Carlos Alberto Severiano Junior

Evolving Spatio-temporal Forecasting Models for Renewable Energy Systems

Belo Horizonte - Minas Gerais

November, 2020

Carlos Alberto Severiano Junior

Evolving Spatio-temporal Forecasting Models for Renewable Energy Systems

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: Miri Weiss Cohen

Belo Horizonte - Minas Gerais

November, 2020

S498e

Severiano Junior, Carlos Alberto.
Evolving spatio-temporal forecasting models for renewable energy systems [recurso eletrônico] / Carlos Alberto Severiano Junior. - 2020.
1 recurso online (126 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.
Coorientadora: Miri Weiss Cohen.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 125-126.

Bibliografia: f. 117-123.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Energia eólica - Teses. 3. Energia renovável - Teses. 4. Energia solar - Teses. 5. Séries temporais - Teses. I. Guimarães, Frederico Gadelha. II. Cohen, Miri Weiss. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica: Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG.

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

"Evolving Spatio-temporal Forecasting Models for Renewable Energy Systems"

Carlos Alberto Severiano Junior

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 27 de novembro de 2020.

Por:

Frederico G. Guimarães
____
Prof. Dr. Frederico Gadelha Guimarães
DEE (UFMG) - Orientador

Miri Weiss Cohen
____
Prof. Dr. Miri Weiss Cohen
Department of Software Engineering (Braude College of Engineering, Israel)

R. Ballini
____
Prof. Dr. Rosângela Ballini
Departamento de Teoria Econômica (UNICAMP)

Eduardo Pestana de Aguiar
____
Prof. Dr. Eduardo Pestana de Aguiar
Departamento de Engenharia de Produção e Mecânica (UFJF)

Rodrigo César P. Silva
____
Prof. Dr. Rodrigo César Pedrosa Silva
Departamento de Computação (UFOP)

Carpi
____
Prof. Dr. Laura Corina Carpi
PPGMMC (CEFET-MG)

Acknowledgements

First of all, I would like to thank my parents, Tereza and Beto. All the admiration and respect I have for them guided me in the decisions that brought me here. I am very grateful for the support, affection and understanding that they and my dear brothers Glaucia and Marcus have always had with me, even in the toughest times.

Gratitude to my dear Luana, who was by my side in the most difficult moments of this trajectory, always giving me the strength to move on. This force was multiplied by a thousand with the greatest gift of our lives, who came close to the end of this thesis! My little Maria Tereza: as you arrived now, you may not yet understand some parts of this work. But know that the brightness of your arrival brought me the energy I needed to close this job. Daddy loves you very much!

I would also like to thank my partners during the writing of this thesis: Favela, Pagode, Gorda, Veveta and Ninja. Always present, they helped a lot, even when accidentally stepped on the keyboard ...

In this work I was fortunate and honored to be mentored by extraordinary people. My advisor, prof. Frederico Gadelha, to whom I am very grateful for the help in conducting this work. His inspiring serenity and genius helped me (and still help!) a lot in the work. And my co-advisor, prof. Miri Cohen, who always kept a regardful eye on my work, helped me to build each stage and continues to inspire me to seek new challenges. I hope to continue collaborating with both in projects here and around the world. :)

I am also very grateful to all the friends who helped me in this step. PPGEE teachers and students for all the support in research and disciplines. The teachers and students of IFMG Sabará, for their support during this work, always with full understanding of the challenges that the conciliation between research and teaching presents us. My dear dr. André Nobre, who asked me in 2015 for help with ANN for solar energy time series and then brought me back to the game. And thanks to all my dear colleagues at the MINDS lab, really special people. Thank you very much for your coexistence, friendship and countless help along this path. Long live to the lab! And I hope that we will be together in the next projects and "pau-de-shot" rounds. :D

“Adaptability is not imitation. It means power of resistance and assimilation.” - Mahatma Gandhi

Resumo

Sistemas de energias renováveis, como energia solar fotovoltaica e eólica, são fontes de energia muito sensíveis a variações climáticas, o que pode afetar seus padrões de geração. É muito importante usar mecanismos que possam ajudar a antecipar tais variações e possibilitar uma tomada de decisão mais informada. Métodos de previsão podem contribuir para essa tarefa e, portanto, sua aplicação nessa área tem sido amplamente estudada. Os métodos de previsão geralmente utilizam dados históricos da série temporal gerada pelo ponto de interesse. Para melhorar a precisão desses resultados, as informações disponíveis no espaço também vem sendo aplicadas aos métodos de previsão. Essas abordagens, chamadas de métodos espaço-temporais, utilizam todos os dados disponíveis coletados em diferentes localidades. Em energias renováveis, variações observadas em localidades vizinhas podem acontecer em algum ponto de interesse em um futuro próximo, dado que muitos desses eventos são resultado de fenômenos climáticos. Isso reforça a possibilidade de que a análise de dados espaço-temporais possa melhorar o desempenho da previsão em sistemas de energias renováveis. Além disso, eventos climáticos tendem a influenciar os padrões observados nas séries temporais relacionadas com a produção de energia no sistema, de modo a apresentarem não-estacionariedade. Tal cenário demanda o desenvolvimento de mecanismos que permitam ao modelo de previsão se adaptar às mudanças nos padrões das series temporais. Nesta tese são apresentadas propostas para o tratamento de tais problemas relacionados com a previsão de energias renováveis. A partir da extensão de modelos de Fuzzy Time Series (FTS), são aplicadas propostas para, primeiramente, lidar com o problema de não-estacionariedade de series temporais de energias renováveis a partir de um mecanismo de adaptação do modelo. Em seguida, um modelo que também propõe um mecanismo de adaptação, alinhado ao processamento de dados multivariados, é apresentado e avaliado quanto à previsão de energias solar e eólica. Os mecanismos de adaptação analisados se mostram capazes de prover um ganho de desempenho para os modelos propostos, assim como o uso de dados multivariados dispostos em um contexto de um problema espaço-temporal. O modelo e-MVFTS, que integra uma técnica evolutiva de clusterização com um modelo FTS para desempenhar previsão espaço-temporal, apresentou resultados comparáveis a modelos de maior complexidade e abrangência. O modelo ainda apresenta como vantagens sua robustez de parâmetros e capacidade de adaptação às mudanças dos dados sem necessidade de novas etapas de treinamento. Seu mecanismo de adaptação provê uma maior flexibilidade a modelos FTS, uma vez que não requer uma configuração prévia de seu particionamento e é capaz de adaptar tal estrutura dinamicamente durante sua execução. Além disso, o algoritmo de clusterização utilizado no modelo foi originalmente desenvolvido para problemas de fluxo de dados, sendo portanto apto a lidar com grandes volumes de informação. Tais características o posicionam como uma extensão dos modelos FTS aplicável ao problema de previsão de energias renováveis. Adicionalmente, sua fundamentação baseada em FTS o tornam um modelo cuja representação de regras é de maior facilidade de entendimento, sendo este um incentivo adicional para que possa ser adotado no suporte a tomada de decisão em sistemas de energias renováveis. O e-MVFTS ainda apresentou bons resultados em bases não-estacionarias geradas artificialmente. Portanto, uma avaliação do modelo aplicado a outros problemas de previsão pode ser uma direção para trabalhos futuros.

Palavras-chave: Energias Renováveis; Energia Solar; Energia Eólica; Séries Temporais Nebulosas; Previsão Espaço-temporal; Modelos Evolutivos.

Abstract

Renewable energy systems such as solar photovoltaics and wind are sources of energy very sensitive to climate variations, which can affect their generation patterns. It is very important to use mechanisms that can help to anticipate such variations and enable more informed decision-making. Forecasting methods can contribute to this task and therefore their application in this area has been widely studied. Forecasting methods usually take as input historical data from the time series generated by the point of interest. For a further improvement in forecasting accuracy, the information available in space has been also added to forecasting methods. These approaches, called spatio-temporal methods, make use of all the available data collected from different locations. In renewables, variations observed at neighbor locations may occur in the near future at some point of interest, since many of these events are result of climatic phenomena. This reinforces the possibility that spatio-temporal data analysis can improve forecasting performance in renewable energy systems. In addition, climatic events tend to influence the patterns observed in the time series related to energy production in the system, thus presenting non-stationarity. Such scenario demands the development of mechanisms that allow the forecasting model to adapt to changes in time series patterns. In this thesis, proposals for the treatment of such problems related to renewable energy forecasting are presented. From the extension of Fuzzy Time Series (FTS) models, proposals are applied to first deal with the non-stationarity problem using a model adaptation mechanism. Then, another model that also proposes an adaptation mechanism, aligned with the processing of multivariate data, is presented and evaluated regarding the forecast of solar and wind energy. The analyzed adaptation mechanisms were able to provide a performance gain for the proposed models, as well as the use of multivariate data arranged in the context of a spatio-temporal problem. The e-MVFTS model, which integrates an evolving clustering technique with an FTS model to perform spatio-temporal forecasting, presented results comparable to models of greater complexity and scope. The model also presents the advantages of its robustness of parameters and the ability to adapt to changes in data without the need for new training steps. Its adaptation mechanism provides greater flexibility to FTS models, since it does not require a previous configuration of its partitioning scheme and is able to adapt this structure dynamically during runtime. In addition, the clustering algorithm used in the model was originally developed for data stream problems, and is therefore able to handle large volumes of information. These characteristics position it as an extension of the FTS models applicable to the renewable energy forecasting problem. In addition, its FTS-based rationale makes it a model whose representation of rules is easier to understand, which is an additional motivation for being adopted in support of decision making in renewable energy systems. The e-MVFTS also showed good results on artificially generated non-stationary data sets. Therefore, an assessment of the model applied to other forecasting problems can be a direction for future work.

Keywords: Renewable Energy; Solar Energy; Wind Energy; Fuzzy Time Series; Spatio-temporal forecasting; Evolving models.

List of Figures

List of Tables

List of Algorithms

List of abbreviations and acronyms

ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Networks
AR	Autoregressive
CNN	Convolutional Neural Networks
FBeM	Fuzzy-set-Based Evolving Modeling
$\mathcal{FIG}$-FTS	Fuzzy Information Granular
FLR	Fuzzy Logical Relationship
FLRG	Fuzzy Logical Relationship Group
FTS	Fuzzy Time Series
HOFTS	High Order FTS
LHS	Left Hand Side
LSTM	Long Short-Term Memory
MF	Membership Function
MLP	Multi Layer Perceptron
NSFS	Non-stationary Fuzzy Sets
NSFTS	Non-stationary Fuzzy Time Series
NWP	Numeric Weather Prediction
PCA	Principal Component Analysis
PV	Photovoltaic
RHS	Right Hand Side
RMSE	Root Mean Square Error

RNN	Recurrent Neural Networks
SMAPE	Symmetric Mean Absolute Percentage Error
SSA	Singular Spectrum Analysis
TEDA	Typicality and Eccentricity Data Analysis
UoD	Universe of Discourse
VAR	Vector Autoregressive
e-MVFTS	Evolving Multivariate FTS
eTS	Evolving Takagi-Sugeno

List of symbols

Y	crisp time series
$y(t)$	crisp time series elements at time t
t	time index
U	Universe of Discourse of Y
$\mu_{Aj}(x)$	membership function for a fuzzy set A_j, given a crisp sample x
F	fuzzy time series
$f(t)$	fuzzy time series elements at time t
Ω	FTS model order
k	number of fuzzy sets
$\mathcal{G}$	fuzzy information granule
$\mathcal{V}$	fuzzy linguistic variable set
$\pi()$	perturbation function
δ	displacement factor
ρ	scale factor
$mp(A_j)$	midpoint value of fuzzy set A_j
ϕ	cumulative proximity
$\xi(x_t)$	eccentricity of a sample x at time t
$\zeta(x_t)$	normalized eccentricity of a sample x at time t
$\tau(x_t)$	typicality of a sample x at time t
$\gamma(x_t)$	normalized typicality of a sample x at time t
S_t^i	number of data samples of a micro-cluster i at time t

μ_t^i	center of a micro-cluster i at time t
${\sigma^2}_t^i$	micro-cluster variance of a micro-cluster i at time t
D_t^i	density of a micro-cluster i at time t
$m_t^i(S_t^i)$	outlier threshold parameter
r_0	MicroTEDAclus variance limit parameter
$\mathfrak{Ch}$	set of changed micro-clusters

Contents

Chapter 1

Introduction

1.1 Renewable Energy Market

Renewable energy systems are consistently reaching greater space in the global energy market. In 2019, estimates by the International Energy Agency (IEA) IEA [2019] for the next 5 years indicated that the power capacity of this energy source was set to expand by 50% between 2019 and 2024. This expansion was led by solar photovoltaics, which alone accounts for almost 60% of the expected growth followed by onshore wind representing 25%. A global energy report by IEA in 2020 IEA [2020] reinforced the estimates. The share of power generation attributed to renewables was 28% in the first quarter of 2020. In comparison, this value was 26% in 2019 for the same period. Solar photovoltaics and wind were accounting for 86% of global renewable capacity.

According to ANEEL [2018], renewable energy corresponds to 83% of the energy matrix in Brazil. Hydropower is still a dominant source accounting for about 63% of participation, followed by wind energy, with 9.3%. Solar energy has 1.4% of participation. Given that the country is located in a region of high solar incidence (the sun belt countries), the market of solar photovoltaic energy can be considered still little explored given its potential.

At the time of the writing of this thesis, the world is affected by an unprecedented shock in peacetime caused by the coronavirus pandemic. In order to reduce the spread of the virus, governments across the world have imposed restrictions on most mobility, social and economic activities. The impact of such restrictions on energy use is highly asymmetrical and affect different energy sources according to the energy use which they are more strongly related. For instance, energy use to power data centers and server farms was not as affected as aviation jet fuel, which had a sharp drop. In a greater scope, restrictions in mobility are an important impact on fossil energy sources, such as oil. Initial IEA evaluations point that during 2020 the energy demand could decline by circa 6%. An analysis of this scenario in different regions is presented in IEA [2020] and indicates that the use of renewable energy sources is almost unaffected and even presented some growth. Figures 1 and 2 show the evolution of the energy mix in China and the European Union, respectively, during the first quarter of 2020. In both cases, renewables have presented an important share of the overall energy use, suggesting their strong application as a source for energy used in essential activities. A projected change in primary energy demand by energy source in 2020 relative to 2019 is presented in Figure 3. The only demand expected to increase is for renewables, especially because of low operating costs and preferential access to many power systems. A future after the pandemic can potentially change the priority of investments and government policies, mainly due to lower budgets and economy reformulation. Given the mentioned aspects, the investments in competitive renewable energy systems can be an interesting alternative, since their low costs may also provide reduced CO2 emissions and foster technology innovation. This reinforces the importance of designing robust and reliable renewable energy systems.

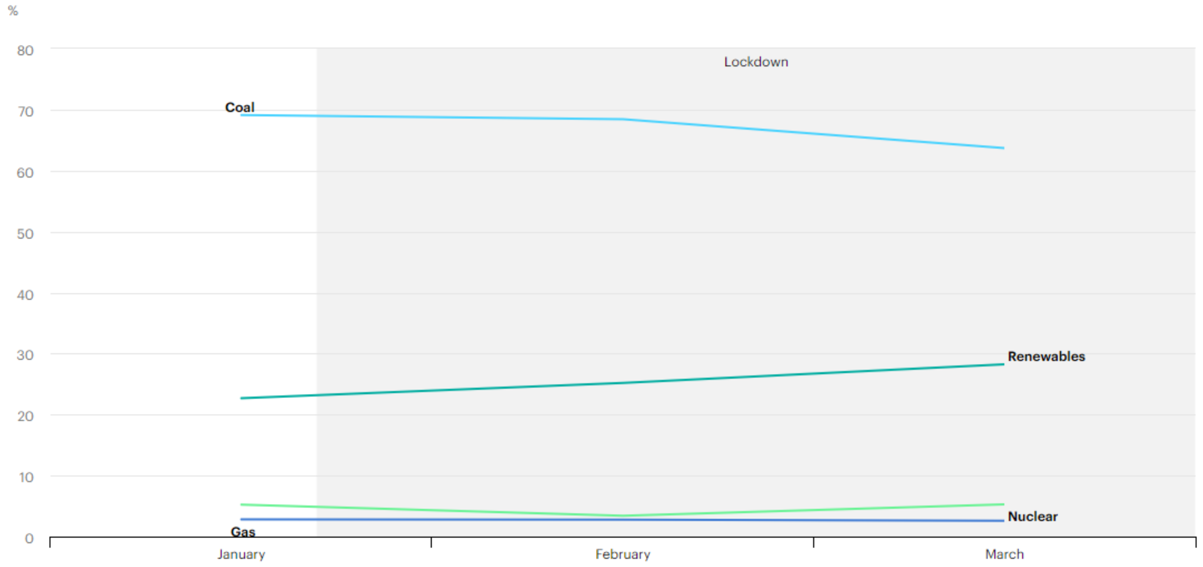


Figure 1 – Energy mix in China during Q1, 2020

Source: [IEA, 2020]

Figures 1 and 2 also suggest that this scenario constitutes a concept drift event in the demand that has been causing changes in the energy sector. For the case of renewable energies, it is possible that drifts also occur in the energy generation. For example, in Peters et al. [2020], a study carried out in New Delhi, India, observed that the mobility restriction caused a reduction in air pollution in the region, resulting in significant increases in solar irradiance for the initial periods of the restrictive measures. That is, there is also a drift that causes an increase in the generation of solar energy. Therefore, the models applied to energy systems must be able to consider and adapt to such events, to better understand their functioning and estimate future values with greater assertiveness.

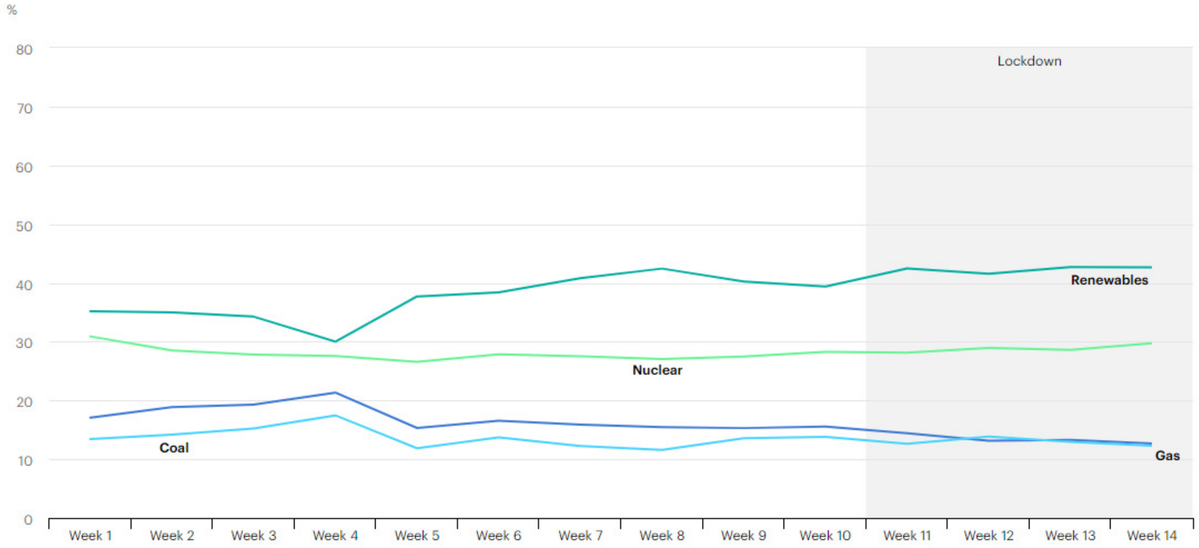


Figure 2 – Energy mix in European Union during Q1, 2020

Source: [IEA, 2020]

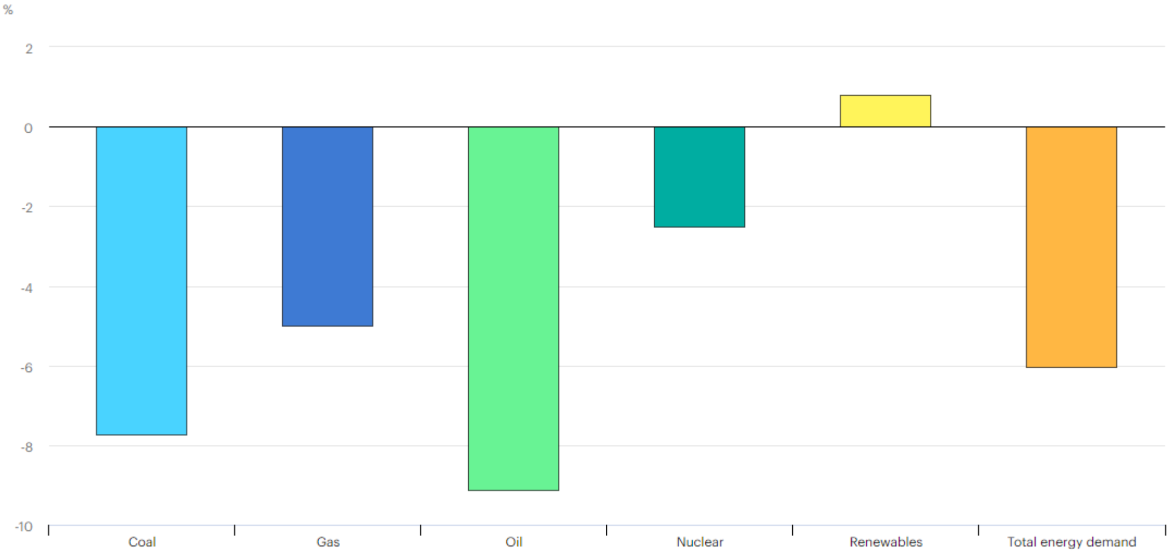


Figure 3 – Projected change in primary energy demand by fuel in 2020 relative to 2019

Source: [IEA, 2020]

1.2 Renewable Energy Forecasting

Renewable energy systems also bring an uncertainty component related to both its supply and demand. The risk around this problem can be mitigated with the application of forecasting methods. Broadly speaking, forecasting is a prediction of some future events. According to Montgomery et al. [2015], forecasting is often classified as short-term, medium-term, and long-term. Short-term problems involve predicting events only for a few time periods into the future (days, weeks, months). At this point there is also the concept of very short-term (or nowcasting), where minutes or hours are predicted.

Medium-term forecasting extends from 1 to 2 year into the future and long-term problems refer to many years ahead. Usually, short and medium-term forecasts are more suitable for activities carried out more frequently, such as operations management, budgeting and trading. Long-term forecasts are more concerned to strategic planning.

As discussed in Sweeney et al. [2020], renewable energy forecasts users can be divided in two main groups: energy market participants and power system operators. The former deals with buying and selling operations, while the latter focuses on the maintenance of power supply. Energy markets work in different (and complementary) timescales. Long-term trading represents agreements for weeks and years ahead, while day-ahead markets guide the schedules in power plants, which can receive fine adjustments according to the events during the intra-day markets. For all the cases, an accurate forecasting is very important to assist the trading operations. Power system operators also make decisions which can be supported by forecasting models. Among their tasks, a mix of energy appropriate to the market demand must be provided. It can be more accurate if a proper estimate of power generation from the different sources is available. The technology used in the operation of energy systems has been evolving consistently, as well as the information systems that support the energy markets. Operations occur more quickly, consequently requiring forecasting models that can return values in smaller intervals. In this context, very short-term forecasting models have gained more space in renewable energy systems. A fast turnaround of results is an important feature to the model, where the time for decision making is reduced and the amount of data to be processed is usually higher. Hence, in addition to good accuracy, good performance is a desirable goal for a very short term forecasting model. In this thesis, the models analyzed focuses on very short-term forecasting.

The solutions presented herein aim to contribute mainly in three aspects for forecasting models in renewable energy systems: a) processing of spatio-temporal information, b) ability to adapt to data and c) model interpretability. These concepts are introduced in the next paragraphs.

The application of spatio-temporal models has been encouraged by the greater availability of data. It is increasingly common to provide near real-time data through different automated sources, such as Supervisory control and data acquisition (SCADA) systems and remote sensing [Sweeney et al., 2020]. From this, spatio-temporal models become more feasible, since consist of the analysis of time series observed from different locations, geographically distributed over a region of study. Two aspects are the object of interest: *(i) spatial*, where given the distance between the event and the point of forecasting, the range of influence is observed and *(ii) temporal*, where the analysis consists of determining time intervals from the event *lags*, that can still exert some influence on a forecasting model. The main idea is that the identification of phenomena occurring in different locations can influence their neighborhood and, therefore, constitute useful information for a more decisive forecasting. Figure 4 exemplifies such situation in a wind farm, in which different climatic events observed in neighboring stations, can influence the performance of the system at different time intervals. To improve the accuracy of this method, a spatio-temporal approach might consider not only historical data collected from a specific site but also analyze relevant data from neighbor stations.

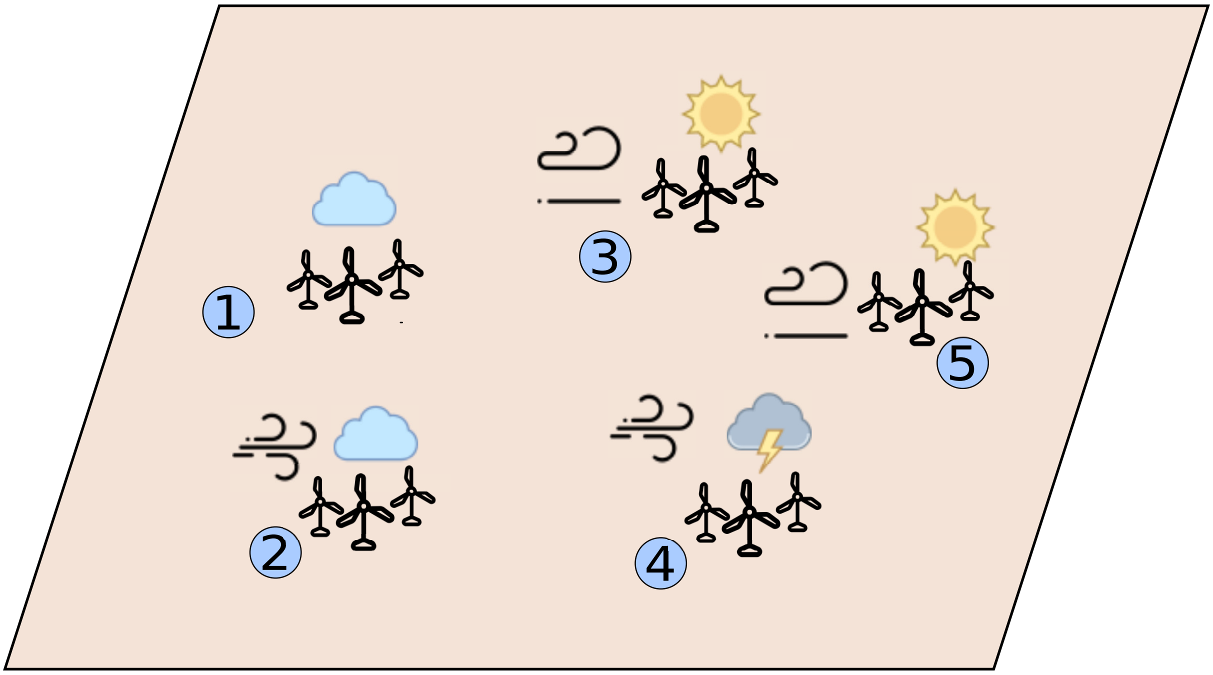


Figure 4 – Wind energy system subject to different climate events

The uncertainty component in renewable energy systems is mainly influenced by climatic conditions that directly affect their generation. This can be seen when related time series are analyzed. Figure 5 shows a time series representing three days of solar irradiance, which is a value directly related to the generation of solar photovoltaic energy. The series present a daily seasonality related to the daily solar cycle. When several months are analyzed, changes in their patterns influenced by the different climatic seasons can be observed. Such patterns are also constantly changing due to other additional events, such as global warming. In Figure 6, monthly solar irradiance boxplots are shown to illustrate such changes throughout the years. Therefore, a model capable of accurately forecasting the power generation today may not be suitable to operate in the coming months or years. Therefore, it is important that a forecasting model for renewables can provide good ability of generalization to deal with different climatic seasons, but still be able to adapt to future changes. Given this motivation, this work also assesses and proposes adaptation mechanisms to address better forecasts.

Adequate accuracy and fast turnaround are characteristics that enable a model to solve forecasting problems. However, in a context where the model has the role of

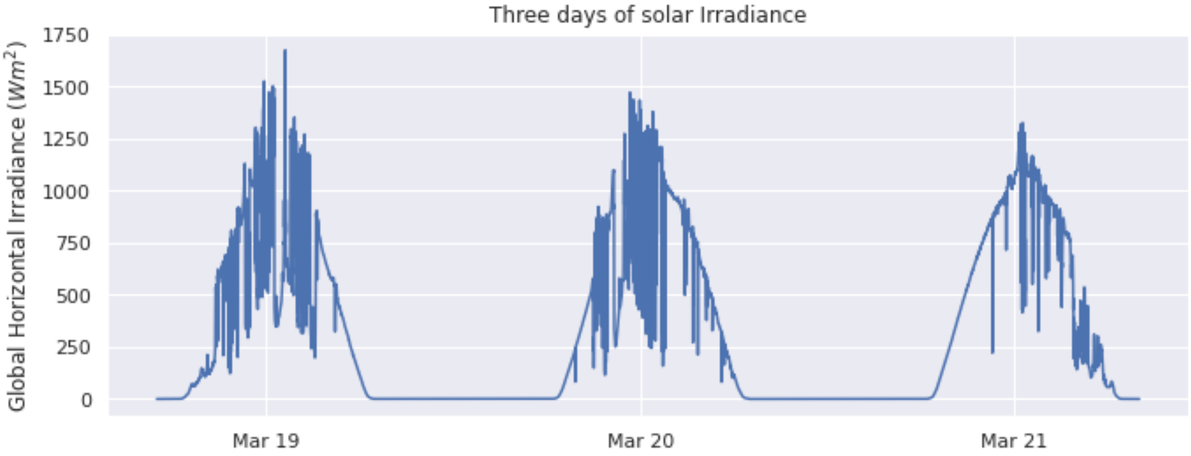


Figure 5 – Solar Irradiance Time Series

Source: [NREL, 2018]

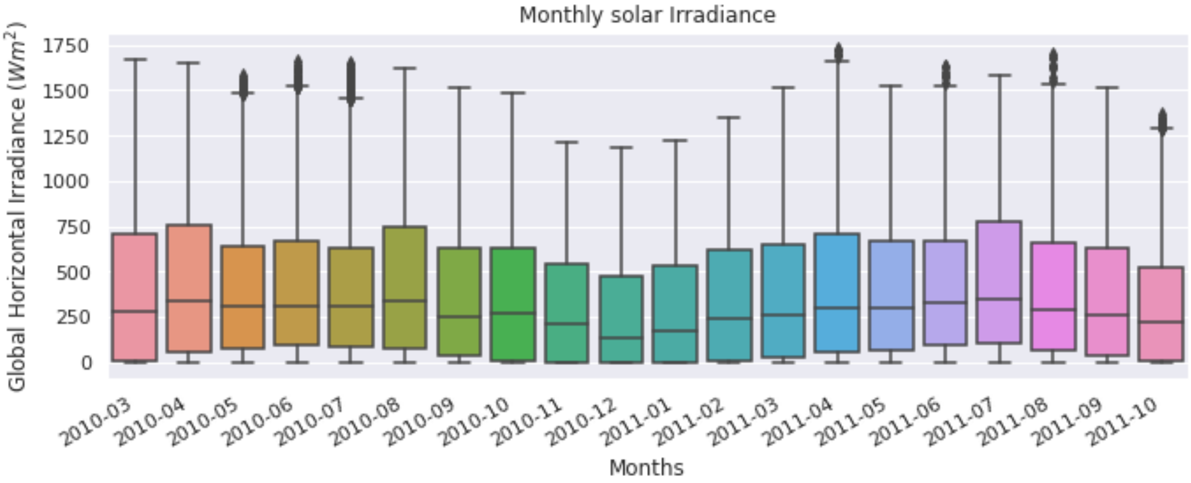


Figure 6 – Monthly Solar Irradiance Boxplots

Source: [NREL, 2018]

assisting in decision making, it is also necessary for the user to trust the returned result. In this sense, machine learning research has been making an effort in interpretability. The concept, discussed in Miller [2019], can be described as the degree to which a human can understand the cause of a decision provided by a machine learning model. The higher the interpretability, the easier the understanding of the output is. In the context of renewable energy, it is expected that, when choosing more interpretable models, the decision made from the result can be of better quality. As will be discussed in the following sections, many of the prediction models for renewables invest in greater complexity, representation and computation, to achieve better performance. In this work, a balance between good performance and interpretability is taken into account, in order to design more useful forecasting models.

