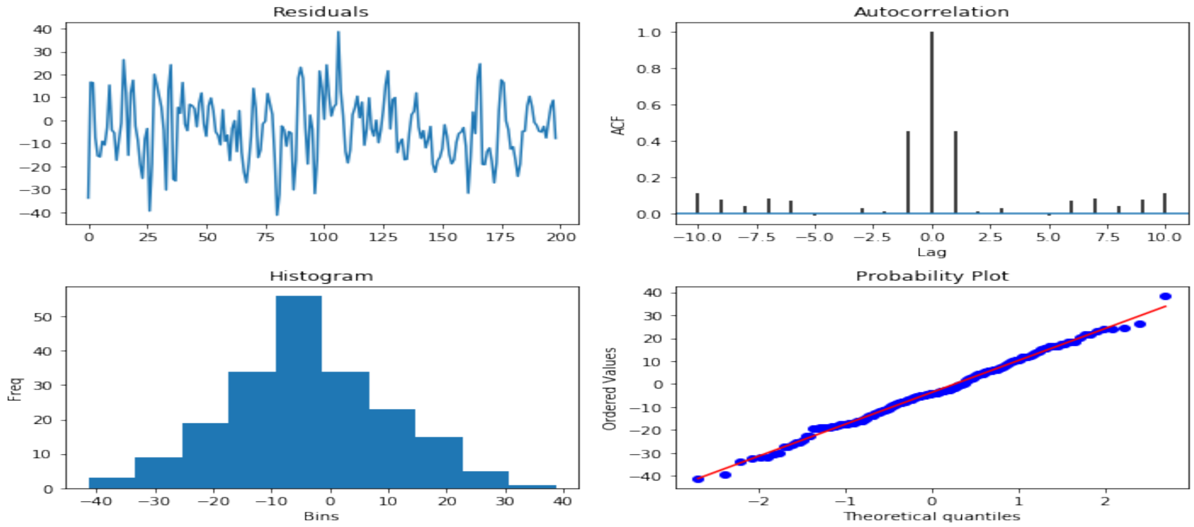


Figure 39 – Residual analysis of PWFTS

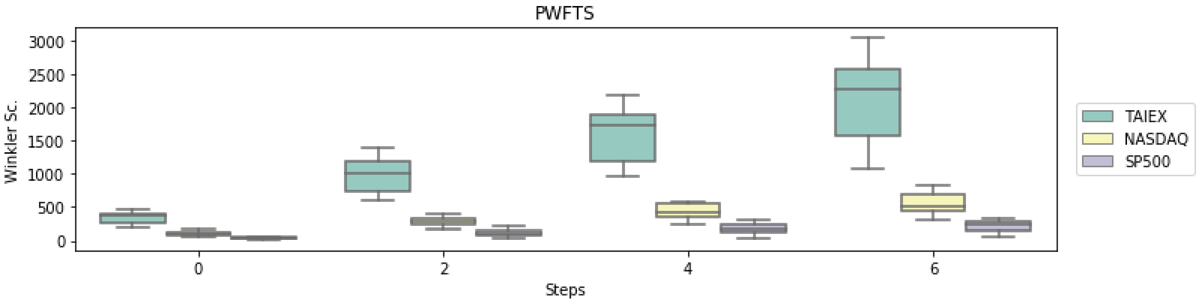


Figure 40 – Impact of the forecasting horizon on Winkler Score accuracy

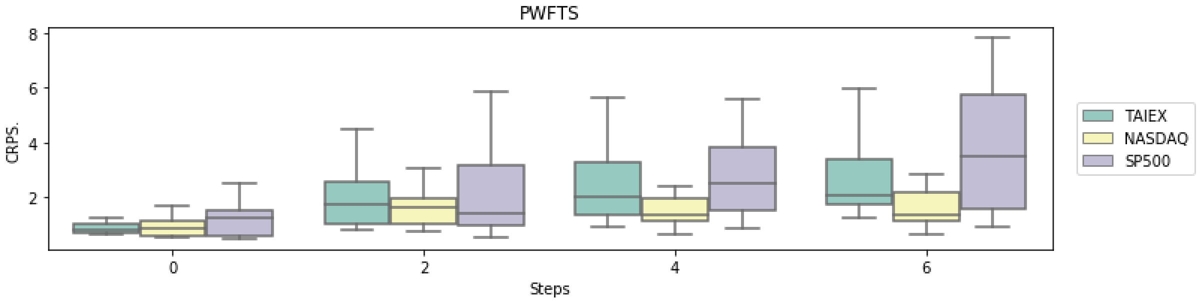


Figure 41 – Impact of the forecasting horizon on CRPS accuracy

Chapter 5

Scalability And Hyperparameter Optimization

"While we cannot accurately predict the course of climate change in the coming decades, the risks we run if we don't change our course are enormous. Prudent risk management does not equate uncertainty with inaction."
— Steven Chu

The previous chapters focused in forecasting tasks and its uncertainties and FTS approaches were presented to deal with theses issues. However, the presented approaches do not optimize its models for a given time series Y. The best hyperparameters must be found by the user and then informed to the methods, task that was delegated until now to an expensive Grid Search optimization.

As reviewed in Chapter 2, several approaches in the literature embody optimization tasks in their training procedures. Nevertheless an holistic hyperparameter optimizer for all FTS components present in Table 1 is still lacking in the literature. The most challenging issue presented by this task is the choose of the partitioning method Π, which may use computationally expensive meta-heuristics (and without known parallel or distributed extensions), and the number of partitions k and order Ω that regulates the parsimony of the model.

The optimization task becomes even harder when dealing with time series with Big Data properties. Many of the traditional forecasting methods, and even some new ones, were not designed to deal with such high volume of data. The most critical issues are the high dimensionality (dozens of hundreds of attributes) and volume (hundreds of millions or billions of samples), as pointed out in Qiu et al. [2016]. There is not a universal consensus about the small, medium and big data sizes. By simplicity it can be considered that it strongly depends on the available hardware capabilities.

In all cases Big Data starts to happen when it does not fit in the memory of a single machine. Such data volume, that cannot be grounded on a single machine memory, demands a distributed architecture of storage and processing. New technologies are emerging to tackle these issues, for instance the Map Reduce based frameworks Dean and Ghemawat [2008], a divide-and-conquer approach which is the basis of Hadoop clusters White [2012], where the processing units also act as storage units of the data subsets.

These distributed computation approaches have been determinant to enable the processing of data-intensive and computationally expensive tasks. Thanks to the distributed computation frameworks, soft-computing methods are now enabled to work with massive datasets using cheap and available hardware infrastructure. Such kind of tasks are spread over several areas in science and engineering, such as in weather and environmental datasets and on smart sensor data of smart grids where, according to Coelho et al. [2016], there are networks of smart sensors continuously monitoring all system components and streaming historical data with high volume and velocity.

Applying Big Data to machine learning algorithms is a trending topic in recent years, as can be seen in Zhou et al. [2017]. But the literature on taming big time series did not considered FTS methods directly, although there exists approaches involving other fuzzy methods, see for instance Singh [2015].

The absence of computationally expensive iterative procedures inside the training and forecasting procedures of the proposed FTS methods facilitates its scalability, as well as the use of white box knowledge models that can be easily distributed and updated. With this design, FTS models $\mathcal{M}$ can be quickly trained in commodity hardware, without major processing requirements, and transferred between cluster nodes.

This chapter aims to propose scalable alternatives to perform training FTS methods using distributed algorithms and exploit these solutions to tackle the hyperparameter optimization of big time series. In Section 5.2 an distributed FTS training approach for big time series is proposed. In Section 5.3 the Distributed Evolutionary Hyperparameter Optimization (DEHO) method is proposed, combining evolutionary algorithms and the previously defined distributed training and forecasting approaches. In Section 5.4 computational experiments are performed to assess the speed up provided by the distributed methods and the convergence of DEHO method for large environmental time series. Finally, in Section 5.5 the results are discussed and synthesized.

5.1 Computational Clusters

According to Baker and Buyya [1999], “A cluster is a type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers working together as a single, integrated computing resource”. Clusters are used for providing high availability, load-balancing, distributed storage, high processing power, among other purposes.

The Distributed Storage Clusters (DSC) systems are employed to keep distributed database systems or distributed file systems, exploiting local storage resources to allow the storage of data volumes that can be handled by a single machine, while providing transparent access to the whole dataset. Diversely, the High Processing Clusters (HPC) systems are employed to solve complex or expensive computational tasks which can be decomposed and parallelized by sub-datasets (Single Instruction / Multiple Data), subtasks (Multiple Instruction / Single Data) or both (Multiple Instruction / Multiple Data). A good example of HPC cluster is the classical Beowulf Cluster[1] architecture, which makes use of message passing middleware like MPI and PVM. In these frameworks the instructions and data are spread across the cluster and, after the local processing on each cluster node finishes, the results are gathered in some master or control node.

With the advent of Big Data, the demand of distributed file systems capable to store large datasets was joined with the demand for simple programming interfaces for processing distributed data. The Map/Reduce, proposed in Dean and Ghemawat [2008], became a popular distribution paradigm due to its high adoption in Big Data literature. In such paradigm the computational cluster contains a master node, which centralizes the management of the tasks, and several slave nodes responsible for working tasks. The distributed execution is divided into two main phases, the map (scattering) and reduce (gathering). The Map phase splits the original dataset into smaller subsets and distributes them to the slave nodes. Each individual slave node will perform the same predefined set of computations on data and send the results back to the master node. The Reduce phase collects the results from the slave nodes and performs final aggregations of results.

The popularity of the Map/Reduce paradigm to tackle Big Data problems imersed after the first open source infrastructure frameworks became available, for instance Apache Hadoop[2]. More recently some infrastructure was developed to allow in-memory processing, turning the processing yet more efficient, as for instance the Spark framework[3]. In the next sections, the distribution strategies for the sequential FTS methods are discussed using HPC middleware and Map/Reduce paradigm.

5.2 Scalable Models With Distributed Execution

Depending on the data size and the capabilities of the available infrastructure, different approaches must be considered for FTS method scalability, specially when dealing with hyperparameter optimization.

[1] The Beowulf Project - http://www.beowulf.org. Access in 15/05/2019

[2] Apache Hadoop Project - https://hadoop.apache.org/. Access in 15/05/2019

[3] Apache Spark - https://spark.apache.org/. Access in 15/05/2019

Small-sized time series (up to 10,000 instances) can be handled easily by a single machine and the costs of distribution (network and middleware overhead) do not pay off. This is the approach presented in all previous chapters.

For middle sized data (from 10,000 to 500,000 instances) the optimization process is more likely task-intensive, the evaluation dataset Y can be split in smaller train/test data windows that can be handled by a single machine, and just the accuracy results need to be gathered and agregated. This is the approach presented in Section 5.2.1.

However, for highly sized data (above 500,000 instances), even the train/test data windows are costly to be trained by a single machine. In this case the training and testing methods need to be distributed themselves. This is the approach presented in Section 5.2.2.

5.2.1 Distributed Testing With Sequential Models

The distributed testing with sequential models aims to speed up iterative optimization processes that require unnumbered evaluations of the objective function with different small to medium datasets. Each evaluation requires a sample of the entire dataset with which an FTS model $\mathcal{M}$ will be trained and evaluated using an accuracy metric ϵ.

This is particularly the case of the hyperparameter optimization where, given a set of hyperparameters Θ and a time series dataset Y, for each combination of hyperparameters values $\theta \in \Theta$ being evaluated, it must perform a rolling window cross validation on the time series dataset. The distributed rolling window cross validation, shown in Figure 42, splits the whole dataset Y in W smaller and overlapping data windows $i = 1..W$ and each data window is divided in train and test subsets. Then, for each data window i, a new model $\mathcal{M}_i$ will be trained and evaluated, generating the local accuracy metric ϵ_i. The average of the accuracy metric is calculated as $\epsilon = W^{-1} \sum_{i=1}^{W} \epsilon_i$.

Parameter	Name	Description
$0 < W_L < \|Y\|$	Window Length	the number of time series instances in each data window
$W_I \in [0,1]$	Window Increment	Percentage of W_L which is used to move the window
$T_S \in [0,1]$	Train/Test split	Percentage of W_L which is used as training set and the remaining as test set.

Table 29 – Distributed Testing Parameters

The distributed testing uses the parameters in Table 29. The number of data windows W is given by $W = \max\{w \mid w(W_L \cdot W_I) + W_L \leq |Y|\}$ where $|Y|$ is the length of Y, and the process illustrated in Figure 42 is executed in each evaluation of optimization engine. The key advantage of this approach is that it does not require any change on the

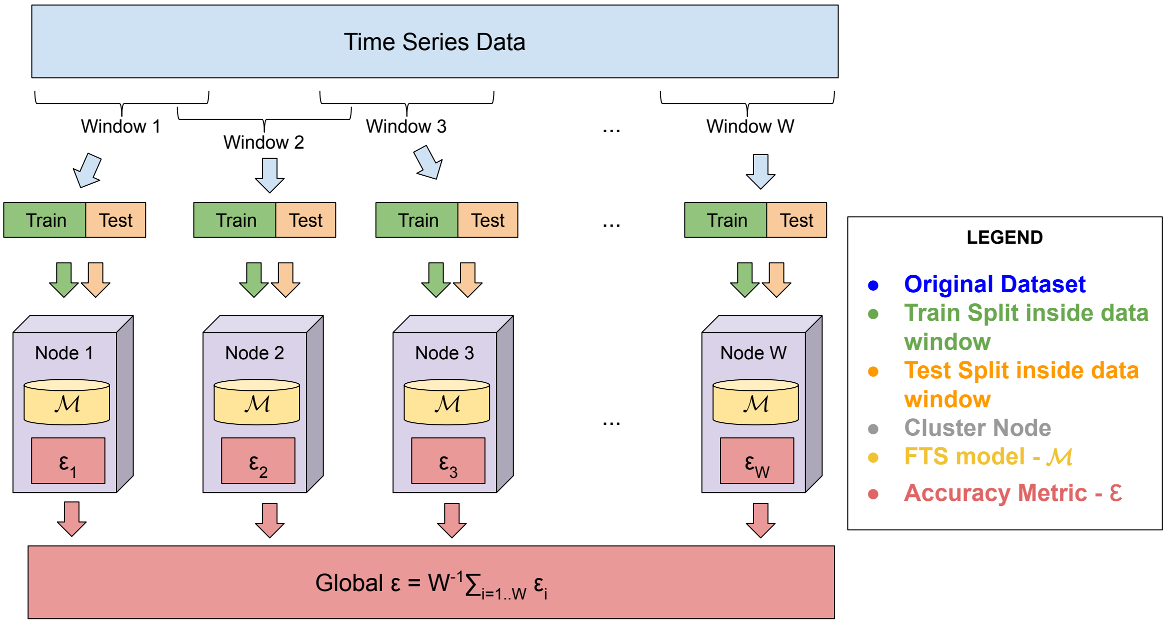


Figure 42 – Distributed Testing With Sequential Models approach

FTS training and testing methods, just an adaption in the way the optimizer performs the evaluations.

However, for big time series it is not enough. Depending on $|Y|$ and the capabilities of the available hardware, choosing of W_L and W_I may lead to three scenarios: a) W_L will be larger than the available memory of the cluster nodes; b) a value of W much greater than the number of cluster nodes, implicating in several rounds of computation for each node; c) W_I too large that leads to the sub-sampling of the data (the windows are not overlapped and let ranges without been tested). None of these scenarios is desirable and thus a new approach must be considered.

5.2.2 Distributed Models

For big time series the choosing of W_L and W_I values that do not lead to sub-sampling or cluster overhead, may fatally lead to a value of W_L greater than the available machine memory. In this case even the windows must be split in smaller ones and the methodology of training and testing the models must change. Several stages of the training processes defined in Sections 2.7.1 and 4.2 can be executed in parallel or distributed, on a Single Instruction/Multiple Data (SIMD) approach, since the data splits preserve the inherent time ordering. This characteristic allows the procedure's distribution to enhance their scalability and enable it to handle big time series.

This new approach changes the training method to first run the sequential procedure on individual cluster nodes with a slice of the original data window, creating a sub-model $\mathcal{M}_i$. Then the locally trained models are transmitted back to a master node where all local models are merged in a unique global model $\mathcal{M}$, as illustrated in Figure 43. In the next sections the distributed training and forecasting methods are presented.

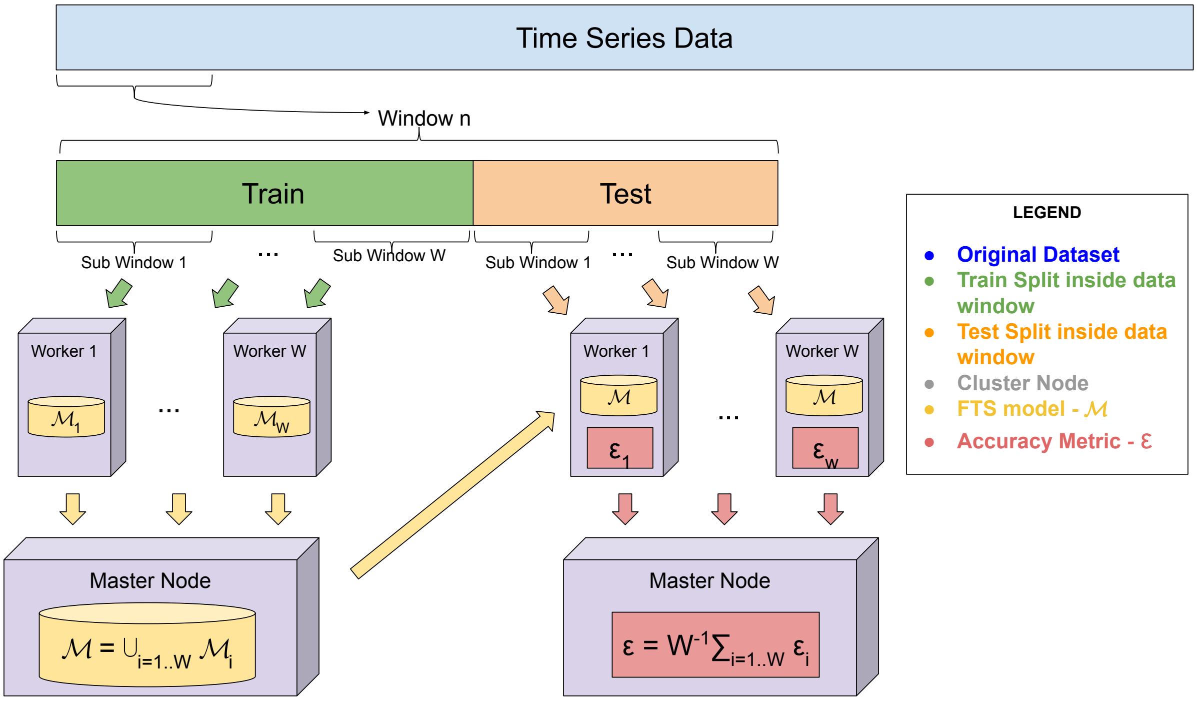


Figure 43 – Distributed Models approach

5.2.2.1 Distributed training Procedure

The adaption of the sequential procedure defined in Section 2.7.1 to the distributed one requires just few interventions. Stage 1 deeply depends on finding the universe of discourse U. The general procedure splits the dataset over the working nodes where the U_i universes of discourse are computed. In the final step a general linguistic variable $\tilde{A}$, is computed by merging the locals U_i as $U = \bigcup U_i$ where $\bigcup$ is the merge step.

On the other hand, the design of FTS allows the complete execution of stages 2 and 3 of training process without changes, as shown in Figure 44. In this way, each computational node will produce its own complete model $\mathcal{M}_i$ using its subset of the data, and on the final step a unique model $\mathcal{M}$ is generated by merging the local models $\mathcal{M}_i$ as $\mathcal{M} = \bigcup \mathcal{M}_i$, where $\bigcup$ is the merge step. The complete distributed training procedure is listed below and illustrated in Figure 44:

1. **Partitioning**:

   a) **Share**: The hyperparameters k and μ are shared across the cluster;

   b) **Map**: Distribute the Y dataset over the slave nodes and find U_i returning it back to the master node;

c) **Reduce**: Collect the U_i from the slave nodes, mixing it on a unique interval as $U = \bigcup U_i$, where the $\bigcup$ will select the smallest lower bound and the greatest upper bound of each given interval;

d) **Create**: Once the universe of discourse U were defined, the creation of the linguistic variable $\tilde{A}$ is performed as the steps 2 and 3 of Stage 1 of the sequential procedure.

2. **Fuzzyfication & Rule Induction**:

a) **Share**: The linguistic variable $\tilde{A}$ and the α hyperparameters are shared across the cluster;

b) **Map**: Distribute the Y dataset over the slave nodes and perform the fuzzyfication and rule induction for each subset, generating a local FTS model $\mathcal{M}_i$ which is returned to the master;

c) **Reduce**: Collect all $\mathcal{M}_i$ models;

d) **Merge**: Create an empty FTS model $\mathcal{M}$. For each rule $LHS \rightarrow RHS$ in all collected models $\mathcal{M}_i$:

i. If $\mathcal{M}$ does not contain the LHS, then append the entire rule on $\mathcal{M}$;

ii. If $\mathcal{M}$ contains the LHS, then for each $w_j \cdot A_j \in RHS$:

A. If the RHS on $\mathcal{M}$ does not contain A_j, then append $w_j \cdot A_j$ on RHS and add w_j on $\#RHS$

B. If the RHS on $\mathcal{M}$ contains A_j, then add w_j on existing weight and add w_j on $\#RHS$

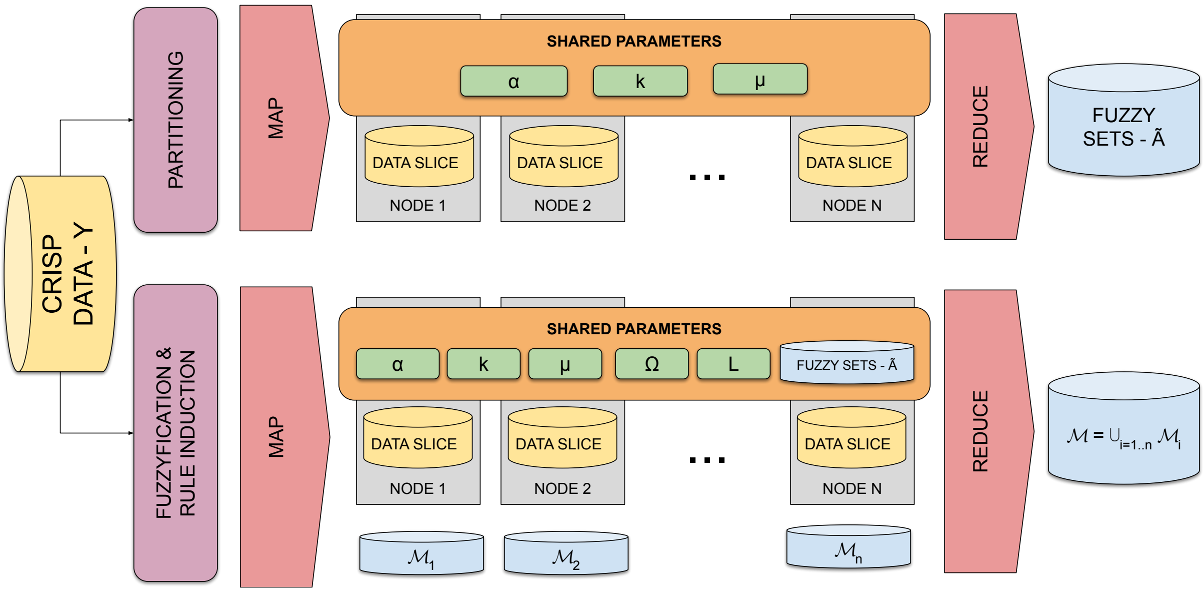


Figure 44 – Training of distributed models

5.2.2.2 Distributed forecasting procedure

The computational cost of the forecasting procedure is low when compared with the training procedure. The forecasting procedure bottlenecks are the fuzzyfication and rule matching steps, and both of them can be optimized using spatial indexes, for instance KD-trees [Muja and Lowe, 2014], which are more efficient when executed locally. Also, as on every distributed procedure, the communication overhead makes this procedure inefficient for data with low volume.

However, there are occasions where the model needs to be used for forecasting in batch, where the input has high volume and one step ahead forecasting will be performed for each data point. This scenario is common in model testing, simulation and hyperparameter optimization. In these cases it is profitable to share the parameters of the model $\mathcal{M}$ across several slave nodes and perform the forecasting on data splits, keeping its time ordering. The steps of the distributed forecasting procedure are listed below:

1. **Share**: The linguistic variable $\tilde{A}$ and the rules $\mathcal{M}$ are shared across the cluster;

2. **Map**: Distribute the Y dataset over the slave nodes, such that each split is labelled with its time ordering;

3. **Forecast** Each slave node receives a data split Y_p and executes the sequential forecasting process, generating the estimates $\hat{Y}_p$, sending it back to the master node with the same time label received with Y_p;

4. **Reduce**: Collect the $\hat{Y}_p$ estimates from the slave nodes;

5. **Merge**: Sort the $\hat{Y}_p$ estimates by their time label and concatenate them on a unique dataset $\hat{Y}$.