1.3 Objectives

The main objective of this thesis is to propose forecasting models for renewable energy generation, more specifically solar and wind. Therefore, the following objectives are addressed in this work:

- Analyze the characteristics of time series related to the generation of solar and wind energy;
- Discuss mechanisms that can adapt the models to changes inherent to the data of the analyzed time series;
- Propose a model able to use the information associated with spatio-temporal data to provide a more accurate forecasting;
- Propose scalable models to deal with higher volume multivariate data;
- Combine such characteristics in models based on less complex concepts, to provide more easily interpretable models and, consequently, with better possibilities of integration to real problems.

1.4 Major Contributions

Guided by the aforementioned objectives, this thesis presented some contributions, the main ones being listed below:

- An analysis of the correlations between spatio-temporal data in solar and wind energy time series, with a focus on the application of a decomposition technique that helps in the identification of such correlations;
- An assessment of the impact of the use of adaptive mechanisms in FTS models applied to renewable energy time series, whose patterns indicate non-stationarity;
- The proposal of evolving forecasting algorithm based on FTS and capable of processing multivariate time series;
- The application of the proposed new model to the problem of renewable energy forecasting, using spatio-temporal data.

1.5 Work Structure

To advance the objectives discussed above, the thesis is organized in the following chapters:

- **Chapter 2 -** Solar and Wind Energy Data: performs an analysis of time series related to solar and wind energy, with the objective of highlighting their characteristics that constitute problems to be worked on during the thesis, in order to obtain adequate forecasting models.

- **Chapter 3 -** Literature Review: presents a review of different approaches present in the literature to solve the renewable energy forecasting problems.

- **Chapter 4 -** Fuzzy Time Series: discusses the concept of the Fuzzy Time Series (FTS), which underlies the solutions proposed in this work.

- **Chapter 5 -** Non-stationary Fuzzy Time Series: details the Non-stationary FTS (NSFTS) method, a solution created with the proposal to adapt to the changes present in non-stationary data. In this thesis, NSFTS is applied to renewable energy forecasting problems.

- **Chapter 6 -** Evolving Clustering Algorithm: presents microTEDAclus, an evolving clustering algorithm originally proposed to deal with data stream problems. In this thesis, it supports the proposal of a spatio-temporal forecasting model.

- **Chapter 7 -** Evolving Multivariate Fuzzy Time Series: introduces the Evolving Multivariate FTS (e-MVFTS), a novel evolving spatio-temporal forecasting model. In this work, it is evaluated regarding its performance in forecasting renewable energy, within a context in which multivariate time series are applied.

- **Chapter 8 -** Literature Review: concludes the thesis, summarizing all the discussions presented. In addition, new directions for future work are proposed based on what was conducted in this thesis.

Chapter 2

Solar and Wind Energy Data

As introduced in the previous chapter, one of the central challenges in the problem of renewable energy forecasting is the uncertainty in time series patterns, resulting from the nature of the components involved, subject to different climatic factors. In addition, the analysis of the influence of previous observations or external events related to the main problem is a common practice in the elaboration of adequate solutions for time series forecasting. In this way, this chapter performs an analysis of time series related to the generation of solar and wind energy in order to draw a profile of the type of data in study. More specifically, the degree of uncertainty associated with these time series as well as the degree of influence of past observations (lags), seasonality and spatio-temporal data. The characteristics observed in this analysis illustrate problems that the solutions proposed in this thesis aim to solve.

2.1 Solar Energy Data

The performance and capacity of a photovoltaic energy system can be assessed from the amount of solar resources available in the study region. In this context, the acquisition of solar irradiance data is of great importance, as it directly represents the solar resource used by a PV system [Nobre, 2015]. Solar irradiance can be divided into three principal components: global horizontal irradiation (GHI), diffuse horizontal irradiation (DHI) and direct normal irradiation (DNI). Also known as solar beam irradiance, DNI refers to the irradiance measured at a surface perpendicular to the Sun, excluding the losses occurred in the atmosphere. DHI, in turn, represents the irradiance at the surface gathered from light scattered in the atmosphere. GHI can be calculated from a sum of DHI and DNI, where the latter is weighted by the cosine of the solar zenith angle z, as depicted below:

$$GHI = DNI \times cos(z) + DHI \tag{2.1}$$

In this thesis, solar irradiance time series used in analysis and experiments refer to GHI.

Variability in Solar Irradiance

Time series related to solar energy often present inconstant patterns. One of the most notable contributors to such variability in a short-term horizon is the cloud motion. Figure 7 compares solar irradiance time series with resolution of 10 min in clear sky and broken clouds days, from a PV station in Oahu, USA [NREL, 2018]. It highlights the significant changes presented when the location is subjected to different climate conditions and the challenges associated to short-term forecasting using such time series.

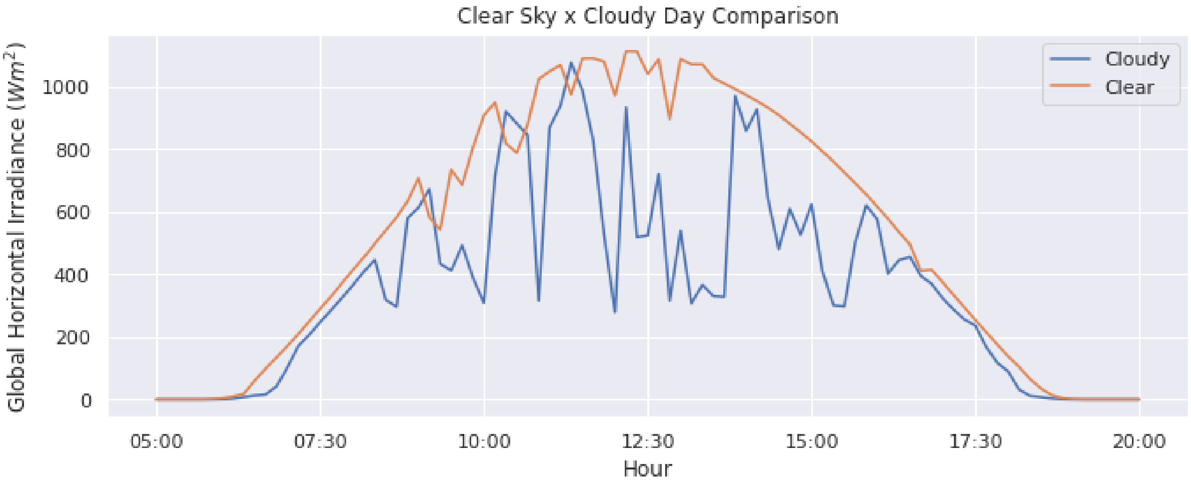


Figure 7 – Solar irradiance measurements in clear sky and cloudy days

The grouped monthly boxplots in Chapter 1, Figure 6, indicate that variations in mean and variance of solar irradiance data also can occur in long term. Such changes can lead to non-stationary patterns in solar irradiance time series, such as those detected by Yang et al. 2013 in Singapore using the Augmented Dickey-Fuller test. As discussed in [Pedro and Coimbra, 2012], it is due to the stochastic process that describes the sky conditions.

Spatial Correlation

Similar patterns of irregularities observed at neighbor stations in close time lags can indicate a cloud movement. Consequently, it can suggest that some impacts on energy supply observed at some station are expected to happen at neighbor stations in a near future. The identification and understanding of such behaviors are a key point for the modeling of a spatio-temporal forecasting system in solar energy. In this section, spatial correlation between PV stations is analyzed through their solar irradiance time series. The region studied is the solar energy network NREL Oahu, whose map is depicted in Figure 8. The red arrow indicates the wind direction at the region.

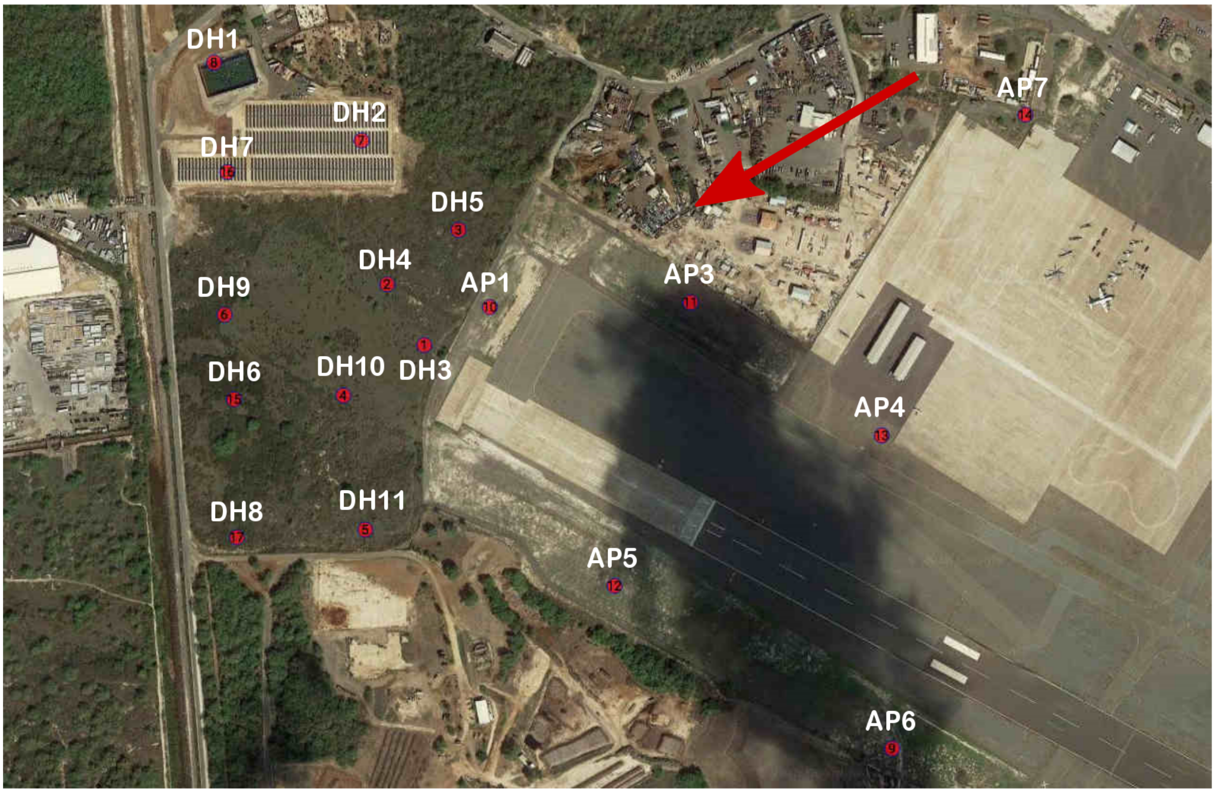
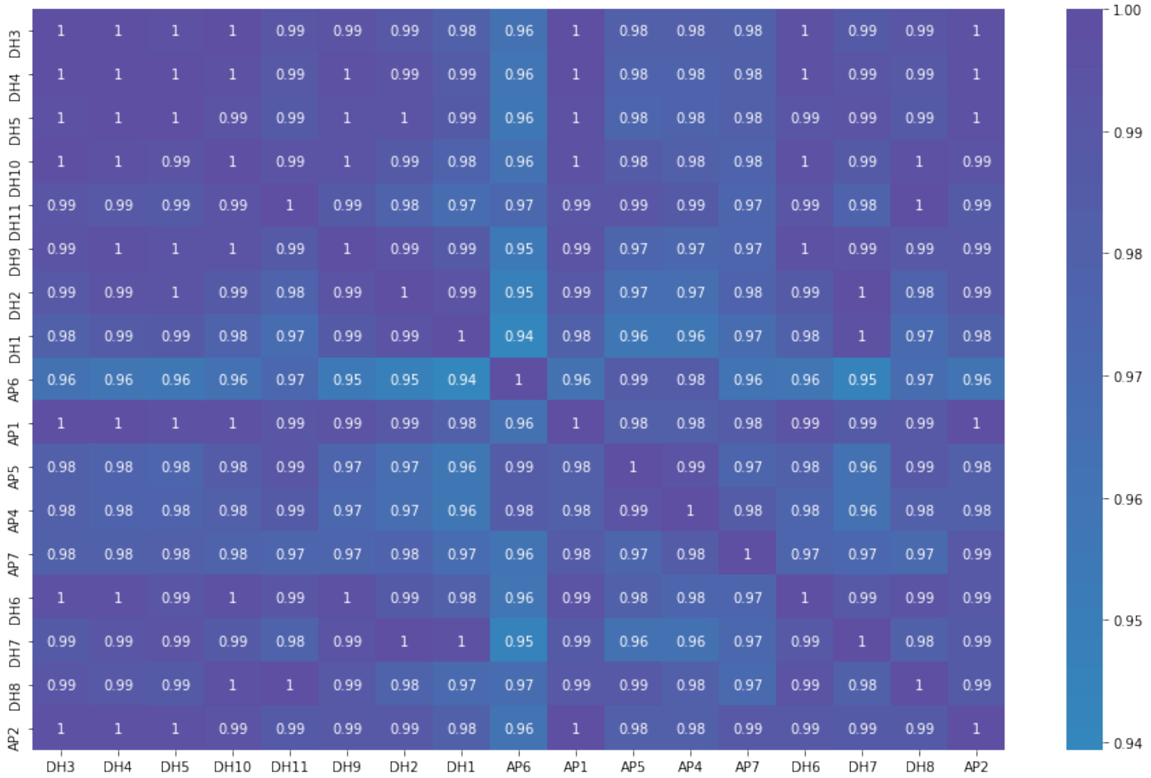


Figure 8 – Map of 17 stations of a PV Network at Oahu. Source: NREL [2018].

There is, however, some challenges for the accurate identification of correlation between stations. For instance, in Figure 7 it can be noticed that, despite the shape differences caused by different cloudiness indexes registered in each of the days, there is still a common trend in both time series, caused by the daily solar cycle. This can result in a false perception of correlation between the time series. Figure 9 presents the cross correlations between time series of solar irradiance measured at the Oahu Network stations. The results suggest highly correlated time series, ranging from .97 to 1.0. However, under a scenario where the stations are dispersed at different distances and subject to various cloud movements, such correlation values would be unlikely. Therefore, it is necessary to remove this trend, leaving only the component associated with the cloudiness, so that the correlation between the stations can be identified in a more adequate way.

There are several approaches in the literature for trend removal in time series of solar energy. The most commonly used is the application of a clear sky model [Inman et al., 2013]. In spatio-temporal models, different clear sky approaches are used [Dambreville et al., 2014], [Aryaputera et al., 2015], [Yang et al., 2014]. Basically they differ by the number of parameters considered, which also influences their accuracy. This kind of model normally uses information from astronomical formulae, combined to fitted parameters. This dependency of such parameters can be a problem in some occasions. For example, when

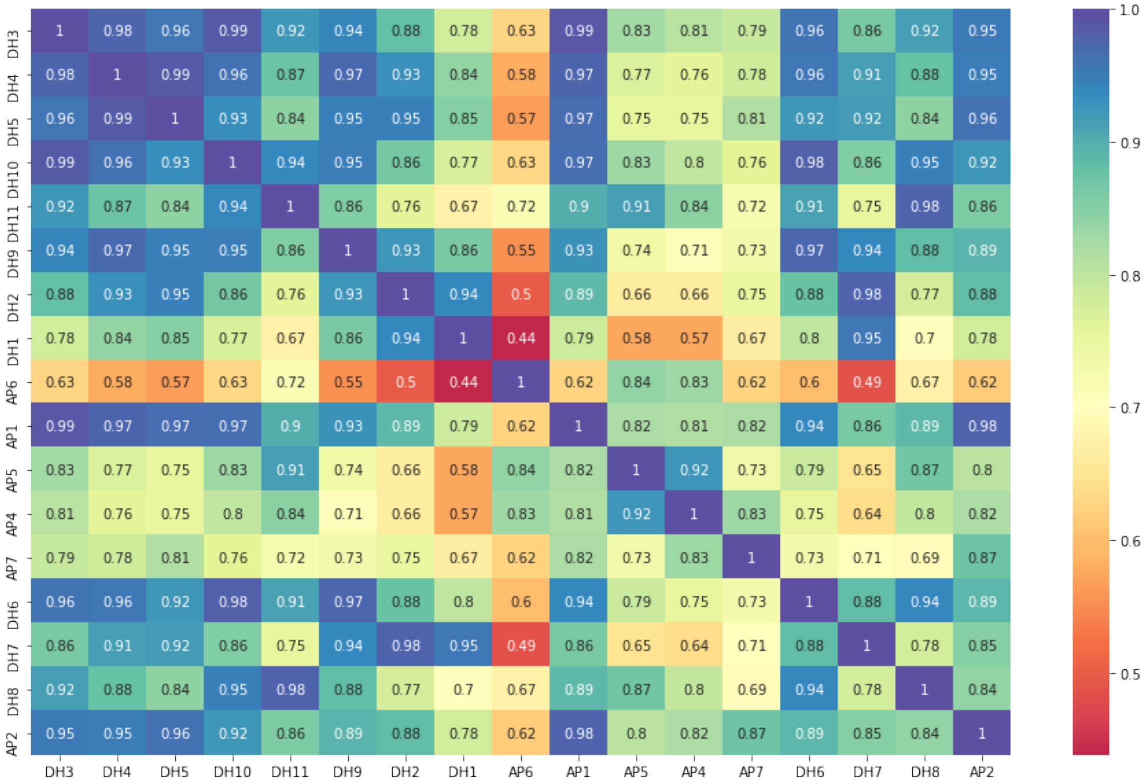


	DH3	DH4	DH5	DH10	DH11	DH9	DH2	DH1	AP6	AP1	AP5	AP4	AP7	DH6	DH7	DH8	AP2
DH3	1	1	1	1	0.99	0.99	0.99	0.98	0.96	1	0.98	0.98	0.98	1	0.99	0.99	1
DH4	1	1	1	1	0.99	1	0.99	0.99	0.96	1	0.98	0.98	0.98	1	0.99	0.99	1
DH5	1	1	1	0.99	0.99	1	1	0.99	0.96	1	0.98	0.98	0.98	0.99	0.99	0.99	1
DH10	1	1	0.99	1	0.99	1	0.99	0.98	0.96	1	0.98	0.98	0.98	1	0.99	1	0.99
DH11	0.99	0.99	0.99	0.99	1	0.99	0.98	0.97	0.97	0.99	0.99	0.99	0.97	0.99	0.98	1	0.99
DH9	0.99	1	1	1	0.99	1	0.99	0.99	0.95	0.99	0.97	0.97	0.97	1	0.99	0.99	0.99
DH2	0.99	0.99	1	0.99	0.98	0.99	1	0.99	0.95	0.99	0.97	0.97	0.98	0.99	1	0.98	0.99
DH1	0.98	0.99	0.99	0.98	0.97	0.99	0.99	1	0.94	0.98	0.96	0.96	0.97	0.98	1	0.97	0.98
AP6	0.96	0.96	0.96	0.96	0.97	0.95	0.95	0.94	1	0.96	0.99	0.98	0.96	0.96	0.95	0.97	0.96
AP1	1	1	1	1	0.99	0.99	0.99	0.98	0.96	1	0.98	0.98	0.98	0.99	0.99	0.99	1
AP5	0.98	0.98	0.98	0.98	0.99	0.97	0.97	0.96	0.99	0.98	1	0.99	0.97	0.98	0.96	0.99	0.98
AP4	0.98	0.98	0.98	0.98	0.99	0.97	0.97	0.96	0.98	0.98	0.99	1	0.98	0.98	0.96	0.98	0.98
AP7	0.98	0.98	0.98	0.98	0.97	0.97	0.98	0.97	0.96	0.98	0.97	0.98	1	0.97	0.97	0.97	0.99
DH6	1	1	0.99	1	0.99	1	0.99	0.98	0.96	0.99	0.98	0.98	0.97	1	0.99	0.99	0.99
DH7	0.99	0.99	0.99	0.99	0.98	0.99	1	1	0.95	0.99	0.96	0.96	0.97	0.99	1	0.98	0.99
DH8	0.99	0.99	0.99	1	1	0.99	0.98	0.97	0.97	0.99	0.99	0.98	0.97	0.99	0.98	1	0.99
AP2	1	1	1	0.99	0.99	0.99	0.99	0.98	0.96	1	0.98	0.98	0.99	0.99	0.99	0.99	1

Figure 9 – Correlation Map between Oahu Network Stations. [1]

not all the required parameters are available. Also, some models need to be readjusted according to the geographic region where the system is located. To address such problems, some data driven approaches are proposed. In Pedro and Coimbra [2012] an approximated clear sky model was calculated by using a smooth surface over historical irradiance time series of an entire year. Boland [2015] uses Fourier transform to identify cycles of seasonality and decompose the original time series into a component represented by a Fourier time series model and residual time series model. In this thesis, Singular Spectrum Analysis (SSA) is applied to decompose the original data into a main and a residual part, so the last one could capture most of the information present in cloud intervals. The method is explained in Appendix A.

After the trend removal with SSA decomposition the residual data is used to analyze spatio-temporal correlation in the data set. Figure 10 shows the correlation map between Oahu stations using residual data. In comparison to Figure 9, it can be noticed that the mean correlation has decreased, ranging from .5 to 1.0. Moreover, the correlations have a more direct relationship with the geographic location of the stations. For example, station *DH1* is more correlated to its neighbor *DH2* than to a more distant station, such as *AP7*. Such information can be useful to determine which slice of data is more beneficial to the forecasting model and moreover, reduces dimensionality.



	DH3	DH4	DH5	DH10	DH11	DH9	DH2	DH1	AP6	AP1	AP5	AP4	AP7	DH6	DH7	DH8	AP2
DH3	1	0.98	0.96	0.99	0.92	0.94	0.88	0.78	0.63	0.99	0.83	0.81	0.79	0.96	0.86	0.92	0.95
DH4	0.98	1	0.99	0.96	0.87	0.97	0.93	0.84	0.58	0.97	0.77	0.76	0.78	0.96	0.91	0.88	0.95
DH5	0.96	0.99	1	0.93	0.84	0.95	0.95	0.85	0.57	0.97	0.75	0.75	0.81	0.92	0.92	0.84	0.96
DH10	0.99	0.96	0.93	1	0.94	0.95	0.86	0.77	0.63	0.97	0.83	0.8	0.76	0.98	0.86	0.95	0.92
DH11	0.92	0.87	0.84	0.94	1	0.86	0.76	0.67	0.72	0.9	0.91	0.84	0.72	0.91	0.75	0.98	0.86
DH9	0.94	0.97	0.95	0.95	0.86	1	0.93	0.86	0.55	0.93	0.74	0.71	0.73	0.97	0.94	0.88	0.89
DH2	0.88	0.93	0.95	0.86	0.76	0.93	1	0.94	0.5	0.89	0.66	0.66	0.75	0.88	0.98	0.77	0.88
DH1	0.78	0.84	0.85	0.77	0.67	0.86	0.94	1	0.44	0.79	0.58	0.57	0.67	0.8	0.95	0.7	0.78
AP6	0.63	0.58	0.57	0.63	0.72	0.55	0.5	0.44	1	0.62	0.84	0.83	0.62	0.6	0.49	0.67	0.62
AP1	0.99	0.97	0.97	0.97	0.9	0.93	0.89	0.79	0.62	1	0.82	0.81	0.82	0.94	0.86	0.89	0.98
AP5	0.83	0.77	0.75	0.83	0.91	0.74	0.66	0.58	0.84	0.82	1	0.92	0.73	0.79	0.65	0.87	0.8
AP4	0.81	0.76	0.75	0.8	0.84	0.71	0.66	0.57	0.83	0.81	0.92	1	0.83	0.75	0.64	0.8	0.82
AP7	0.79	0.78	0.81	0.76	0.72	0.73	0.75	0.67	0.62	0.82	0.73	0.83	1	0.73	0.71	0.69	0.87
DH6	0.96	0.96	0.92	0.98	0.91	0.97	0.88	0.8	0.6	0.94	0.79	0.75	0.73	1	0.88	0.94	0.89
DH7	0.86	0.91	0.92	0.86	0.75	0.94	0.98	0.95	0.49	0.86	0.65	0.64	0.71	0.88	1	0.78	0.85
DH8	0.92	0.88	0.84	0.95	0.98	0.88	0.77	0.7	0.67	0.89	0.87	0.8	0.69	0.94	0.78	1	0.84
AP2	0.95	0.95	0.96	0.92	0.86	0.89	0.88	0.78	0.62	0.98	0.8	0.82	0.87	0.89	0.85	0.84	1

Figure 10 – Correlation Map between Oahu Network Stations using SSA residual data.

2.2 Wind Energy Data

Similarly to solar irradiance time series, series related to wind energy are also subject to climate conditions. However, they present different patterns. A first characteristic to be observed is that they are not directly correlated to the diurnal cycle of the sun, which means that a detrending process as discussed in section 2.1 would not be required to evidence spatial cross correlation. The data set analyzed in this section was previously provided for the Global Energy Forecasting Competition 2012 (GEFCom2012). Aspects of the problems presented in the competition such as details about the data, best forecasting approaches and discussion about the results are presented in [Hong et al., 2014]. For the wind energy forecasting problem, hourly wind power generation time series of seven wind farms from the same region of the world are provided. Figure 11 depicts a three day span of wind power values observed at one of the wind farms, denoted as Wind Farm 1 (WP1).

Variability in Wind Time Series

As discussed in Kariniotakis [2017], different time scales are more suitable for some very short forecasting problems related to wind energy. For example, if the focus is the control of wind turbines, time scales in seconds tend to be more relevant. From 10 minute to 1 hour, the focus of the forecasting is more associated to the power system into which the turbines are integrated, and is commonly used to assist economic dispatch. However,

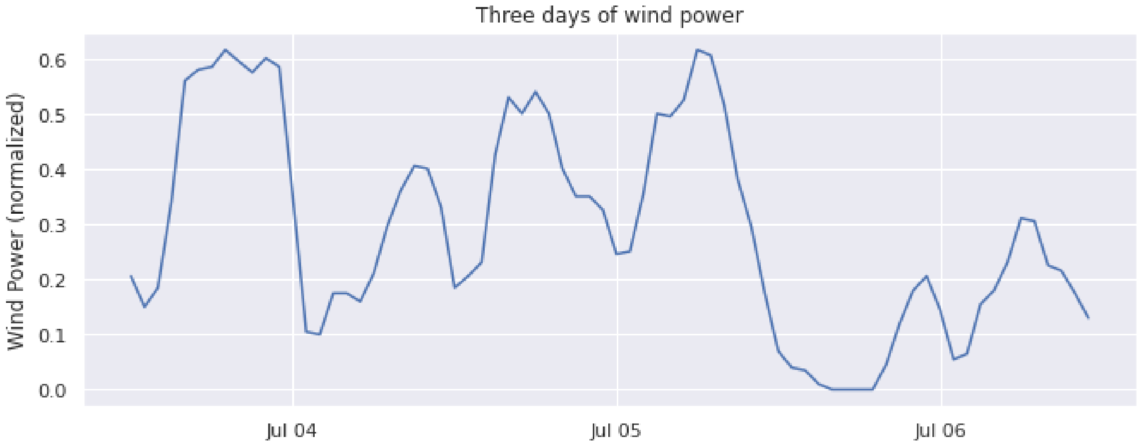


Figure 11 – Wind Power values observed at Wind Farm 1 from GEFCom2012 data set

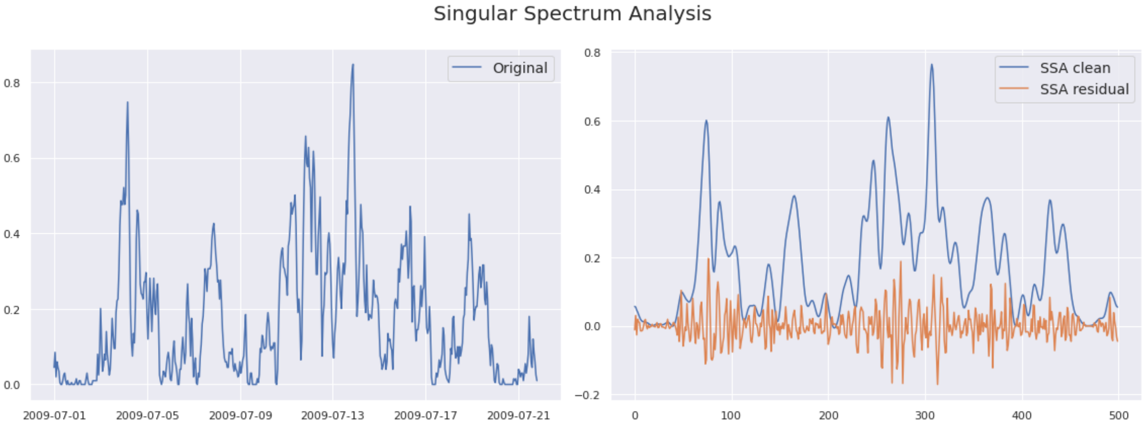


Figure 12 – SSA decomposition applied to Wind Power time series at Wind Farm 1 from GEFCom2012 data set

as pointed by Vincent et al. 2010, in wind energy it is difficult to define any time scale on which the data could be considered stationary. Many physical events are responsible for such variability in wind. Some of them are mentioned by Vincent et al. 2010, such as inertial oscillations, low-level jets, and gravity waves. In this context, the application of techniques to decompose the data, such as transform methods, can be very useful for the analysis of wind energy time series, in order to detect fluctuations in data which eventually are covered by other irregular components. In this thesis, such decomposition is performed by SSA, in a similar way as it is applied in Moreno and dos Santos Coelho [2018], where wind speed time series are decomposed into two components and applied to an ANFIS based model for very short term forecasting. However, in this work this decomposition is used only for data analysis, not integrated with the proposed new models. Figure 12 shows an example of SSA decomposition applied to the time series in Wind Farm 1 presented in Figure 11.