The proposed distributed methods speed up, or even make possible, to tackle high processing tasks, for instance iterative optimization procedures, using big time series and the previously proposed FTS methods. In next section an adaption of evolutionary algorithm is proposed to FTS hyperparameter optimization.

5.3 Distributed Evolutionary Hyperparameter Optimization

This section aims to propose the Distributed Evolutionary Hyperparameter Optimization - DEHO for FTS methods. The DEHO approach combines the FTS distributed training and forecasting methods with evolutionary optimization algorithms, specifically

Genetic Algorithms, in order to optimize FTS models in terms of accuracy and parsimony for big time series.

Given a set of hyperparameters Θ and an accuracy function $f : \Theta \rightarrow \mathbb{R}^+$, the hyperparameter optimization task aims to discover the values of each hyperparameter $\theta \in \Theta$ such that $\hat{\theta} = \arg\min_\theta f(\Theta)$. The hyperparameter optimization is a computational expensive task and, on a Big Data context, its cost may be prohibitive. The general FTS training procedure does not incorporate any kind of optimization, leaving to the user the task of empirically searching for the best hyperparameters. In the meantime, these parameters have great impact on final model performance and their optimization is highly recommended. At this point, it is necessary to take advantage of the speed up provided by the distributed method in Section 5.2 and to employ an efficient optimization method.

It is necessary to advise that the partitioning method Π was taken out of the optimization, and was kept constant as the Grid Partitioning. The reason is that, besides the Grid Partitioning, the other heuristic and metaheuristic Π methods can not be trained with separated data and after merged without compromising its original features. This special aspect must be the subject of future investigations in order to enhance the DEHO method.

Two conflicting goals are sought during the hyperparameter optimization, the accuracy and the parsimony (the structural complexity as usually measured by the number of parameters of the model). In fuzzy time series models, to increase model's accuracy it is common to increase the number of fuzzy sets k and/or the order Ω of the model, which automatically increases the number of fuzzy rules $|\mathcal{M}|$. As the number of rules grows, the computational complexity of the model also increases and the FTS approach becomes less interesting when compared with other standard approaches, such as ARIMA or QAR. The challenge is to find a balance between these two objectives, keeping the model small, fast and accurate.

The FTS hyperparameter optimization problem is formulated below, where the objective function f_1 (5.1) controls the accuracy (represented by the weighted sum of the mean error $\bar{\epsilon}$ (5.4) and the standard deviation of the error σ_ϵ (5.5)) of the model and the objective function f_2 (5.2) controls for parsimony (represented by the weighted sum of the model length $|\mathcal{M}|$ (5.6) and the sum of the lags $|L|$ (5.7)). Some additional restrictions are imposed on the order Ω (5.8), number of partitions k (5.9), α-cut (5.10), and the lags indexes L (5.11).

Optimize:

$$minimize \quad f_1 = \quad 0.6\bar{\epsilon} + 0.4\sigma_\epsilon \tag{5.1}$$

$$minimize \quad f_2 = \quad 0.6|\mathcal{M}| + 0.4|L| \tag{5.2}$$

Where:

$$RMSE = \sqrt{\sum_{t=0}^{n}(y(t) - \hat{y}(t))^2)} \tag{5.3}$$

$$\bar{\epsilon} = W^{-1}\sum_{i=0}^{W} RMSE(i) \tag{5.4}$$

$$\sigma_{\epsilon} = W^{-1}\sum_{i=0}^{W} \bar{\epsilon} - RMSE(i) \tag{5.5}$$

$$|\mathcal{M}| = \sum_{i=0}^{rules} 1 \tag{5.6}$$

$$|L| = \sum_{i=0}^{\Omega-1} L(i) \tag{5.7}$$

Subject to:

$$\Omega \geq 1 \tag{5.8}$$

$$k \geq 3 \tag{5.9}$$

$$\alpha \in [0, 1) \tag{5.10}$$

$$1 \leq L(0) < ... < L(\Omega) \tag{5.11}$$

Genetic Algorithms (GA) are population-based metaheurisc approaches for solving the optimization problems based on a genetic refinement metaphor. Vanilla GA algorithms are intended for mono-objetive optimization, but with few adaptions it is possible to handle multi-objetive problems as well. In this work the adaptions were made on selection operator, which implements the Double Tournament strategy in order to comprise both objectives on balancing the population.

The set of hyperparameters Θ is presented in Table 1, except the partitioning method Π, such that $\Theta = \{\mu, k, \alpha, \Omega, L\}$. Each individual of the population is represented by a vector - the genotype - with the values of each hyperparameter $\theta \in \Theta$. This vector contains both real, categorical and array values, according to hyperparameter type. The GA iterates over steps listed below until the stop criteria is achieved:

1. **Initial Population**: The initial population, with size NP, is generated randomly, except for the hyperparameter L, whose size is constrained by Ω and the lag indexes initially consider the most significant ACF/PACF lags.

2. **Evaluation:** Each genotype is transformed into a trained model - the phenotype - using the distributed method of Section 5.2 and then evaluated. The phenotype evaluation uses the metrics according the objective function f_1 and f_2 and after the evaluation procedure, the population is sorted by f_1 and f_2 in ascending order.

3. **Selection:** The selection operator is responsible to choose one part of the individuals that will survive for the next generation. As the problem is multi-objective, a Double Tournament strategy was implemented to balance the selection between the two objectives. In the double tournament, the first round chooses randomly two pairs of individuals in the population, and each pair will compete with each other based on objective f_1. On the second round the winner individuals of the first round will compete with each other based on the objective f_2. This process is repeated by a rate SR of the population.

4. **Elitism:** As the selection operator is random, there is the possibility that the best individual of the population be discarded. The elitist strategy will keep the best individual of the current generation in the next generation and discard the worst.

5. **Crossover:** The crossover operator combines the genotypes of two individuals (i_1 and i_2) in order to generate a descendent individual (i_N). On crossover, two individuals are randomly selected in the population and ordered as i_1 and i_2 according to their f_1 and f_2 objectives. For all genes the mixing process will give a major contribution for the best ranked individual (with .7 and .3 rates). For the real coded genes a linear combination as $i_N = .7i_1 + .3i_2$ will be performed. For the categorical genes, the value of i_N will be i_1 with probability .7 or i_2 otherwise. For the lag L the individual lags will be also a linear combination of each lag. This process is repeated by a rate CR of the population.

6. **Mutation:** The mutation operator aims to introduce novelties in the population, then an individual is randomly chosen and random perturbations are applied to its genes, taking care to keep the gene values feasible according to the problem restrictions. This process is repeated by a rate MR of the population.

7. **Stop Criteria:** Repeat the steps 2 to 7 until one of these criteria are achieved: NG_{stop} generations without improvement or maximum number of generations NG.

This Genetic Algorithm requires the choice of the parameters presented in Table 30. The complete hyper-parameter optimization method also requires the choosing of the distribution type and the parameters presented in Table 29.

Parameter	Description
$PS \in \mathbb{N}^+$	Population Size
$NG \in \mathbb{N}^+$	Max Number of Generations
$0 < NG_{stop} < NG$	Max Number of Generations without improvement
$SR \in [0,1]$	Selection Rate
$CR \in [0,1]$	Crossover Rate
$MR \in [0,1]$	Mutation Rate

Table 30 – Genetic Algorithm parameters

5.4 Computational Experiments

This section presents an exploratory study of distributed models performance and the DEHO method. The computational experiments employed a large sized environmental time series, the SONDA dataset with 2,000,000 instances, and a medium sized time series, the Malaysia dataset, with 17,000 instances. Both datasets are detailed in Appendix B, where its main characteristics are presented.

In Section 5.4.1 the speed ups provided by the distributed training and forecasting are presented by several cluster configurations. In 5.4.2 the distributed methods are employed in DEHO method, and the convergence of the method is analyzed.

In order to contribute with the replication of all the results in the research, all data and source codes employed in this chapter are available at the URL: http://bit.ly/scalable_probabilistic_fts_chap5

5.4.1 Speed Up Of Distributed Methods

In order to assess the impact of including more processing nodes on training and forecasting processing times of distributed methods, different cluster configurations were evaluated .

The performance of the sequential and distributed methods, on above cited clusters configurations, was measured in terms of execution time (in seconds) and the speed up from the sequential time, such that $S_p = \frac{T_1}{T_p}$, where S_p is the speed up for p nodes, T_1 is the time of the sequential execution and T_p is the time of the distributed execution with p nodes.

The experiments show improvements on performance for each added node on the large sized dataset, but this improvement is smaller in the medium sized dataset. The trade-off between the distribution overhead and the benefit of the distributed computations stops to be profitable above the third node on the cluster for medium sized datasets. Above 3 nodes the network overhead for the length of data makes the distributed algorithm not interesting. However, it can be seen that an average speed up of 2x was achieved on the training procedure for large time series, showing that the performance tends to increase on more robust computational clusters.

Dataset	CPU's	Training Time	Training Speed Up	Forecasting Time	Forecasting Speed Up
SONDA Wind Speed	1	685.25 ± 135.16	-	285.89 ± 57.94	-
	2	383.29 ± 72.50	1.78	164.82 ± 30.10	1.73
	3	330.13 ± 64.55	2.07	138.58 ± 25.37	2.06
	4	342.64 ± 52.86	1.99	151.34 ± 25.54	1.88
	5	300.75 ± 58.19	2.27	130.67 ± 23.20	2.18
	6	348.98 ± 67.41	1.96	153.16 ± 30.0	1.86
	7	361.45 ± 65.61	1.89	160.74 ± 30.68	1.77
SONDA Solar Radiation	1	651.29 ± 121.95	-	274.28 ± 47.72	-
	2	383.24 ± 66.37	1.69	165.98 ± 36.21	1.65
	3	314.10 ± 59.98	2.07	136.92 ± 24.63	2.00
	4	345.55 ± 64.135	1.88	152.31 ± 28.43	1.8
	5	289.38 ± 54.44	2.25	129.52 ± 24.22	2.11
	6	340.35 ± 59.64	1.91	153.09 ± 28.35	1.79
	7	349.70 ± 65.48	1.86	159.16 ± 28.46	1.72
Malaysia Temperature	1	12.28 ± 0.70	-	5.11 ± 0.35	-
	2	7.21 ± 0.48	1.7	3.42 ± 0.23	1.49
	3	6.64 ± 0.45	1.84	3.24 ± 0.24	1.57
	4	7.44 ± 0.18	1.64	3.95 ± 0.23	1.29
	5	6.56 ± 0.29	1.87	3.93 ± 0.47	1.30
	6	7.46 ± 0.24	1.64	4.44 ± 0.23	1.15
	7	8.09 ± 0.27	1.51	5.15 ± 0.01	0.99
Malaysia Load	1	13.05 ± 1.50	-	5.32 ± 0.59	-
	2	7.84 ± 0.9	1.66	3.43 ± 0.24	1.54
	3	7.14 ± 0.75	1.82	3.42 ± 0.21	1.55
	4	8.06 ± 1.01	1.61	4.31 ± 0.39	1.23
	5	8.18 ± 2.31	1.59	4.10 ± 0.41	1.29
	6	8.06 ± 1.01	1.61	4.77 ± 0.46	1.11
	7	8.90 ± 0.86	1.46	6.18 ± 0.71	0.86

Table 31 – Speed up provided by the distributed model by number of CPU's

5.4.2 Convergence of DEHO approach

The DEHO method was employed using a computational cluster with 7 CPU's and the parameters contained in Table 32 using PWFTS as FTS method. The experiment performed 5 executions of DEHO for each dataset and the averaged results are presented in Table 33. A sample of the convergence process of DEHO can be seen in Figure 46, for SONDA Wind Speed dataset.

The results showed that, for the studied time series, the convergence was fast, expending about 16 generations on average. In the trade off between the objectives f_1 and f_2, the accuracy objective showed to be predominant over the parsimony objective during the convergence of the method.

The optimized values for the hyperparameters generated parsimonic and accurated forecasting models, whose samples of their performance can be seen in Figure

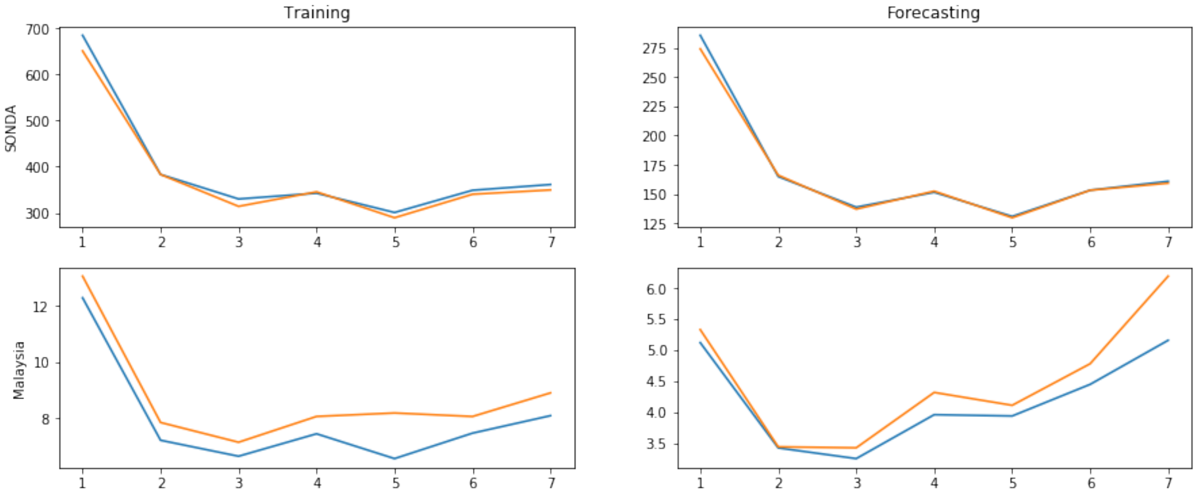


Figure 45 – Speed up provided by the distributed model by number of CPU's

Parameter	**Dataset**	**Value**
W_L	SONDA Wind Speed	600,000
	SONDA Solar Radiation	600,000
	Malaysia Temperature	10,000
	Malaysia Eletric Load	10,000
W_I	All	.5
T_S	All	.9
PS	All	20
NG	All	30
NG_{stop}	All	10
SR	All	.5
CR	All	.5
MR	All	.2

Table 32 – Distributed Evolutive Hyperparameter Optimization parameter values

Dataset	Generations	k	μ	α	Ω	L	Metric	Value
SONDA Solar Radiation	16.4 ± 7.8	50.8 ± 0.7	2	0.24 ± 0.13	2	[1,2]	$\|\mathcal{M}\|$	613 ± 222
							RMSE	93.13 ± 0.62
							Time	3221 ± 1505
SONDA Wind Speed	30.0	50	1	0.13 ± 0.1	1	[1]	$\|\mathcal{M}\|$	24 ± 1.45
							RMSE	$0.34 \pm 74e{-}10^{-4}$
							Time	3058 ± 891
Malaysia Energy Load	12.5 ± 2.5	50.6 ± 1.2	2	0.22 ± 0.23	2	[1,2]	$\|\mathcal{M}\|$	306.9 ± 137.9
							RMSE	2745.5 ± 271.27
							Time	3945.09 ± 800.71
Malaysia Temperature	16.6 ± 10.15	52.8 ± 3.18	1	0.24 ± 0.09	1	[1]	$\|\mathcal{M}\|$	73.21 ± 1.08
							RMSE	1.08 ± 0.06
							Time	3916.58 ± 2042.12

Table 33 – Optimization mean results by dataset

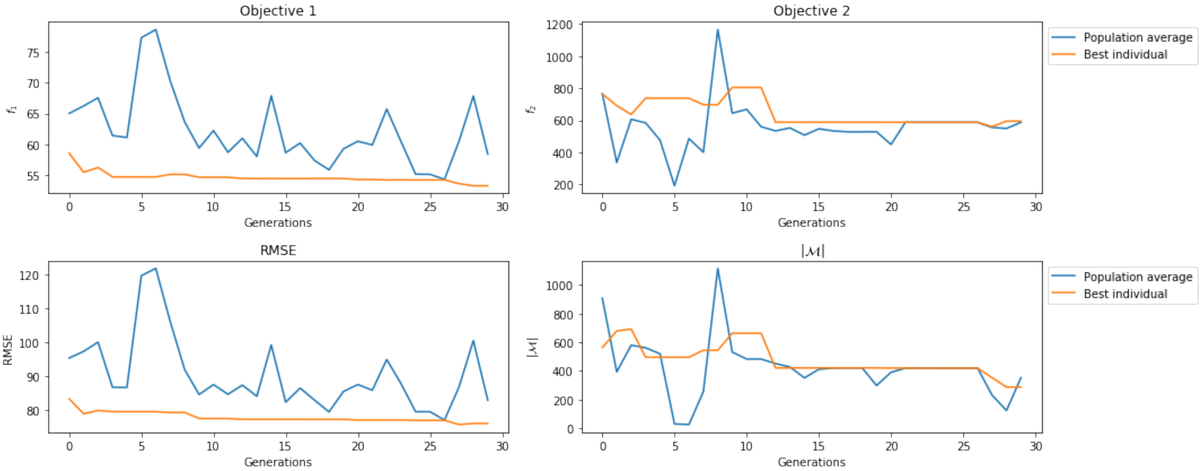


Figure 46 – Sample of DEHO convergence

5.5 Conclusion

Training accurate models for forecasting big time series is a challenging task for traditional and soft-computing methods. Usually the methods are not designed to deal with such high volume of data. When such data volume cannot be grounded on a single machine memory, it demands a distributed architecture of storage and processing. This is particularly problematic when optimizing the hyperparameters of a method, because successive model training and testing processes are required.

This chapter proposed two distributed approaches for FTS model scalability, one for middle sized data and another for big sized data. The first one distributes the data across the nodes of a cluster, where individual models are trained and tested. The second one, the distributed model itself, splits the training of a unique model across several nodes, allowing a big time series model to be trained in pieces and then aggregated into a single model.

These approaches were employed on Distributed Evolutionary Hyperparameter Optimization (DEHO). DEHO method is an adapted genetic algorithm that minimizes two cost functions, the accuracy function f_1 and the parsimony function f_2. An exploratory study was performed in order to measure the feasibility of the proposed distributed models and showed the speed-up provided for big time series. The convergence of DEHO method also was analyzed, showing its effectiveness.

5.5.1 Method limitations

The distributed training method is indicated only for big time series. Using the method for small data might slow down the training time due to the network and model merging overheads. DEHO method took into account only time invariant, rule based, monovariate and high-order methods, not being applicable for time variant, multivariate and first order FTS methods.

Next chapter presents a short review of multivariate methods and proposes a simple approach for extending PWFTS to forecasting multivariate time series.

Chapter 6

Multivariate Models

Despite the existing approaches in the literature, dealing with multivariate and spatio-temporal time series was always a challenging task for FTS methods, specially because of the complexity growth of the rules as the dimension increases. An important gap in the literature is the absence of multiple input and multiple output (MIMO) methods - the majority of FTS literature consists of basically univariate forecasting methods.

A simple approach is to transform multivariate time series into monovariate time series by using Fuzzy Information Granules (FIG). Each FIG acts as a multivariate fuzzy set, or a composition of individual fuzzy sets from different variables, allowing to replace a vector (the values of one data point) by a scalar (the identification of the FIG with the highest membership of that data point).

This approach is employed on Fuzzy Information Granular Fuzzy Time Series ($\mathcal{FIG}$-FTS), a wrapper method that enables PWFTS to tackle multivariate time series. It begins by partitioning the Universe of Discourse of each individual variable. Then the crisp values of each variable are fuzzyfied and the corresponding fuzzy sets are combined to create one Fuzzy Information Granule, such that it can be used as a reference of all data points in that same region. This incremental approach creates the FIGs on demand, according to the training data, and its sensibility can be controlled using the method's hyperparameters.

This chapter presents a short review of multivariate FTS methods and Fuzzy Information Granules in Section 6.1. Section 6.2 presents a conventional method for multivariate FTS (MVFTS), which does not employ FIGs. In Section 6.3 the Fuzzy Information Granule Fuzzy Time Series ($\mathcal{FIG}$-FTS) is proposed to enable PWFTS method be used for multivariate data and allowing the use of its interval and probabilistic forecasting features. In Section 6.4 an exploratory study of the performance of $\mathcal{FIG}$-FTS when compared with previous FTS methods is presented and finally, in Section 6.5, the results are discussed and the conclusions given.

6.1 Multivariate FTS Methods

Multivariate time series are sets of sequential vectors of the form $Y \in \mathbb{R}^n$ where $n = |\mathcal{V}|$ and $\mathcal{V}$ is the set of attributes of Y. Each vector $y(t) \in Y$ contains all attributes $\mathcal{V}_i \in \mathcal{V}$ and there is a temporal dependence between these data points such that their temporal ordering – given by the time index $t \in T$ – must be respected.

In the FTS literature it is common to employ clusterization methods to reduce multivariate data in monovariate ones, as can be seen in Li et al. [2008], Chen and Chang [2010], Sun et al. [2015] which employ Fuzzy C-Means (FCM) clustering algorithm to create multivariate FLRG's.

Chen and Chen [2011] introduced the concept of *Fuzzy Variation Groups* - FVG for bivariate FTS, where each FVG groups the FLRG's of each variable by their co-occurrence. Askari and Montazerin [2015] proposes the *High-Order Multi-Variable FTS* - HMV-FTS algorithm based on FCM clustering to generate the multi-variable FLRG's. Jilani et al. [2008] proposed the *Multivariate Stochastic FTS* - MSFTS based on the exponential smoothing between the diverse variables.

6.1.1 Fuzzy Information Granules

The concept of Fuzzy Information Granules (FIG) was first proposed in Zadeh [1996] as a way to define entities that represent subsets (or granules) of a wider domain. There are some works in the FTS literature where this concept is mixed with the partition of the Universe of Discourse, as discussed in Lu et al. [2014], Chen and Chen [2015a], but there are several ways to define FIG in the literature.

For univariate time series it is common to define a FIG as representative set of subsamples of the data, so each FIG is a common temporal pattern as in Yang et al. [2017b]. The construction of this kind of FIG usually employs the clustering of sub-sequences, as in Magalhães et al. [2008]. In Wang et al. [2014, 2015] we can find a univariate fuzzy time series approach whose FIGs are a combination of unequal partitioning of the UoD and prototype sub-sequences.

For multivariate time series FIGs are usually represented as hyper-boxes or multidimensional clusters in the feature space, as in Reyes-Galaviz [2016], Singh and Dhiman [2018]. In Singh and Dhiman [2018], a multivariate fuzzy time series method is presented, which uses a bio-inspired optimization method to create FIGs by iteratively adjusting the interval lengths of each variable.

Other non-FTS granular approaches can also be found in the literature, as the Granular Functional Forecasting (GFM), proposed in Magalhães et al. [2008], a univariate forecasting method based on Takagi-Sugeno fuzzy system where FIGs are created using

Alias	Parameter	Type	Description
k_i	Number of partitions	$\mathbb{N}^+$	The number of fuzzy sets that will be created in the linguistic variable $\widetilde{\mathcal{V}}_i$
μ	Membership function	$\mu : U \rightarrow [0,1]$	A function that measure the membership of a value $y \in U$ to a fuzzy set
α	α-cut	$[0,1]$	The minimal membership grade to take account on fuzzyfication process

Table 34 – WMVFTS hyperparameters for each variable $\mathcal{V}_i \in \mathcal{V}$

clustering methods. In Leite et al. [2011], the authors propose the fuzzy set based granular evolving modeling (FBeM) approach for time series prediction, later extended in Soares et al. [2018] for spatio-temporal data.

There are some notable drawbacks in the previous methods, namely: a) the absence of multivariate forecast (MIMO); b) the use of optimization methods to create the FIGs, which makes the learning process computationally expensive; c) the absence of multivariate FTS methods that could provide both weighted and high order characteristics. To fix these drawbacks this work proposes the $\mathcal{FIG}$-FTS method, a weighted and high-order FTS method that will be discussed in the next sections.

6.2 The Conventional Multivariate Fuzzy Time Series method

Just as it was done in Chapter 2, this section proposes a consensus model for rule based multivariate FTS that extends the model of Chen [1996] to the multivariate case. The Conventional Multivariate Fuzzy Time Series (MVFTS) method was designed to allow several models to be trained individually with subsets of a greater dataset and later to be merged into a single model, feature that enhances the performance of model creation by enabling its distribution.

For each chosen variable $\mathcal{V}_i \in \mathcal{V}$ on Y, MVFTS also incorporates several features present in the literature, represented by the hyperparameters in Table 34, giving versatility and flexibility to the model. The method is composed of two procedures: the training procedure and the forecasting procedure.

The MVFTS is a first order point forecaster of type Multiple Input/Single Output (MISO), then for the set of variables $\mathcal{V}$ one of them is chosen as the target (or endogenous) variable and the others are referred as the explanatory (or exogenous) variables. From now on, the target variable will be distinguished from the others by an asterisk, as $*\mathcal{V}$.

The training procedure, explained in subsection 6.2.1 and illustrated in Figure 47, is a three stage process responsible to create a multivariate weighted FTS model $\mathcal{M}$. The

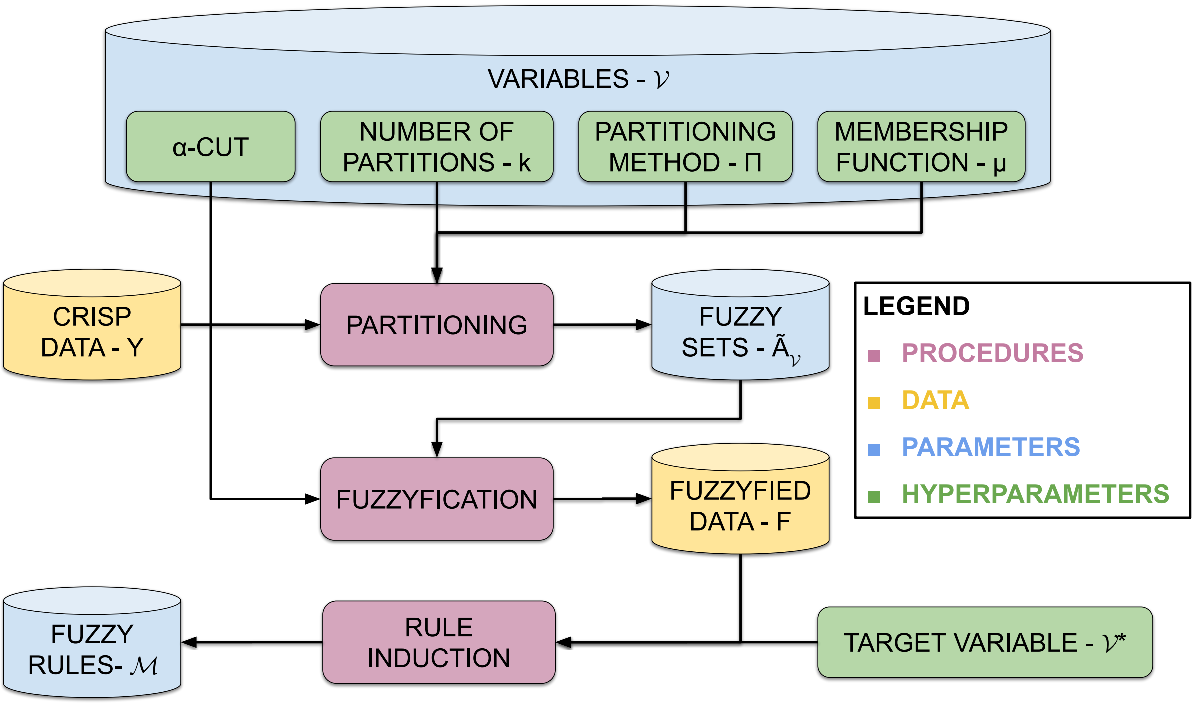


Figure 47 – MVFTS training procedure

final MVFTS model $\mathcal{M}$ consists of a set of variables $\mathcal{V}$, a fuzzy linguistic variable $\widetilde{\mathcal{V}}_i$ for each $\mathcal{V}_i \in \mathcal{V}$ and a set of weighted fuzzy rules over the linguistic variables $\widetilde{\mathcal{V}}_i$. The inputs of the training procedure are the crisp time series training data Y and the set of hyperparameters for each $\mathcal{V}_i \in \mathcal{V}$.

The forecasting procedure, explained in subsection 6.2.2 and illustrated in Figure 48, aims to produce a point estimate $\hat{y}(t+1)$ for the target variable $*\mathcal{V}$, given an input sample Y, using the linguistic variables $\widetilde{\mathcal{V}}_i$ and the induced fuzzy rules on model $\mathcal{M}$.

6.2.1 Training Procedure

Stage 1 *Partitioning*:

a) *Defining $U_{\mathcal{V}_i}$*: The Universe of Discourse $U_{\mathcal{V}_i}$ defines the sample space, i.e., the known bounds of the variable $\mathcal{V}_i$, such that $U_{\mathcal{V}_i} = [\min(Y^{\mathcal{V}_i}) - D_1, \max(Y^{\mathcal{V}_i}) + D_2]$, where $D_1 = \min(Y^{\mathcal{V}_i}) \times 0.2$ and $D_2 = \max(Y^{\mathcal{V}_i}) \times 0.2$ are used to extrapolate the known bounds as a security margin, $\forall \mathcal{V}_i \in \mathcal{V}$.

b) *$U_{\mathcal{V}_i}$ Partitioning*: Split $U_{\mathcal{V}_i}$ in k_i intervals U_j with midpoints c_j, for $j = 0..k_i$, where all the intervals have the same length;

c) *Define the linguistic variable $\widetilde{\mathcal{V}}_i$*: For each interval $U_j \in U_{\mathcal{V}_i}$ create an overlapping fuzzy set $A_j^{\mathcal{V}_i}$, with the membership function $\mu_{A_j^{\mathcal{V}_i}}$. The midpoint of the fuzzy set $A_j^{\mathcal{V}_i}$ will be c_j, the lower bound $l_j = c_{j-1}$ and the upper bound $u_j = c_{j+1}$

$\forall\ j > 0$ and $j < k_i$, and $l_0 = \min U_{\mathcal{V}_i}$, $l_k = \max U_{\mathcal{V}_i}$. Each fuzzy set $A_j^{\mathcal{V}_i}$ is a linguistic term of the linguistic variable $\widetilde{\mathcal{V}}_i$;

Stage 2 *Fuzzyfication*:

Transform the original numeric time series Y into a fuzzy time series F, where each data point $f(t) \in F$ is an $n \times k$ array with the fuzzyfied values of $y(t) \in Y$ with respect to the linguistic terms $A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i$, where the fuzzy membership is greater than the predefined α-cut, i.e., $f(t) = \{A_j^{\mathcal{V}_i} \mid \mu_{A_j^{\mathcal{V}_i}}(y(t)^{\mathcal{V}_i}) \geq \alpha_i\ \forall A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i\}$;

Stage 3 *Rule Induction*:

a) *Generate the temporal patterns*: The fuzzy temporal patterns associate the fuzzyfied values $\mathcal{V}$ to a set of possible values of the target variable $*\mathcal{V}$, such that $\mathcal{V} \rightarrow *\mathcal{V}$, whith the format $A_j^{\mathcal{V}_0}, ..., A_j^{\mathcal{V}_n} \rightarrow A_j^{*\mathcal{V}}$, where the precedent, or left hand side (LHS), is $f(t-1) = A_j^{\mathcal{V}_i}, \forall \mathcal{V}_i \in \mathcal{V}$, and the consequent, or right hand side (RHS), is $f(t+1) = A_j^{*\mathcal{V}}$, $A_j^{*\mathcal{V}} \in \widetilde{*\mathcal{V}}$.

b) *Generate the rule base*: Select all temporal patterns with the same precedent and group their consequent sets creating a rule with the format $\mathcal{V} \rightarrow w_k \cdot A_k^{*\mathcal{V}}, w_j \cdot A_j^{*\mathcal{V}}, ...$, where the LHS is $f(t-1) = A_j^{\mathcal{V}_i}, \forall \mathcal{V}_i \in \mathcal{V}$ and the RHS is $f(t+1) \in \{A_k^{*\mathcal{V}}, A_j^{*\mathcal{V}}, ...\}$. Each rule can be understood as the weighted set of possibilities which may happen on time $t+1$ (the consequent) when a certain precedent $A_{i0}, ..., A_{i\Omega}$ is identified on previous lag (the precedent).

6.2.2 Forecasting Procedure

Step 1 *Fuzzyfication*: Compute the membership grade μ_{ji} for $y(t-1) \in Y$ such that $\mu_{ji} = \mu_{A_j^{\mathcal{V}_i}}(y(t-1))$, for each $A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i$, for each $\mathcal{V}_i \in \mathcal{V}$;

Step 2 *Rule matching*: Select the K rules where all fuzzy sets $A_j^{\mathcal{V}_i}$ on the LHS, for each $\mathcal{V}_i \in \mathcal{V}$, have $\mu_{ji} > \alpha_i$; The rule fuzzy membership grade is shown below, using the minimum function as T-norm.

$$\mu_q = \bigcap_{j \in \widetilde{\mathcal{V}}_i\ ;\ i \in \mathcal{V}} \mu_{ji} \tag{6.1}$$

Step 3 *Rule mean points*: For each selected rule q, compute the mean point mp_q of the target variable $*\mathcal{V}$ as below, where c_j is the c parameter of the μ function from fuzzy set $A_j^{*\mathcal{V}}$:

$$mp_q = \sum_{j \in *\widetilde{\mathcal{V}}_i} c_j \tag{6.2}$$

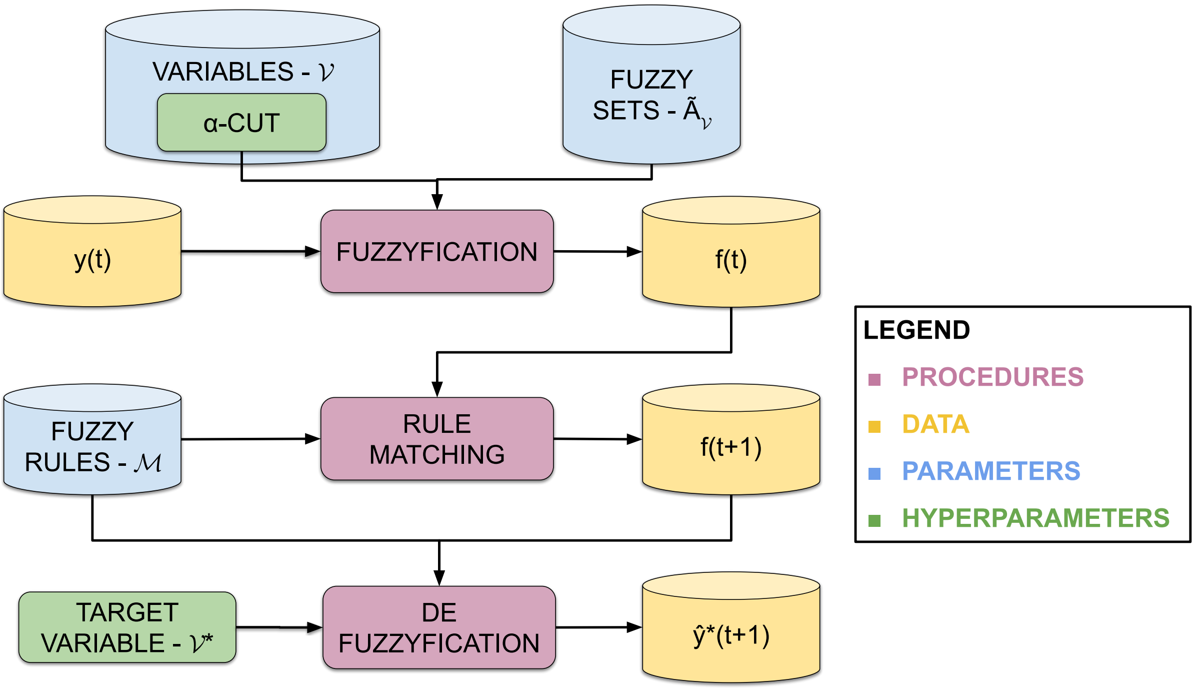

VARIABLES - $\mathcal{V}$
α-CUT
FUZZY SETS - $\tilde{A}_{\mathcal{V}}$
y(t)
FUZZYFICATION
f(t)
FUZZY RULES - $\mathcal{M}$
RULE MATCHING
f(t+1)
TARGET VARIABLE - $\mathcal{V}$*
DE FUZZYFICATION
ŷ*(t+1)
LEGEND
■ PROCEDURES
■ DATA
■ PARAMETERS
■ HYPERPARAMETERS

Figure 48 – MVFTS forecasting procedure

Step 4 *Defuzzyfication*: Compute the forecast as the weighted sum of the rule mid-points mp_q by their membership grades μ_q for each selected rule j:

$$\hat{y}(t+1) = \frac{\sum_{q \in K} \mu_q \cdot mp_q}{\sum_{q \in K} \mu_q} \tag{6.3}$$

6.2.3 Interval forecasting for MVFTS

The MVFTS model can be used for interval forecasting following the same approach of $[\mathbb{I}]FTS$ method. For this it is needed to change the Steps 3 and 4 of the forecasting procedure presented in Section 6.2.2, as presented below:

Step 3 *Rule intervals*: For each selected rule q, compute the interval $\mathbb{I}^q$ of the target variable $*\mathcal{V}$ as below, where $\underline{A_j^{*\mathcal{V}}}$ and $\overline{A_j^{*\mathcal{V}}}$ are respectively the lower and upper bounds of the target fuzzy sets $A_j^{*\mathcal{V}}$:

$$\mathbb{I}^q = [\underline{\mathbb{I}^q_{min}}, \overline{\mathbb{I}^q_{max}}] \tag{6.4}$$

$$\underline{\mathbb{I}^q_{min}} = \min(\underline{A_j^{*\mathcal{V}}} \in *\widetilde{\mathcal{V}}_i) \tag{6.5}$$

$$\overline{\mathbb{I}^q_{max}} = \max(\overline{A_j^{*\mathcal{V}}} \in *\widetilde{\mathcal{V}}_i) \tag{6.6}$$

$$\tag{6.7}$$

Step 4 *Defuzzyfication*: Compute the prediction interval as the extrema of the rule intervals

$\mathbb{I}_q$ by their membership grades μ_q for each selected rule j:

$$\mathbb{I}(t+1) = \frac{\sum_{j \in *\widetilde{\mathcal{V}}_i} \mu_q \mathbb{I}^q}{\sum_{q \in \mathcal{V}} \mu_q} = \frac{\sum_{q \in \mathcal{V}} [\mu_q \underline{\mathbb{I}^q_{min}}, \mu_q \overline{\mathbb{I}^q_{max}}]}{\sum_{q \in \mathcal{V}} \mu_q} \tag{6.8}$$

6.2.4 Weighted Multivariate FTS - WMVFTS

A simple extension of MVFTS to embody weights in its rules can be achieved by changing Stage 3.b of the training procedure presented in Section 6.2.1, where the new step is:

Stage 3.b) *Generate the rule base*: Select all temporal patterns with the same precedent and group their consequent sets creating a rule with the format $\mathcal{V} \rightarrow w_k \cdot A_k^{*\mathcal{V}}, w_j \cdot A_j^{*\mathcal{V}}, ...,$ where the LHS is $f(t-1) = A_j^{\mathcal{V}_i}, \forall \mathcal{V}_i \in \mathcal{V}$ and the RHS is $f(t+1) \in \{A_k^{*\mathcal{V}}, A_j^{*\mathcal{V}}, ...\}$ and the weights $w_j, w_k, ...$ are the normalized frequencies of each temporal pattern such that:

$$w_i = \frac{\#A_j^{*\mathcal{V}}}{\#RHS} \quad \forall A_j^{*\mathcal{V}} \in RHS \tag{6.9}$$

where $\#A_i$ is the number of occurrences of A_i on temporal patterns with the same precedent LHS and $\#RHS$ is the total number of temporal patterns with the same precedent LHS.

It is also need to change the Step 3 of the forecasting procedure presented in Section 6.2.2, as presented below:

Step 3 *Rule mean points*: For each selected rule q, compute the mean point mp_q of the target variable $*\mathcal{V}$ as below, where c_j is the c parameter of the μ function from fuzzy set $A_j^{*\mathcal{V}}$:

$$mp_q = \sum_{j \in *\widetilde{\mathcal{V}}_i} w_j \cdot c_j \tag{6.10}$$

For the interval forecasting method proposed in Section 6.2.3, a new approach is adopted to create the fuzzy rule intervals, as presented below:

$$\mathbb{I}^q = [\underline{\mathbb{I}^q_{min}}, \overline{\mathbb{I}^q_{max}}] \tag{6.11}$$

$$\underline{\mathbb{I}^q_{min}} = \sum_{j \in *\widetilde{\mathcal{V}}_i} w_j \cdot \underline{A_j^{*\mathcal{V}}} \tag{6.12}$$

$$\overline{\mathbb{I}^q_{max}} = \sum_{j \in *\widetilde{\mathcal{V}}_i} w_j \cdot \overline{A_j^{*\mathcal{V}}} \tag{6.13}$$

MVFTS and WMVFTS methods take separated partitionings for each variable and its rules contains references for the different variables. In next section a simple approach is proposed for transforming multivariate time series in monovariate ones, allowing the use of monovariate FTS methods to tackle multivariate time series.

6.3 Fuzzy Information Granule Fuzzy Time Series $\mathcal{FIG}$-FTS

The Fuzzy Information Granule Fuzzy Time Series ($\mathcal{FIG}$-FTS) is a wrapper model which enables a monovariate model (PWFTS) to tackle multivariate time series. In addiction to extending PWFTS features for multivariate data, the $\mathcal{FIG}$-FTS also appends the multivariate forecasting capability, acting as a Multiple Input/Multiple Output (MIMO) method, where all variables are both targets and explanatory variables.

The aim of $\mathcal{FIG}$-FTS is to replace the Partitioning and Fuzzyfication stages of the PWFTS training procedure detailed in Section 4.2, the Fuzzyfication step of the forecasting procedure detailed in Section 4.3 and appends the multivariate forecasting to the extensions presented in Section 4.4.

Given an n-variate time series $Y = (y_1(t), \dots, y_n(t)), t = 0 \dots, T$, corresponding variables $\mathcal{V}_i$ are defined for each $y_i(t)$. The resulting fuzzy time series F is then composed by data points $f(t) \in F$ that represent a sequence of fuzzy information granules $\mathcal{G}_i$. Each granule contains a set of fuzzy linguistic variables $\widetilde{\mathcal{V}}_i$ related to each variable $\mathcal{V}_i$.

The training procedure, described in Section 6.3.1 and illustrated in Figure 49, performs the multivariate partitioning, fuzzyfication and then feeds PWFTS with the fuzzyfied data, whose is responsible for rule induction. The final $\mathcal{FIG}$-FTS model $\mathcal{M}$ consists of a set of variables $\mathcal{V}$, a fuzzy linguistic variable $\widetilde{\mathcal{V}}_i$ for each $\mathcal{V}_i \in \mathcal{V}$, a fuzzy information granule set $\mathcal{FIG}$ and a set of probabilistic weighted high order fuzzy rules over the information granules $\mathcal{G}_i \in \mathcal{G}$. The training procedure employ the hyperparameters listed in Table 35

In the training method the partitioning of each variable is independent from the others. Each variable has its own linguistic variable $\widetilde{\mathcal{V}}_i$. For this it is necessary to inform, for each chosen variable $\mathcal{V}_i \in \mathcal{V}$ on Y, the hyperparameters k_i, μ and α. The order of the model is controlled by the parameter Ω and the lag indexes are controlled by the parameter L.

The global linguistic variable $\mathcal{FIG}$ is the union of all Fuzzy Information Granules $\mathcal{G}_i$, which in turn are the combination of one fuzzy set for each variable, such that $\mathcal{G}_i = \{A_j^{\mathcal{V}_i}\}, \forall \mathcal{V}_i \in \mathcal{V}$ and its membership function is given by $\mu_{\mathcal{G}_i} = \bigcap \mu_{A_j^{\mathcal{V}_i}}$, where $\bigcap$ is the minimum T-norm. The $\mathcal{FIG}$ set is indexed by the midpoints of its internal fuzzy sets, enabling optimized spatial search using KD-trees. With the linguistic variable $\mathcal{FIG}$ the fuzzyfication process transforms each multivariate data point $y(t) \in Y$ into a $\mathcal{G}_i \in \mathcal{FIG}$,

Alias	Parameter	Type	Description
k_i	Number of partitions	$\mathbb{N}^+$	The number of fuzzy sets that will be created in the linguistic variable $\widetilde{\mathcal{V}}_i$
μ	Membership function	$\mu : U \to [0,1]$	A function that measure the membership of a value $y \in U$ to a fuzzy set
α	α-cut	$[0,1]$	The minimal membership grade to take account on fuzzyfication process
Ω	Order	$\mathbb{N}^+$	The number of past lags used in the precedent of each fuzzy rule
L	Lags		A vector of the past lag indexes, with length Ω and $1 \leq L[i] < L[i+1]$ for $t = 0..\Omega$
κ	k-nearest neighbors	$\mathbb{N}^+$	The number of nearest neighbors that the spatial index search on $\mathcal{FIG}$ during the fuzzyfication process

Table 35 – $\mathcal{FIG}$-FTS hyperparameters

such that $f(t) = \mathcal{G}_i$.

The forecasting procedure, explained in subsection 6.3.2 and illustrated in Figure 50, aims to produce a point estimate $\hat{y}(t+1)$ for each variable $\mathcal{V}$, given an input sample Y, using the linguistic variable $\mathcal{FIG}$ and the induced fuzzy rules on model $\mathcal{M}$.

The rule matching procedure can become computationally expensive as the size of the rule base $\mathcal{M}$ grows. Because of this it is advisable that implementations of this model use spatial trees [Muja and Lowe, 2014] to index the rules with the midpoints of each fuzzy set on their LHS, optimizing the search for applicable rules during the forecasting step. This work used the KD-tree implementation of the Scipy Spatial package[1].

The global parameter κ is related with the spatial index search on $\mathcal{FIG}$, and indicates how many $\mathcal{G}_i \in \mathcal{FIG}$ are returned for a given crisp multivariate data point. This parameter has influence on the sensibility and the diversity of the rules considered during the forecasting procedure, such that as κ increases more rules will be accounted on. If $\kappa = 1$, just the closest rule (the rule with the highest membership degree) will be used.

6.3.1 Training Procedure

Stage 1 *Partitioning*:

a) *Defining* $U_{\mathcal{V}_i}$: The Universe of Discourse $U_{\mathcal{V}_i}$ defines the sample space, i.e., the known bounds of the variable $\mathcal{V}_i$, such that $U_{\mathcal{V}_i} = [\min(Y^{\mathcal{V}_i}) - D_1, \max(Y^{\mathcal{V}_i}) + D_2]$, where $D_1 = \min(Y^{\mathcal{V}_i}) \times 0.2$ and $D_2 = \max(Y^{\mathcal{V}_i}) \times 0.2$ are used to extrapolate the known bounds as a security margin, $\forall \mathcal{V}_i \in \mathcal{V}$.

[1] https://docs.scipy.org/doc/scipy/reference/spatial.html. Access in 2019-04-29.