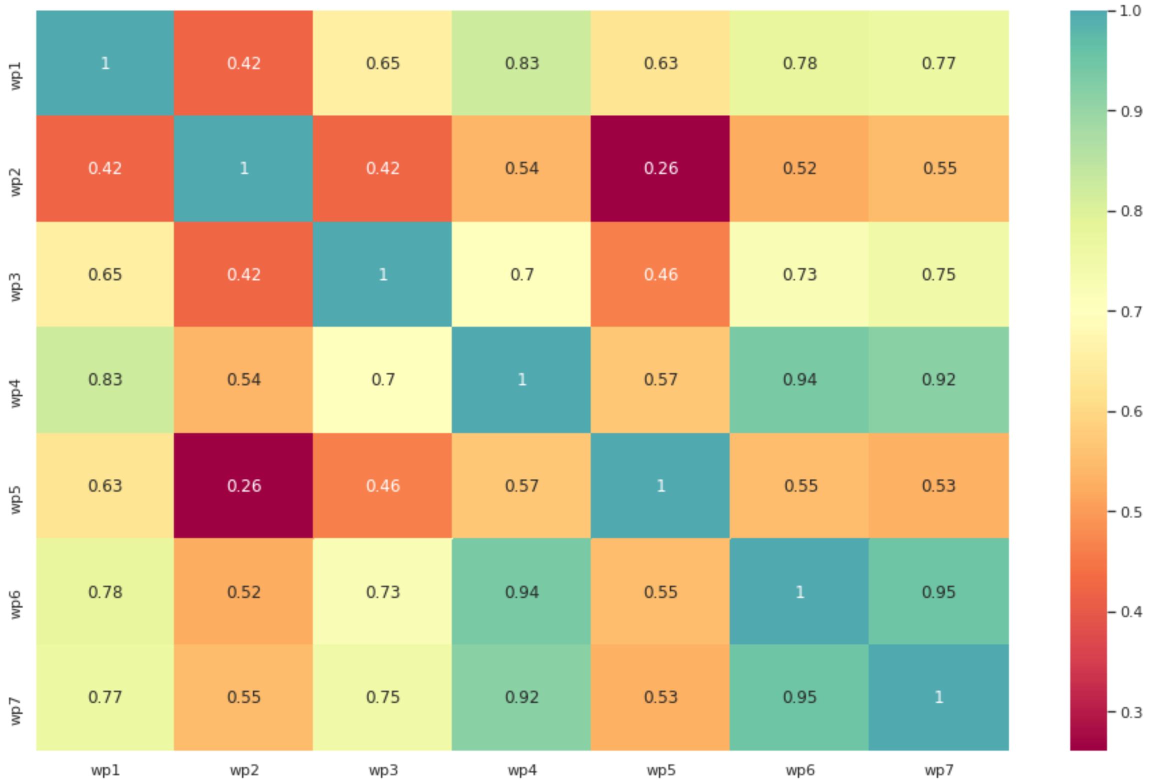


Figure 13 – Wind Speed values observed at a turbine of a wind farm - clean SSA component.[2]

Spatial Correlation

SSA decomposition can isolate the residuals, which helps in the detection of spatial correlations. Figure 13 illustrates the correlation calculated from the clean SSA component from 1 year of hourly data. For this data set, it is possible to identify relevant correlations between wind farms. But in addition, it is also possible to establish a threshold to define which wind farms can contribute to a model, if the focus of forecasting is a specific wind farm.

2.3 Discussion

In this section, time series related to the generation of solar and wind energy were analyzed. For this task, two widely studied data sets were chosen. Although the problems of solar and wind forecasting have different particular characteristics, resulting especially from the type of pattern commonly observed in their time series, there are some major aspects in common. The first is the susceptibility of both to different climatic phenomena, which results in variations in their patterns that lead to non-stationary time series. This suggests the need to use forecasting models with a good ability to adapt to changes. Another point illustrated in the examples is the organization of such systems in multiple integrated stations, arranged in a region with some proximity to each other. When analyzing the time series of all stations together, it is possible to identify correlations between them, which suggests that they contain information relevant to forecasting. Therefore, there is a motivation to investigate models that can cover all of this information and use it with the aim of improving the energy prediction in the system. In this work, the models studied are guided by the search for solutions to these issues, so that the final result can have a better performance.

Chapter 3

Literature Review

The diverse approaches in forecasting for renewable energy present in the literature focus on three main groups: physical modeling, statistical and probabilistic models and intelligent models. There are also hybrid models that make integrated use of some of the techniques of different groups [Ahmed and Khalid, 2019]. The main characteristics of these groups associated with their application to the problem under study are discussed below.

Physical models

Physical modeling is based on numeric weather prediction (NWP), which uses mathematical models to represent the functioning of the atmosphere and the oceans in order to find values that can predict the climate values and, consequently, assist in renewables forecasting. These models have been showing consistent advances, mainly due to improved numerical schemes to solve governing questions, as discussed in [Sweeney et al., 2020]. Other improvement is the fine-grained parameterization schemes, which allow to regionalize their models more appropriately. Such advances appear in line with greater data availability and computational power. Bauer et al. [2015] describe this movement as the quiet revolution of NWP. However, even though more machinery is available, many NWP models are still computationally expensive when dealing with larger databases, to the point of not being a good choice in a very short-term forecasting context [Sweeney et al., 2020]. In this respect, statistical or machine learning models have an advantage as they can only perform forecasting based on recent data, without the need to solve complex mathematical equations at each step.

Statistical models

The basis of many spatio-temporal models are formed by correlations between different variables observed in the problem under study, which makes the application of statistical models a straightforward alternative. Such correlations support the work of Gafurov et al. [2015], in which the spatial correlations between the distributed data sources helped in the forecasting of solar irradiance data on sites whose data were not fully available. Correlations were combined using linear algebra to generate indicators to be applied in regression models. Using this formulation, the model was able to solve the drawback of using synthetic data to replace meteorological variables that were unavailable. In this sense, the model proposed by Aryaputera et al. [2015] applied spatio-temporal kriging to forecast solar irradiance in locations whose historical data was not observed. In both models cited, it is possible to identify a gain in the incorporation of spatio-temporal models in order to improve the quality of analysis of regions where data may be unavailable or unreliable. Such improved robustness of the methods can also prospect opportunities for expansion of the energy system. Another aspect derived from the use of statistical models is the delimitation of regions whose values are correlated and, therefore, relevant to be included as input in the models. For instance, in [Yang et al., 2014], models created from the integration between kriging and autoregressive (AR) methods defined a threshold distance, which specified the maximum separation between regions whose values could be correlated. AR methods are also useful in spatio-temporal problems for the representation of the endogenous and exogenous variables that compound the model. In [Agoua et al., 2017], an AR method outputs photovoltaic power forecasts for a power plant using observations of PV power and a cluster of meteorological variables, such as temperature and wind speed from neighbor plants. To solve the dimensionality problem, the LASSO method is used. There is also a statiotionarity procedure, where the original time series is transformed using an equation similar to the clear sky index, widely used in solar energy time series analysis. The experiments denoted significant improvements with all these techniques combined. Fan et al. 2015 proposed a wind speed forecasting method which combines Vector Autoregressive (VAR) with Kriging. But also in this case, the methods require a data pre-processing step that can provide at least a weak stationarity. A de-trending technique based on fourier series was then applied separately to each time series component of the spatio-temporal model.

In general, statistical models are easier to implement and can solve the problem of computational complexity that affects the physical models. However, there are disadvantages in the ability to represent more complex data distributions, often requiring a preprocessing step so the data can be better understood. Stationarization procedures, standardization, signal decomposition and independent component analysis are examples of preprocessing activities commonly used in statistical models.

Intelligent models

An increase in the ability of dealing with more complex, nonlinear data used in forecasting models for renewable energy was intensified through the application of machine learning techniques. Some literature reviews in solar forecasting such as [Inman et al., 2013] and [Antonanzas et al., 2016] as well as wind forecasting reviews as the work of Jung and Broadwater 2014, suggest a general trend in the application of intelligent models, as discussed in [Ahmed and Khalid, 2019]. Intelligent models use machine learning methods, with Artificial Neural Networks (ANN) being the most prevalent technique in the literature. However, when applying ANN to spatio-temporal problems, it is necessary to observe a balance between complexity and performance. The complexity in ANN is based on its attributes, such as the number of layers and neurons, that define its generalization capacity. More complex models tend to demand more computational effort in training and hyperparameter tuning. To solve this problem, some works propose applying dimensionality reduction methods. In [Lan et al., 2019] and [Licciardi et al., 2015], the dimensionality reduction is performed with the application of principal component analysis (PCA). In both papers, the reduction is applied to spatio-temporal data for solar energy forecasting.

Deep Learning Methods

The application of more complex models of ANN was again considered with the advent of deep learning models [Goodfellow et al., 2016]. The ability to deal with a greater volume of information organized in other formats, such as images and videos, opened new perspectives for the problem. In [Zhang et al., 2018], historical photovoltaic power values and sky images are taken as input to estimate photovoltaic power in a very short term future, using a model based on Convolutional Neural Networks (CNN). Sadaei et al. 2019 proposed an image representation that is used as an input to a load forecasting model which combines CNN and fuzzy time series. In [Yu et al., 2019] intervals of wind power time series are extracted from a wind turbine map, and serve to create multi-channel images training a CNN. Recurrent Neural Networks (RNN), widely used in speech recognition, handwriting recognition and time series forecasting, have been also evaluated in renewable energy forecasting. Ghaderi et al. 2017 proposed a spatio-temporal wind speed forecasting algorithm using Long Short Term Memory (LSTM), a widely used RNN architecture, obtaining forecasts from all the spatial locations simultaneously. In [Khodayar and Wang, 2018], wind speed forecasting is performed by extracting temporal information from each wind site with a LSTM. The wind farm composed by the sites is then represented by a graph convolutional deep learning architecture (GCDLA), responsible to model the spatial information of the system. Although deep learning models have shown promising results, there are still some challenges for their full adoption in renewable energy forecasting. One is the computational complexity of such models, which often requires more powerful equipment. This can be a major problem in time series such as solar irradiance or wind speed, which can vary in their patterns over time. Consecutive adjustments, such as retraining, to a complex model can make it difficult to run in a context of very short term forecasting. Such adjustments can be made at a lower cost using adaptive models. There are solutions proposing the use of evolutionary algorithms, as in [Sun et al., 2018] or neuroevolution, discussed in [Miikkulainen et al., 2019], but even in these scenarios there is still a considerable computational effort. This opens space for the research of adaptive models based on simpler machine learning techniques that can bring comparable results in a shorter time.

Fuzzy based Methods

Fuzzy based models have been extensively used in renewable energy forecasting, since they present some attributes that can be suitable to this problem. It presents a many valued logic, capable of reasoning from uncertainties. Even in simpler models that implement fuzzy logic fundamentals, such characteristics are present, thus providing a powerful representation mechanism. This can be an advantage if compared to neural networks, in the sense of resolving the trade-off between complexity and performance. As discussed in [Suganthi et al., 2015], the use of fuzzy based methods integrated to statistical or machine learning methods yielded higher accuracy to solar and wind forecasting problems. The review also suggests that neuro fuzzy models, especially the Adaptive neuro fuzzy inference system (ANFIS), are the most frequently used approach. An application of ANFIS can be seen in the model proposed by Moreno and dos Santos Coelho 2018. A pre-processing step using singular spectrum analysis is applied to decompose raw wind speed data and the output is sent to the ANFIS model to perform forecasting. The results suggest that such pre-processing step has a positive impact on the model performance. This can be seen more clearly when compared to the results of the same method without SSA pre-processing.

The term "adaptive" in ANFIS refers to preliminary training procedures, responsible for modeling its architecture and parameters from patterns extracted from the training data set. However, it does not include incremental adjustments in structure and parameters demanded by changes over time in the distribution parameters of the data under analysis, providing an evolving capacity to the model. This is a desired feature to deal more properly with some time series of renewable energy systems. As an example of evolving fuzzy based model, in [Soares et al., 2018] an evolving model that uses gaussian fuzzy sets is proposed to forecast hurricane tracks. When compared to ensembles of classical evolving fuzzy methods, such as evolving Takagi-Sugeno (eTS) method [Angelov and Zhou, 2006] and TEDA [Angelov, 2014b], better results were presented. Other fuzzy-based evolving algorithm is presented in [Samanta et al., 2019], where a Self-adaptive spatio-temporal neuro-fuzzy inference system (SPATFIS) is presented. The model includes memory type neurons in its structure, in order to better incorporate spatio-temporal information. It is applied to different spatio-temporal forecasting problems, including wind speed forecasting. In this case, the model was faster and presented better accuracy than other fuzzy inference systems and eTS.

Fuzzy Time Series

The application of fuzzy time series (FTS) models to forecasting problems in renewables is still little explored, but with good prospects for solutions due to their structural flexibility that allows different ways of integrating and organizing architectures, generating hybrid solutions with good performance. Jiang et al. 2019 proposed an example of a hybrid solution using FTS for wind speed forecasting. In it, an FTS method is integrated with a multi-objective optimization algorithm, responsible for adjusting the parameters that define its fuzzy sets and weights. Such integration opens the opportunity to develop an FTS model whose parameters can be adjusted over time using algorithms coupled to the model. The research of FTS models in this thesis was reinforced by promising results previously obtained in [Severiano Junior et al., 2017] with the application of high order FTS for solar irradiance forecasting. FTS models yielded comparable results to statistical and intelligent models commonly used in forecasting problems, such as ARIMA and ANN. Another aspect that encourages the use of FTS is the interpretability associated with such models. Garibaldi [2019] points the need to invest in explainable AI models so that they can be better adopted in different applications and how fuzzy-based models can play a central role in this are due to their nature of abstracting uncertainty components of the problem.

In comparison with other widely used fuzzy based models, such as ANFIS, FTS has an advantage in interpretability because it provides a simpler and more intuitive representation format, based on decision rules. According to Molnar [2019], methods for machine learning interpretability can be classified under different criteria. Based on the way the interpretability is obtained, it can be divided into **intrinsic** or ***post-hoc***. The first is obtained from the use of intrinsically interpretable models. FTS models could be included in this group because of their models based on decision rules usually associated with categorical values, the fuzzy sets. *Post-hoc* refers to the application of interpretation methods after model training. For example, the use of surrogate models. It occurs when an interpretable model (surrogate) is trained from the set of inputs and outputs generated by a more complex model (black box) in order to clarify its decision process. For instance, Ribeiro et al. 2016 proposed a surrogate technique named Local interpretable model-agnostic explanations (LIME) to explain specific predictions of a black box model from modifications in samples originally used for training and observation of the respective outputs. This new modified data set is then applied to an interpretable model, so that the features that influenced the decision can be highlighted. If, on the one hand, *post-hoc* techniques can reconcile interpretability with good precision, since the chosen black box model can have different levels of complexity, on the other hand, there is a higher cost in explaining the model, which probably would need a process of greater computational effort, as it needs to be redone with each new training procedure. There is, therefore, another factor considered when choosing an adaptive FTS model to assist in the problem in question: its interpretability tends to be less costly because it is intrinsic to the model and less impacted by new training steps. In the next chapters the concepts of FTS and some extensions are discussed, as well as proposals to apply such models to spatio-temporal forecasting using adaptability mechanisms.

Chapter 4

Fuzzy Time Series

Fuzzy Time Series can be seen as a way to represent time series from the perspective of fuzzy logic. It basically means that conventional time series values, usually represented by real numbers (or crisp values), are translated to fuzzy logic using fuzzy sets as fundamental components. In this chapter, fundamentals of fuzzy time series are presented as well as the basic procedures for training a model and perform forecasting. Such fundamentals refer to the concepts first proposed by Song and Chissom 1993 and improvements presented in Chen et al. [1996]. Next, some extensions and improvements in FTS are presented.

4.1 Fuzzy Time Series Fundamentals

Given a univariate time series Y with elements $y(t) \in \mathbb{R}^1$, where $t = 0, 1, \ldots, T$ is a time index. The **Universe of Discourse** U of Y is the range of values that the time series can assume, or $U = [min(Y], max(Y)$. In this context, the role of **fuzzy sets** is to divide U into partitions so minor differences and uncertainties in crisp values of Y can be more easily abstracted and explained. Such translation from crisp to fuzzy is intermediated by **membership functions**, each one associated to one fuzzy set. A membership function $\mu(x)$ indicates, within a range of $[0, 1]$, how much a value x belongs to a fuzzy set. For example, given two fuzzy sets, A and B, and their respective membership functions, $\mu_A(x)$ and $\mu_B(x)$. If, for a crisp value x, $\mu_A(x) = 0.7$ and $\mu_B(x) = 0.2$, these results indicate that x belongs to both A and B, although it has a higher membership grade in A. It highlights an important tool in fuzzy logic, where the division of an universe of discourse is done with components with flexible boundaries that can even overlap and thus provide more embracing representations of crisp values. Given these concepts, basic training and forecasting procedures in FTS are described as follows.

Training Procedure

Given a training set Y_t, defined from a subsequence of Y, the following steps are taken to train an FTS model:

Step 1 - Universe of Discourse Partitioning

In this step the U is divided into sub intervals according to a partition scheme. Consider k sub intervals of U, such as $U = u_1, u_2, \cdots, u_k$. In this case, the computational cost for the task is $O(k)$.

Step 2 - Fuzzy Sets Definition

Given the partitioning, corresponding fuzzy sets A_i, where $i = 1, 2, \cdots, k$, are then defined over each sub interval with membership functions $\mu_{A_i} : u_i \longmapsto [0, 1]$. All the defined fuzzy sets compose a linguistic variable $\tilde{A}$. The step also has a linear computational cost of $O(k)$.

Step 3 - Time Series Fuzzification

The time series Y is converted to an FTS F, or the crisp numbers $y(t)$ are translated to fuzzy values $f(t)$, given their membership to the fuzzy sets. In Chen et al. [1996], although a numeric value may have nonzero membership values for multiple fuzzy sets, only the one with maximum membership is selected to represent the fuzzified value of the crisp sample. It means that each value in $f(t)$ corresponds to a fuzzy set A_i. The fuzzification process is illustrated in Figure 14. For a set of samples $y(t)$, with $t = 1, \ldots, T$, the computational cost is $O(T \cdot k)$.

Step 4 - Temporal Patterns Extraction

Temporal patterns are extracted from F according to the number of past observations (lags) that are considered in the model to perform forecasting. The hyperparameter order Ω defines the number of lags in a model. Thus, for a first order model in FTS ($\Omega = 1$), a temporal pattern has the format $P \to C$, where the precedent P is represented by the fuzzy value at $f(t)$ and the consequent C contains the fuzzy value at time $f(t+1)$. Each pattern represent a fuzzy rule, named **Fuzzy Logical Relationship (FLR)**. The computational cost of this step is $O(T \cdot k^{\Omega})$.

Step 5 - Rule Base Creation

From the information extracted, temporal patterns are grouped by their precedents. Each group is a **Fuzzy Logical Relationship Group (FLRG)** and they compose the **rule base**. The rule base is the final representation of an FTS model. The trained model

$\mathcal{M}$ contains the information used to forecast values in FTS, given as input a precedent value.

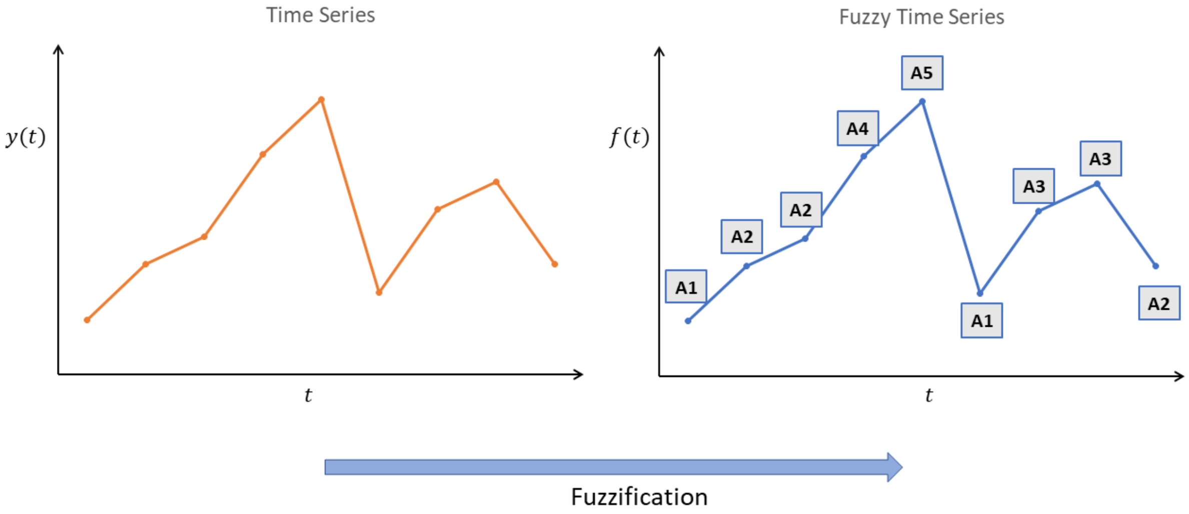


Figure 14 – Fuzzification process diagram

Forecasting Procedure

Given the first order FTS model M, forecasting can be performed by using as input a value from Y at time t to estimate a future value at $t+1$. The following steps describe the forecasting procedure.

Step 1 - Input Value Fuzzification

The input value fuzzification is also related to the membership functions defined in M. The membership value of $y(t)$ is calculated for all the fuzzy sets in $\tilde{A}$. The input value is then associated to the fuzzy set from which the maximum membership was obtained, representing the fuzzy value $f(t)$. The for an order Ω and k fuzzy sets, the computational cost of this step is $O(\Omega \cdot k)$.

Step 2 - Fuzzy Rule Matching

$f(t)$ is used to search in the rule base for the corresponding FLRG. The FLRG whose precedent is equal to the input value is selected and the candidate fuzzy sets in its consequent are applied to estimate the forecast value. The computational cost of this step depends on the order Ω and the number of rules in M (k^Ω), then the complexity is $O(\Omega \cdot k^\Omega)$.

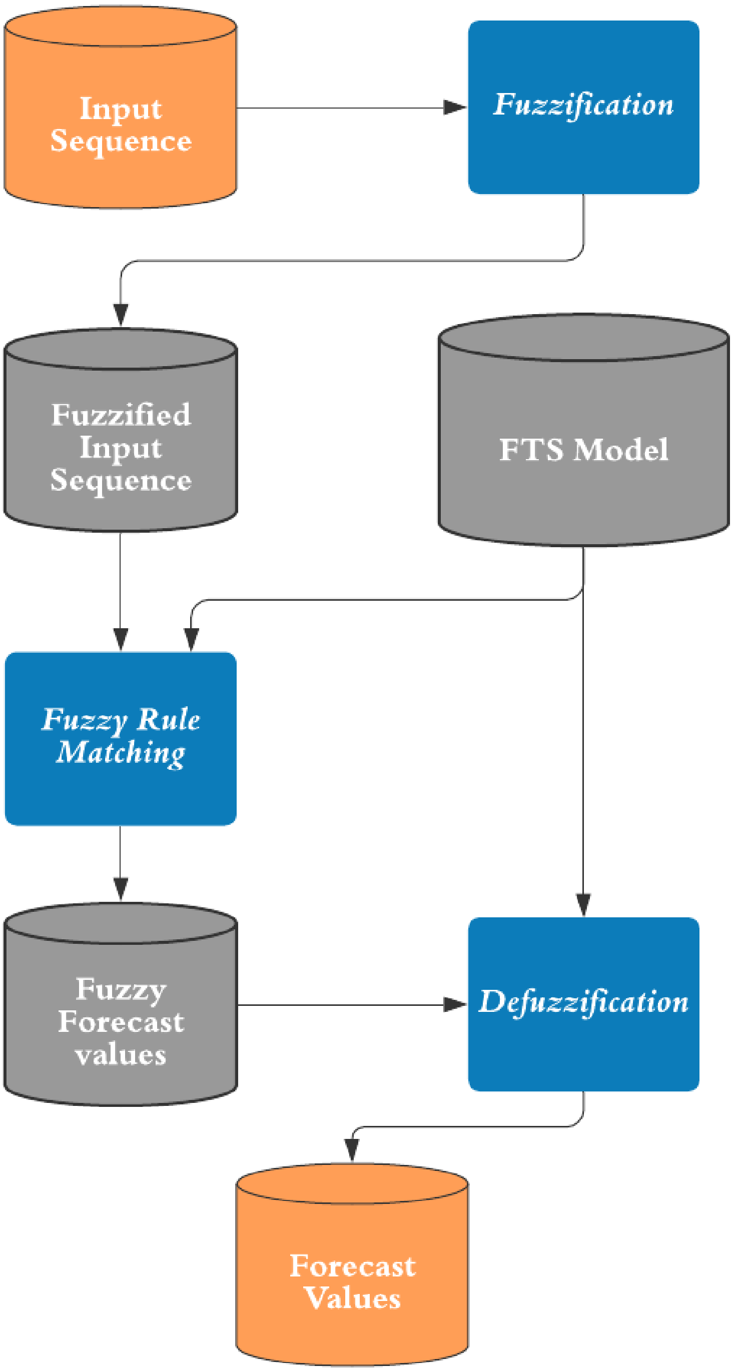


Figure 16 – FTS Forecasting process diagram

Step 3 - Defuzzification

Defuzzification is the conversion of the fuzzy representations to a crisp value. From the selected FLRG, the consequent is used to perform this step. A basic defuzzification procedure proposed by Chen et al. [1996] consists of obtaining the midpoints from all the fuzzy sets in the consequent and then calculate the mean value. Midpoints correspond to the center of the fuzzy sets or the value in the middle of the interval u_i related to a fuzzy set A_i.

Diagrams representing the training and forecasting procedures of FTS models are described in Figures 15 and 16, respectively. From the basic model here described, several expansions and improvements were proposed and some of greater interest for the problem under study in this thesis are highlighted in the next sections. A more comprehensive study on the different approaches in FTS as well as some directions of research are presented in e Silva [2019].

4.2 High Order FTS

For many real problems, including the time series analyzed in this thesis, there is relevant information for forecasting in multiple past observations, which indicates that a first order model, which considers only the first lag, may not be the most suitable in this context. One of the extension aspects of the FTS models is the High Order models, whose hyperparameter $\Omega = p$, where $p \in \mathbb{Z}$ and $p > 1$. Therefore, the FLRGs have a larger number of entries in their precedent, as described below:

$$f(t), f(t-1), f(t-2), \cdots, f(t-(p-1)) \rightarrow f(t+1) \tag{4.1}$$

In this way, consequents whose precedent is the same sequence are grouped into one FLRG. In addition to the hyperparameter of order Ω, high order models may have the flexibility to use past observations that are not the most recent, but with periodic delays that may be related to seasonality. For example, given a hourly time series, lags in the precedent can be the values observed at t and $t-24$, in order to use as information the most recent lag and the observation at the same time in the previous day. Such data modeling decisions are usually extracted from time series analysis techniques, such as the Autocorrelation Function (ACF).

4.3 Multivariate FTS

Multivariate FTS consists of using multiple Fuzzy Time Series to solve a problem, usually related to one of the time series analyzed. In Lee et al. [2006], a model with two

FTS (primary and secondary) is presented. In Jilani et al. [2007] the concept is expanded to multiple FTS. Given an d-variate time series $Y = (Y_1, Y_2, \cdots, Y_d)$, where each time series is represented by a corresponding FTS $F_1, F_2, \cdots, F_d$, composing a model of order p. The purpose of the model is to forecast values in Y_1. The Fuzzy Logical Relationships can be represented as:

$$\begin{aligned}(f_1(t-p), f_2(t-p), \ldots, f_d(t-p)), \ldots,\\ (f_1(t-1), f_2(t-1), \ldots, f_d(t-1)),\\ (f_1(t), f_2(t), \ldots, f_d(t)) \rightarrow f_1(t+1)\end{aligned} \tag{4.2}$$

Where each FTS has its own universe of discourse and a particular partitioning scheme. The FLRs are grouped under the same criterion seen in section 4.2 and the forecasting process takes a defuzzification activity for the consequent analogous to the approaches seen in classical FTS methods, especially the High Order model.

However, the use of multivariate FTS tends to generate multiple combinations of rules, resulting in very complex models and poor performance. First, due to the effort to train a high volume of rules being considerably high. Moreover, there is a tradeoff between the partitioning of the universe of discourse and the resulting rule combinations. As the number of combination increases, the set of rule combinations become so specific that the input data to a forecasting hardly match to any of these combinations. In order to solve some scalability problems, hybrid models are proposed in [Egrioglu et al., 2009] and [Egrioglu et al., 2013], where fuzzy c-means and ANNs are applied to reduce the amount of fuzzy rules. Also in [Chen and Chen, 2011] a clustering technique is used to improve the model. However, improvements in scalability may not prevent the high volume of rules from affecting interpretability. Therefore, an alternative would be to propose solutions for the representation of the information. The model presented in the next section proposes an alternative in this regard.

4.4 Fuzzy Information Granular FTS

$\mathcal{FIG}$-FTS model was proposed by e Silva et al. 2019 with the objective of transforming a multivariate time series into an univariate FTS data by using Fuzzy Information Granules ($\mathcal{FIG}$). Zadeh [1979] defined $\mathcal{FIG}$ as an entity that represents a subset of a wider domain. The representation of such entities may vary according to the problem it is applied. For example, in multivariate time series they are usually represented as multidimensional structures. In Singh and Dhiman [2018], a time series clustering algorithm is applied so each FIG is represented by a multidimensional cluster. In $\mathcal{FIG}$-FTS, a granule can be seen as a multivariate fuzzy set, where each variable is related to a time series. For example, consider an d-variate time series $Y = (Y_1, Y_2, \cdots, Y_d)$. For each time series Y_i, a corresponding linguistic variable $\mathcal{V}_i \in \mathcal{V}$ is defined. Each variable, in turn, is composed by fuzzy sets $A_j^{\mathcal{V}_i}$, created from the partitioning of the universe of discourse of Y_i. The combination of fuzzy sets from all the variables represents a granule $\mathcal{G}_l$, as follows:

$$\mathcal{G}_l = A_j^{\mathcal{V}_i} \quad \forall \mathcal{V}_i \in \mathcal{V} \tag{4.3}$$

The global linguistic variable $\mathcal{FIG}$ is the union of all granules $\mathcal{G}_l$ and defines the univariate $\mathcal{FIG}$-FTS model. It means that each granule G_l behaves as a fuzzy set in $\mathcal{FIG}$-FTS and its membership function is:

$$\mu_{\mathcal{G}_l} = \min \mu_{A_j^{\mathcal{V}_i}} \tag{4.4}$$

The equation describes that the membership corresponds to the minimum value among the fuzzy sets, according to the minimum T-norm, widely applied in fuzzy systems.