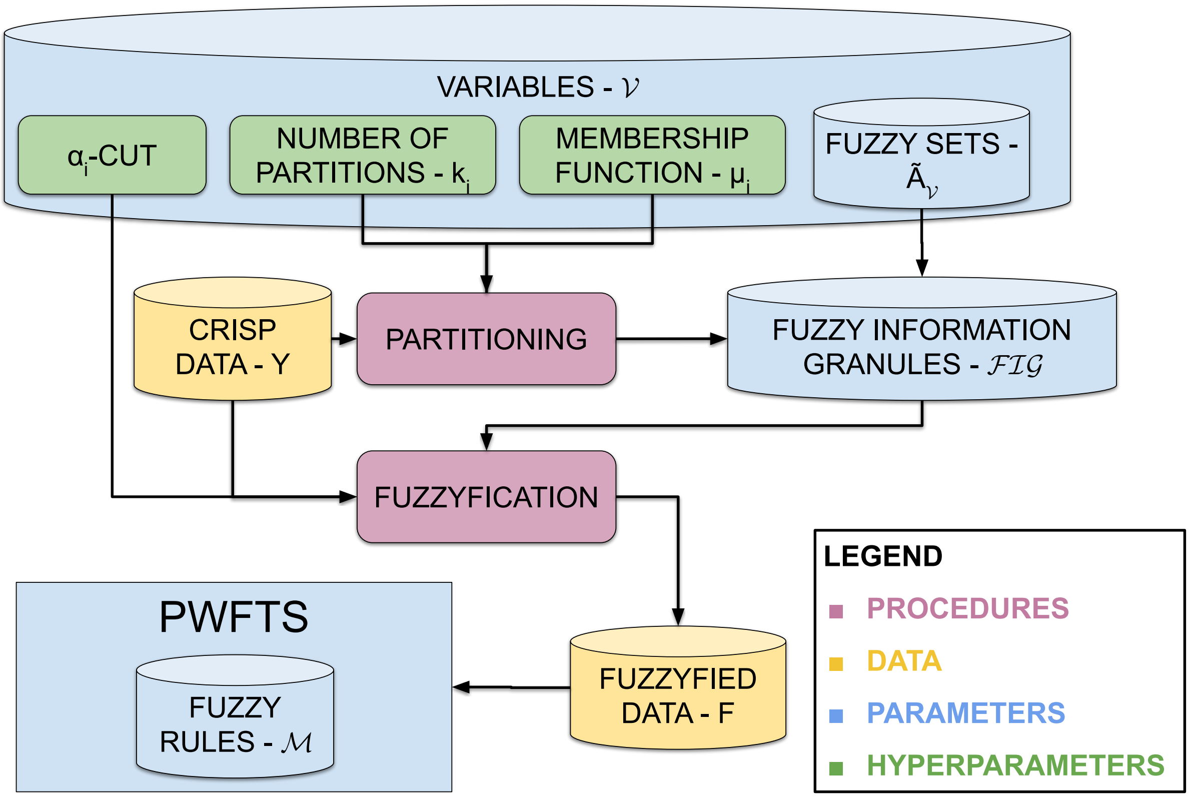

VARIABLES - $\mathcal{V}$
α_i-CUT
NUMBER OF PARTITIONS - k_i
MEMBERSHIP FUNCTION - μ_i
FUZZY SETS - $\tilde{A}_{\mathcal{V}}$
CRISP DATA - Y
PARTITIONING
FUZZY INFORMATION GRANULES - $\mathcal{FIG}$
FUZZYFICATION
PWFTS
FUZZY RULES - $\mathcal{M}$
FUZZYFIED DATA - F
LEGEND
■ PROCEDURES
■ DATA
■ PARAMETERS
■ HYPERPARAMETERS

Figure 49 – $\mathcal{FIG}$-FTS training procedure

b) $U_{\mathcal{V}_i}$ *Partitioning*: Split $U_{\mathcal{V}_i}$ in k_i intervals U_j with midpoints c_j, for $j = 0..k_i$, where all the intervals have the same length;

c) *Define the linguistic variable* $\widetilde{\mathcal{V}}_i$: For each interval $U_j \in U_{\mathcal{V}_i}$ create an overlapping fuzzy set $A_j^{\mathcal{V}_i}$, with the membership function $\mu_{A_j^{\mathcal{V}_i}}(y_{\mathcal{V}_i}(t))$, where $y_{\mathcal{V}_i}(t)$ is the value of the $\mathcal{V}_i$ variable on instance $y(t) \in Y$. The midpoint of the fuzzy set $A_j^{\mathcal{V}_i}$ will be c_j, the lower bound $l_j = c_{j-1}$ and the upper bound $u_j = c_{j+1}$ $\forall\ j > 0$ and $j < k_i$, and $l_0 = \min U_{\mathcal{V}_i}$, $l_k = \max U_{\mathcal{V}_i}$. Each fuzzy set $A_j^{\mathcal{V}_i}$ is a linguistic term of the linguistic variable $\widetilde{\mathcal{V}}_i$;

Stage 2 *Fuzzyfication*:

Transform the original numeric time series Y into a fuzzy time series F, where each data point $f(t) \in F$ is a $\mathcal{G}_i \in \mathcal{FIG}$. For each $y(t) \in Y$ the following steps must be executed:

a) *Individual variable fuzzyfication*: For each variable $\mathcal{V}_i \in \mathcal{V}$, find the linguistic terms $A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i$, where the fuzzy membership is greater than the predefined α-cut, i.e., $f_{\mathcal{V}_i}(t) = \{A_j^{\mathcal{V}_i} \mid \mu_{A_j^{\mathcal{V}_i}}(y_{\mathcal{V}_i}(t)) \geq \alpha_i\ \forall A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i\}$;

b) *Search in $\mathcal{FIG}$*: For each combination of fuzzy sets $A_j^{\mathcal{V}_i}$ in $f_{\mathcal{V}_i}(t)$ verify if there is a $\mathcal{G}_i \in \mathcal{G}$ where $\mathcal{G}_i \supset \{A_j^{\mathcal{V}_i}\}, \forall A_j^{\mathcal{V}_i} \in f_{\mathcal{V}_i}(t)$. If it exists, then the fuzzyfied value

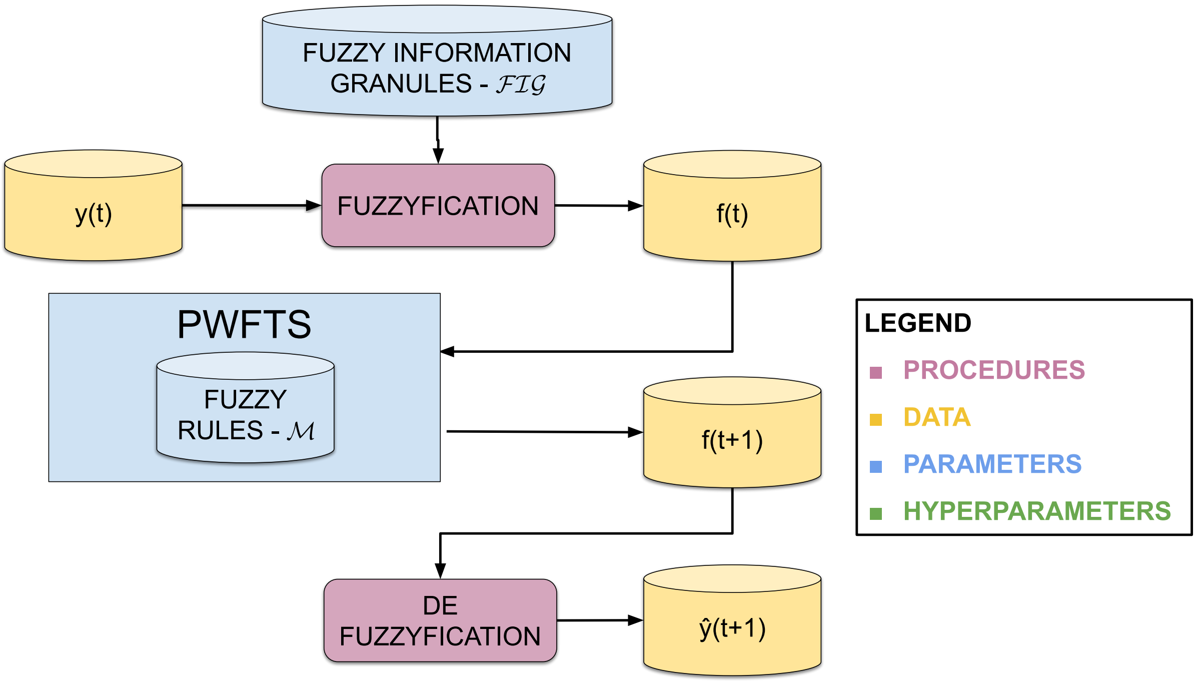


Figure 50 – $\mathcal{FIG}$-FTS forecasting procedure

of $y(t)$ is $\mathcal{G}_i$. This search is performed with KD-trees, comparing the midpoints of the fuzzyfied data and the midpoints of the fuzzy sets in the $\mathcal{FIG}$.

c) *Create new $\mathcal{G}_i$ in $\mathcal{FIG}$*: If no $\mathcal{G}_i$ was found in the previous step, new ones are added to $\mathcal{FIG}$. For each combination of fuzzy sets $A_j^{\mathcal{V}_i}$ in $f_{\mathcal{V}_i}(t)$ create a fuzzy information granule $\mathcal{G}_i$ such that $\mathcal{G}_i = \{A_j^{\mathcal{V}_i}\}, \forall A_j^{\mathcal{V}_i} \in f_{\mathcal{V}_i}(t)$ and $\mu_{\mathcal{G}_i} = \bigcap \mu_{A_j^{\mathcal{V}_i}}$. The created $\mathcal{G}_i$ is then the fuzzyfied value of $y(t)$.

Stage 3 *Rule Induction*:

a) The fuzzyfied data F where $f(t) = [(\mathcal{G}_0, \mu_{\mathcal{G}_0}), \ldots, (\mathcal{G}_i, \mu_{\mathcal{G}_i})]$ is passed to the Rule Induction stage of PWFTS method, which create the PWFTPG model $\mathcal{M}$. Each high-order PWFTG rule now will have the format $\pi_j \mathcal{G}_{i0}, ..., \mathcal{G}_{i\Omega} \rightarrow w_{j0} \cdot \mathcal{G}_{i0}, \ldots w_{ji} \cdot \mathcal{G}_{ji}$, where the LHS is $f(t - L(\Omega)) = \mathcal{G}_{i0}$, $f(t - L(\Omega - 1)) = \mathcal{G}_{i1}$, ..., $f(t - L(0)) = \mathcal{G}_{i\Omega}$ and the RHS is $f(t+1) \in \{\mathcal{G}_k, \mathcal{G}_j, ...\}$ and the weights π_j, w_{jk} are the fuzzy empirical probabilities.

6.3.2 Forecasting Procedure

Step 1 *Fuzzyfication*: Find the κ closest $\mathcal{G}_{i\Omega}, ..., \mathcal{G}_{i0} \in \mathcal{FIG}$ to the input sample $y(t - \Omega), ..., y(t)$.

Step 2 *Rule matching*: The fuzzyfied input sample is transfered to PWFTS that search for the applicable rules. For each PWFTPG rule j found, its fuzzy membership is given by:

$$\mu_j = \bigcap_{t \in L} \bigcap_{i \in \mathcal{FIG}} \mu_{\mathcal{G}_{it}} \quad (6.14)$$

Step 3 *Defuzzyfication*:

a) *Target variable selection*: For non multivariate forecasts a target variable $*\mathcal{V}$ must be chosen between the variables $\mathcal{V}$.

b) *PWFTPG adaption*: After the target variable $*\mathcal{V}$ be selected, the RHS of all PWFTPG's are modified, replacing the $figi$ by the fuzzy sets $A_j^{*\mathcal{V}} \in *\widetilde{\mathcal{V}}_i$, keeping the weights untouched;

c) *Deffuzyfication*: The PWFTS deffuzyfication methods (point, interval and probabilistic) can be invoked without modifications;

Step 4 *Multivariate forecasting*:

a) If a target variable was not specified, compute a point forecast $\hat{y}^{\mathcal{V}_i}(t+1)$ invoking the PWFTS point forecasting by taking each variable $\mathcal{V}_i \in \mathcal{V}$ as a target variable;

b) Merge the individual variable forecastings to create the multivariate forecast $\hat{y}(t+1) = \bigcup_{\mathcal{V}_i \in \mathcal{V}} \hat{y}^{\mathcal{V}_i}(t+1)$

Step 5 *Forecasting horizon*: Given the number m of steps ahead to forecast (the forecast horizon), repeat the Steps 1 to 4 m times, appending the output $\hat{y}$ of the previous Step 4 at the end of the $y(t)$ input for the next Step 1.

6.3.3 Method discussion

The main insight of the $\mathcal{FIG}$-FTS is that the linguistic variables $\widetilde{\mathcal{V}}_i$ work as feature extraction layers and each fuzzy information granule $\mathcal{G}$ is a small cluster prototype of these features, simplifying the representation of temporal patterns and aiding pattern identification and rule induction. Each $\mathcal{G}$ also helps the multivariate defuzzyfication process, working as a final output layer for the model.

The learning procedure of $\mathcal{FIG}$-FTS is controlled by its hyperparameters that directly affect its accuracy and parsimony. The number of partitions of each variable k_i affects the number of rules directly, given the maximum number of rules (in the worst case) is a Cartesian product of the fuzzy sets $A_j^{\mathcal{V}_i} \in \widetilde{\mathcal{V}}_i$, for each $\mathcal{V}_i \in \mathcal{V}$. The α_i-cut, on the other hand, controls the fuzzyfication sensibility by eliminating, in the rule induction stage, values with lower membership grades. It reduces the number of rules by preventing the capture of spurious patterns, generated by insignificant memberships or noise. The α_i-cut also enhances the forecasting process by eliminating lower related rules on rule search.

The parameter κ has influence on the forecasting accuracy. There is also a balance between the use of too few or too many rules on forecasting procedure, such that too few rules may not have enough patterns to describe the correct time series behavior and too many may bring patterns that are not closely related with the current behavior.

In the next section the empirical results of the proposed method are presented, showing its effectiveness for complex artificial and natural dynamic processes.

6.4 Computational Experiments

This section presents an exploratory study of multivariate FTS methods and $\mathcal{FIG}$-FTS. The computational experiments employed two multivariate time series, the SONDA dataset with 2,000,000 instances and the Malaysia dataset, with 17,000 instances. Both datasets are detailed in Appendix B, where its main characteristics are presented.

The multivariate models were testes for point, interval and probabilistic forecasting (in the case of $\mathcal{FIG}$-FTS) using the presented FTS methods as competitor models. For each dataset, with exception to timestamp variables, each variable $\mathcal{V}_i \in \mathcal{V}$ was used as target variable $*\mathcal{V}$ once, allowing the comparation with the monovariate FTS methods.

In order to optimize the forecasting accuracy an specific configuration of variables was researched for each $*\mathcal{V} \in \mathcal{V}$, and it is shared among MVFTS, WMVFTS and $\mathcal{FIG}$-FTS. The specific values of k_i, μ_i and α_i for each variable $\mathcal{V}_i \in \mathcal{V}$ were obtained using DEHO method on the isolated variables.

In subsections 6.4.1 and 6.4.2 the details about the variables of the multivariate methods are presented. In subsection 6.4.3 the results of the experiments are presented for point, interval and probabilistic forecasting, and samples of methods performances are provided.

In order to contribute with the replication of all the results in the research, all data and source codes employed in this chapter are available at the URL: `http://bit.ly/scalable_probabilistic_fts_chap6`

6.4.1 SONDA models settings

The SONDA dataset is composed by 3 variables `DateTime` (timestap of each instance), `glo_avg` (solar radiation) and `ws_10m` (wind speed). The details of this dataset and its variables are presented in Appendix B.

The Solar Radiation variable, is independent to Wind Speed variable and then this last can be discard. The Solar Radiation has two main seasonal components: yearly and hourly. These two seasonalities can be extracted from the DateTime variable. A model to forecast the Solar Radiation variable based on SONDA multivariate dataset contains the set up presented in Table 36 and illustrated in Figure 51.

$\mathcal{V}_\mathrm{i}$	**Data Source**	$\mathbf{k_i}$	μ_i	α_i
Hour	DateTime	24	Triangular	.3
Month	DateTime	12	Triangular	.3
Solar	glo_avg	5	Gaussian	.25

Table 36 – Variables and partitioning for SONDA Solar Radiation

The Wind Speed variable is independent to Solar Radiation variable and then this last can be discard. The Wind Speed has a yearly seasonal components that can be extracted from the DateTime variable. A model to forecast the Wind Speed variable based on SONDA multivariate dataset contains the set up presented in Table 37 and illustrated in Figure 52.

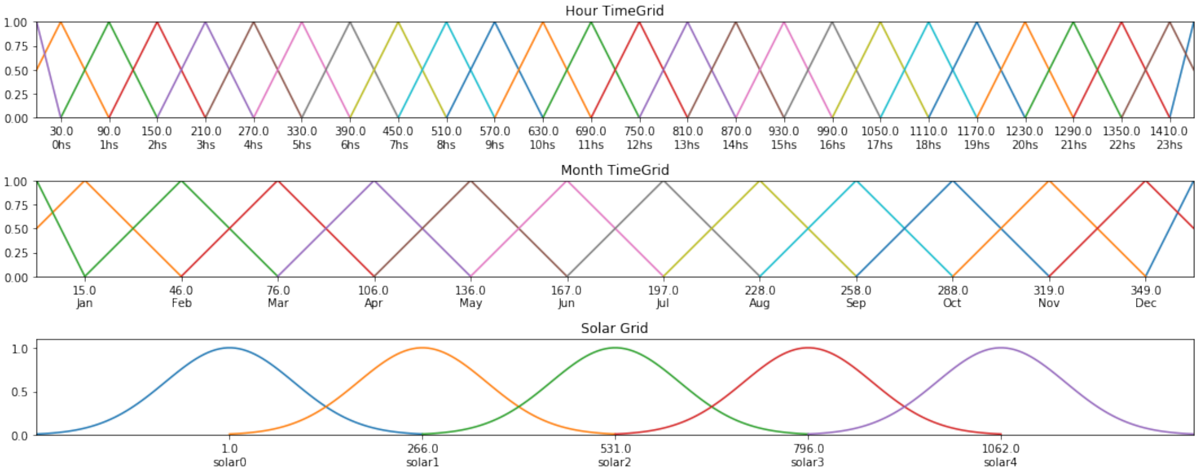


Figure 51 – Variables and partitioning for SONDA Solar Radiation

$\mathcal{V}_\mathrm{i}$	**Data Source**	$\mathbf{k_i}$	μ_i	α_i
Month	DateTime	12	Triangular	.3
Wind	ws_10m	15	Gaussian	.25

Table 37 – Variables and partitioning for SONDA Solar Radiation

6.4.2 Malaysia models settings

The Malaysia dataset is composed by 3 variables `DateTime` (timestap of each instance), `temperature` and `load` (electric load). The details of this dataset are presented in Appendix B.

The Load variable is a hourly seasonal variable (e. g. dependent of DateTime variable) and also known to be dependent of the temperature variable. A model to forecast

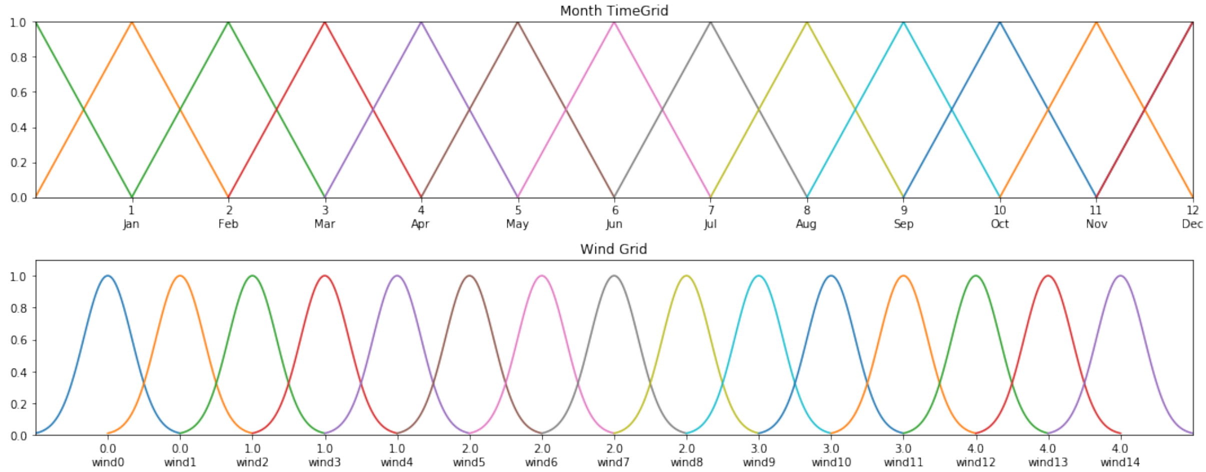


Figure 52 – Variables and partitioning for SONDA Wind Speed

$\mathcal{V}_\mathbf{i}$	**Data Source**	$\mathbf{k_i}$	$\mu_\mathbf{i}$	$\alpha_\mathbf{i}$
Hour	DateTime	24	Triangular	.3
Temperature	temperature	10	Gaussian	.3
Load	load	10	Gaussian	.25

Table 38 – Variables and partitioning for SONDA Solar Radiation

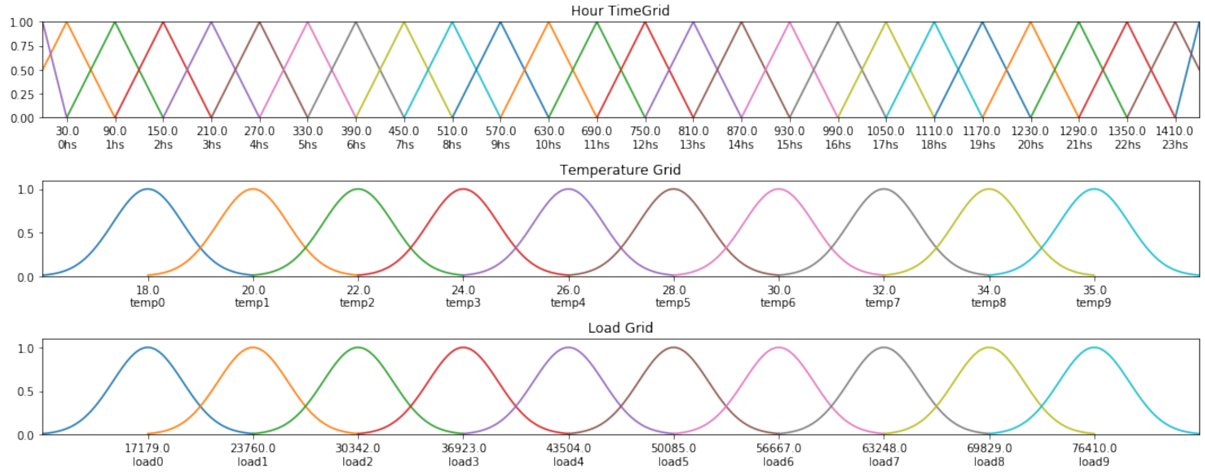


Figure 53 – Variables and partitioning for Malaysia Eletric Load

the Load variable based on Malaysia multivariate dataset contains the set up presented in Table 38 and illustrated in Figure 53.

The Temperature variable, in other hand, is independent in relation of Load variable and then it can be discarded. The temperature has two main seasonal components: yearly and hourly. These two seasonalities can be extracted from the DateTime variable. A model to forecast the Temperature variable based on Malaysia multivariate dataset contains the set up presented in Table 39 and illustrated in Figure 54.

$\mathcal{V}_\mathbf{i}$	**Data Source**	$\mathbf{k_i}$	μ_i	α_i
Hour	DateTime	24	Triangular	.3
Month	DateTime	12	Triangular	.3
Temperature	temperature	10	Gaussian	.3

Table 39 – Variables and partitioning for Malaysia Temperature

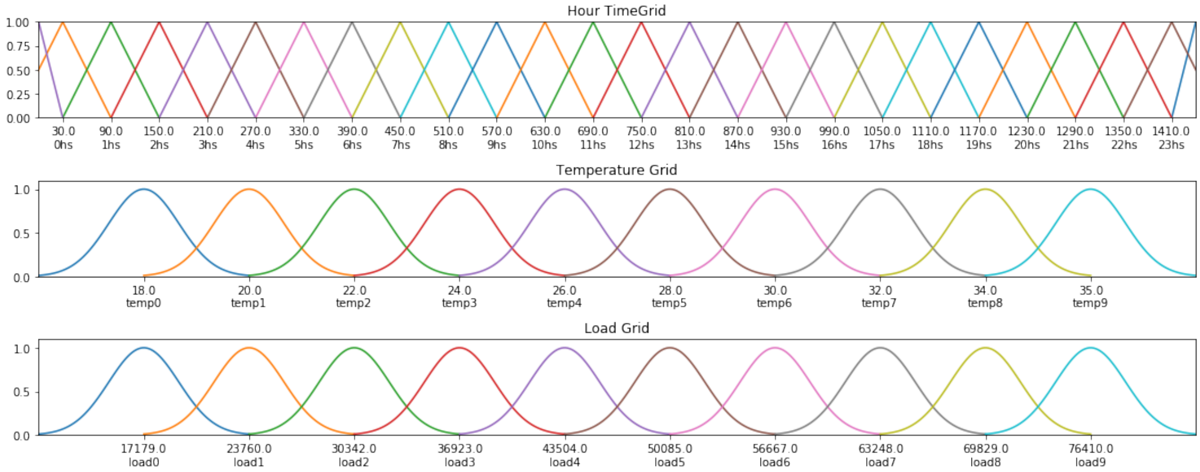


Figure 54 – Variables and partitioning for Malaysia Temperature

6.4.3 Results

The RMSE accuracy for one step ahead point forecasting is presented in Figure 55, by method and dataset. Samples of the multivariate methods point forecasting performance are also illustrated in Figure 56 for one step ahead forecasting and in Figure 57 for many steps ahead forecasting.

The interval forecasting accuracy using the Winkler Score metric, for one step ahead is presented in Figure 58, by method and dataset. Samples of the multivariate methods interval forecasting performance are also illustrated in Figure 56 for one step ahead forecasting and in Figure 57 for many steps ahead forecasting.

The probabilistic forecasting accuracy using the CRPS metric, for one step ahead, is presented in Figure 59, by method and dataset. Samples of $\mathcal{FIG}$-FTS probabilistic forecasting performance are also illustrated in Figures 60 and 61 for one step ahead forecasting and in Figures 62 and 63 for many steps ahead forecasting.