From this modeling, $\mathcal{FIG}$-FTS also starts to work in a similar way to a High Order FTS model as mentioned in section 4.2, with the difference that its fuzzy sets are granules which contain information from multiple time series. The defuzzification process is also analogous to that of a conventional FTS, in which each fuzzy set of the granule returns a value based on its midpoint. There is a difference when compared to the models discussed in section 4.3, in which multivariate time series were used to forecast values to a specific, and univariate, time series. In $\mathcal{FIG}$-FTS, all the time series that compound the model have their future values estimated. Another key point of the model is in the fuzzification process, which is guided by an threshold hyperparameter, named α-cut, whose range is of $[0, 1]$. This value defines a minimum membership threshold from which fuzzy sets $A_j^{\mathcal{V}_i}$ can be considered in the composition of a granule. It is, therefore, a hyperparameter that defines the granularity of the model, since lower values make the model more comprehensive and, consequently, more complex by increasing the number of rules. There is, therefore, a trade-off in this model related to granularity, but that can be reduced from the search of an optimal value to α-cut.

4.5 Discussion

In this chapter, some extensions in FTS present in the literature were highlighted. They reinforce the flexibility of the concept and tend to contribute to better results in time series forecasting for renewable energy. Firstly, the ability presented in High Order models to represent models that involve more than one past observations, in addition to the flexibility to involve lags of different seasonality. From the correlation analysis of time series related to renewable energy and used in this work, it is possible to notice the relevance of information in past observations to understand the model. Observations of time series from neighboring locations also tend to contribute to the forecasting model, being applied in a context of spatio-temporal prediction in this thesis. Such models can be represented from the Multivariate FTS concept. Therefore, there is a conceptual basis for the representation of spatio-temporal FTS models based on High Order and Multivariate methods. However, the high volume of data involved in this type of problem requires solutions that can balance aspects such as complexity of the model, accuracy of forecasts, computational performance and understanding of rules and decision making. $\mathcal{FIG}$-FTS presents an alternative that aims to deal with such problems using the concept of fuzzy information granules to simplify their representation, while maintaining a good ability to provide forecasts, as will be shown in experiments in this thesis. The model still presents trade-offs between complexity and results, but its central idea of translating a multivariate problem to univariate is an important contribution for FTS models that deal with such problem. A similar approach is proposed in a novel spatio-temporal model presented in this thesis.

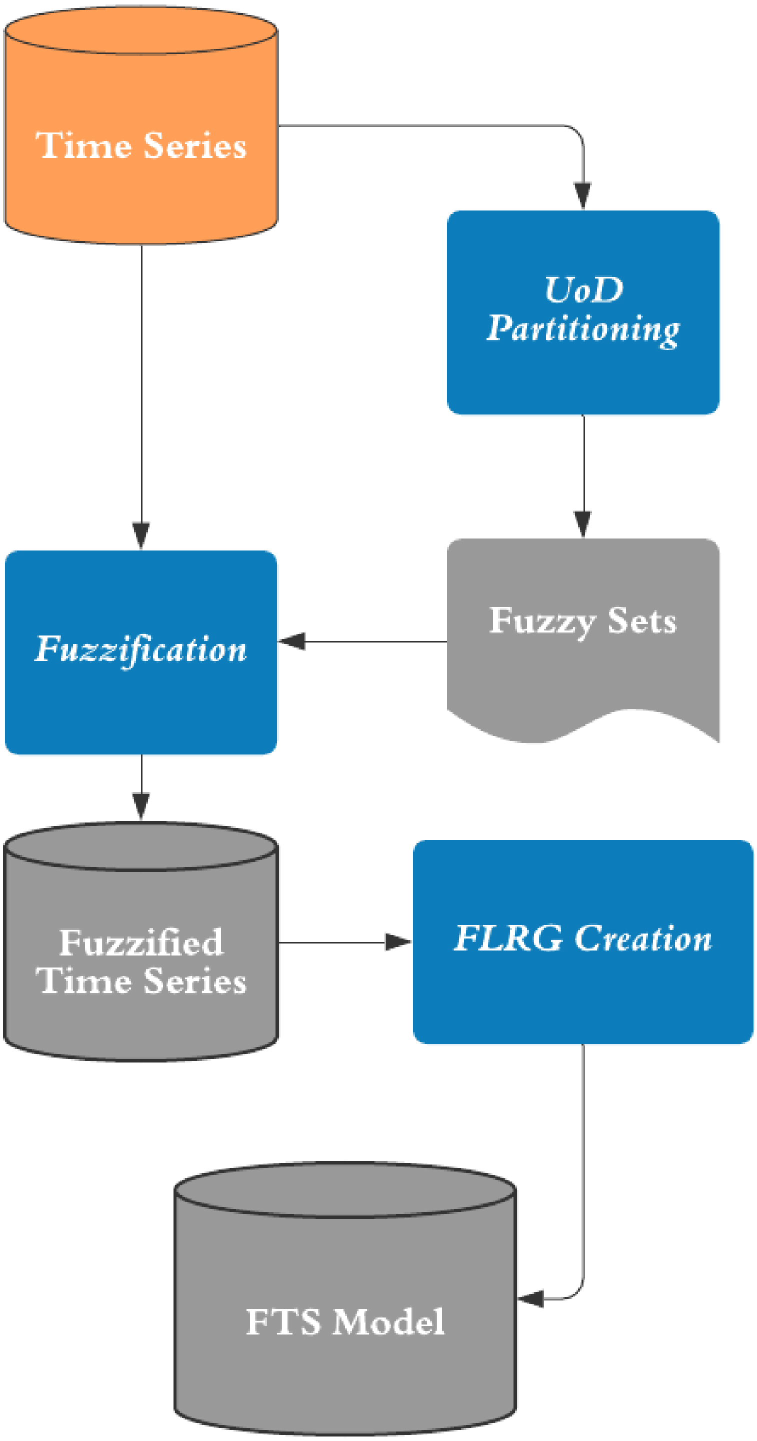


Figure 15 – FTS Training process diagram

Chapter 5

Non-stationary Fuzzy Time Series

The previous chapter highlighted some reasons why FTS models may be an interesting alternative to a spatio-temporal forecasting problem. However, as discussed in chapter 2, time series related to solar and wind energy tend to show non-stationary patterns. This can become a challenge to an FTS-based application, since a previously established model may not have the necessary scope to represent a problem over time. For example, a Universe of Discourse delimited from a training data set may not understand subsequent data that is outside its range. Furthermore, its fuzzy sets may no longer be suitable for the problem. Thus, for an FTS model to be able to have a good performance in non-stationary processes, it is desirable to incorporate adaptation mechanisms. More specifically, adjustments in their membership functions, fuzzy sets or Universe of Discourse.

The first efforts to provide adaptability to FTS were the use of time variant models. The approach was initially proposed by Song and Chissom 1993 and consists of performing retraining along sliding windows, so that the model is constantly rebuilt from scratch. A model with such mechanisms, when appropriately configured for the problem, is able to adapt to changes. Its implementation, however, has some disadvantages, such as the high computational cost and memory loss of the data processed in previous windows. In [Alves et al., 2018], an alternative was proposed to avoid the high cost of retraining. The model starts from another premise, in which its adaptation mechanisms do not demand a total reconstruction of the model, but incremental changes with less computational complexity. It is based fundamentally on the concepts of Non-stationary Fuzzy Sets (NSFS), presented by Garibaldi et al. [2008], which are applied in the context of FTS. The model is capable of forecasting in non-stationary series, but in very specific contexts, in which the variance of the data distribution changes in a predictable way. This limits its performance to more complex changes, such as concept drift events.

The Non-Stationary FTS (NSFTS) described in this chapter is a proposal for improving the adaptability of FTS models. It was originally presented in [e Silva et al., 2020], where its design and performance issues are further explored. In the following sections, its operation and the main concepts are described. Then, the model is evaluated in terms of its performance for time series of wind and solar energy, where non-stationary patterns are frequently present.

5.1 Non-stationary FTS Method

The NSFTS method is based on the concept of non-stationary fuzzy sets, which describes fuzzy sets that are adjustable over time. The mechanism responsible for the adjustments is the perturbation function, which change the parameters that define the membership function. Two parameters are central to this adjustment: **displacement** and **scale**. The first refers to the displacement of the FS along the Universe of Discourse. When applied alone, it only performs the displacement, without changing the shape of the fuzzy set. Scale, on the other hand, is responsible for changing the FS coverage area, either by stretching or contracting its shape.

In the version examined in this thesis, a fuzzy set use the triangular membership function, which contains three parameters that demarcate its lower and upper bounds, in addition to the central point of the fuzzy set, where the maximum membership value is returned. The parameters, respectively denoted as l, u and c compose the membership function for a given value x, as shown below:

$$\mu(x,l,c,u) = \begin{cases} 0 & if \quad (x < l) \; or \; (x > u) \\ \frac{x-l}{c-l} & if \quad l \leq x \leq c \\ \frac{u-x}{u-c} & if \quad c \leq x \leq u \end{cases} \tag{5.1}$$

According to the NSFS concept and given a perturbation function π, with displacement δ and scale ρ, the membership function for an NSFTS fuzzy set is defined by:

$$\mu(x, \pi(l,c,u,\delta,\rho)) \tag{5.2}$$

Where the perturbation function π is:

$$\pi(l,c,u,\delta,\rho) = \left\{ \frac{\rho}{2} - (l+\delta) \; , \; c+\delta \; , \; \frac{\rho}{2} + (u+\delta) \right\} \tag{5.3}$$

Examples of the effects of the perturbation function on the membership function are illustrated in Figure 17.

Similar to the FTS models discussed in the chapter 4, the NSFTS has a training procedure, in which its fuzzy sets are initially defined and its FLRG created. Based on the rule base, the model can perform a forecasting procedure similar to other conventional FTS

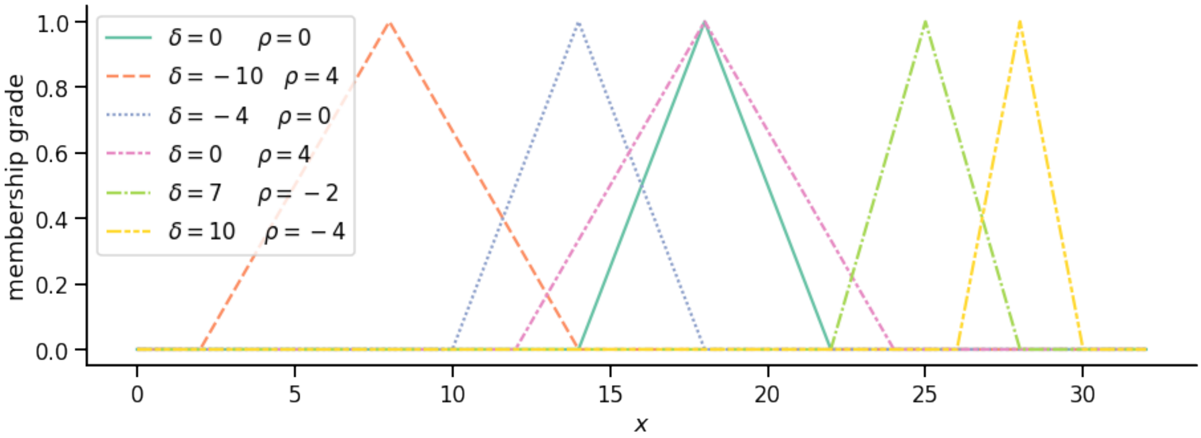


Figure 17 – Triangular Membership Function with $l = 14$, $c = 18$ and $u = 22$ and perturbation functions with different parameters.

Source: e Silva et al. [2020]

models. The model also has a parameter adaptation procedure, where displacement and scale parameters are updated according to the forecasting error observed for the model, so that the membership functions of each fuzzy set are adjusted. The procedures are detailed below.

5.1.1 Training Procedure

Given a training set Y_t, extracted from a univariate time series Y with elements $y(t) \in \mathbb{R}^1$, the training procedure takes the following steps:

Step 1 - Universe of Discourse Partitioning

Given a Universe of Discourse $U = [lb, ub]$ where the lower bound $lb = \min(Y_t) - \min(Y_t) \times 0.2$ and the upper bound $ub = \max(Y_t) + \max(Y_t) \times 0.2$. The factors 0.2 are default values applied to prevent an underestimation of the bounds from data available in training set. In this step, U is divided into k parts, each one corresponding to a fuzzy set. The partitioning is oriented by the midpoints of each fuzzy set, denoted as $c_i, i = 0, 1, \ldots, k-1$ calculated as follows:

$$c_i = lb + i \times \frac{(ub - lb)}{(k-1)} \tag{5.4}$$

Step 2 - Fuzzy Sets Definition

As previously mentioned, this method adopts triangular membership functions. For each fuzzy set A_i, a MF μ_{A_i} is defined as in equation 5.5:

$$\mu_{A_i}(x) = \begin{cases} 0 & if \quad x < l \text{ or } x > u \\ \frac{x-l_i}{c_i-l_i} & if \quad l_i \leq x \leq c_i \\ \frac{u_i-x}{u_i-c_i} & if \quad c_i \leq x \leq u_i \end{cases} \tag{5.5}$$

where $l_i = c_{i-1}$ and $u_i = c_{i+1}$.

Additionally, a perturbation function π_i, as described in equation 5.3, is assigned to A_i, so the fuzzy set can be considered an NSFS. The parameters displacement and scale are initialized as $\delta_i = 0$ and $\rho_i = 0$, respectively.

Step 3 - Time Series Fuzzification

The time series $Y_t = \{y(0), y(1), \ldots, y(T)\}$ is then transformed into an FTS $F_t = \{f(0), f(1), \ldots, f(T)\}$, where each element $f(t)$ is represented by its membership functions:

$$f(t) = \{\mu_{A_0}(y(t)), \mu_{A_1}(y(t)), ..., \mu_{A_{k-1}}(y(t))\} \tag{5.6}$$

Step 4 - Temporal Patterns Extraction

Given temporal patterns with the format $A_p \rightarrow A_c$, where A_p is the precedent and A_c the consequent, both are related to the fuzzy set with maximum membership, as described in the following equations:

$$\begin{aligned} A_p &= \arg\max_{A_i} (\mu_{A_i}(y(t-1))) \\ A_c &= \arg\max_{A_i} (\mu_{A_i}(y(t))) \end{aligned} \tag{5.7}$$

Step 5 - Rule Base Creation

In this step, the rule groups (FLRG) that compound the rule base are created by grouping the temporal patterns with the same precedent, creating rules with the format $A_p \rightarrow A_a, A_b, \ldots$.

Step 6 - Residuals Computation

The residuals are the difference between estimated values from the forecasting method and the observed values for the same timestamp. NSFTS method uses the residuals to update the perturbation function parameters. In the training procedure, the residuals are calculated by applying the forecasting procedure, to be described in section 5.1.3, to the training set Y_t. The last w items are forecasted to calculate the set of residuals $\mathcal{E}$ defined as:

$$\mathcal{E} = \{\epsilon(t-w), \epsilon(t-(w-1)), ..., \epsilon(t)\} \tag{5.8}$$

where $\epsilon(t) = y(t) - \hat{y}(t)$ and $\hat{y}(t)$ is the estimated value.

5.1.2 Parameter Adaptation Procedure

Given a set of residuals $\mathcal{E}$, as calculated in equation 5.8, the forecast value $\hat{y}(t+1)$ and its corresponding observed value $y(t+1)$, the following steps are taken to update the parameters.

Step 1 - Out of Range Detection

The displacement parameter is updated from changes in its midpoint, which in turn are detected when $y(t)$ is outside the range of the UoD. Thus, if $y(t)$ is below the lower bound lb then $d_l = lb - y(t)$, else $d_l = 0$. If $y(t)$ is above the upper bound ub then $d_u = y(t) - ub$, else $d_u = 0$. The displacement range r and the displacement midpoint d_{mp} are then calculated as in equations 5.9 and 5.10.

$$r = d_u - d_l \tag{5.9}$$

$$d_{mp} = r/2 \tag{5.10}$$

Step 2 - Mean and Variance of Residuals

In this step, mean $\overline{\mathcal{E}}$ and variance $\sigma_{\mathcal{E}}$ of the set $\mathcal{E}$ are calculated. The values will be used to adjust position and length of the fuzzy sets.

Step 3 - Displacement Calculation

For each fuzzy set A_i a corresponding displacement δ_i is calculated:

$$\delta_i = \overline{\mathcal{E}} + \left(i\frac{r}{k-1} - d_{mp}\right) + \left(i\frac{2\sigma_{\mathcal{E}}}{k-1} - \sigma_{\mathcal{E}}\right) \tag{5.11}$$

Step 4 - Scale Calculation

After the calculation of all the displacements δ_i for each fuzzy set A_i, corresponding scaling factors ρ_i are calculated as follows:

$$\rho_i = |\delta_{i-1} - \delta_{i+1}| \tag{5.12}$$

The updated parameters δ_i and ρ_i are then used by the perturbation function π_i, described in equation 5.3, in order to adapt the parameters of the membership function according to the observed error.

Step 5 - Residuals Set Update

The final step in this procedure is to update the residuals set $\mathcal{E}$. It is done by calculating the error term for the forecast value $\epsilon(t+1) = y(t+1) - \hat{y}(t+1)$. This value is inserted in $\mathcal{E}$, while the oldest value is removed.

$$\mathcal{E} = \mathcal{E} \setminus \epsilon(t-w) \cup \epsilon(t+1) \tag{5.13}$$

5.1.3 Forecasting Procedure

Given a first order model and an input value $y(t)$. The following steps are taken to forecast $\hat{y}(t+1)$.

Step 1 - Input Value Fuzzification

The membership grade μ_{A_i} is calculated for each fuzzy set A_i using the membership function described in equation 5.1, with the parameters updated by the perturbation function. The fuzzy sets whose membership grade is greater than zero are selected.

Step 2 - Fuzzy Rule Matching

The selected fuzzy sets A_j are used as input in the rule base to match rules according to their precedent. A rule set $\mathcal{S}$ is then defined as follows:

$$\mathcal{S} = \{A_j \rightarrow C_j | \mu_{A_j}(y(t)) > 0\} \tag{5.14}$$

where A_j is the fuzzy sets that represents the precedent of the rule and C_j is the consequent, composed by the fuzzy sets grouped by this precedent.

Step 3 - Defuzzification

The forecast value $\hat{y}(t+1)$ is then computed as the weighted sum of the rule midpoints mp by their membership grades μ_j, as follows:

$$\hat{y}(t+1) = \sum_{A_j \to C_j \in \mathcal{S}} \mu_{A_j}(y(t)) \cdot mp(C_j) \tag{5.15}$$

where

$$mp(C) = \frac{\sum_{A_i \in C} c_{A_i}}{|C|} \tag{5.16}$$

5.1.4 Computational cost

In section 4.1, the computational complexity of a basic FTS model is analyzed. Since NSFTS presents similar training and forecasting steps their cost can be considered analogous, with the difference of its adaptation procedure, described in section 5.1.2. Such procedure, for an input of size T, memory window length W, refreshing interval R and k fuzzy sets, have a cost of $O(T/R \cdot W \cdot \log k)$. For the same input a retraining procedure would have a cost of $O(T/R \cdot W \cdot (\log k)^{\Omega})$. Such comparison demonstrates the reduction of computational cost with the use of the adaptation mechanism.

5.2 Experiments

In this section, NSFTS is assessed and compared to other forecasting models when applied to solar and wind energy time series. The details of the experiments carried out and the results obtained are described below.

5.2.1 Data sets

For the experiments in this work, the data sets discussed in chapter 2 were used. The solar energy data set is detailed in Table 1 and the wind energy data set is described in Table 2.

The data sets presented some corrupted or missing values at some timestamps, probably due to measurement failures. All data in such conditions was removed.

5.2.2 Experiments Design

In Machine Learning experiments, a given data set can be divided into training and test sets, where the first is used to construct the learning models and the last one to evaluate it. When the problem is related to time series data, the data set cannot be

Table 1 – Solar Energy Dataset

NREL Oahu Solar Measurement Grid	
Variable	Global Horizontal Irradiance (GHI)
Frequency	10 minutes
Start Date	2010-06-01
End Date	2011-07-31
Number of sites	1
Samples	38766

Table 2 – Wind Energy Dataset

Wind Power Time Series - GEFCOM2012	
Variable	Wind Power
Frequency	Hourly
Start Date	2009-07-01
End Date	2010-08-30
Number of sites	1
Samples	8761

split randomly, since the temporality of the data needs to be taken into account. There are different approaches to design tests for forecasting methods and the proper choice of an approach is an important task, as discussed in Tashman [2000]. In these experiments, the rolling window method is chosen, and the following steps are performed:

1. Define a rolling window size where a rolling window is a sequence of observations from the data set which will be split into training and test subsets.
2. Define the size of training and test subsets
3. Define the number of increments between successive rolling windows
4. For each rolling window:
   - Train the model using the training set
   - Apply the model to the test set
   - Compute forecasting errors

The method is also described in Figure 18. The rolling window size for both data sets was 4 weeks, where the first 3 weeks were assigned for training and the last one was used for testing. It means that about 75% of data was separated for training and 25% for testing. The number of increments between successive windows is the equivalent to 1 week of observations so that the test subset of window i is part of the training subset of the next window $i + 1$. Both data sets comprise one year and one month of observations, or

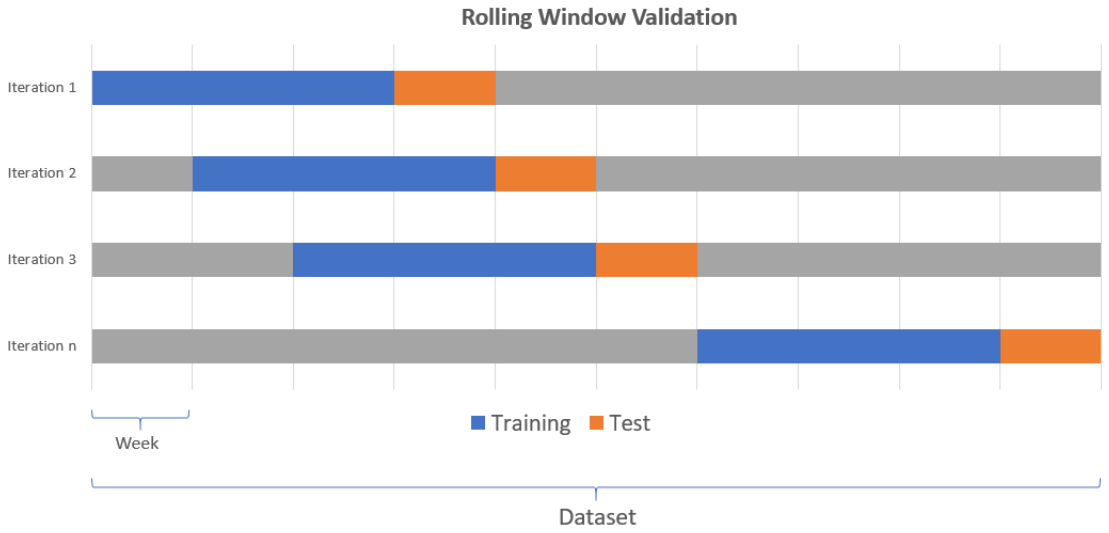


Figure 18 – Rolling Window Validation[2]

13 months. For the rolling window experiment, only the 12 last months are used. The first month is used in hyperparameter tuning, where it is split only once into training and test subsets (with sizes of 3 and 1 weeks, respectively), and the best configuration for each forecasting model is searched. The search for the best hyperparameter is performed using the Tree of Parzen Estimators (TPE) algorithm implemented in **Hyperopt library**[1].

It is worth noting that NSFTS has as one of their differentials being an adaptive model, which has the ability to adjust to the data during the forecasting process, without the need for new training steps. Thus, unlike the other models used in the comparison, NSFTS performed training only in the first iteration of the experiments, while the other models followed the procedure described for the rolling window validation.

For the comparison with the NSFTS, were chosen: the persistence (or naive) method, which basically indicates as forecast the present value; a conventional FTS model and a high order version (HOFTS), as described in chapter 4; and a Multi Layer Perceptron (MLP).

5.2.3 Metrics

The output of the forecasting models are the estimated $\hat{Y}$ time series for each point of interest. During the parameter tuning step, the choice of the best model was guided by the calculation of the **Root Mean Square Error (RMSE)**. The final model for each rolling window was evaluated under its RMSE value, the **Normalized Root Mean Square Error (nRMSE)** and the **Symmetric Mean Absolute Percentage**

[1] http://hyperopt.github.io/hyperopt/

Error (SMAPE). The formulas are described below:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n}} \tag{5.17}$$

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|Y_t - \hat{Y}_t|}{|\hat{Y}_t| + |Y_t|} \tag{5.18}$$

where Y_i are the measured observations at the point of interest at time t, $\hat{Y}_t$ denotes the predicted values and n is the length of dataset.

5.2.4 Forecasting Results

The performance of the models was evaluated from the average error value measured in all windows used for forecasting in the experiments. Figure 19 and Table 3 describe the values of RMSE and SMAPE for the experiments on the solar energy data set.

Forecasting Errors		
	RMSE	SMAPE
Persistence	477.04 ± 71.67	99.78 ± 0.06
MLP	101.81 ± 20.79	26.29 ± 5.37
FTS	194.32 ± 32.33	40.32 ± 4.09
HOFTS	110.44 ± 21.04	31.48 ± 4.76
NSFTS	120.79 ± 16.55	34.28 ± 5.29

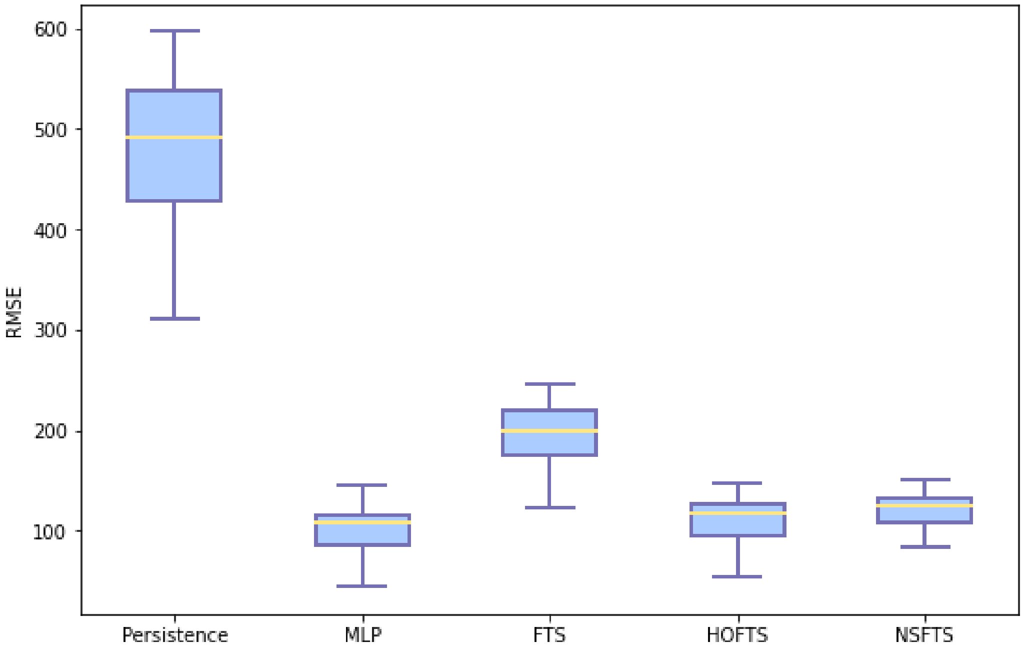
Table 3 – Solar Energy Dataset - NSFTS Experiments

The experiments with the wind energy data set are described in Figure 20 and Table 4.

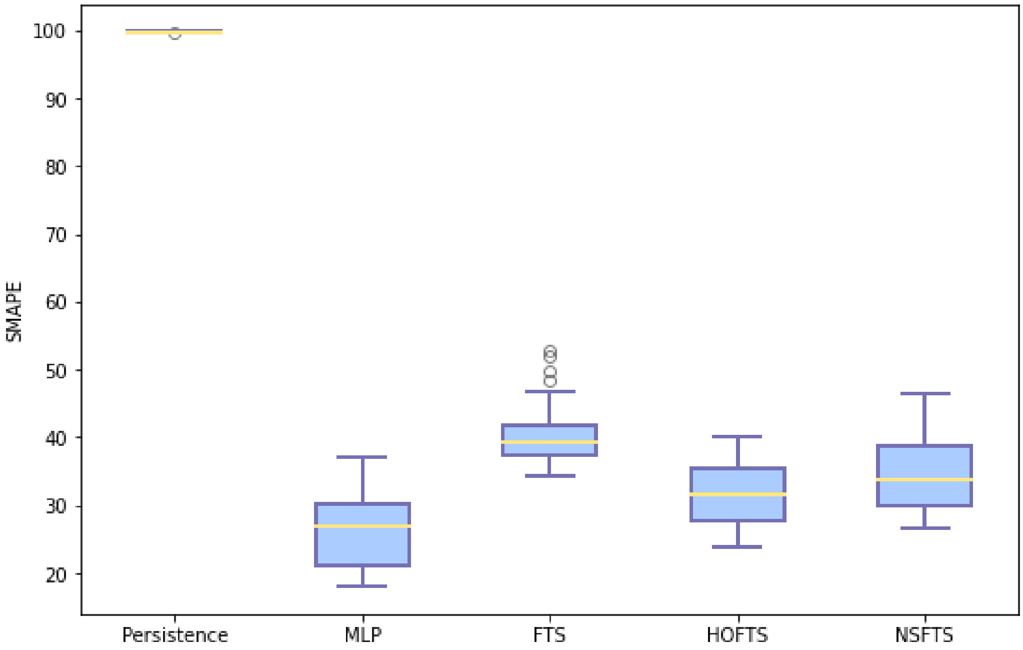
Forecasting Errors		
	RMSE	SMAPE
Persistence	0.07 ± 0.02	22.24 ± 12.97
MLP	0.07 ± 0.02	29.13 ± 14.38
FTS	0.24 ± 0.06	48.11 ± 14.16
HOFTS	0.07 ± 0.02	29.91 ± 14.81
NSFTS	0.08 ± 0.02	30.81 ± 14.64

Table 4 – Wind Energy Dataset - NSFTS Experiments

The results indicate a significant improvement in the performance of the models when applying the SSA decomposition. In all experiments, MLP obtained the best results,

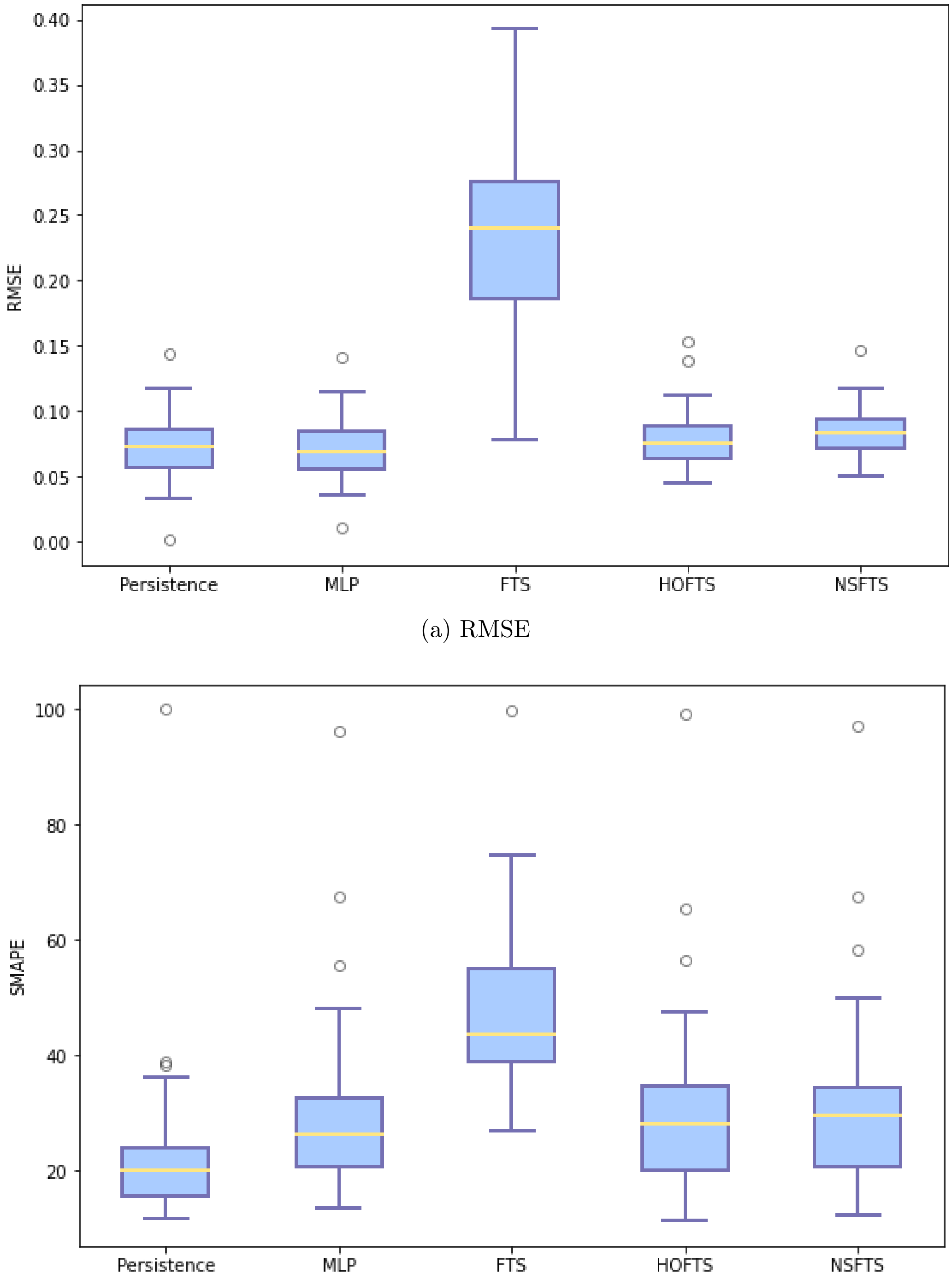


(a) RMSE



(b) SMAPE

Figure 19 – Solar Energy Dataset - NSFTS Experiments



(a) RMSE

(b) SMAPE

Figure 20 – Wind Energy Dataset - NSFTS Experiments

followed by the HOFTS model. NSFTS presented results close to HOFTS, with the advantage of not requiring training at each iteration of the experiments. The conventional FTS model had difficulties in both data sets, being sometimes overcome by the persistence model, which is often used as a minimum benchmark for choosing models that are eligible for forecasting in the context of renewable energy time series.