6.5 Conclusion

Accurate forecasting of complex dynamics systems, as several natural and social processes, is a challenging task specially when the underlying system is composed by many interacting variables. For FTS methods, dealing with multivariate and spatio-temporal time series was always a challenging task, specially because of the complexity growth of

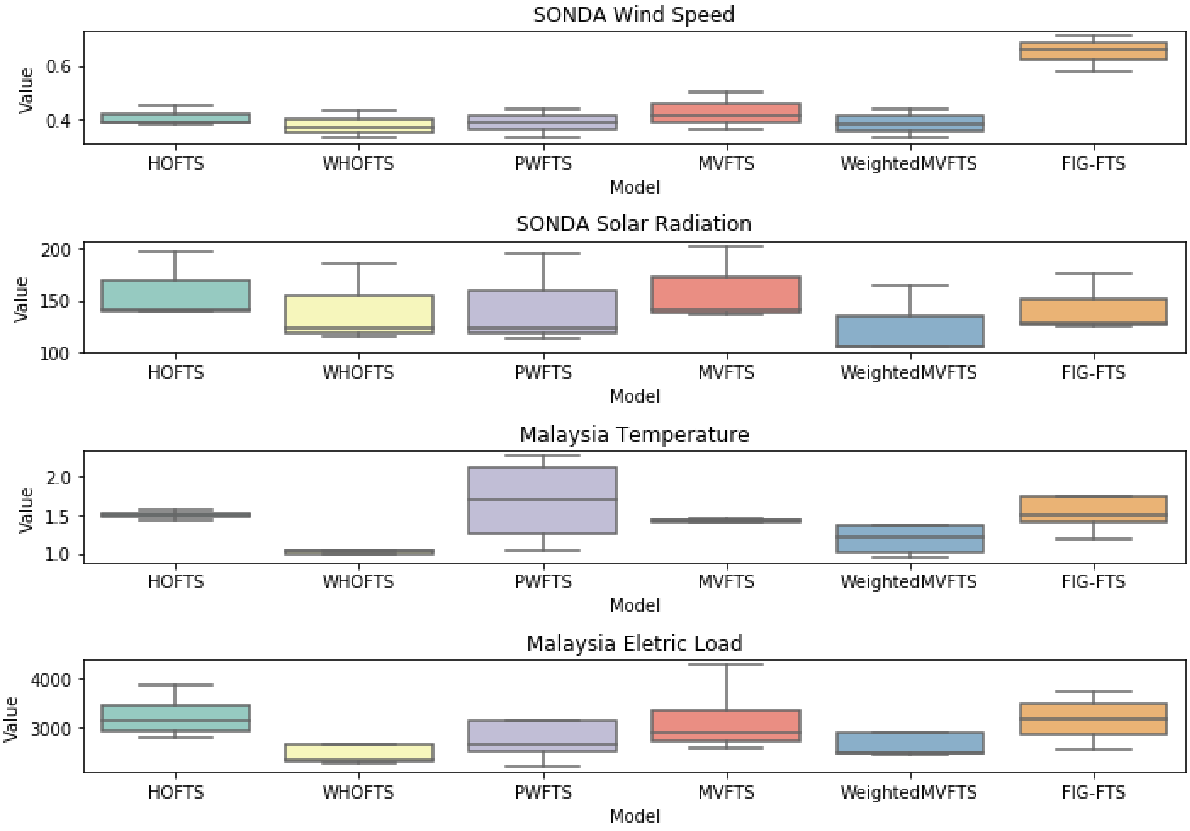


Figure 55 – RMSE point forecasting accuracy for one step ahead

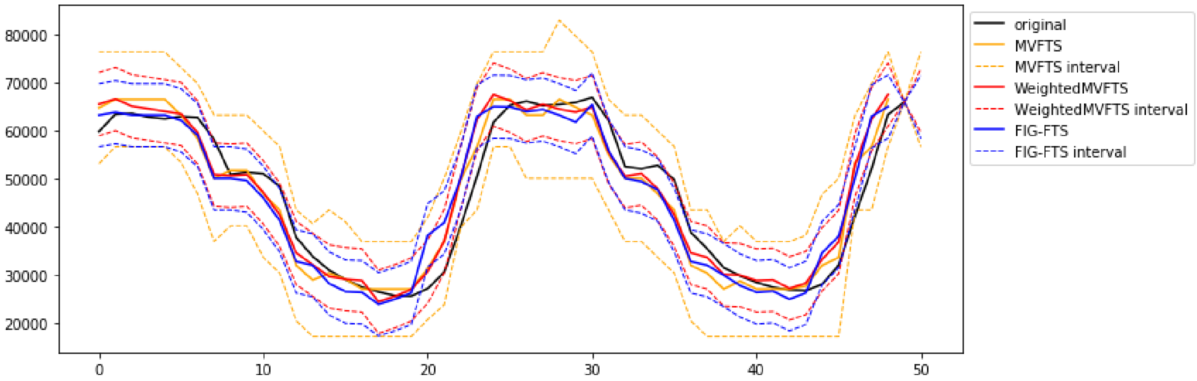


Figure 56 – Sample of point and interval forecasts for one step ahead by multivariate method

the rules as the number of variables increases.

This section presented a short overview of multivariate FTS methods, focusing on the rule based conventional Multivariate Fuzzy Time Series (MVFTS) and its weighted version WMVFTS.

In order to extend the PWFTS method to multivariate time series, the method Fuzzy Information Granule FTS (FIG-FTS) was proposed. FIG-FTS is a wrapper method that pre-process the multivariate input translating it onto a monovariate and allowing

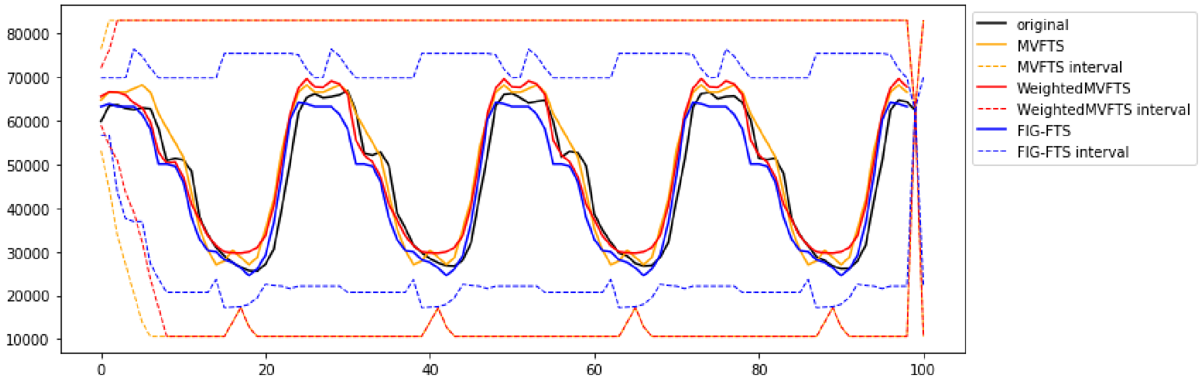


Figure 57 – Sample of point and interval forecasts for many steps ahead by multivariate method

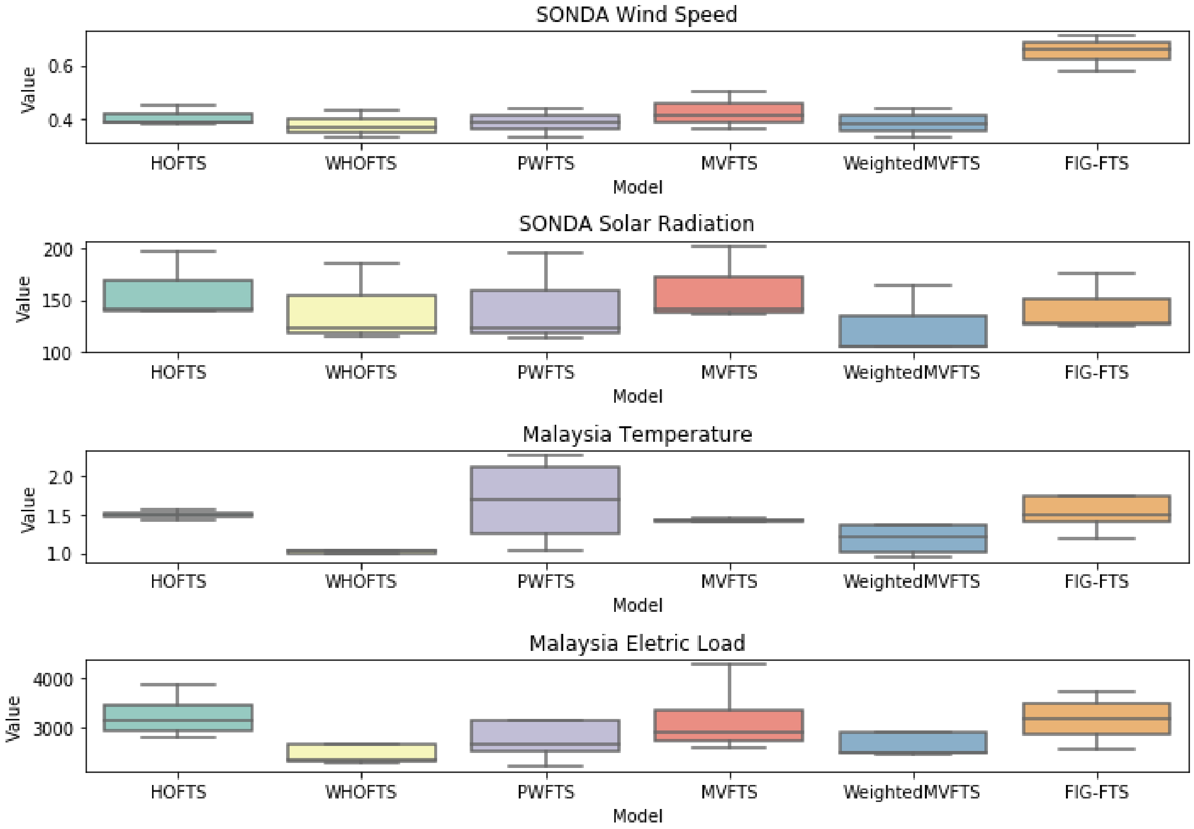


Figure 58 – Winkler interval forecasting accuracy for one step ahead

its use by monovariate methods. $\mathcal{FIG}$-FTS makes use of Fuzzy Information Granules (FIG), which in this work is a multivariate fuzzy set incrementally created during the fuzzyfication stage.

With $\mathcal{FIG}$-FTS, the PWFTS extends its foreasting capabilities to multivariate data, being the first multivariate FTS method to forecast points, intervals and probability distributions.

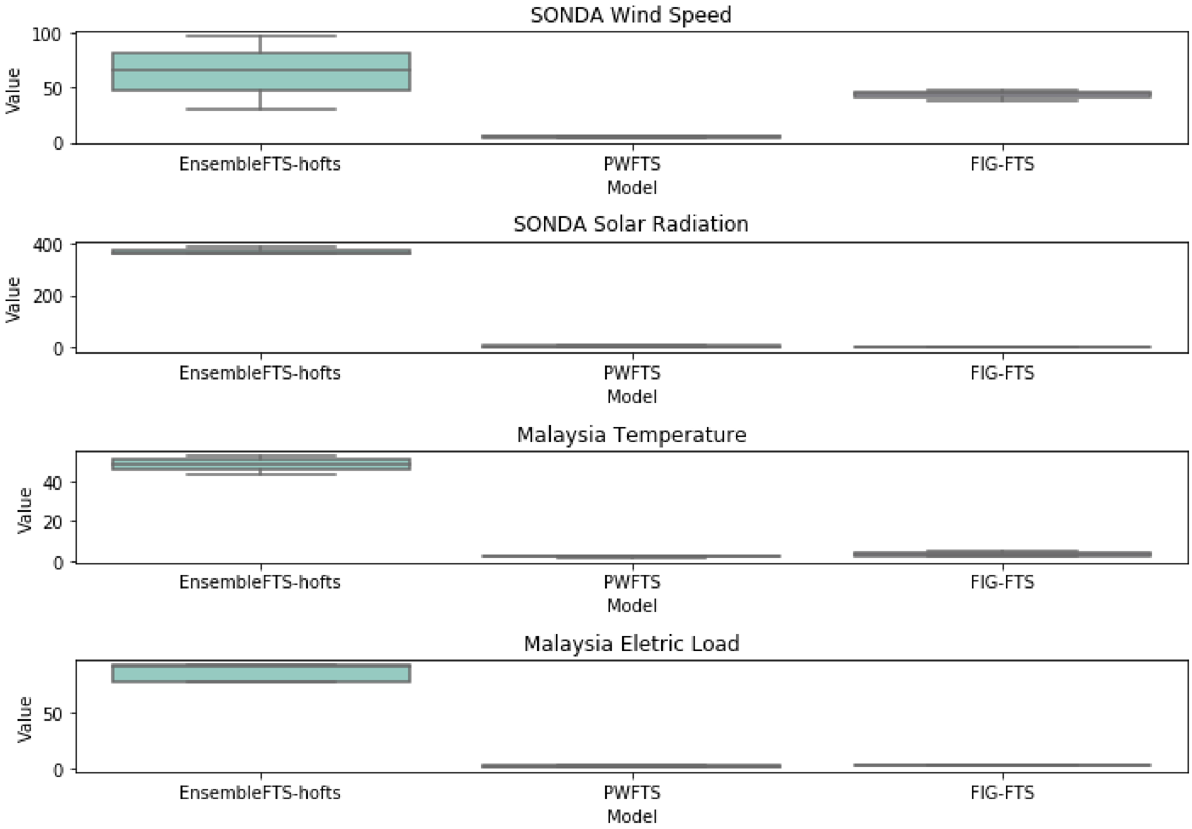


Figure 59 – CRPS probabilistic forecasting accuracy for one step ahead

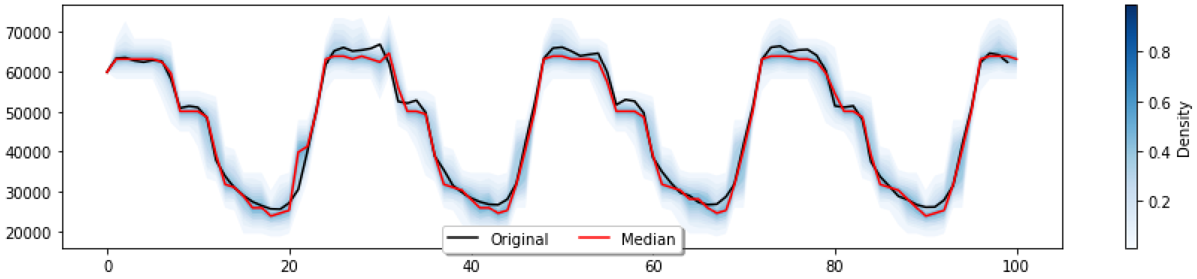


Figure 60 – Sample of $\mathcal{FIG}$-FTS probabilistic forecasting for one step ahead

6.5.1 Method limitations

$\mathcal{FIG}$-FTS produces non-parsimonious methods that can be computationally expensive. In order to optimize the models, both in terms of accuracy and parsimony, it is advisable to fine tunning the hyperparameters of each variable, as well as to optimize the choose of the best variables of the model.

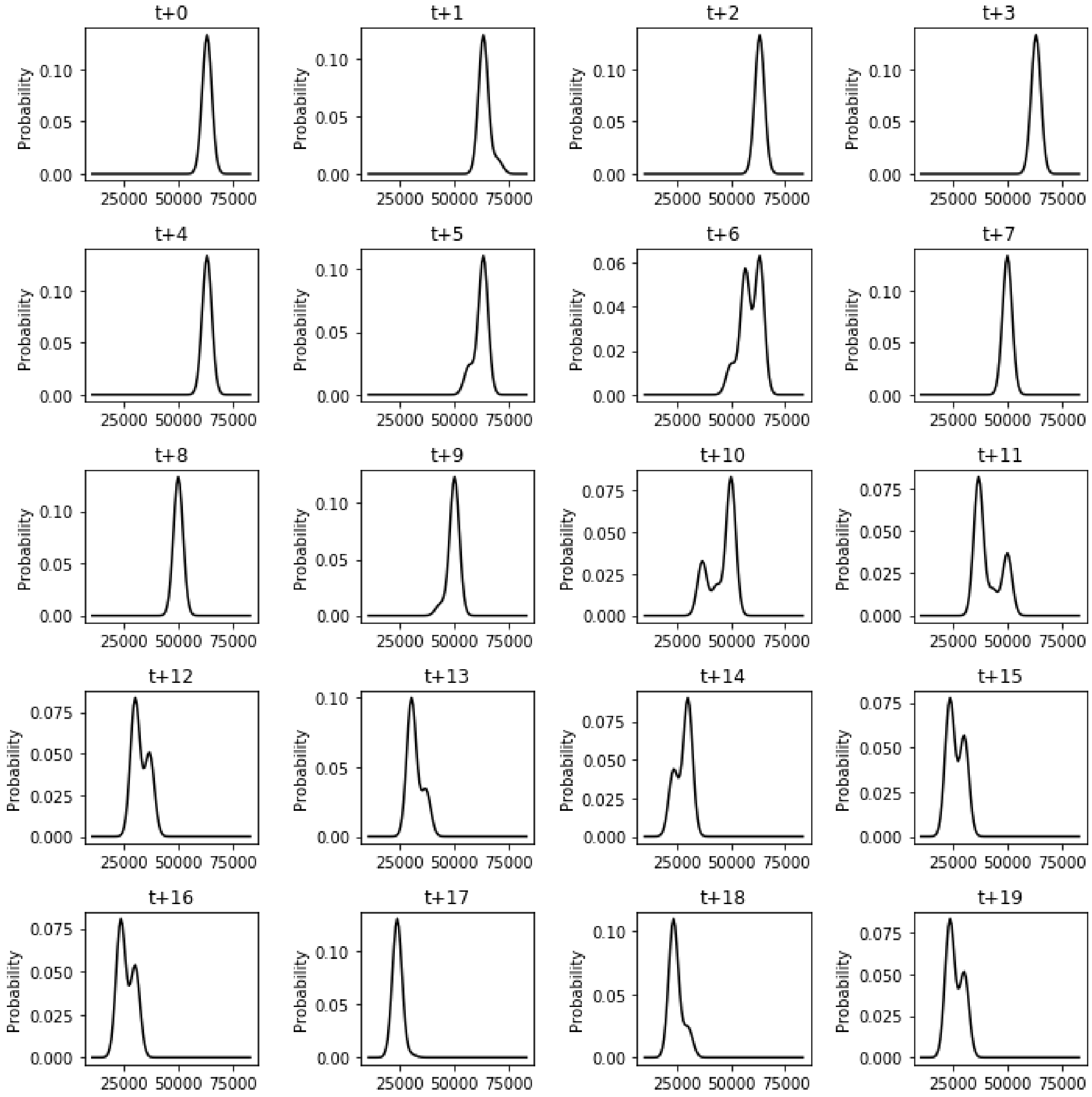


Figure 61 – Shape of $\mathcal{FIG}$-FTS probabilistic distributions for one step ahead

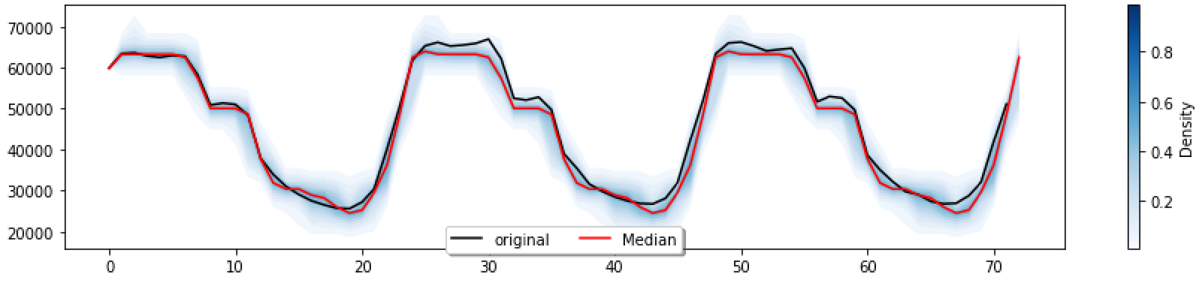


Figure 62 – Sample of $\mathcal{FIG}$-FTS probabilistic forecasting for many steps ahead

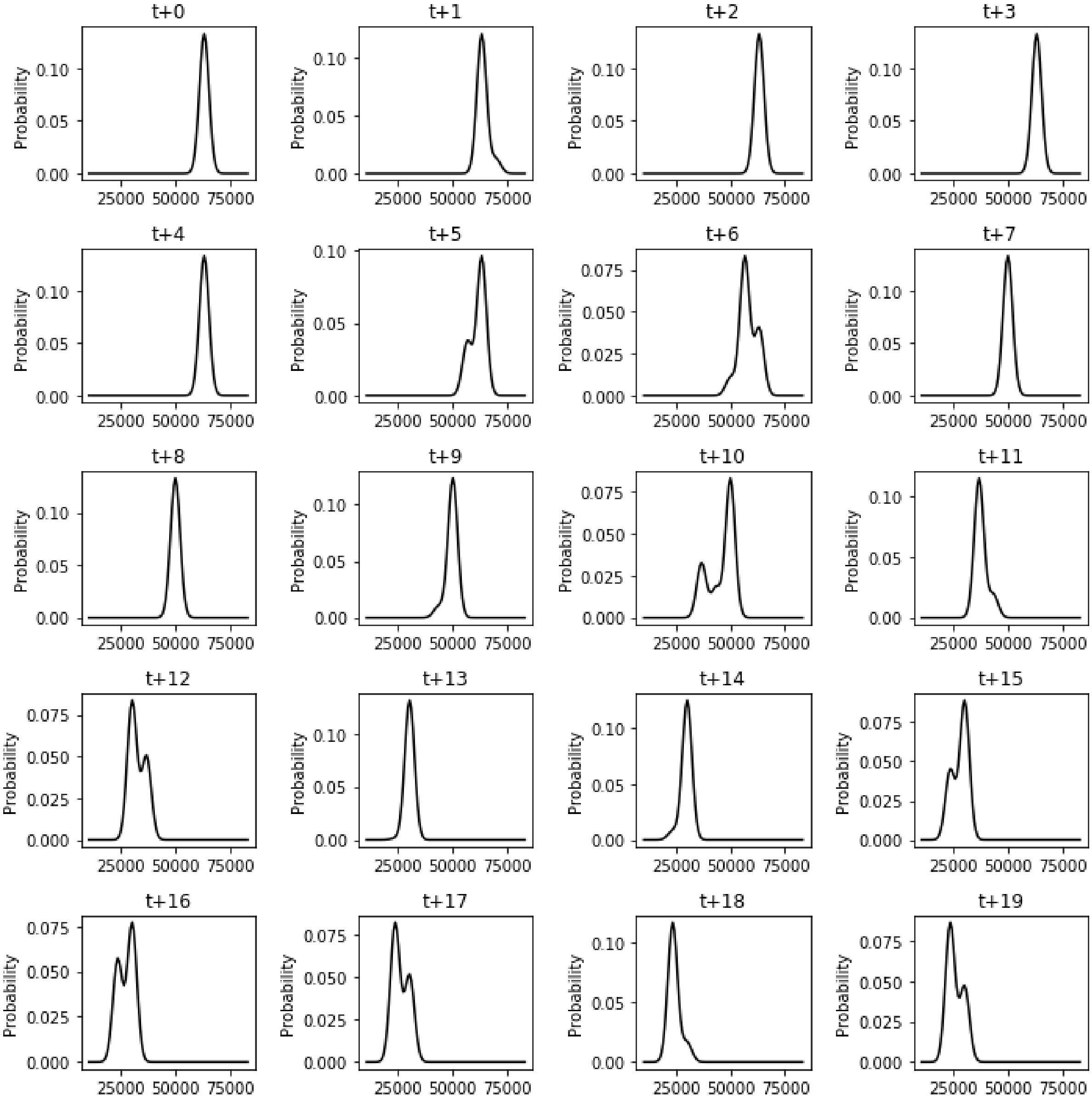


Figure 63 – Shape of $\mathcal{FIG}$-FTS probabilistic distributions for many steps ahead

Chapter 7

Conclusion

"...it is good to have measured myself, to recognize my limitations."

— Charles Francis Richter

On the forecasting research field, dealing with uncertainties is somehow mandatory, but still many of the forecasting methods are only concerned with point forecasting. The point forecasting methods have as their main general drawback the inability to measure the uncertainty of their results and, depending on the field of application, this is a crucial information. The direct alternative are the probabilistic methods as intervals and probability distributions Gneiting and Katzfuss [2014].

There are statistical forecasting methods available for probabilistic and interval forecasting, as instance Auto Regressive Integrated Moving Average (ARIMA), Quantile Auto-Regression (QAR), Bayesian Structural Time Series (BSTS), k-Nearest Neighbors (k-NN), among others. However these methods suffer from several limitations as lack of scalability, parametric assumptions, explainability or computational performance.

In the other hand, the Fuzzy Time Series (FTS) methods represent a growing field that has been gaining more attention in recent years. FTS forecasting methods produce data driven and non-parametric models, and have become attractive due to their simplicity, versatility, forecasting accuracy and computational performance, and it also produces human readable representations of the time series patterns, making its knowledge transferable, auditable, easily reusable and updatable. The variants of Fuzzy Time Series methods were investigated on Chapter 2. Within these variants, this work delimited its scope on time invariant rule-based FTS methods. The rule-based conventional High-Order Fuzzy Time Series (HOFTS) and the Weighted High-Order Fuzzy Time Series (WHOFTS) were studied and its accuracy was assessed and compared with conventional statistical forecasting methods which showed accuracy equivalence between the methods.

However, the FTS methods also suffer from lack of forecasting uncertainty representation, more specifically the absence of probabilistic forecasting methods. To deepen the discussion about the probabilistic forecasting, the Chapter 3 presented a review of the classical methods for interval and probability distribution forecasting and its main features. In order to fill the probabilistic forecasting gap at FTS field, the three first FTS methods for probabilistic forecasting in literature were proposed : $[\mathbb{I}]FTS$, $W[\mathbb{I}]FTS$ and Ensemble FTS.

$[\mathbb{I}]FTS$ and $W[\mathbb{I}]FTS$ extends HOFTS and WHOFTS methods, enabling the generating of predict intervals that represents the fuzzy uncertainty around the point forecasts. The Ensemble FTS method aims to represent the parameter uncertainty by embodying internally several FTS models with variations in their parameters. Ensemble FTS is capable to forecast intervals and probability distributions for one to more steps ahead. The accuracy of these methods was assessed and compared with the main statistical probabilistic methods which showed their accurate performance.

Nevertheless, until this point still missing a method that incorporate all uncertainties, capable to forecast points, intervals and probability distributions, for one to more steps ahead. To fill this lack the Probabilistic Weighted Fuzzy Time Series (PWFTS) method were proposed on Chapter 4. The PWFTS method use empirical fuzzy probabilities associated with their rules to represent the ontological uncertainty of data, and propose new deffuzyfication methods that exploit this probabilities. The PWFTS accuracy was assessed with computational experiments, compared with the previous FTS methods and the classical methods which showed the effectivenes of the method.

The main contribution of PWFTS is to combine versatility, accuracy and human readability. PWFTS is a versatile data driven, non parametric approach which integrates point, interval and probabilistic forecasting for one or multiple steps ahead, for first or higher orders. The measured accuracy shows its compatibility with, when it is not better than, standard approaches in the literature. The PWFTPG rule model is human-readable, easy to understand and interchangeable, which allows its assessment by experts and also non technical people. The PWFTPG rule set can be viewed as the conditional probability distribution of the fuzzy sets, and its visualization can even be used for data description and comprehension tasks.

Once flexible and accurate FTS models were proposed, new questions arise as result of its employment in real world problems, as instance big data scalability, model optimization and multivariate time series. The first question, discussed in Chapter 5, concerns in the impact of the data volume on FTS training and forecasting performances. The optimization of machine learning models for big time series is a challenging task to execute with sequential procedures or even parallel ones executed on a single machine. Thanks to the distributed computation frameworks, these methods are now enabled to work with massive datasets using cheap and available hardware infrastructure.To tackle this problem a distributed training method was proposed for computational clusters of commodity hardware using the Map/Reduce paradigm.

The second question concerns in the hyperparameter optimization which search for accurate and simultaneously parsimonic models. Generally, in fuzzy time series models, the increase of the number of rules leads to improvement on accuracy. But there is a trade off between the increase of the number of rules and the model overfitting. The Distributed Evolutionary Hyperparameter Optimization (DEHO) method is proposed embracing the distributed training and genetic algorithms, producing accurate and parsimonic models in feasible time.

The last question concerns in the forecasting of complex dynamics systems composed by many interacting variables. Dealing with multivariate and spatio-temporal time series was always a challenging task for FTS methods, specially because of the complexity growth of the rules as the dimension increases. To acomplish this task, in Chapter 6, the Fuzzy Information Granular Fuzzy Time Series method ($\mathcal{FIG}$-FTS) is proposed, an approach that incorporates Fuzzy Information Granules (FIG) to the FTS methodology in order to simplify the processing of the multivariate crisp data. First, individual Universe of Discourse partitioning schemes are provided for each variable and then Fuzzy Information Granules $\mathcal{G}_i$ are created as combinations of the fuzzy sets of the variables. Each $\mathcal{G}$ is created on demand, on the fuzzyfication phase, by selecting one fuzzy set of each variable. After that, each multivariate data point can be replaced by an univariate one, identified with a corresponding $\mathcal{G}$.

This work performed computational experiments to assess the $\mathcal{FIG}$-FTS method performance, and applied the proposed method to model and forecast the

In this way, the proposed method family is useful for a wide range of applications and user needs due its flexibility and customizability. The experimental analysis showed the effectiveness of the proposed methods and their flexibility on several scenarios.

7.1 Summary of contributions

- First interval forecasting approaches for FTS methods: Interval Fuzzy Time Series ($[\mathbb{I}]FTS$), Weighted Interval Fuzzy Time Series($W[\mathbb{I}]FTS$), Ensemble FTS, Probabilistic Weighted Fuzzy Time Series (PWFTS) and Fuzzy Information Granule Fuzzy Time Series ($\mathcal{FIG}$-FTS);

- First probabilistic forecasting approaches for FTS methods: Ensemble FTS, PWFTS and $\mathcal{FIG}$-FTS;

- The PWFTS method, an high-order integrated method capable to produce point, interval and probabilistic forecasts for one and many steps ahead, with a white-box model;

- Two new scalability approaches for FTS distributed training and forecasting using clusters of commodity hardware;

- The Distributed Evolutionary Hyperparameter Optimization (DEHO) method, an optimization engine for FTS models;

- $\mathcal{FIG}$-FTS an extension of PWFTS for multivariate data, bringing all features of PWFTS method to the multivariate time series.

- pyFTS - An free and open source library for Fuzzy Time Series in Python language to grant the research reproducibility and easy employment.

7.2 Summary of methods limitations

This research limited its scope to rule based time-invariant methods, which reduced the applicability of the proposed methods to stationary and well behaved time series with or without data pre-processing.

The presented methods lacks abilities on forecasting with trend and demands previous data transformations to deal with this kind of time series. It also lacks mechanisms to deal with concept drifts a heteroskedastic time series. Despite being easily upgradable, the models produced by the proposed methods needs to be frequently updated to follow new data behaviors. For the presented non-weighted methods, outliers may be hard to trick and can reduce the accuracy of the methods. It is advisable to perform outlier removal pre-processing tasks before train the models.

On PWFTS method, as the order and number of partitions increases the a priori probabilities may vanish to very low numbers, limited to the computational numerical precision.

The tuning of multivariate models is an open issue, demanding new hyperparameter optimization strategies. Without tuning, the models produced by FIG-FTS methods are not parsimonious and can be computationally expensive.

7.3 Future Investigations

Some future research directions must be pointed, some of them extracted from methods limitations:

- Time variant extensions for the proposed methods should be investigated, including the use of non-stationary fuzzy sets proposed by Garibaldi et al. [2008];

- The use of Approximate Bayesian Methods will be examined for the substitution of the π_k fixed probabilities for probability distributions, to embrace the uncertainty of these quantities;
- Extension of DEHO method for MVFTS and $\mathcal{FIG}$-FTS should be investigated;
- A new probabilistic forecasting method that produces joint probability distributions for multivariate forecasting in $\mathcal{FIG}$-FTS should be investigated.

7.4 Publications

From this research methods the following publications were extracted:

7.4.1 Journal Papers

1. SILVA, Petrônio C. L.; SADAEI, Hossein J. ; BALLINI, Rosângela ; GUIMARÃES, Frederico G. . Probabilistic Forecasting With Fuzzy Time Series. IEEE Transactions on Fuzzy Systems, v. 1, p. 1-1, 2019. DOI: 10.1109/tfuzz.2019.2922152
2. SADAEI, Hossein J.; SILVA, Petrônio C. L.; GUIMARÃES, Frederico G.; LEE, Muhammad H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. ENERGY, v. 174, p. 1, 2019. DOI: 10.1016/j.energy.2019.03.081

7.4.2 Conference Papers

1. ALVES, M. A.; ALMEIDA, L. V. V. B.; REZENDE, T. M.; SILVA, P. C. L. S.; SEVERIANO, C. A.; SILVA, R.; GUIMARÃES, F. G. Otimização Dinâmica Evolucionária para Despacho de Energia em uma Microrrede usando Veículos Elétricos. In 14^o Simpósio Brasileiro de Automação Inteligente - SBAI'19, Ouro Preto, 2019.
2. LUCAS, P. O. E.; SILVA, P. C. L. S.; GUIMARÃES, F. G. Otimização Evolutiva de Hiperparâmetros para Modelos de Séries Temporais Nebulosas. In 14^o Simpósio Brasileiro de Automação Inteligente - SBAI'19, Ouro Preto, 2019.
3. SILVA, Petrônio C. L.; SEVERIANO Jr., Carlos A.; ALVES, Marcos A. ; COHEN, Miri W.; GUIMARÃES, Frederico G. A New Granular Approach for Multivariate Forecasting. 2nd Latin American Workshop on Computational Neuroscience. Communications in Computer and Information Science, 2019.
4. SILVA, Petrônio C. L.; LUCAS, Patrícia O. ; GUIMARÃES, Frederico G. A Distributed Algorithm for Scalable Fuzzy Time Series. Lecture Notes in Computer

Science. 1ed.: Springer International Publishing, 2019, v. , p. 42-56. DOI: 10.1007/978-3-030-19223-5_4

5. ALVES, Marcos A. ; SILVA, Petrônio C. L. ; SEVERIANO JR., Carlos A. ; VIEIRA, Gustavo L. ; GUIMARAES, Frederico G. ; SADAEI, Hossein J. . An extension of nonstationary fuzzy sets to heteroskedastic fuzzy time series. In: 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2018, Bruges, Bélgica. 26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, 2018.

6. SILVA, Petrônio C. L.; ALVES, Marcos A. ; SEVERIANO JR., Carlos A. ; VIEIRA, Gustavo L. ; GUIMARAES, Frederido G. ; SADAEI, Hossein J. . Probabilistic Forecasting with Seasonal Ensemble Fuzzy Time-Series. In: XIII Brazilian Congress on Computational Intelligence, 2017, Niterói. Anais do XIII Brazilian Congress on Computational Intelligence, 2017.

7. COSTA, Francirley R. B. ; SILVA, Petrônio C. L.; GUIMARAES, Frederico G. ; BATISTA, Lucas S. . Regressão Linear Aplicada na Predição de Séries Temporais Fuzzy. In: XIII Simpósio Brasileiro de Automação Inteligente, 2017, Porto Alegre. Anais do XIII Simpósio Brasileiro de Automação Inteligente, 2017.

8. SEVERIANO Jr, Carlos A.; SILVA, Petrônio C.; SADAEI, Hossein J.; GUIMARÃES, Frederico G. Very Short-term Solar Forecasting using Fuzzy Time Series. 2017 IEEE Conference on Fuzzy Systems. DOI: 10.1109/fuzz-ieee.2017.8015732

9. SILVA, Petrônio C. L.; SADAEI, Hossein J.; GUIMARÃES, Frederico G. Interval Forecasting with Fuzzy Time Series. In Computational Intelligence (SSCI), 2016 IEEE Symposium Series on (pp. 1-8). IEEE. DOI: 10.1109/ssci.2016.7850010

7.4.3 Software Libraries

Silva, P. C. L, et al. pyFTS: Fuzzy Time Series for Python. Source Code: http://pyfts.github.io/pyFTS DOI: 10.5281/zenodo.597359.

7.4.4 Short Courses and Talks

1. SILVA, Petrônio C. L.; GUIMARÃES, Frederico G. Séries Temporais Nebulosas (STN). In 14º Simpósio Brasileiro de Automação Inteligente - SBAI'19, Ouro Preto, 2019.

2. SILVA, Petrônio C. L.; GUIMARÃES, Frederico G. Fuzzy Time Series. In pyDATA BH, Belo Horizonte, 2019.

3. SILVA, Petrônio C. L.; GUIMARÃES, Frederico G. pyFTS Quick Start. In Avenue Code Meetup, Belo Horizonte, 2019.

4. SILVA, Petrônio C. L.; GUIMARÃES, Frederico G. Introdução às Séries Temporais Nebulosas com Aplicações em Energia Solar. In 2^a Semana de Informática do IFNMG Campus Pirapora, Pirapora, 2018.

References

M. A. Alves, P. C. D. L. Silva, C. A. J. Severiano, G. L. Vieira, F. G. Guimaraes, and H. J. Sadaei. An extension of nonstationary fuzzy sets to heteroskedastic fuzzy time series. In *26th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, 2018.

S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil. Fast Direct Methods for Gaussian Processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 3 2014. doi: 10.1109/TPAMI.2015.2448083. URL https://arxiv.org/abs/1403.6015.

S. Askari and N. Montazerin. A high-order multi-variable Fuzzy Time Series forecasting algorithm based on fuzzy clustering. *Expert Systems with Applications*, 42(4):2121–2135, 2015. ISSN 09574174. doi: 10.1016/j.eswa.2014.09.036.

M. Bahrepour, M. R. Akbarzadeh-T., M. Yaghoobi, and M. B. Naghibi-S. An adaptive ordered fuzzy time series with application to FOREX. *Expert Systems with Applications*, 38(1):475–485, 2011. ISSN 09574174. doi: 10.1016/j.eswa.2010.06.087. URL http://dx.doi.org/10.1016/j.eswa.2010.06.087.

Y. Bai, J. Sun, and J. Luo. Forecasting Financial Time Series with Ensemble Learning. In *2010 International Symposium on Intelligent Signal Processing and Communication Systems (lSPACS 2010)*. IEEE, 2010.

N. S. Bajestani and A. Zare. Forecasting TAIEX using improved type 2 fuzzy time series. *Expert Systems with Applications*, 38(5):5816–5821, 2011. ISSN 09574174. doi: 10.1016/j.eswa.2010.10.049. URL http://dx.doi.org/10.1016/j.eswa.2010.10.049.

M. Baker and R. Buyya. Cluster computing at a glance. In *High performance cluster computing: Architectures and systems*, volume 1, chapter 1, pages 3–47. Prentice Hall, Upper Saddle River, NJ, USA, 1999.

D. Barber, A. T. Cemgil, S. Chiappa, Y. Atchadé, G. Fort, E. Moulines, and P. Priouret. *Bayesian Time Series Models*. Cambridge University Press, 2011. ISBN 9780521196765. doi: 10.1017/CBO9780511984679.

E. Bas, E. Egrioglu, C. H. Aladag, and U. Yolcu. Fuzzy-time-series network used to forecast linear and nonlinear time series. *Applied Intelligence*, 43(2):343–355, 2015. doi: 10.1007/s10489-015-0647-0.

E. Bas, C. Grosan, E. Egrioglu, and U. Yolcu. High order fuzzy time series method based on pi-sigma neural network. *Engineering Applications of Artificial Intelligence*, 2018. ISSN 09521976. doi: 10.1016/j.engappai.2018.04.017.

K. Bisht and S. Kumar. Fuzzy time series forecasting method based on hesitant fuzzy sets. *Expert Systems With Applications*, 64:557–568, 2016. doi: 10.1016/j.eswa.2016.07.044. URL https://doi.org/10.1016/j.eswa.2016.07.0.

M. Bose and K. Mali. A novel data partitioning and rule selection technique for modeling high-order fuzzy time series. *Applied Soft Computing*, 63(17):87–96, 2017. ISSN 15684946. doi: 10.1016/j.asoc.2017.11.011. URL http://doi.org/10.1016/j.asoc.2017.11.011.

G. E. P. Box and D. A. Pierce. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970. doi: 10.1080/01621459.1970.10481180. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1970.10481180.

G. W. Brier. VERIFICATION OF FORECASTS EXPRESSED IN TERMS OF PROBABILITY. *Monthly Weather Review*, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.

J. Bröcker and L. A. Smith. From ensemble forecasts to predictive distribution functions. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 60 A(4):663–678, 2008. ISSN 02806495. doi: 10.1111/j.1600-0870.2008.00333.x.

G. Brown. Ensemble Learning, 2010. URL http://www.cs.man.ac.uk/~gbrown/research/brown10ensemblelearning.pdf.

Q. Cai, D. Zhang, W. Zheng, and S. C. H. Leung. A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. *Knowledge-Based Systems journal*, 74:61–68, 2015. doi: 10.1016/j.knosys.2014.11.003.

J. Carvalho Jr and C. Costa Jr. Identification method for fuzzy forecasting models of time series. *Applied Soft Computing*, 50:166–182, 2017. doi: 10.1016/j.asoc.2016.11.003.

P.-T. Chang. Fuzzy seasonality forecasting. *Fuzzy Sets and Systems*, 90(1):1–10, 1997. ISSN 01650114. doi: 10.1016/S0165-0114(96)00138-8. URL https://doi.org/10.1016/S0165-0114(96)00138-8.

C. Chatfield. Calculating Interval Forecasts. *Journal of Business & Economic Statistics*, 11(2):121, 4 1993. ISSN 07350015. doi: 10.2307/1391361. URL http://doi.org/10.2307/1391361.

C. Chatfield. Prediction Intervals for Time-Series Forecasting. In J. Armstrong, editor, *Principles of forecasting*, pages 475–494. Springer US, 1 edition, 2001. ISBN 978-0-306-47630-3. doi: 10.1007/978-0-306-47630-3.

C.-S. Chen, Y.-D. Jhong, W.-Z. Wu, and S.-T. Chen. Fuzzy time series for real-time flood forecasting. *Stochastic Environmental Research and Risk Assessment*, 2019. ISSN 1436-3240. doi: 10.1007/s00477-019-01652-8.

C.-T. Chen, C.-T. Lin, S.-F. Huang, L. Da, and K.-C. Li. A fuzzy approach for supplier evaluation and selection in supply chain management. *Int. J. Production Economics*, 102:289–301, 2006. doi: 10.1016/j.ijpe.2005.03.009.

D.-W. Chen and J.-P. P. Zhang. Time Series Prediction Based On Ensemble ANFIS. *Proceedings of the Fourth International Conference on Machine Learning and Cybernetics*, pages 18–21, 2005.

M.-Y. Chen. A high-order fuzzy time series forecasting model for internet stock trading. *Future Generation Computer Systems*, 37:461–467, 2014. doi: 10.1016/j.future.2013.09.025. URL http://dx.doi.org/10.1016/j.future.2013.09.025.

M.-Y. Chen and B.-T. Chen. A hybrid fuzzy time series model based on granular computing for stock price forecasting. *Information Sciences*, 2015a. ISSN 00200255. doi: 10.1016/j.ins.2014.09.038.

S.-M. Chen. Forecasting enrollments based on fuzzy time series. *Fuzzy Sets and Systems*, 81(3):311–319, 1996. doi: 10.1016/0165-0114(95)00220-0. URL https://doi.org/10.1016/0165-0114(95)00220-0.

S.-M. Chen. FORECASTING ENROLLMENTS BASED ON HIGH-ORDER FUZZY TIME SERIES. *Cybernetics and Systems*, 33(1):1–16, 1 2002. ISSN 0196-9722. doi: 10.1080/019697202753306479. URL https://doi.org/10.1080/019697202753306479.

S. M. Chen and Y. C. Chang. Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques. *Information Sciences*, 180(24):4772–4783, 2010. ISSN 00200255. doi: 10.1016/j.ins.2010.08.026. URL http://dx.doi.org/10.1016/j.ins.2010.08.026.

S. M. Chen and C. D. Chen. TAIEX forecasting based on fuzzy time series and fuzzy variation groups. *IEEE Transactions on Fuzzy Systems*, 19(1):1–12, 2011. ISSN 10636706. doi: 10.1109/TFUZZ.2010.2073712.

S.-M. Chen and N.-Y. Chung. Forecasting Enrollments Using High-Order Fuzzy Time Series and Genetic Algorithms. *INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS,*, 21:485–501, 2006. doi: 10.1002/int.20145. URL https://doi.org/10.1002/int.20145.

S.-M. Chen and J.-R. Hwang. Temperature Prediction Using Fuzzy Time Series. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 30(2), 2000. doi: 10.1109/3477.836375. URL https://doi.org/10.1109/3477.836375.

S. M. S.-W. W. S.-M. Chen and S. M. S.-W. W. S.-M. Chen. Fuzzy Forecasting Based on Two-Factors Second-Order Fuzzy-Trend Logical Relationship Groups and the Probabilities of Trends of Fuzzy Logical Relationships. *IEEE TRANSACTIONS ON CYBERNETICS*, 45(3), 2015b. ISSN 21682267. doi: 10.1109/TCYB.2014.2326888.

Y.-S. Chen, C.-H. Cheng, W.-L. Tsai, Y.-S. Chen, C.-H. Cheng, and W.-L. Tsai. Modeling fitting-function-based fuzzy time series patterns for evolving stock index forecasting. *Appl Intell*, 41:327–347, 2014. doi: 10.1007/s10489-014-0520-6.

C. H. Cheng and C. H. Chen. Fuzzy time series model based on weighted association rule for financial market forecasting. *Expert Systems*, 2018. ISSN 14680394. doi: 10.1111/exsy.12271.

C.-H. Cheng, T. L. Chen, H. J. Teoh, and C. H. Chiang. Fuzzy time-series based on adaptive expectation model for TAIEX forecasting. *Expert Systems with Applications*, 34:1126–1132, 2008a. ISSN 09574174. doi: 10.1016/j.eswa.2006.12.021.

C.-H. Cheng, J.-W. Wang, and C.-H. Li. Forecasting the number of outpatient visits using a new fuzzy time series based on weighted-transitional matrix. *Expert Systems with Applications*, 34(4):2568–2575, 2008b. doi: 10.1016/j.eswa.2007.04.007. URL https://doi.og/10.1016/j.eswa.2007.04.007.

C.-H. Cheng, Y.-S. Chen, and Y.-L. Wu. Forecasting innovation diffusion of products using trend-weighted fuzzy time-series model. *Expert Systems with Applications*, 36(2): 1826–1832, 2009.

C. H. Cheng, S. F. Huang, and H. J. Teoh. Predicting daily ozone concentration maxima using fuzzy time series based on a two-stage linguistic partition method. *Computers and Mathematics with Applications*, 2011. doi: 10.1016/j.camwa.2011.06.044. URL https://doi.org/10.1016/j.camwa.2011.06.044.

C.-H. H. Cheng, J.-R. R. Chang, and C.-A. A. Yeh. Entropy-based and trapezoid fuzzification-based fuzzy time series approaches for forecasting IT project cost. *Technological Forecasting and Social Change*, 73(5):524–542, 2006. doi: 10.1016/j.techfore.2005.07.004. URL https://doi.org/10.1016/j.techfore.2005.07.004.

Y.-C. C. Cheng and S.-T. T. Li. Fuzzy Time Series Forecasting With a Probabilistic Smoothing Hidden Markov Model. *IEEE Transactions on Fuzzy Systems*, 20(2):291–304, 2012. doi: 10.1109/TFUZZ.2011.2173583. URL https://doi.org/10.1109/TFUZZ.2011.2173583.

P. F. Christoffersen. Evaluating Interval Forecasts. *International Economic Review*, 39 (4):841, 11 1998. ISSN 00206598. doi: 10.2307/2527341. URL http://www.jstor.org/stable/2527341?origin=crossref.

H.-c. Chuang, W.-s. Chang, and S.-t. Li. A Multi-Factor HMM-based Forecasting Model for Fuzzy Time Series. In *IMMM 2014 : The Fourth International Conference on Advances in Information Mining and Management*, pages 17–23. IARIA, 2014. ISBN 9781612083643.

R. T. Clemen. Combining forecast: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583, 1989. ISSN 01692070. doi: 10.1016/0169-2070(89) 90012-5.

V. N. Coelho, I. M. Coelho, B. N. Coelho, A. J. R. Reis, R. Enayatifar, M. J. F. Souza, and F. G. Guimarães. A self-adaptive evolutionary fuzzy model for load forecasting problems on smart grid environment. *Applied Energy*, 169:567–584, 2016. doi: 10.1016/j.apenergy.2016.02.045.

S. Davari, M. H. F. Zarandi, and I. B. Turksen. An improved Fuzzy Time Series forecasting model based on Particle Swarm intervalization. In *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 1–5, 2009. doi: 10.1109/NAFIPS.2009.5156420. URL https://doi.org/10.1109/NAFIPS.2009.5156420.

J. Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1:3–18, 2011. doi: 10.1016/j.swevo.2011.02.002.

N. G. Dincer and O. Akkuş. A new fuzzy time series model based on robust clustering for forecasting of air pollution. *Ecological Informatics*, 43:157–164, 1 2018. doi: 10.1016/J.ECOINF.2017.12.001.

D. Dubois and H. Prade. Fuzzy sets, probability and measurement. *European Journal of Operational Research*, 40:135–154, 1989.

O. Duru and S. Yoshida. Modeling principles in fuzzy time series forecasting. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, New York, NY, USA, 2012. IEEE. doi: 10.1109/CIFEr.2012.6327767. URL http://doi.org/10.1109/CIFEr.2012.6327767.

EC AI HLEG. Ethics guidelines for trustworthy AI. Technical report, European Commission on High-Level Expert Group on Artificial Intelligence (AI HLEG), 2019. URL https://ec.europa.eu/futurium/en/ai-alliance-consultation.