5.3 Discussion

In this chapter a new adaptive FTS model is described and evaluated as a forecasting model applied to renewable energy. This study was motivated by the good results obtained with the HOFTS model and the expectation that adaptation mechanisms can open new perspectives for improvements for FTS models. In the experiments, the NSFTS showed performance close to the HOFTS, which, in turn, had a performance comparable to MLP. The difference in accuracy between HOFTS and MLP has the counterpoint of HOFTS providing a model of less complexity and easier to understand. When the comparison is made between NSFTS and HOFTS, the first presents a model that requires less information in the past, given that the evaluated version is a first order model, and less computational complexity, since only in the first iteration of each experiment a training procedure is performed. When comparing the performance of NSFTS with the conventional FTS, the improvement resulting from the inclusion of an adaptive mechanism is even more evident.

The analysis of NSFTS, therefore, indicates as a way to improve FTS-based models applied to non-stationary data the inclusion of mechanisms that can adapt to the changes observed in the data. An NSFTS expansion would demand an increase in the scope of the model, so that it could generate rules from patterns with more relevant information about the data. More specifically, a greater number of lags (a high order model) and multivariate time series (a spatio-temporal model). Also, the NSFTS adaptive mechanism also has an improvement point, which is the creation of new rules during runtime. The model defines its rule base in the initial training and only adjusts their parameters during the forecasting process, which means that new patterns that require additional rules may not be represented adequately by the model. The following chapters describe a proposal for a high order multivariate FTS model which presents an adaptive mechanism that allows, in addition to adjusting the rules, the inclusion of new ones to the rule base without the need for a new training step. At the kernel of the mechanism is an evolving algorithm originally applied to clustering problems, to be described in detail in chapter 6. The evolving multivariate FTS model is developed from the integration of this clustering algorithm with a high order FTS model, and its details are discussed in the chapter 7. Its performance is also evaluated when applied to spatio-temporal forecasting problems in renewables, as well as other non-stationary data sets.

Chapter 6

Evolving Clustering Algorithm

The development of the clustering model presented in this chapter was motivated by the growing presence of streaming data in the most diverse applications. As this kind of data is often non-stationary and with high frequency, there is a need to propose solutions that can fit and adapt to this data in a online setting. Nowadays, online clustering is an important tool for many applications such as knowledge discovery [Gama, 2010], process fault detection [Lemos et al., 2011, Costa et al., 2015], recommendation systems and anomaly detection [Angelov, 2014b]. As analyzed in chapter 2, renewable energy systems also produces non-stationary time series related to their energy generation, often arranged in increasingly shorter intervals, such as minutes or seconds. Such time series may then present similar patterns to streaming data, also requiring online settings. Therefore, there is a convergence of objectives that, as shown in the next chapter, will result in applying this evolving clustering algorithm to the problems addressed in this thesis.

Many approaches have been proposed for clustering data streams as discussed in a survey by Silva et al. 2013. The algorithm here presented follows a widely used approach of first divide the streaming data into micro-clusters and then find the final clusters based on micro-clusters that share common properties. One of the first algorithms to use this approach is Clustream [Aggarwal et al., 2003] which uses samples within a time window to create and update micro-clusters incrementally. However, Clustream has problems when dealing with non-spherical clusters. A solution for this problem is proposed in density based algorithms like DBstream [Hahsler and Bolaños, 2016], Denstream [Cao et al., 2006] which can create clusters of arbitrary shapes. A disadvantage in both methods is that the large number of free parameters makes the search for the best model very difficult in some applications. For example, in non-stationary data such fixed structures and predefined parameters that should be known in advance might not be able to handle future changes in data distribution. In this sense, evolving clustering algorithms have gained increasing importance. These algorithms have the ability to evolve their structure and parameters over time as new data arrives. An example is CEDAS algorithm, proposed by Hyde et al.

2017. It can cluster data from arbitrary shapes that arrive in random order by creating micro and macro-clusters and also can adapt to non stationary data. However, its output is only cluster assignment and it is not possible to discover the densities of the regions in the data space.

This chapter describes MicroTEDAclus, an evolving clustering algorithm that aims to address the gaps previously discussed. It uses micro-clusters based on the concept of Typicality and Eccentricity Data Analysis (TEDA), which defines the criteria for their creation or update as the data arrives. There is also a second offline stage, in which the micro-clusters are grouped into macro-clusters, representing the final clustering scheme of the algorithm. In the next sections, TEDA concept is introduced as well as the clustering algorithm is detailed in its creation processes of micro and macro clusters and evolution mechanism. Experiments with non-stationary data sets are presented in addition to a discussion about the algorithm and the results obtained.

6.1 TEDA

TEDA was designed to incrementally model a non-parametric data distribution based on information obtained from the proximity between data samples. One of its key concepts is the **cumulative proximity**. Given a d-dimensional input vector $\boldsymbol{x_t} \in \mathbb{R}^d$, in the timestamp t, the cumulative proximity $\phi(.)$ of $\boldsymbol{x}_t$ with respect to all existing data samples, is calculated as:

$$\phi_t(\boldsymbol{x}) = \sum_{i=1}^{t} dist(\boldsymbol{x}_t, \boldsymbol{x}_i) \ , \tag{6.1}$$

where $dist(\boldsymbol{a}, \boldsymbol{b})$ is the distance between data points $\boldsymbol{a}$ and $\boldsymbol{b}$, t is the timestamp when the data point $\boldsymbol{x}$ is sampled.

From $\phi_t(\boldsymbol{x})$ the eccentricity $\xi_t(\boldsymbol{x})$ is calculated , which is a measure of the dissimilarity between the data point $\boldsymbol{x}_t$ with respect to all the data samples received until timestamp t.

$$\xi_t(\boldsymbol{x}) = \frac{2\phi_t(\boldsymbol{x})}{\sum_{i=1}^{t} \phi_t(\boldsymbol{x}_i)} \ , \ \sum_{i=1}^{t} \phi_t(\boldsymbol{x}_i) > 0 \ , \ t > 2. \tag{6.2}$$

For the case of euclidean distance, eccentricity can be calculated recursively as described in Equation 6.3, whose formulation is discussed in [Angelov, 2014b].

$$\xi(\boldsymbol{x}_t) = \frac{1}{t} + \frac{(\boldsymbol{\mu}_t - \boldsymbol{x}_t)^\intercal(\boldsymbol{\mu}_t - \boldsymbol{x}_t)}{t\sigma_t^2}, \tag{6.3}$$

where $\boldsymbol{\mu}_t$ and σ_t^2 are the mean and variance respectively, that can also be recursively updated:

$$\boldsymbol{\mu}_t = \frac{t-1}{t}\boldsymbol{\mu}_{t-1} + \frac{\boldsymbol{x}_t}{t} \ , \ t \geq 1 \ , \ \boldsymbol{\mu}_1 = \boldsymbol{x}_1. \tag{6.4}$$

$$\sigma_t^2 = \frac{t-1}{t}\sigma_{t-1}^2 + \frac{1}{t-1}\|\boldsymbol{x}_t - \boldsymbol{\mu}_t\|^2 \ , \ \sigma_1^2 = 0 \tag{6.5}$$

The typicality $\tau(\boldsymbol{x}_t)$ is the dual of the eccentricity and represents how typical an arbitrary data point $\boldsymbol{x}_t$ is with respect to all the data points received until the timestamp t.

$$\tau(\boldsymbol{x}_t) = 1 - \xi(\boldsymbol{x}_t) \ , \ t \geq 2 \tag{6.6}$$

Typicality and Eccentricity concepts are illustrated in Figure 21. The data point "A" is more distant from the rest of the data set than the data point "B", therefore "A" has higher eccentricity and lower typicality than "B".

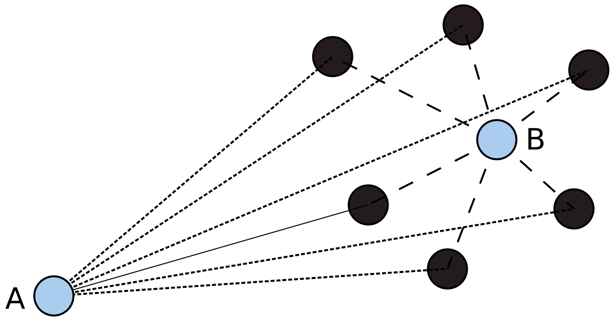


Figure 21 – Illustration of Typicality and Eccentricity concepts in TEDA

From the concepts described, the normalized versions of eccentricity $\zeta(\boldsymbol{x}_t)$ and typicality $\gamma(\boldsymbol{x}_t)$ can be obtained as follows:

$$\zeta(\boldsymbol{x}_t) = \frac{\xi(\boldsymbol{x}_t)}{2} \tag{6.7}$$

$$\gamma(\boldsymbol{x}_t) = \frac{\tau(\boldsymbol{x}_t)}{t-2} \tag{6.8}$$

The normalized eccentricity $\zeta(\boldsymbol{x}_t)$ is used to define a threshold that determines whether a data sample can be considered an outlier. However, so that the algorithm can adapt to changes in data distribution, this threshold is adjusted in the algorithm according to the processed data. In the next section, the application of these concepts in the development of MicroTEDAclus are described in detail.

6.2 MicroTEDAclus

MicroTEDAclus performs clustering in two steps. The first refers to micro-clusters management. As the data samples are processed, new micro-clusters are created or existing ones are updated. In the second step, micro-clusters are connected according to their proximity and grouped into macro-clusters, which are the final clustering representation of the algorithm. From this representation, data samples are assigned to a macro-cluster according to their membership degrees. These steps are described as follows.

6.2.1 Micro-clusters Step

Given a model with a set of micro-clusters $\mathfrak{m}_i$, $i = 1 \ldots M$, each micro-cluster $\mathfrak{m}_\mathrm{i}$ is defined by the following parameters:

- S_t^i: number of data samples;
- $\boldsymbol{\mu}_t^i$: center;
- $(\sigma_t^i)^2$: variance;
- $\xi^i(\boldsymbol{x}_t)$, $\zeta^i(\boldsymbol{x}_t)$: eccentricity and normalized eccentricity;
- $\tau^i(\boldsymbol{x}_t)$, $\gamma^i(\boldsymbol{x}_t)$: typicality and normalized typicality;
- $D_t^i = \frac{1}{\zeta^i(\boldsymbol{x}_t)}$: density;
- $m_t^i(S_t^i)$: outlier threshold parameter;

When the first data sample $\boldsymbol{x}_1$ is processed by the model, the first micro-cluster $\mathfrak{m}_1$ is created with the following parameters:

$$M = 1 \ , \ S_1^1 = 1 \ , \ \boldsymbol{\mu}_1^1 = \boldsymbol{x}_1 \ , \ (\sigma_1^1)^2 = 0 \tag{6.9}$$

where n is the number of micro-clusters. Note that just a few parameters are calculated when $S_t^i = 1$ because typicality and eccentricity can be only calculated with 2 or more data samples.

For the next data samples $\boldsymbol{x}_t$, the algorithm calculates the typicality and eccentricity of $\boldsymbol{x}_t$ to all the existing micro-clusters using Equations (6.6) and (6.3), respectively and check if $\boldsymbol{x}_t$ is an outlier or not :

$$\zeta^i(\boldsymbol{x}_t) > \frac{m_t^i(S_t^i)^2 + 1}{2S_t^i} \tag{6.10}$$

$$m_t^i(S_t^i) = \frac{3}{1 + e^{-0.007(S_t^i - 100)}} \tag{6.11}$$

It is worth noting that the m_t^i function, which establishes the outlier threshold value, is a sigmoid that tends to saturate due to an increase in S_t^i. The objective is for a micro-cluster to be more restrictive in its initial stage, in order to prevent any data sample from being part of the first micro-clusters, generating very comprehensive partitions. A discussion on the formulation of the outlier mechanism is detailed in [Maia et al., 2020].

The outlier condition rule also includes a special case where $S_t^i = 2$. In order to prevent a cluster to grow indefinitely, a parameter r_0 is added to limit the variance of each micro-cluster. Thus, for $S_t^i = 2$:

$$\left[\zeta_2^i(\boldsymbol{x}_2) > \frac{(m^i(2))^2 + 1}{4}\right] \textbf{ AND } \left[(\sigma_2^i)^2 < r_0\right] \tag{6.12}$$

In the experiments performed in this thesis, r_0 was set to **0.001**. Next, one of two conditions can take place:

Condition 1: $\boldsymbol{x}_t$ is not an outlier for at least one micro-cluster **then** update all the micro-clusters for which this condition holds.

$$\begin{aligned} S_t^i =& S_{t-1}^i + 1 \\ \boldsymbol{\mu}_t^i =& \frac{S_t^i - 1}{S_t^i} \boldsymbol{\mu}_{S_t^i - 1} + \frac{\boldsymbol{x}_t}{S_t^i} \\ (\sigma_t^i)^2 =& \frac{S_t^i - 1}{S_t^i} (\sigma_{t-1}^i)^2 + \frac{1}{S_t^i - 1} \left(\frac{2\|\boldsymbol{x}_t - \boldsymbol{\mu}_t^i\|}{d} \right)^2 \\ \xi^i(\boldsymbol{x}_t) =& \frac{1}{S_t^i} + \frac{2(\boldsymbol{\mu}_t^i - \boldsymbol{x}_t)^\intercal (\boldsymbol{\mu}_t^i - \boldsymbol{x}_t)}{S_t^i (\sigma_t^i)^2 d} \end{aligned} \tag{6.13}$$

where d is the dimensionality of the data set.

Condition 2: $\boldsymbol{x}_t$ is an outlier for all existing micro-clusters **then** create a new micro-cluster.

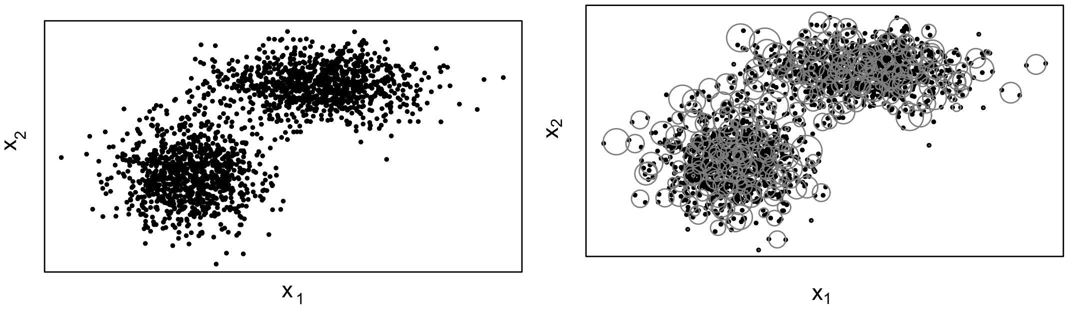
$$M = M + 1;\ S_t^M = 1;\ \boldsymbol{\mu}_t^M = \boldsymbol{x}_t;\ (\sigma_t^M)^2 = 0 \tag{6.14}$$

The micro-cluster update procedure is detailed in Algorithm 1

As an example of how the micro-clusters are represented, Figure 22b shows the micro-clusters defined for the data set in Figure 22a.

Algorithm 1: Micro-cluster update

```
Input: $\boldsymbol{x}_t$, $r_0$
Output: $\mathfrak{m}_i,\ i = 1, 2, \ldots, M$
begin
    while new samples available do
        if $t == 1$ then
            Set $\mathfrak{m}_1$ parameters as defined in Eq. 6.9;
        else
            $flag \leftarrow true$;
            for $i = 1 : M$ do
                $\mathfrak{m} \leftarrow \mathfrak{m}_i$;
                if $S_t^i == 2$ then
                    $outlier \leftarrow$ condition of Eq. 6.12;
                else
                    $outlier \leftarrow$ condition of Eq. 6.10;
                end
                if $outlier == false$ then
                    Update $\mathfrak{m}_i$ according to Eq. 6.13;
                    $flag \leftarrow false$;
                end
            end
            if $flag == true$ then
                Create a new micro-cluster with the parameters of Eq. 6.14;
            end
        end
    end
end
```



(a) (b)

Figure 22 – (a) Data set. (b) Micro-clusters.

6.2.2 Macro-clusters Step

In this step, the micro-clusters are grouped into macro-clusters, according to their proximity. microTEDAclus represents this organization as an intersection graph, whose adjacency matrix have dimensions equal to the number of micro-cluters. Each element in the matrix is set as 1 if two micro-clusters intersects and 0 otherwise. The condition to check if two micro-clusters intersect considers their mean and variance values. It is shown by Equation (6.15).

$$dist(\boldsymbol{\mu}_t^i, \boldsymbol{\mu}_t^j) < 2(\sigma_t^i + \sigma_t^j)\ ,\ \forall i \neq j \tag{6.15}$$

To avoid that an overlapping data set result in an unique big macro-cluster, the algorithm also includes a micro-cluster activation mechanism, which activates or not the micro-clusters based on their density. Let $\mathfrak{M}_j = \{\mathfrak{m}_1^j, \mathfrak{m}_2^j, \ldots, \mathfrak{m}_l^j\},\ j = 1, 2, \ldots, N$ be the jth macro-cluster composed by a set of connected micro-clusters $\mathfrak{m}^j$. The set of active micro-clusters of the macro-cluster $\mathfrak{M}_j$ are the ones for which the density D_t^l is greater or equal to the average density calculated over all the micro-clusters that belong to $\mathfrak{M}_j$:

$$active(\mathfrak{m}_l^j) = D_t^l \geq mean(D_t^l),\ l = 1, \ldots, |\mathfrak{M}_j| \tag{6.16}$$

The activation mechanism deactivates micro-clusters at low density regions while the ones at high density regions will be active. In other words, the deactivated micro-clusters tend to create borders between the macro-clusters, avoiding the unwanted effect of joining them.

The density estimate of each macro-cluster is calculated as a sum of the normalized typicalities of its active micro-clusters, weighted by their normalized density such as a density mixture model, as discussed in [Angelov, 2014a]:

$$\mathcal{T}_j(\boldsymbol{x}_t) = \sum_{l \in \mathfrak{M}_j} w_t^l t_t^l(\boldsymbol{x}_t) \tag{6.17}$$

$$w_t^l = \frac{D_t^l}{\sum\limits_{l \in \mathfrak{M}_j} D_t^l} \tag{6.18}$$

A new data point $\boldsymbol{x}_t$ is assigned to the macro-cluster for which it has the highest mixture of typicalities score $\mathcal{T}_j(\boldsymbol{x}_t)$. The detailed procedure to calculate macro-clusters is presented in Algorithm 2.

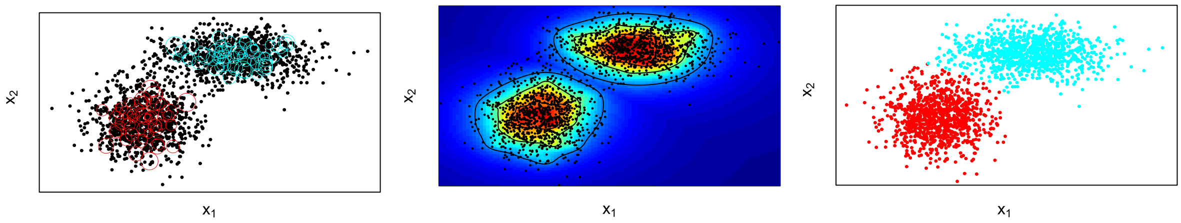
An example of how the macro-cluster step works is illustrated in Figure 23. Figure 23a shows the micro-clusters arranged in two different macro-clusters (blue and red). In Fig.23b, the density regions are highlighted according to the mixture of typicalities.Figure 23c shows the final cluster structures identified by the algorithm in the data set.

6.2.3 Clustering new data

After completing the steps described in Sections 6.2.1 and 6.2.2, MicroTEDAclus is able to cluster new data. Thus, for a new data sample $\boldsymbol{x}_t$, the algorithm updates

Algorithm 2: Macro-clusters update

```
Input: $\boldsymbol{x}_t$, $\mathfrak{m}$
Output: membership degree of $\boldsymbol{x}_t$ for each macro-cluster
begin
    while new samples available do
        $\mathfrak{M} \leftarrow$ Group the micro-clusters that intersect each other according the
          condition expressed in Eq. 6.15;
        Find the set of active micro-clusters of each macro-cluster
          $\mathfrak{M}_j,\ j = 1, 2, \ldots, N$ according Eq. 6.16;
        Calculate $\mathcal{T}_j(\boldsymbol{x}_t)$ for $j = 1, 2, \ldots N$ according Eq. 6.17;
        Assign $\boldsymbol{x}_t$ to the cluster $j$ it has highest $\mathcal{T}_j(\boldsymbol{x}_t)$;
    end
end
```



(a) (b) (c)

Figure 23 – (a) Macro-Clusters. (b) The mixture of typicalities. (c) Cluster assignment.

micro-clusters, recalculates the macro-clusters and returns the macro-cluster for which $\boldsymbol{x}_t$ is more compatible with respect to the mixture of typicalities $\mathcal{T}_j(\boldsymbol{x}_t)$.

6.3 Experiments

This section presents experiments applied to MicroTEDAclus to assess its ability to adapt to changes in the distribution of data streams. Another aspect observed is its scalability, which in the experiments is evaluated based on its clustering performance in face of an increase in dimensionality. The evaluation of such aspects is relevant within the work conducted in this thesis to assist in the decision of using this algorithm in a solution for a spatio-temporal forecasting problem, whose data have variations in distribution and eventually greater dimensionality. A more in-depth evaluation of the algorithm, involving other aspects of the clustering problem, is carried out in Maia et al. 2020 where the model is introduced.

6.3.1 Data sets

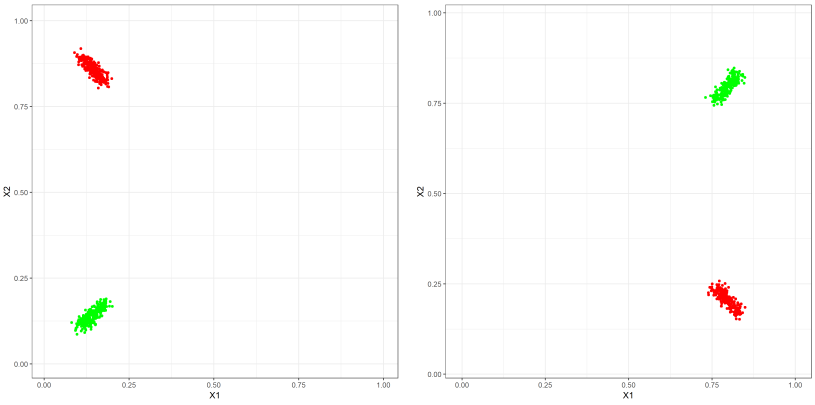
The data sets used to observe the adaptability of the algorithm were artificially generated by the R tool **stream**, developed by Hahsler et al. 2017. For each data set created, different concept drift events were inserted, as described below:

Concept Drift Data sets			
Data set	Clusters	Dimension	Description
STR-B1	2	2	One clusters moves from top left to bottom right while the other one moves from bottom left to top right. Both clusters overlap when they meet exactly in the center of the data space.
STR-B2	3	2	2 static clusters and a third one moving from the first to the second
RBF	1-9 [a]	2	Starts with a fixed number of centroids representing each cluster, which has a random position, a single standard deviation, class label and weight. During the sampling, the cluster attributes are changed, and events such as cluster splitting/merging and deletion/creation are set.

Table 5 – Concept Drift Data sets

[a] RBF experiments start with 3 clusters and according to the events of concept evolution, can change the number of clusters. The range is from 1 to 9 clusters.

Examples of the initial and final states of **STR-B1** and **STR-B2** data sets are shown in Figures 24 and 25, respectively. Figure 26 depicts an output of **RBF**. Figure 26a shows the initial state with 3 clusters. In Figure 26b an event of cluster creation occurs. In Figure 26c a cluster is split. Figure 26d indicates an event of cluster deletion.



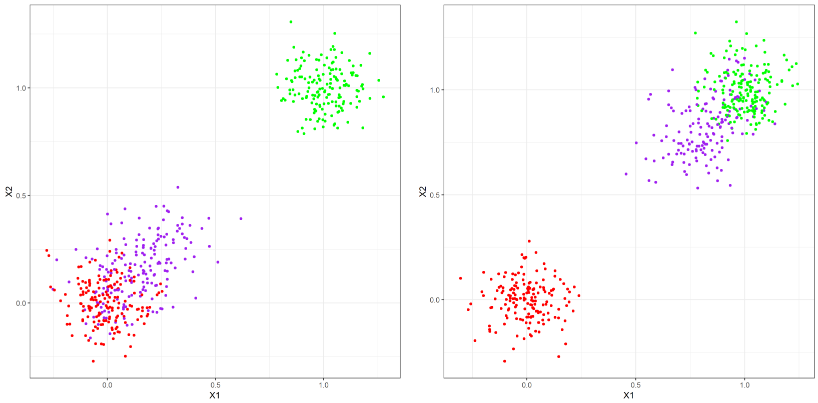
(a)

(b)

Figure 24 – Stream Benchmark 1. (a) Before Concept Drift. (b) After Concept Drift.

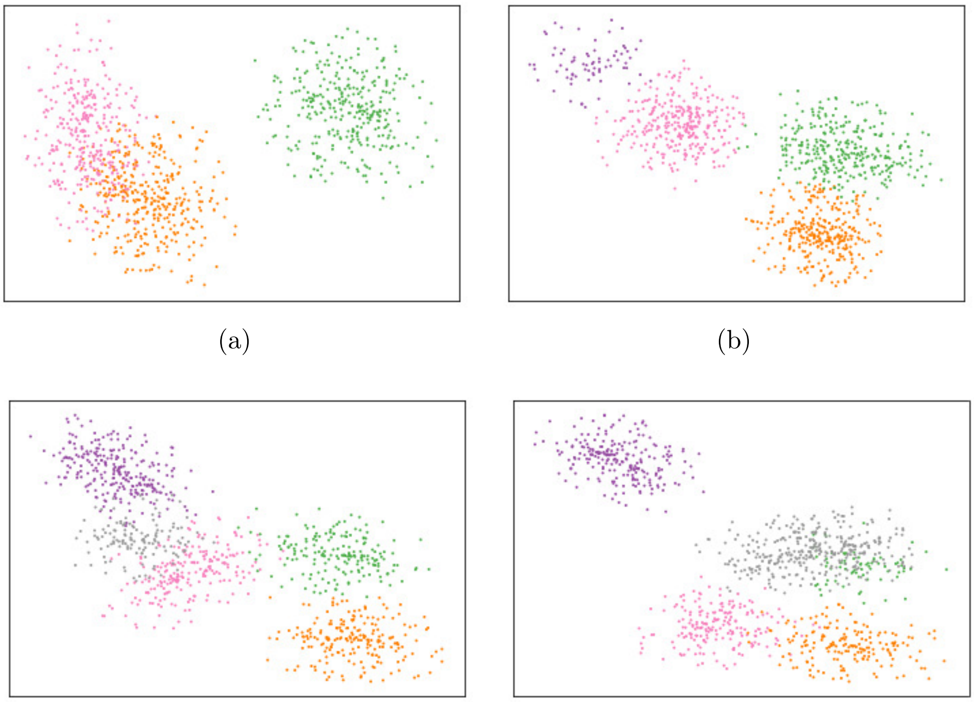
6.3.2 Experiments Design

The clustering algorithms were evaluated using the prequential method. In this method, also known as first-test-then-train method, the instances are used to test and then for training one by one, at the same iteration. If compared to a regular cross-validation



(a) (b)

Figure 25 – Stream Benchmark 2. (a) Before Concept Drift. (b) After Concept Drift.