R. Efendi, Z. Ismail, and M. M. Deris. Improved weight Fuzzy Time Series as used in the exchange rates forecasting of US Dollar to Ringgit Malaysia. *International Journal of Computational Intelligence and Applications*, 12(01):1350005, 2013. doi: 10.1142/S1469026813500053. URL http://doi.org/10.1142/S1469026813500053.

E. Egrioglu, C. H. Aladag, U. Yolcu, V. R. Uslu, and M. A. Basaran. A new approach based on artificial neural networks for high order multivariate fuzzy time series. *Expert Systems with Applications*, 36(7):10589–10594, 9 2009. doi: 10.1016/J.ESWA.2009.02.057. URL https://doi.org/10.1016/j.eswa.2009.02.057.

E. Egrioglu, C. H. Aladag, U. Yolcu, V. R. Uslu, and M. A. Basaran. Finding an optimal interval length in high order fuzzy time series. *Expert Systems with Applications*, 37 (7):5052–5055, 7 2010. doi: 10.1016/J.ESWA.2009.12.006. URL https://doi.org/10.1016/j.eswa.2009.12.006.

R. Enayatifar, H. J. Sadaei, A. H. Abdullah, and A. Gani. Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (RHWFTS-ICA) for short term load forecasting. *Energy Conversion and Management*, 76:1104–1116, 2013. ISSN 01968904. doi: 10.1016/j.enconman.2013.08.039.

Everette S. Gardner. A Simple Method of Computing Prediction Intervals for Time Series Forecasts. *Management Science*, 34(4):541–546, 1998.

H. Finner. On a Monotonicity Problem in Step-Down Multiple Test Procedures. *Journal of the American Statistical Association*, 88(423):920–923, 1993. doi: 10.1080/01621459.1993.10476358. URL http//doi.org/10.1080/01621459.1993.10476358.

C. Fraley, A. E. Raftery, and T. Gneiting. Probabilistic weather forecasting in R. *The R Journal*, 3(June):55–63, 2011. ISSN 2073-4859. doi: 10.1198/jasa.2009.ap07184. URL http://doi.org/10.1198/jasa.2009.ap07184.

C. Fraley, A. E. Raftery, T. Gneiting, and J. M. Sloughter. ensembleBMA : An R Package for Probabilistic Forecasting using Ensembles and Bayesian Model. Technical report, Department of Statistics, University of Washington, Seattle, WA, 2013.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. ... *Computing Surveys (...*, 46(4):1–37, 2014. ISSN 03600300. doi: 10.1145/2523813.

S. S. Gangwar and S. Kumar. Probabilistic and Intuitionistic Fuzzy Sets–Based Method for Fuzzy Time Series Forecasting. *Cybernetics and Systems*, 45(4):349–361, 2014. ISSN 0196-9722. doi: 10.1080/01969722.2014.904135. URL https://doi.org/10.1080/01969722.2014.904135.

S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180:2044–2064, 2010. doi: 10.1016/j.ins.2009.12.010.

J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan. Nonstationary Fuzzy Sets. *IEEE TRANSACTIONS ON FUZZY SYSTEMS*, 16(4), 2008. doi: 10.1109/TFUZZ.2008.917308.

V. Georgescu. Fuzzy time series estimation and prediction: Criticism, suitable new methods and experimental evidence. *Studies in Informatics and control*, 19(3):230, 2010.

V. Georgescu. Joint propagation of ontological and epistemic uncertainty across risk assessment and fuzzy time series models. *Computer Science and Information Systems*, 2014. ISSN 18200214. doi: 10.2298/CSIS121215048G. URL http://doi.org/10.2298/CSIS121215048G.

T. Gneiting. Editorial: Probabilistic forecasting. *Journal of the Royal Statistical Society*, 171(2):319–321, 2008. URL http://www.jstor.org/stable/30130759.

T. Gneiting and M. Katzfuss. Probabilistic Forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014. ISSN 2326-8298. doi: 10.1146/annurev-statistics-062713-085831. URL http://doi.org/10.1146/annurev-statistics-062713-085831.

T. Gneiting and A. E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

T. Gneiting, A. E. Raftery, A. H. Westveld, T. Goldman, A. H. Westveld Iii, T. Goldman, A. H. Westveld, T. Goldman, A. H. Westveld Iii, and T. Goldman. Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation. *Monthly Weather Review*, 133(5):1098–1118, 2005. ISSN 0027-0644. doi: 10.1175/MWR2904.1.

T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic Forecasts, Calibration and Sharpness. *Source Journal of the Royal Statistical Society. Series B (Statistical Methodology) Journal of the Royal Statistical Society. Series B (Statistical Methodology J. R. Statist. Soc. B*, 69(2):243–268, 2007. URL http://www.jstor.org/stable/4623266.

I. J. Good. Rational Decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114, 1952. URL http://www.jstor.org/stable/2984087.

G. Grmanová, P. Laurinec, V. Rozinajová, A. B. Ezzeddine, M. Lucká, P. Lacko, P. Vrablecová, P. Návrat, G. Kosková, P. Laurinec, V. Rozinajová, A. B. Ezzeddine, M. Lucká, P. Lacko, P. Vrablecová, and P. Návrat. Incremental ensemble learning for electricity load forecasting. *Acta Polytechnica Hungarica*, 13(2):97–117, 2016. ISSN 17858860. doi: 10.1177/1542305015616103[doi].

H. Guney, M. A. Bakir, and C. H. Aladag. A Novel Stochastic Seasonal Fuzzy Time Series Forecasting Model. *International Journal of Fuzzy Systems*, 2018. ISSN 21993211. doi: 10.1007/s40815-017-0385-z. URL http://doi.org/10.1007/s40815-017-0385-z%0AA.

B. E. Hansen. Interval forecasts and parameter uncertainty. *Journal of Econometrics*, 135:377–398, 2006. doi: 10.1016/j.jeconom.2005.07.030.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. doi: 10.2307/2334940.

J. L. Hodges and E. L. Lehmann. Rank Methods for Combination of Independent Experiments in Analysis of Variance. *Ann. Math. Statist.*, 33(2):482–497, 5 1962. doi: 10.1214/aoms/1177704575. URL https://doi.org/10.1214/aoms/1177704575.

W. Homenda and A. Jastrzebska. Clustering techniques for Fuzzy Cognitive Map design for time series modeling. *Neurocomputing*, 2017. ISSN 18728286. doi: 10.1016/j.neucom.2016.08.119.

W. Homenda, A. Jastrzebska, and W. Pedrycz. Time Series Modeling with Fuzzy Cognitive Maps: Simplification Strategies. In *Computer Information Systems and Industrial Management*, pages 409–420. Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-662-45237-0{_}38. URL http://doi.org/10.1007/978-3-662-45237-0_38.

T. Hong and S. Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32:914–938, 2016. doi: 10.1016/j.ijforecast.2015.11.011. URL http://doi.org/10.1016/j.ijforecast.2015.11.011.

T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *International Journal of Forecasting*, 32:896–913, 2016. doi: 10.1016/j.ijforecast.2016.02.001. URL http://doi.org/10.1016/j.ijforecast.2016.02.001.

L.-Y. Hsu, S.-J. Horng, T.-W. Kao, Y.-H. Chen, R.-S. Run, R.-J. Chen, J.-L. Lai, and I.-H. Kuo. Temperature prediction and TAIFEX forecasting based on fuzzy relationships and MTPSO techniques. *Expert Systems with Applications*, 37(4):2756–2770, 4 2010. doi: 10.1016/J.ESWA.2009.09.015. URL https://doi.org/10.1016/j.eswa.2009.09.015.

Y.-L. L. Huang, S.-J. J. Horng, M. He, P. Fan, T.-W. W. Kao, M. K. Khan, J.-L. L. Lai, and I.-H. H. Kuo. A hybrid forecasting model for enrollments based on aggregated fuzzy time series and particle swarm optimization. *Expert Systems with Applications*, 38(7):8014–8023, 7 2011. ISSN 09574174. doi: 10.1016/j.eswa.2010.12.127. URL https://doi.org/10.1016/j.eswa.2010.12.127.

K. Huarng. Effective lengths of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets and Systems*, 123(3):387–394, 2001. ISSN 01650114. doi: 10.1016/S0165-0114(00)00057-9.

K. Huarng and H.-K. Yu. A Dynamic Approach to Adjusting Lengths of Intervals in Fuzzy Time Series Forecasting. *Intell. Data Anal.*, 8(1):3–27, 1 2004. ISSN 1088-467X.

K. Huarng and H.-K. Yu. A type 2 fuzzy time series model for stock index forecasting. *Physica A: Statistical Mechanics and its Applications*, 353:445–462, 2005. doi: 10.1016/j.physa.2004.11.070.

K. Huarng and T. H.-K. Yu. The application of neural networks to forecast fuzzy time series. *Physica A: Statistical Mechanics and its Applications*, 363(2):481–491, 5 2006. doi: 10.1016/J.PHYSA.2005.08.014. URL https://doi.org/10.1016/j.physa.2005.08.014.

Z. Ismail and R. Efendi. Enrollment forecasting based on modified weight fuzzy time series. *Journal of Artificial Intelligence*, 4(1):110–118, 2011. ISSN 19945450. doi: 10.3923/jai.2011.110.118.

Z. Ismail, R. Efendi, and M. M. Deris. Application of Fuzzy Time Series Approach in Electric Load Forecasting. *New Mathematics and Natural Computation*, 11(3):229–248, 2015. doi: 10.1142/S1793005715500076.

Jeng-Ren Hwang, Shyi-Ming Chen, Chia-Hoang Lee, J.-R. Hwang, S.-M. Chen, and C.-H. Lee. Handling forecasting problems using fuzzy time series. *Fuzzy Sets and Systems*, 100(1):217–228, 1998. doi: 10.1016/S0165-0114(97)00121-8.

P. Jiang, Q. Dong, P. Li, and L. Lian. A novel high-order weighted fuzzy time series model and its application in nonlinear time series prediction. *Applied Soft Computing*, 55:44–62, 2017. ISSN 15684946. doi: 10.1016/j.asoc.2017.01.043.

T. A. Jilani and S. M. A. Burney. Multivariate stochastic fuzzy forecasting models. *Expert Systems with Applications*, 35(3):691–700, 2008. ISSN 09574174. doi: 10.1016/j.eswa.2007.07.014.

T. A. Jilani, A. S. Burney, and C. Ardil. A New Quantile Based Fuzzy Time Series Forecasting Model. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 2(3):995–1001, 2008. URL http://waset.org/publications/14214/a-new-quantile-based-fuzzy-time-series-forecasting-model.

E. P. Klement, W. Schwyhla, and R. Lowen. Fuzzy Probability Measures. *Fuzzy Sets and Systems*, 5:21–30, 1981.

G. Klir and B. Yuan. *Fuzzy sets and fuzzy logic*, volume 4. Prentice hall New Jersey, 1995.

R. Koenker and K. F. Hallock. Quantile Regression. *Journal of Economic Perspectives—Volume*, 15(4):143–156, 2001.

R. Koenker and Z. Xiao. Quantile Autoregression. *Journal of the American Statistical Association*, 101(405):980–990, 2006. doi: 10.1198/016214506000000672. URL https://doi.org/10.1198/016214506000000672.

B. Kosko. Fuzzy Cognitive Maps. *International Journal of Man-Machine Studies*, 24(1): 65–75, 1986. doi: 10.1016/S0020-7373(86)80040-2.

R. Krzysztofowicz. The case for probabilistic forecasting in hydrology. *Journal of Hydrology*, 249:2–9, 2001.

I. H. Kuo, S. J. Horng, T. W. Kao, T. L. Lin, C. L. Lee, and Y. Pan. An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization. *Expert Systems with Applications*, 36(3 PART 2):6108–6117, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2008.07.043. URL http://dx.doi.org/10.1016/j.eswa.2008.07.043.

F. Laio and S. Tamea. Verification tools for probabilistic forecasts of continuous hydrological variables. *Hydrol. Earth Syst. Sci*, 11:1267–1277, 2007. doi: 10.5194/hess-11-1267-2007. URL https://doi.org/10.5194/hess-11-1267-2007.

C.-H. L. Lee, A. Liu, and W.-S. Chen. Pattern Discovery of Fuzzy Time Series for Financial Prediction. *IEEE Transactions On Knowledge And Data Engineering*, 18(5):613–625, 2006. doi: 10.1109/TKDE.2006.80.

M. H. Lee and H. Javedani. A Weighted Fuzzy Integrated Time Series for Forecasting Tourist Arrivals. In *International Conference on Informatics Engineering and Information Science*, pages 206–217, Berlin Heidelberg, 2011. Springer.

M. H. Lee, H. J. Sadaei, and Suhartono. Improving TAIEX forecasting using fuzzy time series with Box–Cox power transformation. *Journal of Applied Statistics*, 40(11): 2407–2422, 11 2013a. ISSN 0266-4763. doi: 10.1080/02664763.2013.817548.

M. H. Lee, H. J. Sadaei, and Suhartono. Introducing polynomial fuzzy time series. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 25(1): 117–128, 2013b.

W.-J. Lee, H.-Y. Jung, H. Y. Jin, and S. H. Choi. A Novel Forecasting Method Based on F-Transform and Fuzzy Time Series. *International Journal of Fuzzy Systems*, 2017. doi: 10.1007/s40815-017-0354-6.

D. Leite, F. Gomide, R. Ballini, and P. Costa. Fuzzy granular evolving modeling for time series prediction. *IEEE International Conference on Fuzzy Systems*, pages 2794–2801, 2011. ISSN 10987584. doi: 10.1109/FUZZY.2011.6007452.

D. Leslie. *Understanding artificial intelligence ethics and safety: A guide for the responsible design and implementation of AI systems in the public sector*. The Alan Turing Institute, 6 2019. doi: 10.5281/ZENODO.3240529.

M. Leutbecher and T. N. Palmer. Ensemble forecasting. *Journal of Computational Physics*, 227(7):3515–3539, 2008. ISSN 00219991. doi: 10.1016/j.jcp.2007.02.014. URL http://doi.org/10.1016/j.jcp.2007.02.014.

S.-T. Li, Y.-C. Cheng, and S.-Y. Lin. A FCM-based deterministic forecasting model for fuzzy time series. *Computers & Mathematics with Applications*, 56(12):3052–3063, 12 2008. ISSN 08981221. doi: 10.1016/j.camwa.2008.07.033. URL https://doi.org/10.1016/j.camwa.2008.07.033.

B. Liu, J. Nowotarski, T. Hong, and R. Weron. Probabilistic Load Forecasting via Quantile Regression Averaging on Sister Forecasts. *IEEE Transactions on Smart Grid*, 2015. doi: 10.1109/TSG.2015.2437877.

G. M. Ljung and G. E. P. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978. doi: 10.1093/biomet/65.2.297.

E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20: 130–141, 1963.

W. Lu, W. Pedrycz, X. Liu, J. Yang, and P. Li. The modeling of time series based on fuzzy information granules. *Expert Systems with Applications*, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2013.12.005.

J. Luo and S. M. Bridges. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems*, 15(8):687–703,

2000. doi: 10.1002/1098-111X(200008)15:8<687::AID-INT1>3.0.CO;2-X. URL https://doi.org/10.1002/1098-111X(200008)15:8%3C687::AID-INT1%3E3.0.CO;2-X.

C. Lynch. Big data: How do your data grow? *Nature*, 455(7209):28, 2008.

M. H. Magalhães, R. Ballini, and F. A. C. Gomide. Granular Models for Time-Series Forecasting. In *Handbook of Granular Computing*, chapter 45, pages 949–967. John Wiley & Sons, Ltd, 2008. ISBN 9780470724163. doi: 10.1002/9780470724163.ch45. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470724163.ch45.

S. Makridakis and N. Bakas. Forecasting and uncertainty: A survey. *Risk and Decision Analysis*, 6(1):37–64, 2016. ISSN 18759173. doi: 10.3233/RDA-150114.

S. Makridakis and N. Taleb. Living in a world of low levels of predictability. *International Journal of Forecasting*, 25:840–844, 2009. doi: 10.1016/j.ijforecast.2009.05.008.

S. Makridakis, R. M. Hogarth, and A. Gaba. Why forecasts fail. What to do instead. *MIT Sloan Management Review*, 51(2):83–90, 2010. ISSN 15329194.

N. Metropolis and S. Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81–97, 1956. doi: 10.1037/h0043158.

A. A. Mohammed and Z. Aung. Ensemble Learning Approach for Probabilistic Forecasting of Solar Power Generation. *Energies*, 9(12), 2016. ISSN 19961073. doi: 10.3390/en9121017.

A. A. Mohammed, W. Yaqub, and Z. Aung. Probabilistic Forecasting of Solar Power: An Ensemble Learning Approach. In R. Neves-Silva, L. C. Jain, and R. J. Howlett, editors, *Intelligent Decision Technologies: Proceedings of the 7th KES International Conference on Intelligent Decision Technologies (KES-IDT 2015)*, pages 449–458, Cham, 2015. Springer International Publishing. ISBN 978-3-319-19857-6. doi: 10.1007/978-3-319-19857-6{_}38.

G. Moyse and M.-J. Lesot. Linguistic summaries of locally periodic time series. *Fuzzy Sets and Systems*, 285:94–117, 2016. doi: 10.1016/j.fss.2015.06.016. URL www.elsevier.com/locate/fss.

M. Muja and D. G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2227–2240, 11 2014. ISSN 0162-8828. doi: 10.1109/TPAMI.2014.2321376. URL file:///home/petronio/Downloads/06809191.pdf.

G. G. Netto, A. C. Barbosa, M. N. Coelho, A. R. L. Miranda, V. N. Coelho, M. J. F. Souza, F. G. Guimarães, A. J. R. Reis, F. G. Guimar, and A. J. R. Reis. A hybrid evolutionary probabilistic forecasting model applied for rainfall and wind power forecast. In *2016 IEEE Conference on Evolving and Adaptive Intelligent Systems*, pages 73–78. IEEE, 2016. ISBN 9781509025831. doi: 10.1109/EAIS.2016.7502494.

V. Novák. Linguistic characterization of time series. *Fuzzy Sets and Systems*, 285:52–72, 2016a. ISSN 01650114. doi: 10.1016/j.fss.2015.07.017.

V. Novák. Mining information from time series in the form of sentences of natural language. *International Journal of Approximate Reasoning*, 78:192–209, 2016b. doi: 10.1016/j.ijar.2016.07.006. URL www.elsevier.com/locate/ijar.

I. Perfilieva. Fuzzy transforms: Theory and applications. *Fuzzy Sets and Systems*, 157: 993–1023, 2006. doi: 10.1016/j.fss.2005.11.012. URL http://doi.org/10.1016/j.fss.2005.11.012.

P. Pinson and H. Madsen. Ensemble-based probabilistic forecasting at Horns Rev. *Wind Energy*, 12:137–155, 2009. ISSN 10954244. doi: 10.1002/we.309.

P. Pinson, G. Kariniotaki, H. Aa Nielsen, T. S. Nielsen, and H. Madsen. Properties of Quantile and Interval Forecasts of Wind Generation and their Evaluation. In *Proceedings of the European Wind Energy Conference & Exhibition*, Athens, 2006. doi: 10.1.1.583.9758.

J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 67, 2016. doi: 10.1186/s13634-016-0355-x.

W. Qiu, X. Liu, and H. Li. A generalized method for forecasting based on fuzzy time series. *Expert Systems with Applications*, 38(8):10446–10453, 2011. ISSN 09574174. doi: 10.1016/j.eswa.2011.02.096. URL http://dx.doi.org/10.1016/j.eswa.2011.02.096.

W. Qiu, X. Liu, and H. Li. High-order fuzzy time series model based on generalized fuzzy logical relationship. *Mathematical Problems in Engineering*, 2013, 2013. ISSN 1024123X. doi: 10.1155/2013/927394.

A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review*, 133(5):1155–1174, 2005. ISSN 0027-0644. doi: 10.1175/MWR2906.1.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. ISBN 0-262-18253-X.

O. F. Reyes-Galaviz. *Granular Fuzzy Models: Construction, Analysis, and Design.* PhD thesis, Department of Electrical and Computer Engineering, University of Alberta, 2016.

S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 2 2013. ISSN 1364503X. doi: 10.1098/rsta.2011.0550.

I. Rodríguez-Fdez, A. Canosa, M. Mucientes, and A. Bugarín. STAC: a web platform for the comparison of algorithms using statistical tests. In *Proceedings of the 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Istanbul, Turkey, 2015. IEEE. doi: 10.1109/FUZZ-IEEE.2015.7337889. URL https://github.com/citiususc/stac.

A. Rubio, J. D. Bermúdez, and E. Vercher. Forecasting portfolio returns using weighted fuzzy time series methods. *International Journal of Approximate Reasoning*, 75:1–12, 2016. doi: 10.1016/j.ijar.2016.03.007.

H. Saberi, A. Rahai, and F. Hatami. A fast and efficient clustering based fuzzy time series algorithm (FEFTS) for regression and classification. *Applied Soft Computing*, 61:1088–1097, 12 2017. doi: 10.1016/J.ASOC.2017.09.023. URL https://doi.org/10.1016/J.ASOC.2017.09.023.

H. J. Sadaei. *Improved models in Fuzzy Time Series for forecasting.* PhD thesis, Universiti Teknologi Malaysia, 2013.

H. J. Sadaei and M. H. Lee. Multilayer Stock Forecasting Model Using Fuzzy Time Series. *The Scientific World Journal*, 2014. doi: 10.1155/2014/610594.

H. J. Sadaei, R. Enayatifar, A. H. Abdullah, and A. Gani. Short-term load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search. *International Journal of Electrical Power & Energy Systems*, 62(from 2005):118–129, 2014. ISSN 01420615. doi: 10.1016/j.ijepes.2014.04.026.

H. J. Sadaei, R. Enayatifar, F. G. Guimarães, M. Mahmud, and Z. A. Alzamil. Combining ARFIMA models and fuzzy time series for the forecast of long memory time series. *Neurocomputing*, 175:782–796, 2016a. ISSN 09252312. doi: 10.1016/j.neucom.2015.10.079. URL https://doi.org/10.1016/j.neucom.2015.10.079.

H. J. Sadaei, R. Enayatifar, M. H. Lee, and M. Mahmud. A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting. *Applied Soft Computing Journal*, 40:132–149, 2016b. doi: 10.1016/j.asoc.2015.11.026.

H. J. Sadaei, F. G. Guimarães, C. J. d. Silva, M. H. Lee, and T. Eslami. Short-term load forecasting method based on fuzzy time series, seasonality and long memory process. *International Journal of Approximate Reasoning*, 83:196–217, 2017. doi: 10.1016/j.ijar.2017.01.006.

H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimarães, and M. H. Lee. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 2019. ISSN 0360-5442. doi: https://doi.org/10.1016/j.energy.2019.03.081. URL http://www.sciencedirect.com/science/article/pii/S0360544219304852.

S. L. Scott and H. R. Varian. Predicting the Present with Bayesian Structural Time Series. *International Journal of Mathematical Modelling and Numerical Optimisation (IJMMNO)*, 5(1), 2014. doi: 10.2139/ssrn.2304426.

C. A. C. Severiano, P. P. C. L. Silva, H. J. H. Sadaei, and F. F. G. Guimarães. Very Short-term Solar Forecasting using Fuzzy Time Series. In *2017 IEEE International Conference on Fuzzy Systems*, Naples, Italy, 2017. ISBN 9781509060344. doi: 10.1109/FUZZ-IEEE.2017.8015732.

S. J. Sheather and M. C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 683–690, 1991.

P. C. d. L. Silva, P. O. Lucas, H. J. Sadaei, and F. G. Guimarães. pyFTS: Fuzzy Time Series for Python, 2018. URL https://doi.org/10.5281/zenodo.597359.

B. W. Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

J. P. Singh and S. Prabakaran. Quantum computing through quaternions. *Electronic Journal of Theoretical Physics*, 2008. ISSN 17295254.

P. Singh. Big Data Time Series Forecasting Model: A Fuzzy-Neuro Hybridize Approach. In D. P. Acharjya, S. Dehuri, and S. Sanyal, editors, *Computational Intelligence for Big Data Analysis: Frontier Advances and Applications*, pages 55–72. Springer International Publishing, Cham, 2015. ISBN 978-3-319-16598-1. doi: 10.1007/978-3-319-16598-1{_}2. URL https://doi.org/10.1007/978-3-319-16598-1_2https://www.researchgate.net/publication/279985721.

P. Singh and G. Dhiman. A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. *Journal of Computational Science*, 2018. doi: 10.1016/j.jocs.2018.05.008. URL https://doi.org/10.1016/j.jocs.2018.05.008.

A. Smith. *Sequential Monte Carlo methods in practice.* Springer Science & Business Media, 2013.

L. A. Smith. Predictability and chaos. In J. R. Holton, J. Pyle, and J. A. Curry, editors, *Encyclopedia of Atmospheric Sciences*, pages 1777–1785. Academic Press, London, 2003. ISBN 9780122270901.