(a) (b)

(c) (d)

Figure 26 – Random RBF Generator. (a) Initial State. (b) After First Event (cluster creation). (c) After Second Event (cluster split). (d) After Third Event (cluster deletion).

method, prequential has the advantages of requiring a smaller number of runs and also maintaining the temporal order of the data stream. The experiments define, for each evaluated clustering algorithm, an initial training set and subsequent prequential windows (where test-then-train procedure occurs). Figure 27 illustrates how the iterations work. The initial training set has the same size of a prequential window. The prequential error

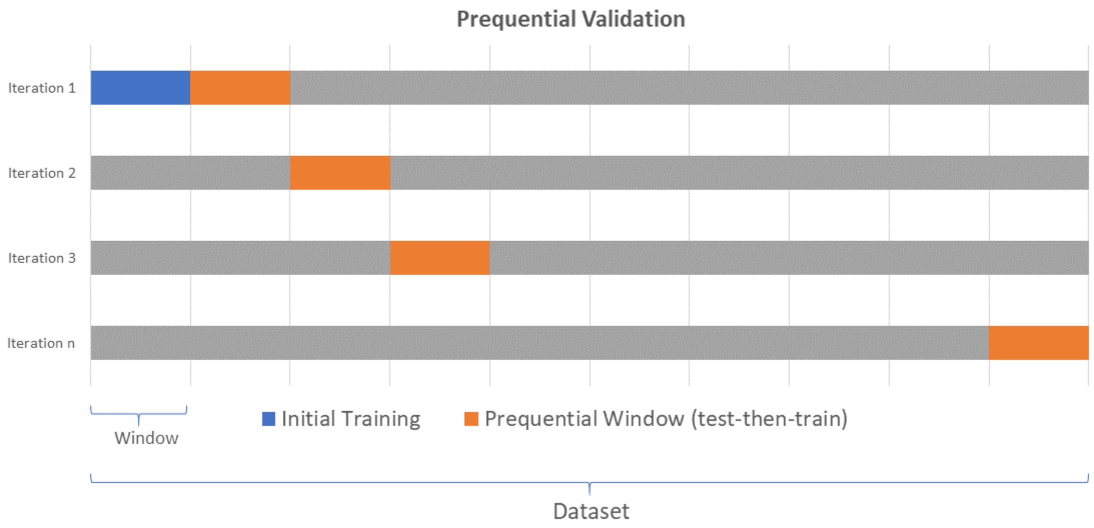


Figure 27 – Prequential Validation

for each window w is calculated as follows:

$$P_w(t) = \frac{1}{w} \sum_{i=t-w+1}^{t} L(\hat{y}_i, y_i) \tag{6.19}$$

The settings for the experiments using each data set are described in Table 6. For each combination of method and data set, the experiment was performed 30 times.

Table 6 – Experiment settings for the data sets

Data set	**STR-B1**	**STR-B2**	**RBF**
Samples	4000	4000	10000 [a]
Window Size	100	100	100
Windows	40	40	100

[a] After the generation of 1000 samples, one concept drift event occurs, which results in 10 events during each RBF experiment.

MicroTEDAclus was compared with the following data stream clustering algorithms: **Denstream** Cao et al. [2006], **CluStream** Aggarwal et al. [2003] and **StreamKM++** Ackermann et al. [2012]. The benchmark algorithms used in the experiments are available in the R package **streamMOA** Hahsler and Forrest [2019]. The hyperparameters tuning of the algorithms was performed by using grid search. For each data set, the samples used in the initial training were applied to the tuning. Table 7 lists the range of parameters where the grid search occurred for all the methods.

Table 7 – Range of parameters for grid search

Method	Parameter	Description	Range
DenStream	ϵ	Micro-cluster radius	$\{0.001, \dots, 0.01\}$
	μ	Weight limit	$\{1, \dots, 20\}$
	β	Outlier threshold	$\{0.2, 0.4\}$
CluStream	m	Micro-clusters	$\{10, \dots, 1000\}$
	h	Time window	$\{100, \dots, 1000\}$
	t	Boundary factor	$\{1, \dots, 10\}$
StreamKM++	s	Size of Coreset	$\{10, \dots, 1000\}$
	k	Number of clusters	$\{1, \dots, 10\}$

6.3.3 Metrics

The loss function chosen for the experiments was the Adjusted Rand Index. It calculates the similarity between two data clusterings. The similarity measured by the function is based on the structures of the clusters, which helps in the evaluation of clustering algorithms that use different labeling mechanisms for the clusters found.

Given a set S of n samples $S = \{o_1, \dots, o_n\}$ and two data clusterings of S to compare, $X = \{X_1, \dots, X_r\}$, a partition of S into r clusters, and $Y = \{Y_1, \dots, Y_s\}$, a partition of S into s clusters, and the contingency table:

Table 8 – Contingency Table

	Y_1	Y_2	$\dots$	Y_s	Sums
X_1	n_{11}	n_{12}	$\dots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\dots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\dots$	n_{rs}	a_r
Sums	b_1	b_2	$\dots$	b_s	

The Adjusted Rand Index ARI is:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{6.20}$$

The function returns a value close to 0.0 for random labeling and exactly 1.0 when the partitions X and Y are identical.

6.3.4 Clustering Results

6.3.4.1 Prequential Evaluation

Figures 28 and 29 show the results of prequential validation for the data sets **STR-B1** and **STR-B2**, respectively. In both cases, concept drifts that mainly change the distribution means occurs at most prequential windows During the evaluation, such events affect the clustering methods in a similar way. The performance is unstable, having an improvement when some clusters merge. The prequential validation for data set **RBF**, illustrated in Figure 30, also presents situations where concept drifts occur at all time steps, including events such as cluster creation, deletion, merge or split which are set to happen after each 1000 samples. The performance of all the clustering algorithms decrease during the sequence of concept drift events. However, while the benchmark algorithms present a tendency to drop in performance, MicroTEDAclus adapt to such events over time.

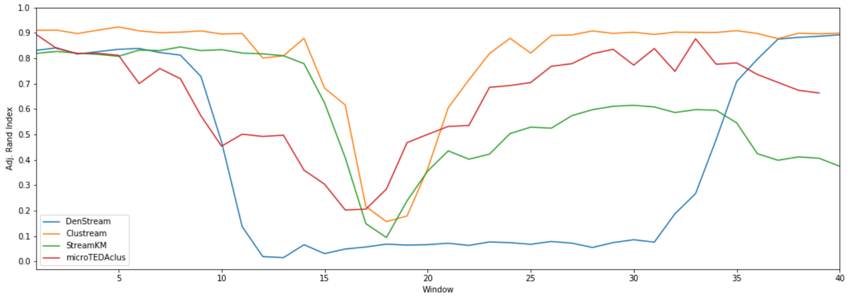


Figure 28 – Prequential Evaluation - STR-B1

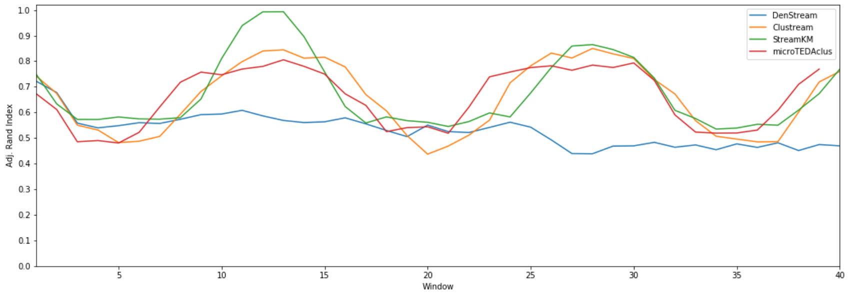


Figure 29 – Prequential Evaluation - STR-B2

Table 9 lists the results of ARI values for each clustering method and data set. In terms of accuracy, Clustream, StreamKM++ and MicroTEDAclus presented comparable results, with small differences between them depending on the experiment and the data set.

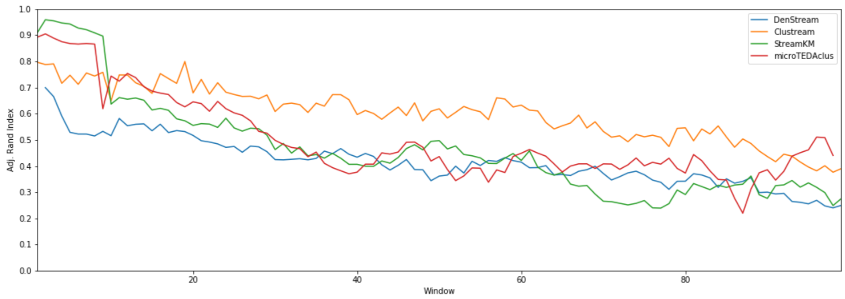


Figure 30 – Prequential Evaluation - RBF

Table 9 – Adjusted Rand Index for Clustering Methods

Data set	DenStream	CluStream	StreamKM	MicroTEDAclus
STR-B1	0.61 ± 0.31	0.57 ± 0.33	0.66 ± 0.27	**0.68 ± 0.21**
STR-B2	0.65 ± 0.13	0.65 ± 0.13	0.55 ± 0.04	**0.69 ± 0.12**
RBF	0.42 ± 0.10	**0.61 ± 0.06**	0.46 ± 0.03	0.50 ± 0.07

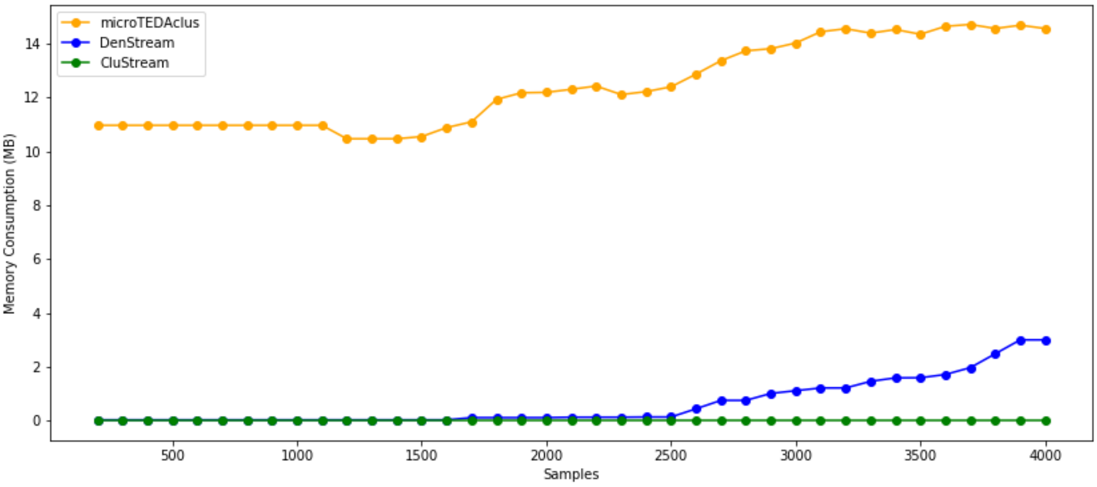
6.3.4.2 Memory Consumption Evaluation

The data set STR-B1 was used in this experiment to monitor the memory consumption of the algorithms. In this experiment, only the Clustream and DenStream algorithms were chosen as a benchmark, as they implement approaches similar to MicroTEDAclus, using micro-clusters structures. For this experiment, Python versions of the algorithms were used, in order to prevent differences between programming languages from influencing the results. Figure 31a shows the evolution of memory consumption, in megabytes, for each model during data set processing. The plots represent the difference between the amount of memory allocated before the start of the algorithms execution and during the processing of each window. Figure 31b represents the Adjusted Rand Index values observed during the experiments.

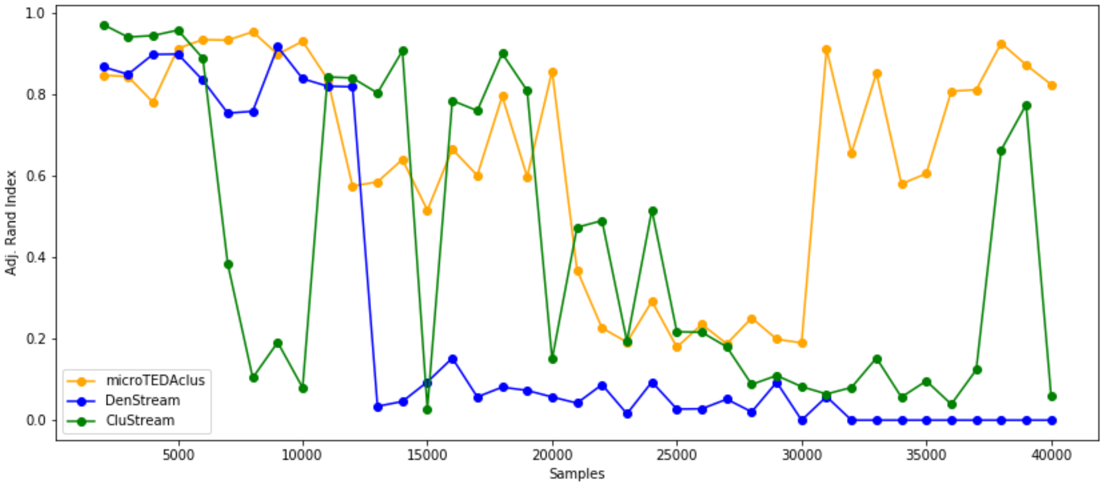
In the experiments, MicroTEDAclus presented a proportionally higher memory consumption if compared with DenStream and CluStream. This is due to its internal data structures for representing macro-clusters and micro-clusters. However, this consumption after an increase to fit the concept drift remains stable at the end of the experiment. In addition, MicroTEDAclus presented a more effective ability to adapt to the drift, as shown in Figure 31b. While MicroTEDAclus was able to recover from the events, the other algorithms had their performance reduced when trying to recover from data variations.

6.3.4.3 Processing Speed Evaluation

In this last experiment, the algorithms were assessed when confronted data sets of larger dimensionality. Each data set was composed of three clusters generated from



(a)

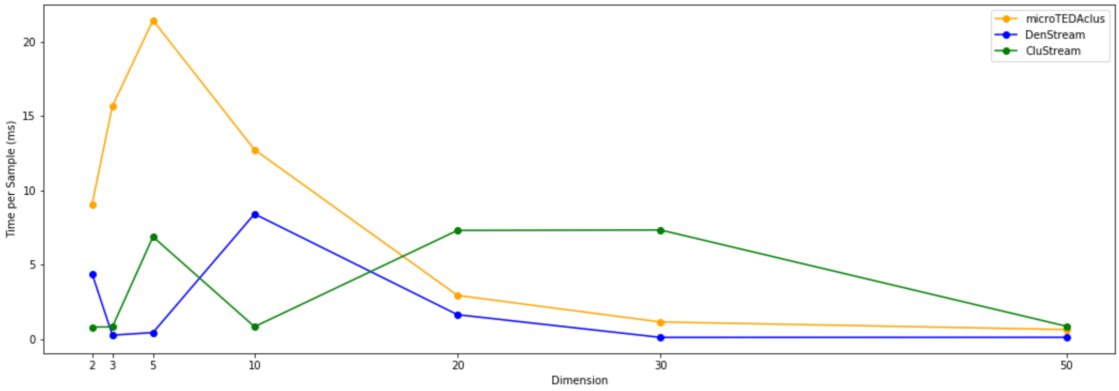


(b)

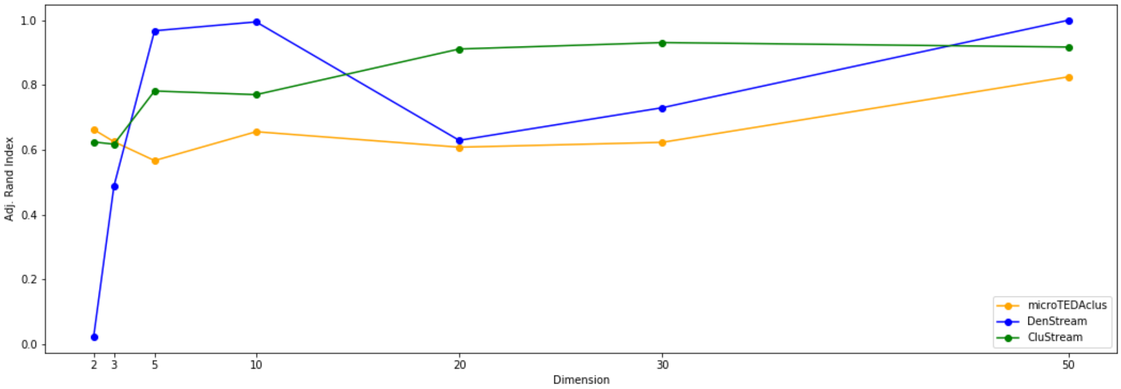
Figure 31 – Memory Consumption Experiments. (a) Memory in megabytes. (b) Adjusted Rand Index.

Gaussian distributions and their dimensions ranged from 2 to 50. 4,000 data samples were sampled for each experiment with dimension d, which was run ten times and the mean values were reported. The experiments used the prequential evaluation described in Section 6.3.2 with window size of 100. Figure 32a compares the average processing time per data sample in milliseconds of CluStream, Denstream and MicroTEDAClus. Figure 32b shows the accuracy of each algorithm throughout the experiments.

The results indicated that, on average, CluStream and DenStream are faster than MicroTEDAclus to deal with the problems proposed in the experiment. This difference is even more evident in smaller dimensional problems. However, MicroTEDAclus is able to maintain its accuracy, even in larger data sets. It is also worth mentioning that, unlike the



(a)



(b)

Figure 32 – Scalability Experiments. (a) Average processing time per sample in milliseconds. (b) Adjusted Rand Index.

benchmark algorithms, which required adjustments to the hyperparameters at each data set, MicroTEDAclus kept its only adjustable hyperparameter fixed at 0.001. This reinforces the robustness of the algorithm when faced with different dimensionality problems.

6.4 Discussion

In this chapter the evolving algorithm MicroTEDAclus for clustering data streams was introduced. It was evaluated for its ability to handle data streams with different concept drift events. In the experiments, it was compared in terms of clustering accuracy with algorithms developed for the same purpose, some of them using similar concepts such as online clustering using micro-clusters. From the results, it was possible to observe that MicroTEDAclus had a competitive performance. However, it has the advantage of being very robust in terms of hyperparameters. The only one to be adjusted, r_0, remained the same for all the experiments. By avoiding the expensive and sometimes time-consuming hyperparameter tuning tasks, the algorithm presents a valuable differential when applied to data stream problems that usually require quick responses.

Another aspect observed in the experiments is the good ability of the algorithm to deal with data of higher dimensionality. MicroTEDAclus showed good accuracy in clustering larger dimension data sets. There is a drawback in memory consumption, which, despite being constant in the experiments, was still higher than the other benchmark algorithms. The memory consumption of MicroTEDAclus is directly linked to the number of micro-clusters created. Some of these micro-clusters may be underused by the algorithm over time, due to changes in data distribution parameters or situations with overlapping micro-clusters. Therefore, an improvement point for the algorithm could be the development of mechanisms that optimize the number of micro-clusters used, in order to reduce the memory consumption.

The observed results present MicroTEDAclus as a clustering algorithm capable of adapting to various changes in the distribution of data, in addition to having a good ability to deal with high dimensionality. It also presented the advantage of requiring few adjustments to its hyperparameters to work. Such characteristics enable it to be applied to the problems studied in this thesis, which involve a large volume of non-stationary data, possibly of higher dimensionality. In the next chapter, an application of MicroTEDAclus to a forecasting problem is introduced, from the integration of it to a method intended for this purpose.

Chapter 7

Evolving Multivariate Fuzzy Time Series

This chapter introduces a novel forecasting model named **evolving multivariate FTS (e-MVFTS)** It was conceived from the integration of the Fuzzy Time Series, in its High Order version, with MicroTEDAclus clustering algorithm, described in chapter 6. The model deals with spatio-temporal data in the format discussed in this thesis, in addition to presenting a dynamic adaptation mechanism.

A central point to be considered in models applied to the renewable energy forecasting problems studied in this thesis is their capacity to deal with a high volume of non-stationary spatio-temporal data, arranged as multivariate time series. In this regard, e-MVFTS proposes some contributions:

- **Adaptation** The ability to adapt to data online as it is received. In this case, these are adjustments that involve not only its parameters, but also other internal structures, such as sets and rules.

- **Scalability.** The application of a clustering technique to provide a new representation of the data, so that each multivariate sample is internally represented by a cluster, which in turn is used as univariate information in an FTS model. Such translation can contribute to the scalability of the model.

- **Robustness.** Also regarding clustering, the use of an evolving technique, which, as noted in the experiments presented in chapter 6, presents good robustness of hyperparameters, which require few adjustments when applied to different data sets. Within a context in which there is a need to process large volumes of data, avoiding generally computationally expensive tasks such as hyperparameter tuning, may bring a considerable performance gain.

The concepts that underlie the proposed model were detailed throughout this thesis. The next sections of this chapter details the proposed model, its training, adaptation and forecasting procedures. Next, experiments performed on the model are demonstrated, whose performance was observed in time series forecasting problems with concept drift events, as well as to the solar and wind energy data sets studied in this thesis.

7.1 e-MVFTS method

From the integration between MicroTEDAclus and High Order FTS, e-MVFTS presents an extension of the FTS models, in which an adaptive clustering mechanism provides a more scalable solution to treat multivariate data and allows incremental adjustments in its universe of discourse, fuzzy sets and rules. Innitially, data points from multivariate time series are organized based on the temporal index, which is called in this work the **embedding** procedure. Figure 33 depicts this process. For a given time step t, the values observed in each time series are grouped, to generate a multidimensional sample. These samples compose the data set used by e-MVFTS. Figure 34 illustrates the process of transforming the data that are then used by the FTS model. It is possible to note that, for each micro-cluster created by MicroTEDAclus, a corresponding fuzzy set is defined based on its parameters. In this way, the data samples are fuzzified so that Fuzzy Logical Relationship Groups can be generated, following the procedures described in Section 4.2. This set of rules is then used for forecasting, similarly to an High Order FTS model.

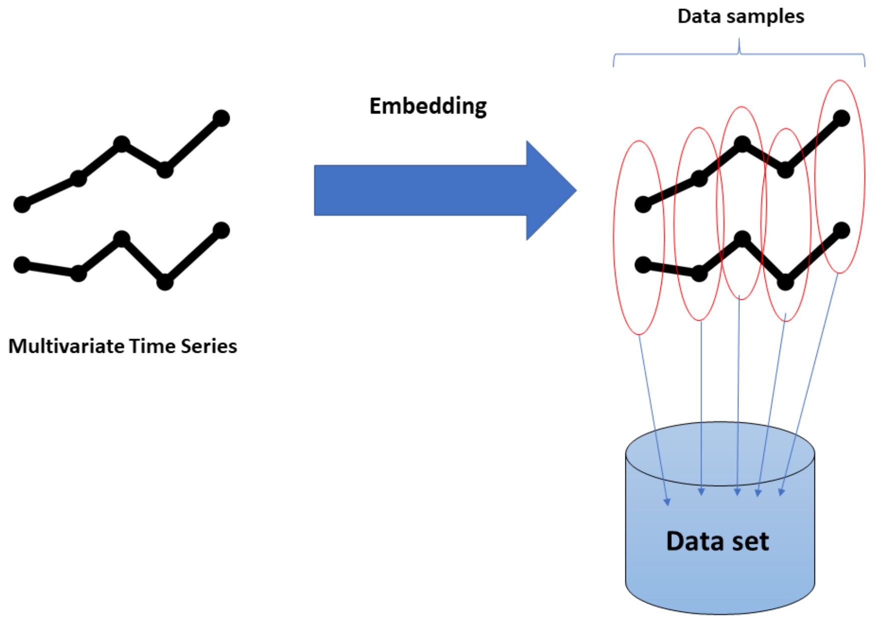


Figure 33 – Illustration of e-MVFTS embedding procedure

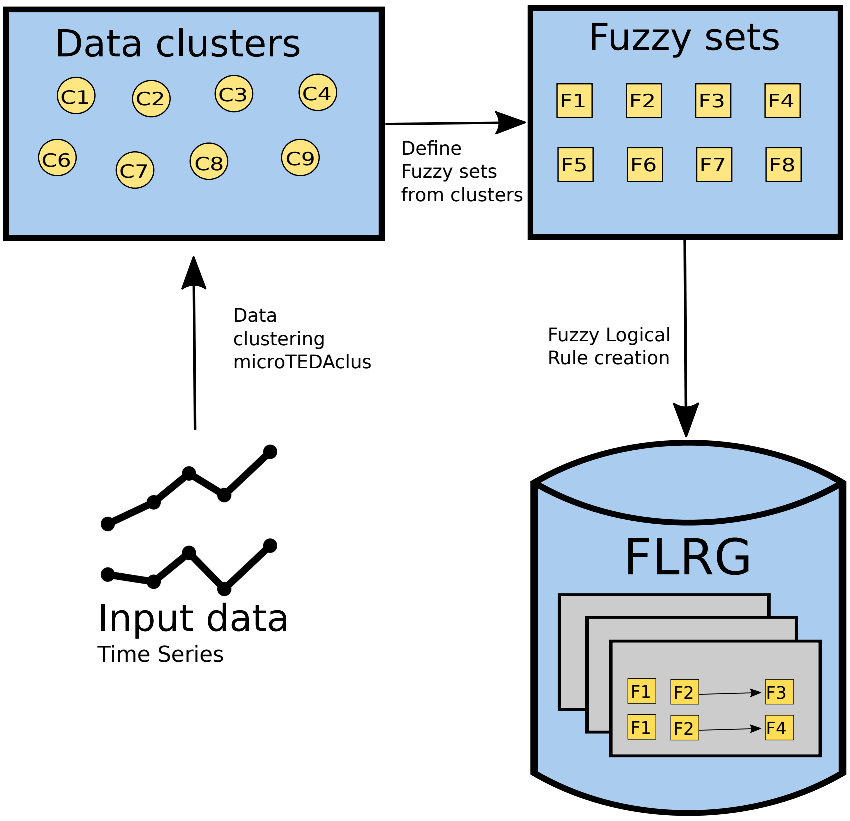


Figure 34 – Illustration of e-MVFTS main flow

In the next sections, the training, updating and forecasting procedures are described in detail.

7.1.1 Training Procedure

Given a d-dimensional time series Y, where each sample is a value $y(t) \in \mathbb{R}^d$:

Step 1 - Data Clustering

MicroTEDAclus is applied to data clustering. For each data sample, micro-clusters are created or updated, according to the procedure described in section 6.2.1. Macro-cluster update step defined in section 6.2.2 is not performed, since only micro-clusters are used in e-MVFTS. After the data is processed, a set $\mathfrak{C}$ of micro-clusters c_i, where $i = 1 \dots M$, is created.

It is important to note that this step replaces the Universe of Discourse partitioning process, present in conventional FTS models. Therefore, the model does not require an initial representation of the Universe of Discourse, with lower and upper bounds, nor a partitioning scheme. This process occurs incrementally as the data is processed by the algorithm.

Step 2 - Fuzzy Sets Definition

Each micro-cluster c_i is used to define a multi-dimensional fuzzy set $f_i \in \mathfrak{F}$, in a one-to-one relationship. Its membership function $\mu_{f_i}()$ is represented by the normalized typicality function $\gamma()$, described in section 6.1, equation 6.8, from its corresponding micro-cluster c_i. Thus, given a sample $y(t)$, its membership to a fuzzy set f_i can be calculated as follows:

$$\mu_{f_i}(y(t)) = \gamma(y(t)) \tag{7.1}$$

Step 3 - Time Series Fuzzification

Since the membership function based on typicality can return nonzero values to all the fuzzy sets, a **cumulative membership threshold** is applied to limit the number of candidates. Given a model with fuzzy sets $f_i \in \mathfrak{F}$, a set $FC_{yt} \subset \mathfrak{F}$ of candidates for a crisp sample $y(t)$ is $FC_{yt} = f_j | \sum_j \mu_{f_j} \geq \alpha$ and $|FC_{yt}|$ is minimal. α is the cumulative membership threshold and its *default* value is set to **0.6**.

Step 4 - Temporal Patterns Extraction

In this step, a High Order FTS is formulated from fuzzy temporal patterns. The fuzzy temporal patterns have the format $A_p \rightarrow A_c$, where:

The precedent (or Left Hand Side - LHS), denoted by A_p, contains fuzzy values of lags. For instance, given a FTS of order $\Omega = 2$, each temporal pattern A_p has two elements, corresponding to fuzzy values of $y(t)$ and $y(t-1)$.

Analogously to precedent, the consequent A_c (or Right Hand Side - RHS) represents the corresponding fuzzy values of $y(t+1)$.

Each temporal pattern consists of a combination of fuzzy values between the fuzzy candidates of each element.

Step 5 - Rule base generation

In this step, temporal patterns with the same precedent are grouped creating FLRGs with the format $A_p \rightarrow A_a, A_b, \ldots$. The rules can be understood as the set of possibilities which may happen at future time t (the consequent) when a pattern A_p is identified.

7.1.2 Model Update Procedure

The update procedure occurs after a data window is processed. Thus, given a window of size w the following steps occur:

Step 1 - Micro-clusters update

Each data sample in the window is processed by MicroTEDAclus as occurs in the training procedure. During this step, each micro-cluster created or updated is added to a set of changed micro-clusters $\mathfrak{Ch}$.

Step 2 - Fuzzy sets update

The fuzzy sets defined in $\mathfrak{F}$ are updated according to the changed micro-clusters c_j in $\mathfrak{Ch}$. If c_j is a new micro-cluster, a new corresponding fuzzy set is created and added to $\mathfrak{F}$. If c_j is an updated micro-cluster, its new parameters are used to adjust the membership function and the midpoint of the corresponding fuzzy set f_j.

Step 3 - Rule base update

Temporal patterns are formed using the data window. Patterns whose precedent already exists into the rule base update their corresponding consequent and new temporal patterns are grouped according their precedents creating a new FLRG.

7.1.3 Forecasting Procedure

For a model with order $\Omega = p$ the following steps are taken to forecasting:

Step 1 - Input Value Fuzzification

Each element in a sequence of crisp data points $y(t), y(t-1), \ldots, y(t-(p-1))$ is assigned to fuzzy sets according to their membership grades and the cumulative membership threshold, as discussed in the training procedure, section 7.1.1, step **2**.

Step 2 - Fuzzy Rule matching

The selected fuzzy sets are then combined to create a set $\mathfrak{P}$ of A_p (precedent) patterns. The precedents $A_{pj} \in \mathfrak{P}$ are searched in the rule base, composing a rule set $\mathfrak{S}$ with rules $A_{pj} \rightarrow A_{cj}$.

Step 3 - Defuzzification

The forecast value $\hat{y}_{t+1}$ is calculated according to Equation (7.2) as the weighted sum of the rule midpoints, mp, by their membership grades μ_j for each selected rule j:

$$\hat{y}_{t+1} = \sum_{A_{pj} \rightarrow A_{cj} \in \mathfrak{S}} \mu_{A_{pj}}(y_t) \cdot mp(A_{cj}) \tag{7.2}$$

where

$$\mu_{A_{pj}}(y_t) = min(\{\mu_{f_i} | f_i \in A_{pj}\}) \tag{7.3}$$

and

$$mp(A_{cj}) = \frac{\sum_{f_i \in A_{cj}} mp(f_i)}{|A_{cj}|} \tag{7.4}$$

7.1.4 Computational Cost

The computational cost of the training procedure is concentrated in the data clustering and rule base creation steps. Data clustering is dimensionally dependent during the calculation of variance and eccentricity. For samples with dimension d, both calculations have complexity of $O(d)$. Therefore, given that n is the number of samples and k is the number of micro- clusters, time complexity for this step is $O(ndk)$. Each micro-cluster represents one fuzzy sets, then for k sets and order Ω, the fuzzification step has a cost of $O(nk)$. Using a binary search tree structure to organize the k fuzzy sets, time complexity can be optimized. A search among them decreases from $O(k)$ to $O(logk)$, which impacts on the rule base generation, whose process costs $O(n(logk)^{\Omega})$.

Model update has a computational cost analogous to some steps in training procedure, but proportionally reduced. For an update window with size w, the data clustering step has time complexity of $O(wdk)$. After the clustering, considering c the number of updated micro-clusters, rule base update has a cost of $O(w(logc)^{\Omega})$.

In the forecasting procedure a fuzzification step is taken, followed by rule matching. For a single input sample fuzzification is $O(k)$ and the rule matching costs $O(\Omega(logk)^{\Omega})$.

The time complexity presented by the algorithm during training and forecasting is comparable to that of a classic High Order FTS model, with the addition of the data clustering step, which deals with the dimensionality of the problem. The overall performance improvement is focused in the process of model update, and depends on the number of changed micro-clusters during the task. The number of new clusters created at each update remains at lower levels, but without reaching the opposite extreme case, in which large micro-clusters are created, resulting in few fuzzy sets and a model with low capacity to represent the data. Such trade-off is centered in the hyperparameter r_0, discussed in chapter 6, section 6.2.

7.1.5 Method Discussion

The details of the e-MVFTS procedures, done in the previous sections of this chapter, highlight the differentials of the use of an evolving clustering algorithm for the representation of the data used in a forecasting model. In addition to the already discussed conversion of multivariate data to univariate FTS rules, MicroTEDAclus's recursive formulations also help in the good ability of e-MVFTS to deal with large volumes of data. Such formulations, responsible for updating information that characterize micro-clusters, such as typicality and eccentricity, besides presenting a reduced computational cost, allow the algorithm to be able to deal with big data without increasing its processing load proportionally.

Another point is the inclusion of an adaptation mechanism as a strategy to deal with data variations. As already discussed in the chapter 5, adapting the model during its execution tends to be a more interesting alternative from the point of view of computational cost than applying new training steps. Such adaptation in e-mVFTS has the ability to adjust an FTS model both in the scope of its fuzzy sets and in the partition of its UoD, which is done incrementally and even allowing the inclusion of new partitions, from the creation of new fuzzy sets during runtime. This mechanism, therefore, presents an alternative for the extension and flexibility of FTS models.

7.2 Experiments

The experiments aim to evaluate the performance of e-MVFTS when applied to time series with concept drift events and problems of solar and wind energy forecasting. For the first aspect, artificially generated data sets were used to observe the behavior of the algorithm in the face of different data distribution changes. Additionally, the data sets analyzed in the chapter 2, used in the experiments of the NSFTS model, in chapter 5, were chosen for this experiments. However, the evaluation of e-MVFTS was for a spatio-temporal problem and therefore multivariate time series from the data sets were analyzed. The details of the experiments and their results are described below.

7.2.1 Data sets

7.2.1.1 Concept Drift Data sets

The model was first evaluated on its ability to handle different types of concept drift events. Such events, discussed in Gama et al. [2014], were simulated in 8 univariate time series containing 2000 samples each. In some data sets, more than one event were combined. The time series and their respective concept drift events are illustrated in Figure 35.

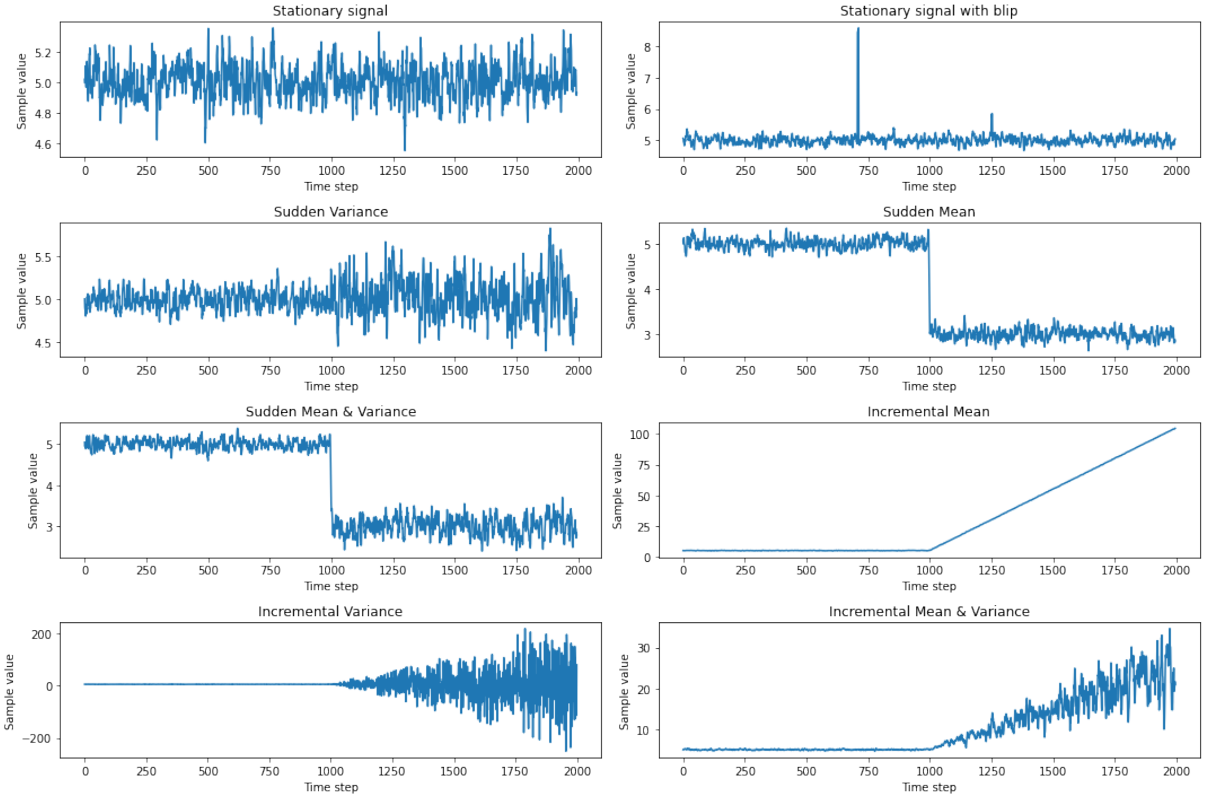


Figure 35 – Time Series artificially generated with concept drift events. The x-axis denotes the time step when each data point was sampled and the y-axis represents their corresponding values.

7.2.1.2 Renewable Energy Data sets

The solar energy data set is detailed in Table 10. The wind energy data set is described in Table 11. The time intervals of both data sets are the same used in the NSFTS method experiments, in chapter 5. The difference is the number of time series used in each database, which correspond to the sites of the observed energy system. The solar energy database originally had 17 sites, but was reduced to 10. The criterion for reducing the sites was the SSA residual correlation value with the DH3 site. Sites with values above 85 % were maintained. DH3 was chosen as a reference because it is a more central site in the map, as described in the Figure 8, chapter 2. The SSA residual correlation is also discussed in chapter 2, with correlation map presented in Figure 10. In the wind energy data set, all available sites were used.

7.2.2 Experiments Design

For the concept drift experiments, e-MVFTS was compared to NSFTS, discussed in chapter 5 and Fuzzy-set-Based evolving Modeling (FBeM) an evolving fuzzy model proposed by Leite et al. 2012. Both benchmark methods were chosen because also provides adaptive mechanisms and are based on fuzzy concepts. The objective is to compare

Table 10 – Solar Energy Data set

NREL Oahu Solar Measurement Grid	
Variable	Global Horizontal Irradiance (GHI)
Frequency	15 minutes
Start Date	2010-06-01
End Date	2011-07-31
Number of sites	10
Samples per site	38766

Table 11 – Wind Energy Data set

GEFCom 2012 Data set	
Variable	Wind Power
Frequency	Hourly
Start Date	2009-07-01
End Date	2010-08-30
Number of sites	7
Samples per site	8761

e-MVFTS to other fuzzy-based models in terms of adaptability to drift events.

All the concept drift data sets in this experiments contain 2000 samples, in which the drift events were triggered from the thousandth sample. The experiments were run under the prequential (First-Test-Then-Train) scenario, illustrated in Figure 36. The first 1000 samples were used for eventual adaptations of the models used in the experiment, the forecast error not being measured. From the triggering of the event, the error of each method is calculated. The objective is to observe how each model adapts to the event inserted during the processing of the dataset. It is worth noting that the adaptation step is not necessary for e-MVFTS, since the model can evolve from scratch.

The validation method chosen for the experiments using renewable energy data sets was the rolling window, the same used in the experiments in the chapter 5. The window set up was 4 weeks for training and one week for testing each iteration. In addition, the first month of each data set was used for parameter tuning before, and is also supported by the Hyperopt tool.

From the second iteration, e-MVFTS used only its model update mechanism, described in section 7.1.2. For each iteration, the data previously used for testing were used for this procedure. The other benchmark algorithms followed the routine proposed by the rolling window validation, of performing training and testing sequentially at each iteration.

The models chosen for comparison with e-MVFTS are all multivariate, in order to be able to process the same data sets. The statistical model Vector Autoregressive (VAR)

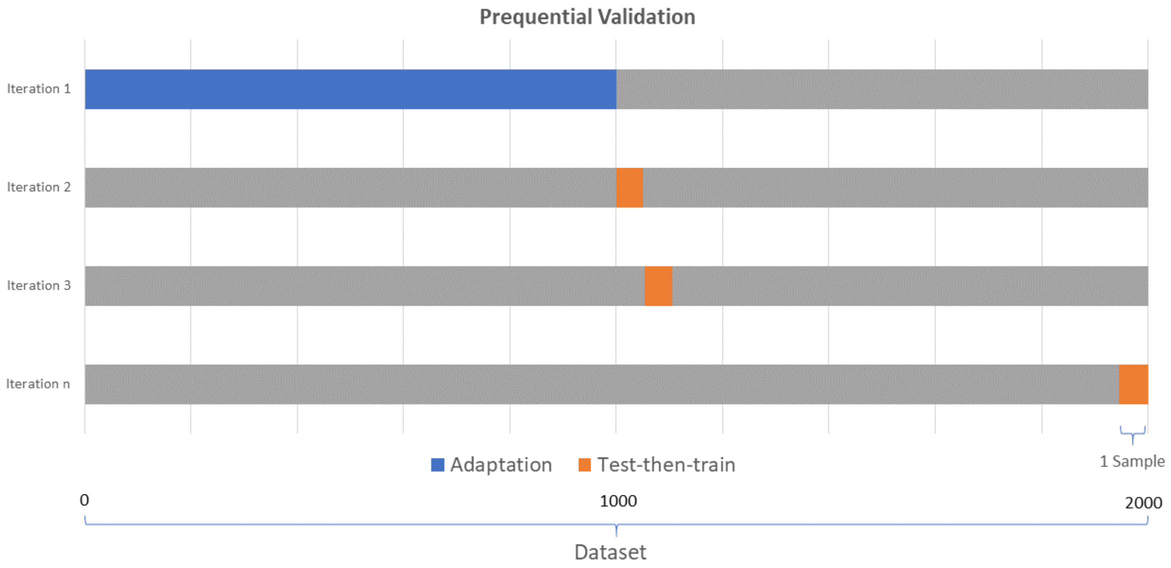


Figure 36 – Prequential Validation

Table 12 – List of hyper-parameters for the methods used in the solar and wind forecasting.

Method	**Parameter**	**Values**
All	Lags	1-4
MLP	Layers	2-3
	Neurons per layer	8, 16, 64, 128, 256, 512
	Dropout Regularization	0-1
	Batch size	28, 64, 128, 256, 512
$\mathcal{FIG}$-FTS	KNN	1-5
	α-cut	0-0.3
	Fuzzy sets	100-250

was used. The Multi Layer Perceptron (MLP) was also applied to the problem, but it was modeled in order to receive as input the multivariate lags (high order) and return as forecast the values of all the variables analyzed. The last model chosen as benchmark is a multivariate version of FTS, the $\mathcal{FIG}$-FTS, described in chapter 4, section 4.4. The ranges of values used during the hyperparameter tuning step are described in Table 12.

The hyperparameters of e-MVFTS, namely variance limit r_0 and cumulative distance α, were set to their default values, **0.001** and **0.6**, respectively, during all the experiments.

7.2.3 Metrics

For these experiments, the metrics RMSE, nRMSE and SMAPE were also applied, according to the equations below.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}{n}} \tag{7.5}$$

$$nRMSE = \frac{RMSE}{\frac{\sum_{t=i}^{i+z} Y_t}{z}} \tag{7.6}$$

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|Y_t - \hat{Y}_t|}{|\hat{Y}_t| + |Y_t|} \tag{7.7}$$

where Y_i are the measured observations at the point of interest at time t, $\hat{Y}_t$ denotes the predicted values and n is the length of dataset. Even though the Y_i values are multivariate in this experiment, the final calculated value is univariate. In both metrics, conversion occurs at the time the data is averaged. As there are no variables that represent different quantities or units of measure, the values are calculated among all the variables used.

After calculating the forecasting errors, the **Diebold-Mariano** test was used to assess the difference in the accuracy between e-MVFTS and each benchmark model. The null hypothesis for this test is that there is no difference. If the p-value returned in the test is less than a significance level, it indicates that there is a significant difference between the results, therefore rejecting the null hypothesis. For the experiments, a significance level of **0.01** was used.

7.2.4 Forecasting Results

The average RMSE and SMAPE for the concept drift datasets are shown in Table 13. The errors are calculated for a one step ahead forecasting $(t + 1)$, which takes as input the observations in t and $t - 1$ **(lag order 2)**. The process was run 10 times for each dataset.

Table 13 – Forecasting - Concept drift datasets

Data set	**NSFTS**		**FBeM**		**e-MVFTS**	
	RMSE	**SMAPE**	**RMSE**	**SMAPE**	**RMSE**	**SMAPE**
Stationary signal	0.1539 ± 0.0103	13.7053 ± 1.4931	0.1542 ± 0.0101	13.7120 ± 1.4426	0.1410 ± 0.0088	12.8664 ± 1.5018
Stationary signal with blip	0.0554 ± 0.0303	17.3365 ± 4.3588	0.0610 ± 0.0487	14.0222 ± 2.6778	0.0454 ± 0.0297	12.9779 ± 2.0690
Incremental Mean	0.0032 ± 0.0001	0.7025 ± 0.0667	0.0032 ± 0.0000	0.8882 ± 0.0342	0.0028 ± 0.0003	1.4462 ± 0.3211
Incremental Mean and Variance	0.0967 ± 0.0067	9.8406 ± 0.4281	0.0934 ± 0.0060	9.7133 ± 0.3889	0.0782 ± 0.0057	11.7267 ± 1.9077
Incremental Variance	0.1485 ± 0.0176	10.2827 ± 1.7948	0.1321 ± 0.0147	9.5040 ± 1.6785	0.1658 ± 0.0190	12.1366 ± 2.2337
Sudden Mean	0.3813 ± 0.0060	56.7437 ± 2.2169	0.0468 ± 0.0023	13.0528 ± 0.9806	0.0431 ± 0.0040	12.0293 ± 1.0649
Sudden Mean and Variance	0.3531 ± 0.0127	43.4510 ± 3.3795	0.0759 ± 0.0047	14.6874 ± 2.3680	0.1005 ± 0.0705	14.8817 ± 3.8664
Sudden Variance	0.1650 ± 0.0098	14.2679 ± 1.1008	0.1653 ± 0.0109	14.5184 ± 1.1335	0.1457 ± 0.0079	13.1385 ± 1.1568

The experiments involving concept drift data sets presented comparable results for all the forecasting methods. Table 14 shows the results of the Diebold-Mariano test for the errors calculated using RMSE. The p-values lower than 0.01 indicates significant differences between e-MVFTS and the compared models. E-MVFTS presented better results mostly when dealing with incremental variations. It suggests that the model update procedure, which incrementally updates the mean and variance of the micro-clusters, presents a dynamic capable of adapting to such changes appropriately. In addition, good results of the method were observed at blip events. This can be explained by the applicability of TEDA framework in outlier detection problems, as discussed by Angelov 2014b.

Table 14 – Forecasting - Concept drift datasets - p-values for Diebold-Mariano Test

Data set	**e-MVFTS vs NSFTS**	**e-MVFTS vs FBeM**
Stationary signal	0.00∗	0.00∗
Stationary signal with blip	0.15	0.00∗
Incremental Mean	0.11	0.10
Incremental Mean and Variance	0.00∗	0.00∗
Incremental Variance	0.00∗	0.00∗
Sudden Mean	0.00∗	0.23
Sudden Mean and Variance	0.00∗	0.04
Sudden Variance	0.01	0.01

*Indicates significant difference

The error for the renewable energy data sets were reported as the average of the rolling window evaluations. For each window used for testing, the error metrics are calculated. Figures 41 and 42 illustrate the nRMSE observed at each validation window. From these illustrations of the temporal evolution of the models, it is possible to notice how each method adapts to the changes occurred in the data. e-MVFTS managed to maintain a consistent adaptation in the face of concept drift events that affected the performance of all models, managing in some cases to maintain a better performance than the other models after adaptation.

The summarized RMSE and SMAPE results for the experiments using the solar energy data set are presented in Figures 37 and 38, while wind energy experiments are depicted in 39 and 40. Table 15 summarizes the results.

For the solar energy experiment, e-MVFTS presented the best results when SMAPE metric was applied, while VAR was the best algorithm for RMSE. The difference in the results observed for the two metrics can be understood by some of their characteristics. The RMSE tends to be more impacted by residues of greater magnitude, as it is based on their standard deviation. On the other hand, because it is based on the absolute average of the errors, SMAPE gives similar importance to all calculated residues. Therefore, the results of this experiment suggest that, in general, residues of greater magnitude were less observed in the VAR algorithm, which may explain its slightly better performance for RMSE. However, their low occurrence throughout the experiment contributed to the lowest SMAPE in the results for e-MVFTS. For the wind energy data set, VAR presented the best forecasting results, while the other algorithms had comparable results. For these results, it is worth highlighting the greater error ranges for both metrics. This indicates that even the best performing methods had difficulties in forecasting, given the unstable patterns presented by the wind power time series. Table 16 shows the results for the Diebold-Mariano test applied to both renewable energy data sets. The p-values indicate significant accuracy differences between e-MVFTS and the benchmark forecasting models for all experiments.

After processing the 4 weeks of data used for initial training, e-MVFTS achieved a number of fuzzy sets and rules that remained stable throughout the experiments. For the solar energy data set, **160** fuzzy sets and **448** FLRGs were generated, and for the wind energy data set, the model consisted of **230** fuzzy sets and **261** FLRGs. It is worth noting that $\mathcal{FIG}$-FTS was configured, through hyperparameter tuning, with similar values. **150** and **200** fuzzy sets were defined for the solar and wind energy experiments, respectively. One of the central problems in the FTS models is the definition and partitioning of its Universe of Discourse. And these values suggest that e-MVFTS is capable of achieving an adequate configuration without the cost of a tuning process.

The models were also evaluated for multi-step ahead forecasting. For all models, the multi-step forecasting mechanism was recursive, in which the output of a 1-step ahead forecast is used as input to the model in order to generate the forecast for the next horizon. For both data sets, the horizons of 1, 4, and 8 steps ahead were observed. Figures 43 and 44 show the values measured using nRMSE. In Figures 45, 46, 47, 48, 49, 50, the residues presented in the experiments are shown. Residual analysis indicates that, for shorter forecast horizons, the residues present zero average and are normally distributed, suggesting that they are white noise. From larger horizons, such behavior starts to change for some models, suggesting that all models have difficulty forecasting such time series. The observation of these results suggests a great variability in the time series analyzed in this work, which makes it difficult to carry out a consistent multi-step forecast. It is, therefore, another aspect that reinforces the need to develop models with mechanisms that allow an adaptation to the changes that occur in these time series.

7.3 Discussion

The forecasting model presented in this chapter proposes some extensions to FTS-based models. First, its ability to dynamically adapt to data provides a relevant alternative of flexibility for the original model structure and its rules at a reduced computational cost, since it does not require an entire model reconstruction process. In addition, the

Table 15 – Forecasting - Renewable Energy data sets

Data set	Model	RMSE	SMAPE
Solar Energy	e-MVFTS	0.0854 ± 0.0192	21.2244 ± 3.9193
	VAR	0.0796 ± 0.0173	25.1752 ± 3.2458
	MLP	0.1099 ± 0.0489	27.1560 ± 5.1009
	FIG-FTS	0.0862 ± 0.0208	26.6996 ± 4.0626
Wind Energy	e-MVFTS	0.1061 ± 0.0201	33.7018 ± 11.7134
	VAR	0.0799 ± 0.0169	28.4566 ± 8.9350
	MLP	0.1401 ± 0.0335	35.4485 ± 8.5438
	FIG-FTS	0.1179 ± 0.0257	32.2817 ± 6.4839

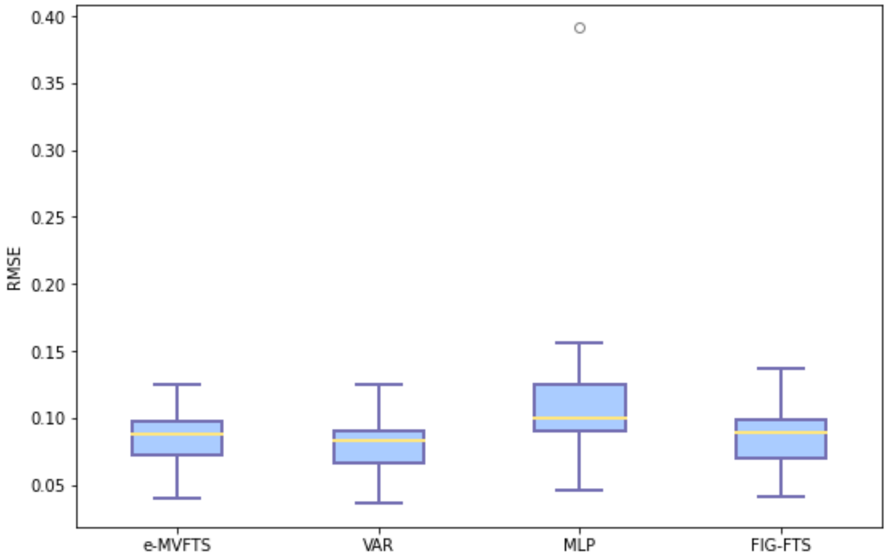


Figure 37 – Solar Energy data set - RMSE

Table 16 – Forecasting - Renewable Energy datasets - p-values for Diebold-Mariano Test

Data set	e-MVFTS vs VAR	e-MVFTS vs MLP	e-MVFTS vs FIG-FTS
Solar Energy	0.00*	0.00*	0.00*
Wind Energy	0.00*	0.00*	0.00*

*Indicates significant difference

data representation format, which uses an integrated clustering algorithm, allows greater scalability when representing data of greater dimensionality.

The model was applied to different concept drift events to assess its robustness to changes in data. In the observed experiments, e-MVFTS showed good results, especially in incremental changes in mean and variance. These characteristics enable it to be applied and analyzed within the context of renewable energy forecasting. Therefore, the model was applied to the solar and wind energy data sets used in this thesis, being compared with different approaches that deal with multivariate time series. e-MVFTS presented a good performance in both data sets, being the best overall result for the solar energy

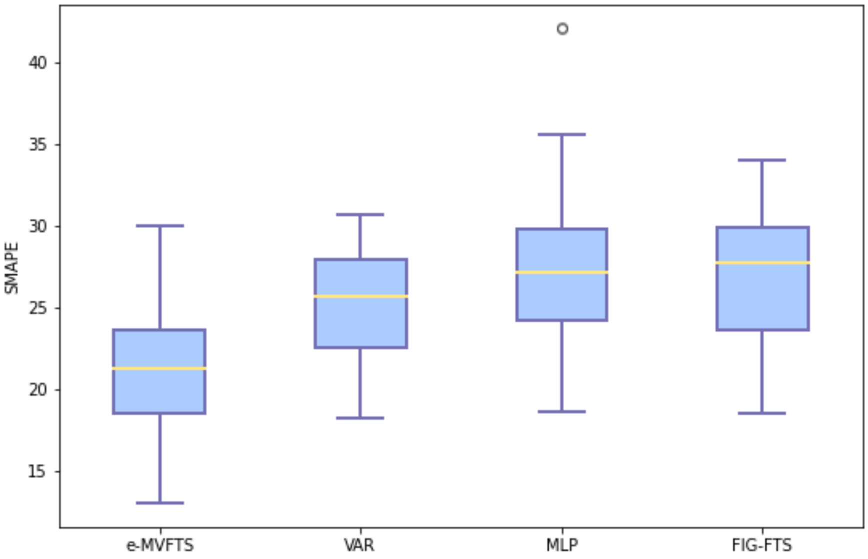


Figure 38 – Solar Energy data set - SMAPE

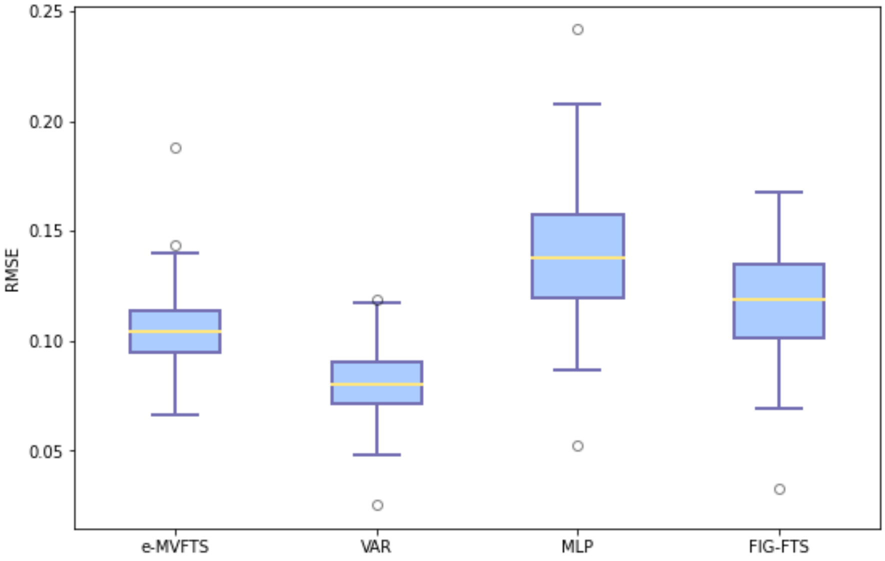


Figure 39 – Wind Energy data set - RMSE

data set. The results suggest that it is a good alternative to the problem studied in this thesis. As it is a new algorithm, the application of e-MVFTS to other problems involving non-stationary multivariate data can be an interesting activity to have a more complete assessment of its performance and eventually point out improvements in its formulation.