E. A. Soares, H. A. Camargo, S. J. Camargo, and D. F. Leite. Incremental Gaussian Granular Fuzzy Modeling Applied to Hurricane Track Forecasting. In *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2018. URL https://www.researchgate.net/publication/326731587.

Q. Song. Seasonal forecasting in fuzzy time series. *Fuzzy sets and systems*, 107:235–236, 1999. doi: 10.1016/S0165-0114(98)00266-8.

Q. Song and B. S. Chissom. Forecasting Enrollments with Fuzzy Time Series - part I. *Fuzzy Sets And Systems*, 54(1):1–9, 1993a.

Q. Song and B. S. Chissom. Fuzzy time series and its models. *Fuzzy Sets and Systems*, 54 (3):269–277, 1993b. doi: 10.1016/0165-0114(93)90372-O.

Q. Song and B. S. Chissorn. Forecasting enrollments with fuzzy time series-part II. *Fuzzy Sets and Systems*, 62:1–8, 1994.

Q. Song, R. P. Leland, and B. S. Chissom. Fuzzy stochastic fuzzy time series and its models. *Fuzzy sets and systems*, 88:333–341, 1997.

I. Steinwart and A. Christmann. Estimating conditional quantiles with the help of the pinball loss. *Bernoulli*, 17(1):211–225, 2011. doi: 10.3150/10-BEJ267.

Steven M. Kay. *Intuitive probability and random processes using MATLAB.* Springer, 2006. ISBN 9780387241579. doi: 10.1198/tas.2008.s104.

B. Sun, H. Guo, H. Reza Karimi, Y. Ge, S. Xiong, H. R. Karimi, Y. Ge, and S. Xiong. Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series. *Neurocomputing*, 151:1528–1536, 3 2015. ISSN 18728286. doi: 10.1016/j.neucom.2014.09.018. URL https://doi.org/10.1016/j.neucom.2014.09.018.

I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola. Nonparametric Quantile Estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

F. M. Talarposhti, H. J. Sadaei, R. Enayatifar, F. G. Guimarães, M. Mahmud, and T. Eslami. Stock market forecasting by using a hybrid model of exponential fuzzy time series. *International Journal of Approximate Reasoning*, 70:79–98, 2016. doi: 10.1016/j.ijar.2015.12.011.

L. J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437–450, 2000. ISSN 01692070. doi: 10.1016/S0169-2070(00)00065-0. URL http://doi.org/10.1016/S0169-2070(00)00065-0.

J. W. Taylor. Forecasting Daily Supermarket Sales Using Exponentially Weighted Quantile Regression. *European Journal of Operational Research*, 178:154–167, 2007.

N. Tran, T. Nguyen, M. Nguyen, and G. Nguyen. A Multivariate Fuzzy Time Series Resource Forecast Model for Clouds using LSTM and Data Correlation Analysis. *Procedia Computer Science*, 126:636–645, 2018. doi: 10.1016/j.procs.2018.07.298. URL www.sciencedirect.comwww.sciencedirect.com.

B. Trawiński, M. Smetek, Z. Telec, and T. Lasota. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Int. J. Appl. Math. Comput. Sci*, 22(4):867–881, 2012. doi: 10.2478/v10006-012-0064-z. URL https://doi.org/10.2478/v10006-012-0064-z.

F.-M. Tseng, G.-H. Tzeng, and H.-C. Yu. Fuzzy Seasonal Time Series for Forecasting the Production Value of the Mechanical Industry in Taiwan. *Technological Forecasting and Social Change*, 60:263–273, 1999. doi: 10.1016/S0040-1625(98)00047-X.

C. Wan, J. Lin, J. Wang, S. Member, Y. Song, and Z. Yang Dong. Direct Quantile Regression for Nonparametric Probabilistic Forecasting of Wind Power Generation. *IEEE Transactions on Power Systems*, 2014. doi: 10.1109/TPWRS.2016.2625101.

L. Wang, X. Liu, W. Pedrycz, and Y. Shao. Determination of temporal information granules to improve forecasting in fuzzy time series. *Expert Systems with Applications*, 2014. ISSN 09574174. doi: 10.1016/j.eswa.2013.10.046. URL http://dx.doi.org/10.1016/j.eswa.2013.04.026.

W. Wang, W. Pedrycz, and X. Liu. Time series long-term forecasting model based on information granules and fuzzy clustering. *Engineering Applications of Artificial Intelligence*, 2015. ISSN 09521976. doi: 10.1016/j.engappai.2015.01.006.

T. White. *Hadoop: The definitive guide.* " O'Reilly Media, Inc.", 2012.

R. L. Winkler. A Decision-Theoretic Approach to Interval Estimation. *Journal of the American Statistical Association*, 67(337):187–191, 1972. doi: 10.1080/01621459.1972.10481224.

W.-K. K. Wong, E. Bai, A. Wai, C. Chu, and A. W. C. Chu. Adaptive time-variant models for fuzzy-time-series forecasting. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 40(6), 2010. ISSN 10834419. doi: 10.1109/TSMCB.2010.2042055.

J. Xie and T. Hong. GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation. *International Journal of Forecasting*, 32:1012–1016, 2016. doi: 10.1016/j.ijforecast.2015.11.005.

D. Yang, Z. Dong, L. Hong, I. Lim, and L. Liu. Analyzing big time series data in solar engineering using features and PCA. *Solar Energy*, 153:317–328, 2017a. doi: 10.1016/j.solener.2017.05.072.

S. Yang and J. Liu. Time Series Forecasting based on High-Order Fuzzy Cognitive Maps and Wavelet Transform, 2018. ISSN 10636706.

X. Yang, F. Yu, and W. Pedrycz. Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system. *International Journal of Approximate Reasoning*, 2017b. ISSN 0888613X. doi: 10.1016/j.ijar.2016.10.010.

Y. Yang, S. Li, W. Li, and M. Qu. Power load probability density forecasting using Gaussian process quantile regression. *Applied Energy*, 213:499–509, 2018. doi: 10.1016/j.apenergy.2017.11.035.

F. Ye, L. Zhang, D. Zhang, H. Fujita, and Z. Gong. A novel forecasting method based on multi-order fuzzy time series and technical analysis. *Information Sciences*, 2016. ISSN 00200255. doi: 10.1016/j.ins.2016.05.038.

O. C. Yolcu and H. K. Lam. A combined robust fuzzy time series method for prediction of time series. *Neurocomputing*, 2017. doi: 10.1016/j.neucom.2017.03.037.

H.-K. Yu. Weighted fuzzy time series models for TAIEX forecasting. *Physica A: Statistical Mechanics and its Applications*, 349(3):609–624, 2005. doi: 10.1016/j.physa.2004.11.006.

L. A. Zadeh. Fuzzy Sets. *Infomation and Control*, 8:338–353, 1965. doi: 10.1016/S0019-9958(65)90241-X. URL https://doi.org/10.1016/S0019-9958(65)90241-X.

L. A. Zadeh. Probability Measures of Fuzzy Events. *Journal of Mathematical Analysis and Applications*, 23(2):421–427, 1968.

L. A. Zadeh. Fuzzy Probabilities. *Information Processing & Management*, 20(3):363–372, 1984.

L. A. Zadeh. FUZZY SETS AND INFORMATION GRANULARITY. In *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*, pages 433–448. World Scientific Publishing Co., 1996. ISBN 981-02-2422-2. doi: 10.1142/9789814261302{_}0022.

S. S. Zhang, S. S. Zhang, D. Yu, W. Zhang, N. Huang, S. S. Zhang, S. S. Zhang, D. Yu, and N. Huang. A novel method based on FTS with both GA-FCM and multifactor BPNN for stock forecasting. *Soft Computing*, 2018a. doi: 10.1007/s00500-018-3335-2. URL https://doi.org/10.1007/s00500-018-3335-2.

W. Zhang, S. Zhang, and S. Zhang. Two-factor high-order fuzzy-trend FTS model based on BSO-FCM and improved KA for TAIEX stock forecasting. *Nonlinear Dynamics*, 2018b. doi: 10.1007/s11071-018-4433-5.

L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos. Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 2017. ISSN 18728286. doi: 10.1016/j.neucom.2017.01.026.

Appendix A

Monovariate Benchmark Datasets

The Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)[1] is a well known economic time series data commonly used in the FTS literature. This dataset is sampled from 1995 to 2014 time window, and has the averaged daily index by business day. This is a stationary time series dataset whose Augmented Dickey-Fuller (ADF) statistic is -2.65 where the critical value for $\alpha = 0.05$ is -2.86.

The National Association of Securities Dealers Automated Quotations - Composite Index (NASDAQ ÎXIC)[2] is an economical index already used in the FTS literature . The historical data was sampled from 2000 to 2016 time window, and has the averaged daily index by business day. This is a stationary time series dataset whose ADF statistic is 0.04 where the critical value for $\alpha = 0.05$ is -2.86.

The S&P500 - Standard & Poor's 500 [3] is a market index composed by 500 assets quoted on New York Stock Exchange and Nasdaq. This dataset contains the averaged daily index, by business day, from 1950 to 2017 with 16000 instances. This is a stationary dataset whose ADF Statistic is 0.00 where critical value for $\alpha = 0.05$ is -2.86.

In order to contribute with the research reproducibility, all data and source codes are available in the following URL `http://bit.ly/scalable_probabilistic_fts_appA`.

[1] `http://www.twse.com.tw/en/products/indices/Index_Series.php`. Access in 23/05/2016

[2] `http://www.nasdaq.com/aspx/flashquotes.aspx?symbol=IXIC&selected=IXIC`. Access in 23/05/2016

[3] `https://finance.yahoo.com/quote/%5EGSPC/history?p=%5EGSPC`. Access in 19/03/2017

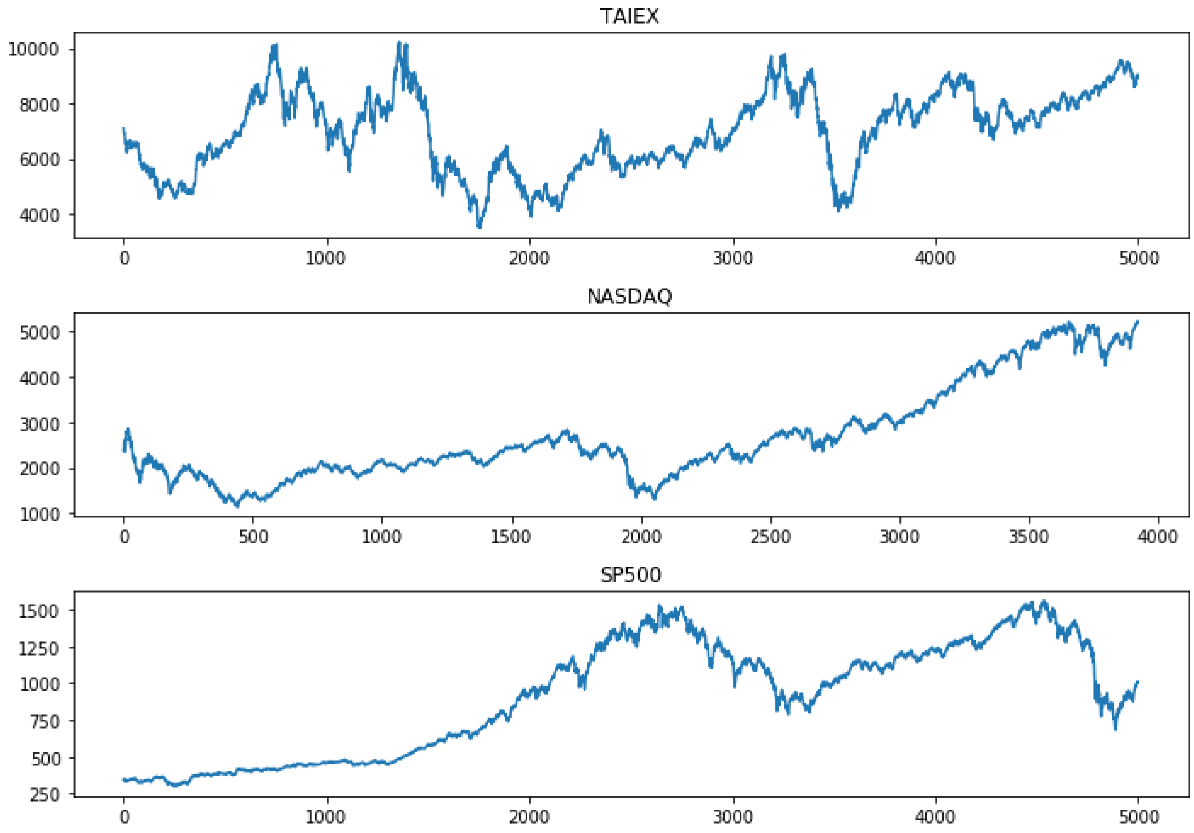


Figure 64 – Benchmark datasets

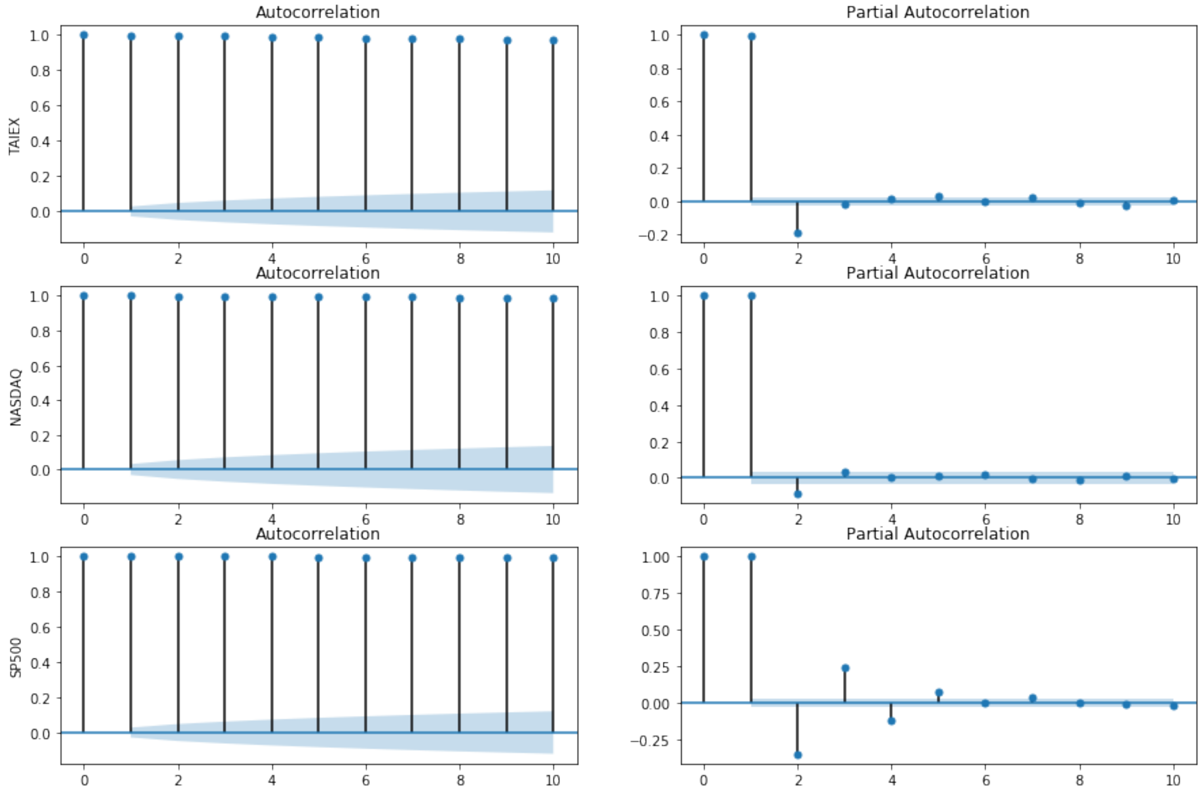


Figure 65 – Autocorrelation and Partial Autocorrelation plots for each benchmark dataset

Appendix B

Multivariate and Large Benchmark Datasets

In recent years the search for renewable energy sources has grown and, as most of them are not perennial power sources, its integration on smart environments will require ability to predict their output generation.

In order to contribute with the research reproducibility, all data and source codes are available in the following URL http://bit.ly/scalable_probabilistic_fts_appB.

B.1 SONDA dataset

The Project SONDA - Sistema de Organização Nacional de Dados Ambientais (Brazilian National System of Environmental Data Organization), is a governmental project which groups environmental data (solar radiance, wind speed, precipitation, etc) from INPE - Instituto Nacional de Pesquisas Espaciais (Brazilian Institute of Space Research). The chosen variables are the global solar horizontal radiation and the wind speed at 10 meters, both from the Brasilia telemetry station[1], recorded between 2012 and 2015, by minute, summing 2 million instances. This dataset was retrieved directly from the SONDA Project page at http://sonda.ccst.inpe.br/[2]

Variable	**Type**	**Description**
DateTime	Time Stamp	yyyy-MM-dd HH:MM
glo_avg	Real	Global average solar radiation in
ws_10m	Real	Wind speed in meters by second (m/s)

Table 40 – SONDA dataset variables

[1] Code: BRB. Coordinates: 15°36’ 03" S 47°42’47" O. Alt.: 1023m

[2] Access in 19/05/2019

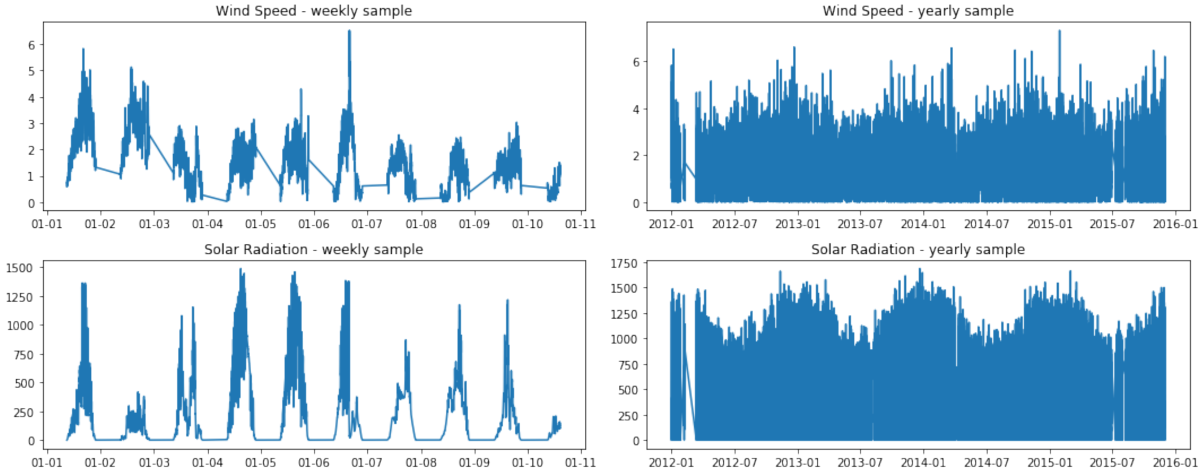


Figure 66 – SONDA dataset samples

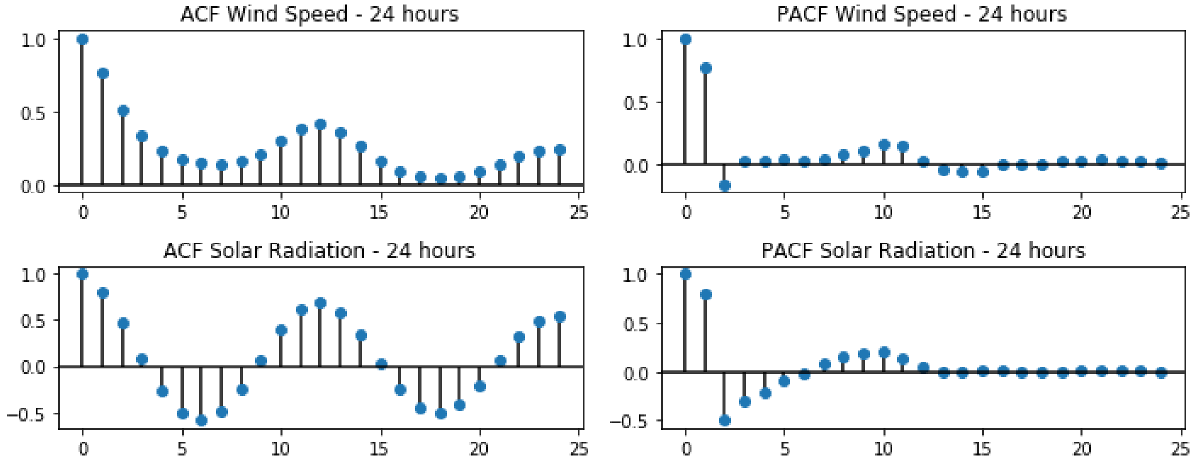


Figure 67 – Autocorrelation and Partial Autocorrelation plots for SONDA dataset

B.2 Malaysia dataset

Hourly electric load and temperature data of the power supply company of the city of Johor in Malaysia sampled between 2009 and 2010, with 17,519 instances. This dataset was retrieved from Sadaei et al. [2019].

Variable	Type	Description
DateTime	Time Stamp	yyyy-MM-dd HH:MM
temperature	Real	Temperature in Celcius degrees (^{o}C)
load	Integer	Eletric load in Mega Watts by hour (MW/h)

Table 41 – Malaysia dataset variables

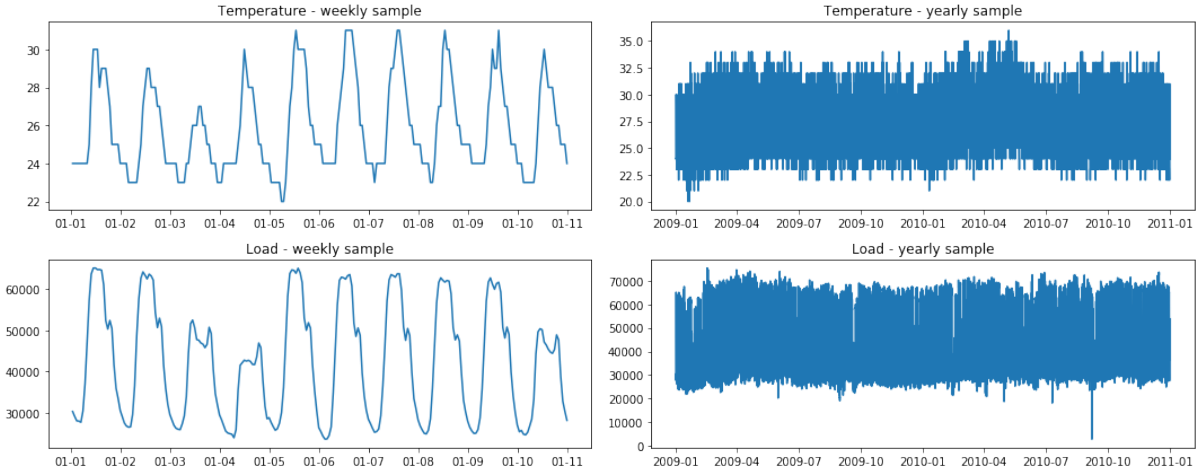


Figure 68 – Malaysia dataset samples

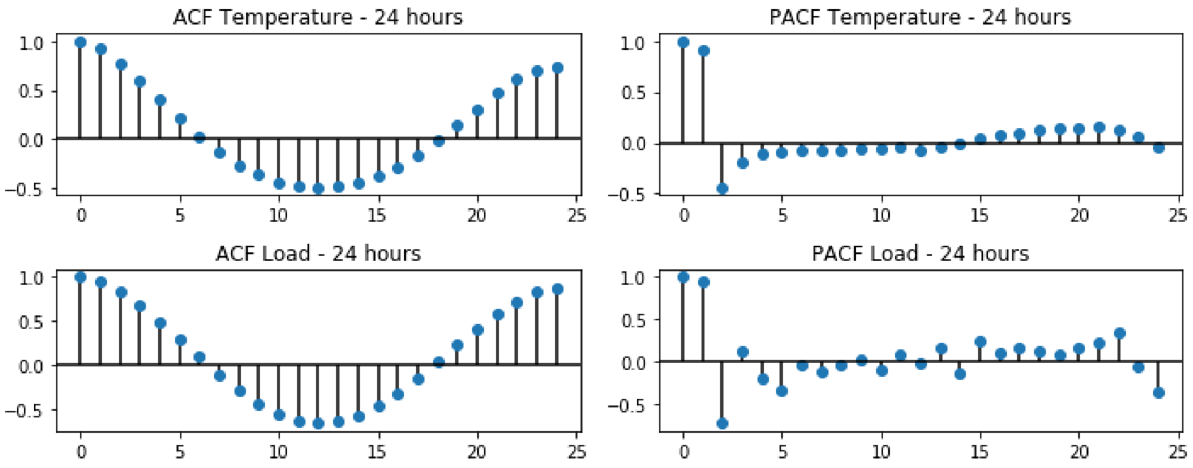


Figure 69 – Autocorrelation and Partial Autocorrelation plots for Malaysia dataset