E-MVFTS integrates MicroTEDAclus with essential elements of an FTS model, such as fuzzy sets and membership functions. An advantage is that eventual integrations with more advanced FTS models tend to be straightforward. Investigating e-MVFTS expansions in this direction may be a direction for future work. Another point that can be observed is its behavior in the face of data that changes over time its data distribution in a more permanent way. In this case, there would be no need to maintain in your

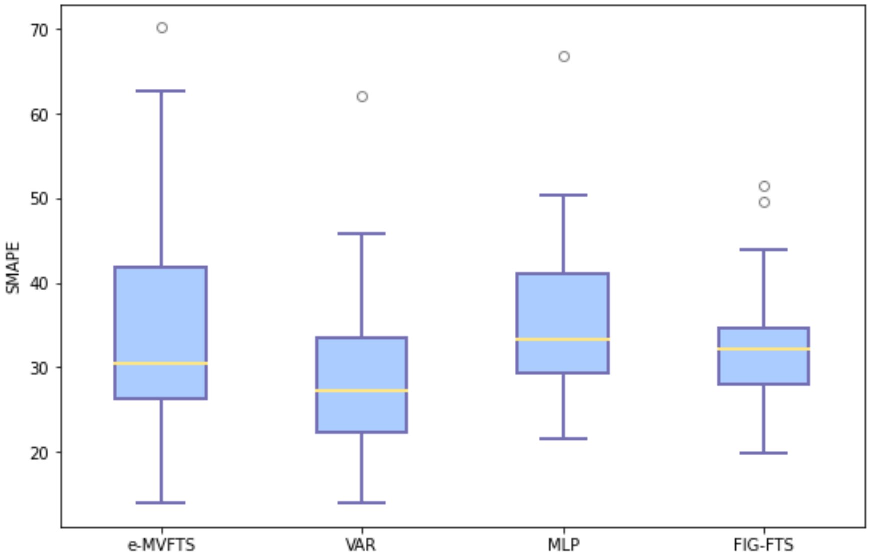


Figure 40 – Wind Energy data set - SMAPE

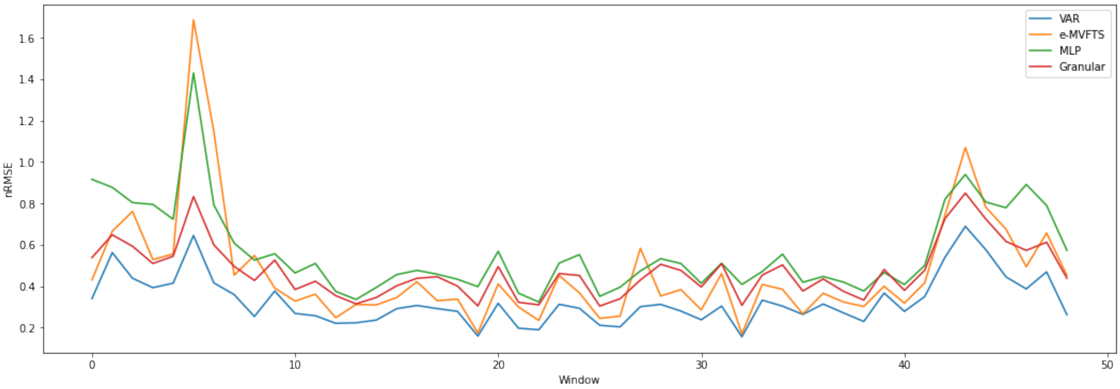


Figure 41 – Solar Energy data set - Temporal evolution of nRMSE

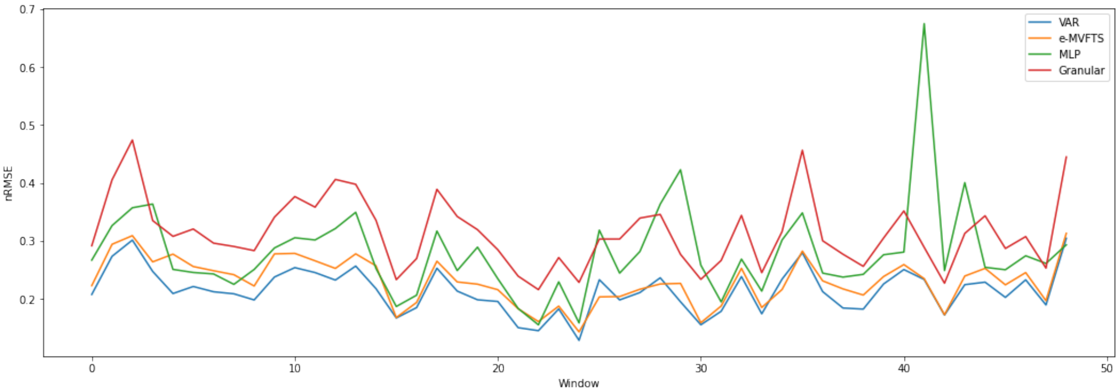


Figure 42 – Wind Energy data set - Temporal evolution of nRMSE

model representations and rules referring to older standards that are no longer accessed. Alternatives to deal with such situations tend to improve mainly the computational performance of the model, since its set of rules may be smaller. Techniques to deal with such situations, such as pruning mechanisms, can provide an important improvement to

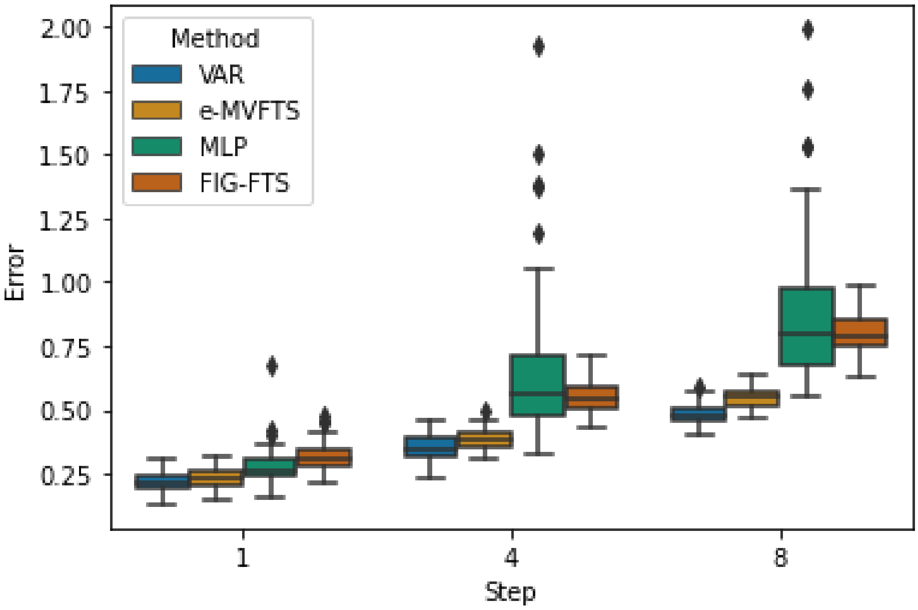


Figure 43 – Solar Energy data set - Grouped Boxplots for Multistep Forecating - nRMSE

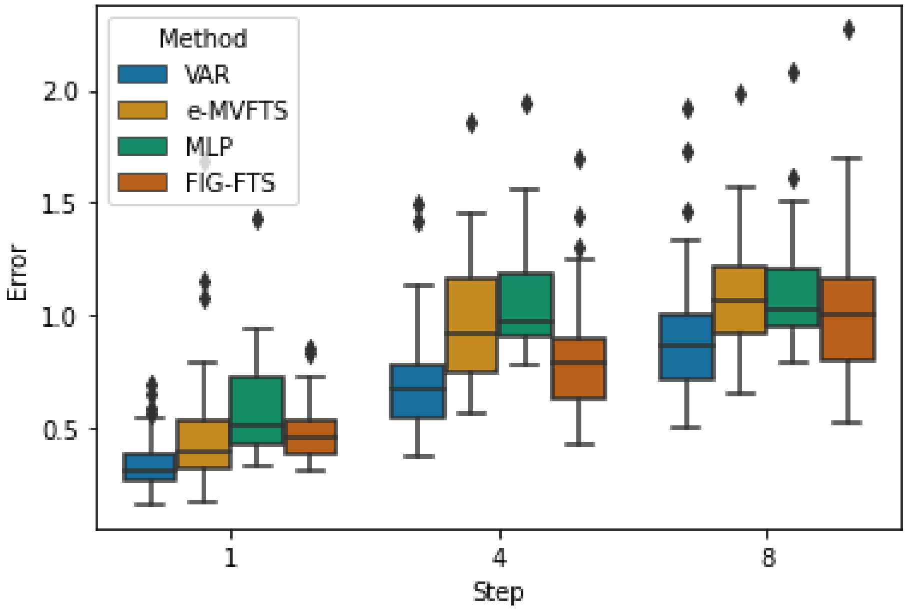
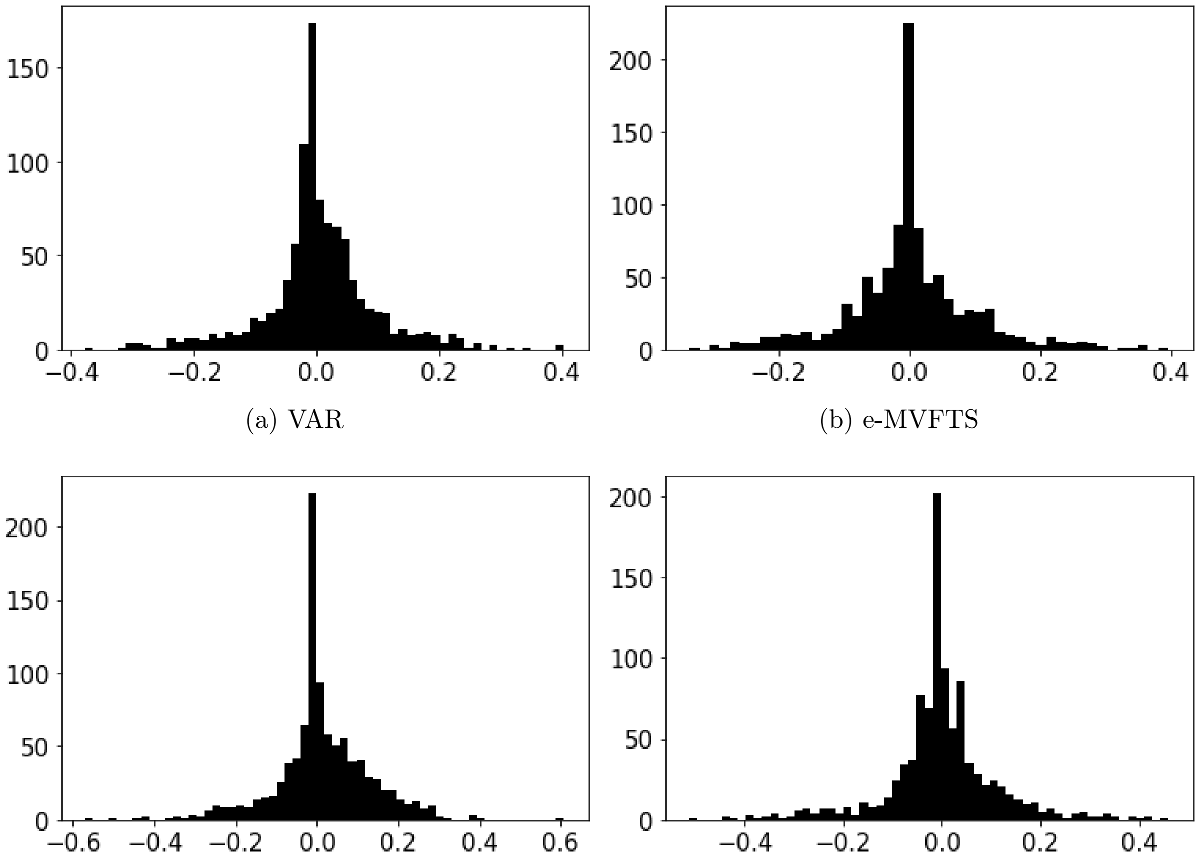


Figure 44 – Wind Energy data set - Grouped Boxplots for Multistep Forecating - nRMSE

the model.



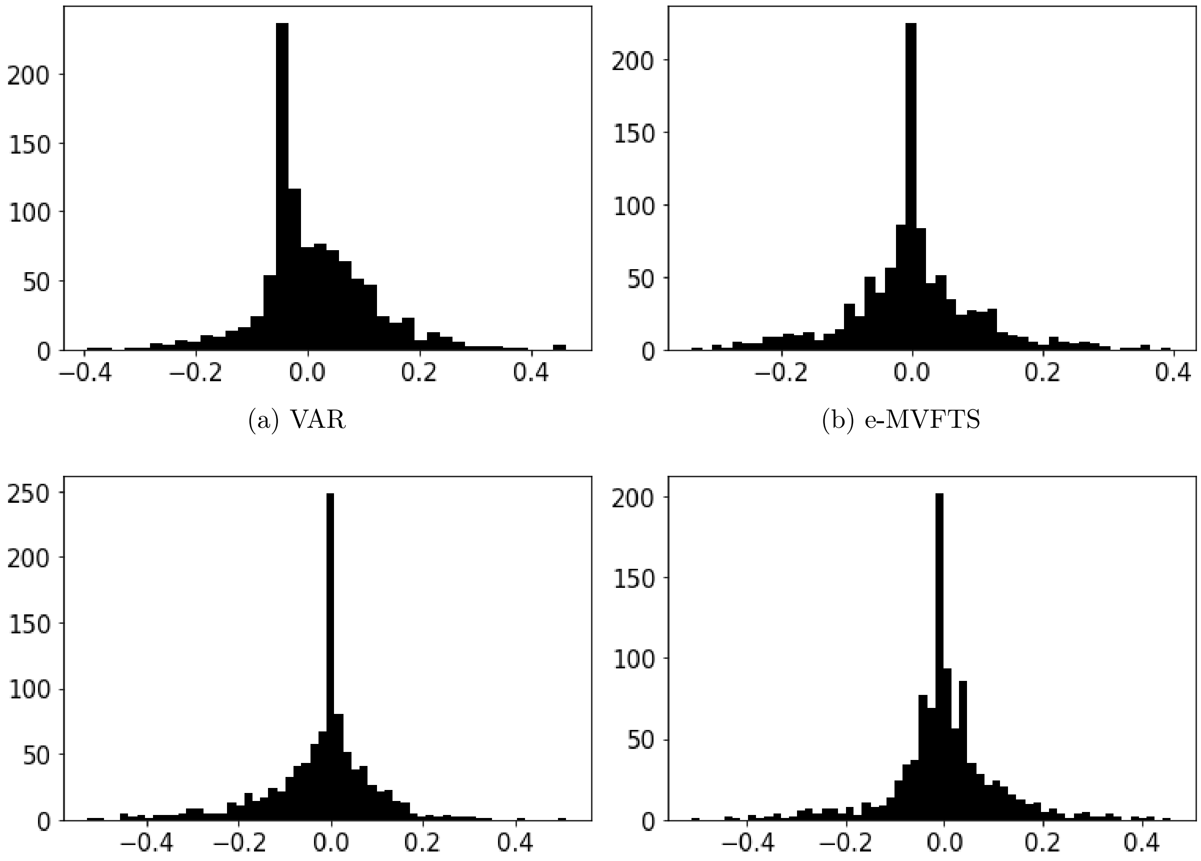
(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 45 – Solar Energy Dataset - Residual Analysis - 1 step ahead Forecasting



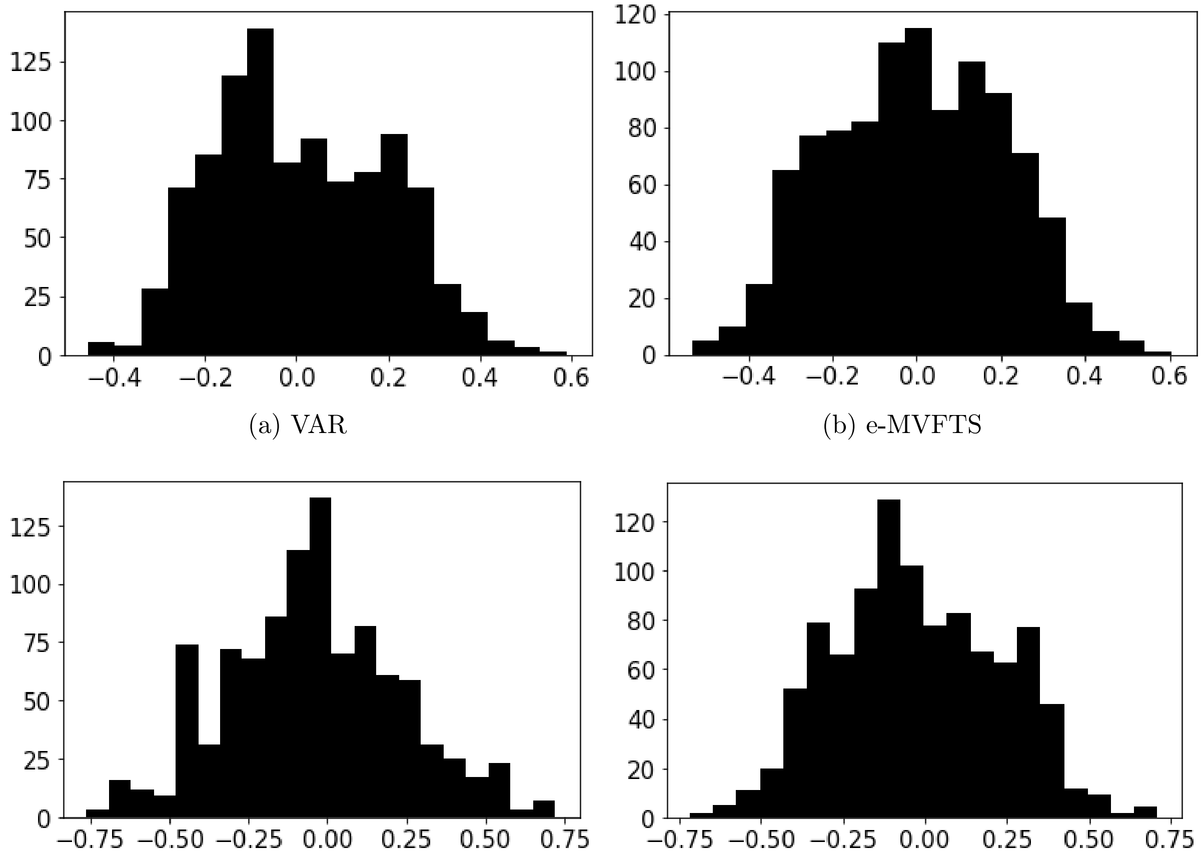
(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 46 – Solar Energy Dataset - Residual Analysis - 4 steps ahead Forecasting



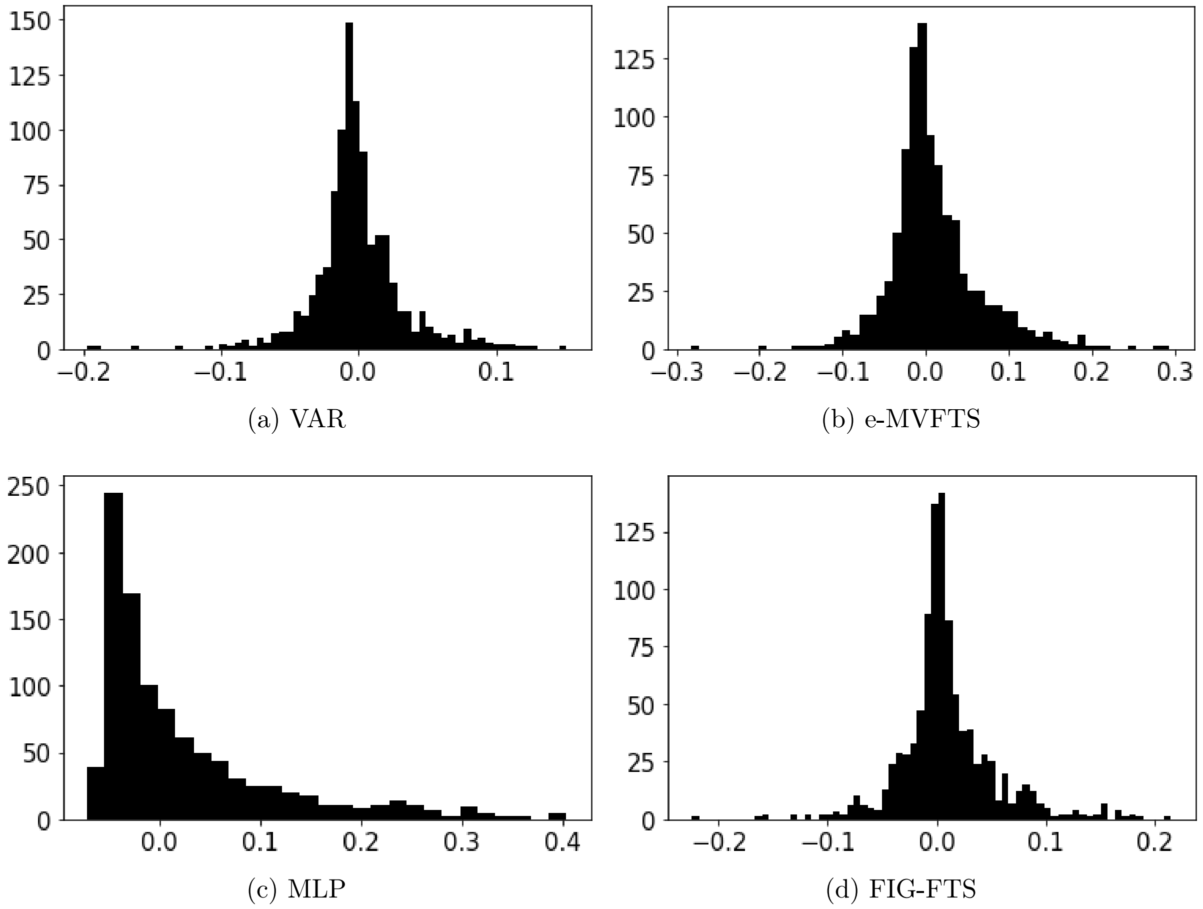
(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 47 – Solar Energy Dataset - Residual Analysis - 8 step ahead Forecasting



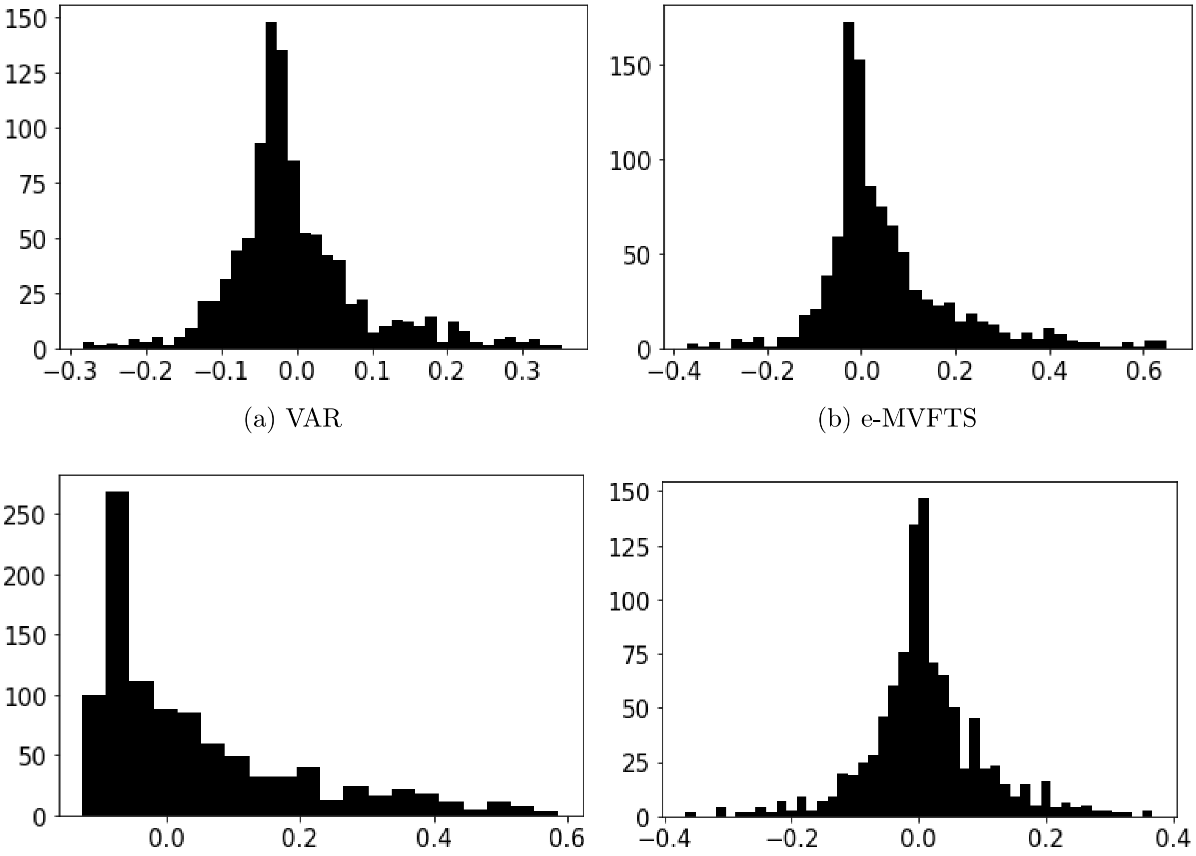
(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 48 – Wind Energy Dataset - Residual Analysis - 1 step ahead Forecasting



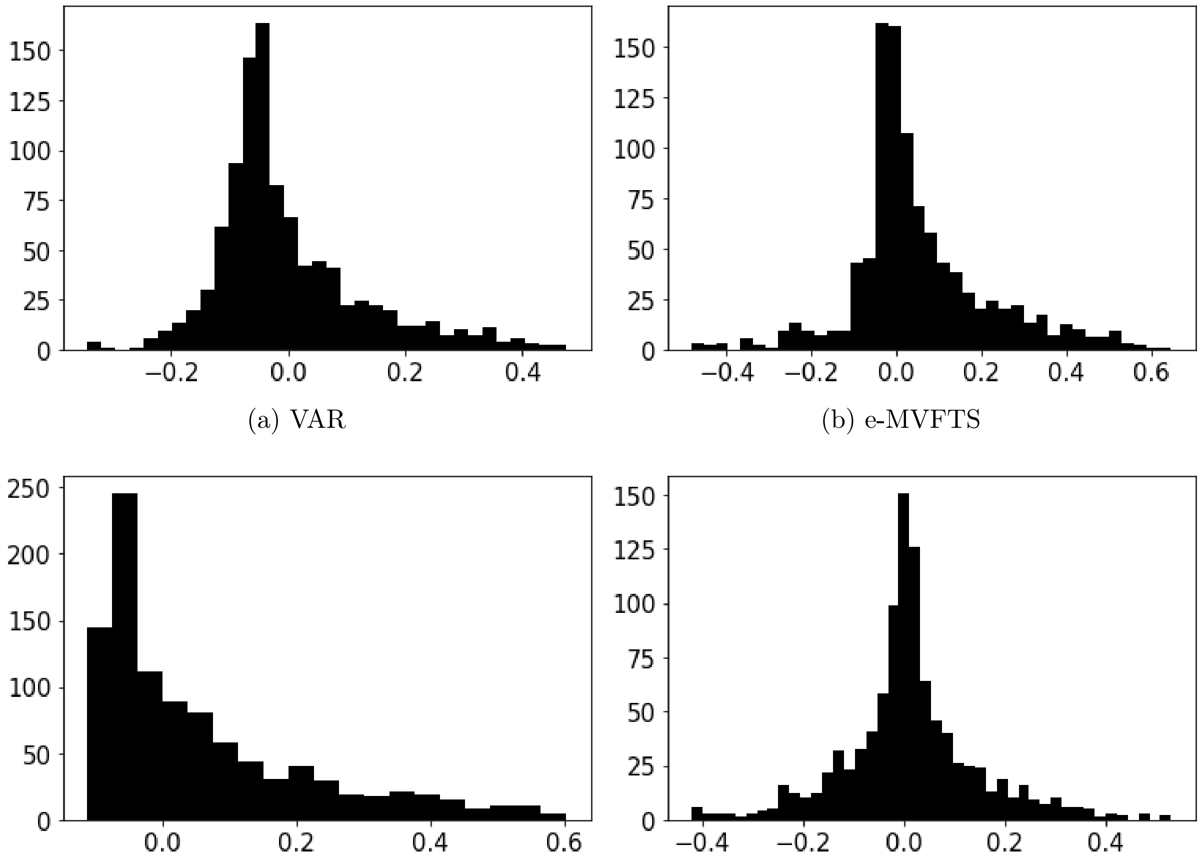
(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 49 – Wind Energy Dataset - Residual Analysis - 4 steps ahead Forecasting



(a) VAR

(b) e-MVFTS

(c) MLP

(d) FIG-FTS

Figure 50 – Wind Energy Dataset - Residual Analysis - 8 step ahead Forecasting

Chapter 8

Conclusion

In this thesis, the problem of renewable energy forecasting was addressed, with a focus on solar and wind energy. Despite their specifics, both energy sources have in common the strong influence of climatic conditions, which result in non-stationary patterns in the time series related to energy production. If, on the one hand, such patterns make it difficult to develop a model that can adequately predict future values, on the other, we have a worldwide growth in demand for renewables, which encourages the modernization of energy systems. Systems with modern technologies may offer a greater capacity for monitoring the plants, based on increasingly detailed and accurate records of the sites. It is from this greater availability of data that is proposed in this thesis the development of a solution that could take into account the correlation between the values recorded on the sites of a system distributed geographically. Such information was modeled within the context of a spatio-temporal problem. In addition, the previously discussed issue of non-stationarity was also addressed, with the development of adaptive mechanisms for the forecasting method.

Among the different approaches present in the literature to deal with the problem, FTS models were chosen in the thesis, due to their flexibility that allows reformulations that can cover both the processing of multivariate series and the implementation of adaptation mechanisms of the model during execution. In addition, their representation of the rules is made in a more interpretable way than other techniques of greater complexity, such as neural networks. Such characteristic is important to facilitate its understanding and improve its reliability, which can assist in the adoption of the technique to support decision making.

First, the problem of non-stationary data was approached, from the application of NSFTS, a novel FTS technique that provides a mechanism of adaptation of the model to deal with the changes in the parameters of data distribution, common to non-stationary time series. When evaluated for renewable energy forecasting, the model was less accurate than other models of greater complexity. However, when compared to a classic FTS model, the NSFTS presents a substantial performance gain, which suggests that a mechanism for adapting the model has great potential to assist in the renewable energy forecasting.

Along with the problem of non-stationarity, the initial premise also involves the understanding of spatio-temporal data, in order to provide a more robust model. The proposed solution to meet all these aspects was the e-MVFTS, an evolving FTS model that represents the data using multivariate clusters, translating them into univariate components, to be used by a High Order FTS model. The model presented performance comparable to other more complex multivariate forecasting models, with the additional advantage of being able to adapt to the data received, without the need for new training steps. Its adaptation mechanism also presents features that provide a more flexible FTS-based representation. For example, the model does not require a prior setup of the number of fuzzy sets, which are created on demand, and have their parameters adjusted during runtime. In addition, the model requires the definition of a few hyper-parameters, thus simplifying eventual hyper-parameter tuning processes, which are usually computationally expensive. The experiments also indicated a robustness of such hyper-parameters, when applied to large volumes of data. Therefore, from the elaboration of this model, a new extension of the FTS models is presented, which reinforces an initial impression of this research, that the treatment of non-stationarity and the correlations present in spatio-temporal data are points of improvement for renewable energy forecasting problems.

Future work

In view of the promising results of the e-MVFTS, we envision some initiatives that may be addressed in order to improve its performance. One of them is the development of mechanisms that can optimize the model, especially regarding the number of micro-clusters. In addition, the evaluation of different distance metrics for the clustering algorithm and, subsequently, their fuzzy sets is a point to be analyzed. The model was initially conceived in the context of renewable energy time series. The evaluation of its performance when applied to other problems can be a line of study with the potential to highlight other positive and negative aspects. For example, forecasting problems involving multivariate data streams, given that the TEDA framework, which underlies the model's clustering mechanism, was originally conceived within this context.

One of the motivations cited in this thesis for the adoption of FTS models is the interpretability of their models. For the e-MVFTS model, the challenge is presenting a better visual representation of multivariate clusters. For that, the application of techniques capable of multidimensional data visualization, such as t-SNE [Maaten and Hinton, 2008], can be an alternative for the representation of fuzzy sets. From these projected components, the rule base that describes the model could be more easily represented, indicating in a more simplified way the decisions that support the returned prediction values.

In addition to interpretable rules, another improvement of the model that can assist in the decision-making process is the output of probabilistic forecasting. Therefore, another future work is the integration of e-MVFTS with FTS-based probabilistic forecasting models, as described in [de Lima Silva et al., 2020]. This extension can assist both in the decision-making process for the generation of energy from an established system and in the prospecting of new installations based on data from neighboring sites. Therefore, it may open new applications for the e-MVFTS model.

Publications

From the research conducted for this thesis, the following publications were produced.

Journal Papers

- J. Maia, C. A. Severiano Junior, F. G. Guimarães, C. L. de Castro, A. P. Lemos, J. C. F. Galindo, and M. W. Cohen. Evolving clustering algorithm based on mixture of typicalities for stream data mining. *Future Generation Computer Systems*, 106: 672–684, 2020

- P. C. d. L. e Silva, C. A. Severiano Junior, M. A. Alves, R. Silva, M. W. Cohen, and F. G. Guimarães. Forecasting in non-stationary environments with fuzzy time series. *Applied Soft Computing*, page 106825, 2020

- C. A. Severiano Junior, P. C. d. L. e Silva, M. W. Cohen, and F. G. Guimarães. Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, 171:764–783, 2021

Conference Papers

- C. A. Severiano Junior, F. G. Guimarães, and M. W. Cohen. Very short-term solar forecasting using multi-agent system based on extreme learning machines and data clustering. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016

- C. A. Severiano Junior, P. C. Silva, H. J. Sadaei, and F. G. Guimarães. Very short-term solar forecasting using fuzzy time series. In *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2017

- P. Silva, M. A. Alves, C. Severiano Jr, G. Vieira, F. Guimarães, and H. Sadaei. Probabilistic forecasting with seasonal ensemble fuzzy time-series. In *XIII Brazilian Congress on Computational Intelligence, Rio de Janeiro*, 2017

- M. A. Alves, P. C. d. L. Silva, C. A. Severiano Junior, G. L. Vieira, F. G. Guimarães, 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*, pages 591–596. Springer LNCS Series, 2018

- P. C. d. L. e Silva, C. A. Severiano Jr, M. A. Alves, M. W. Cohen, and F. G. Guimarães. A new granular approach for multivariate forecasting. In *Latin American Workshop on Computational Neuroscience*, pages 41–58. Springer, 2019

References

M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler. Streamkm++: A clustering algorithm for data streams. *Journal of Experimental Algorithmics (JEA)*, 17:2–4, 2012.

C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29*, VLDB '03, pages 81–92. VLDB Endowment, 2003. ISBN 0-12-722442-4.

X. G. Agoua, R. Girard, and G. Kariniotakis. Short-Term Spatio-Temporal Forecasting of Photovoltaic Power Production. *IEEE Transactions on Sustainable Energy*, (September): 1–1, 2017. ISSN 1949-3029. doi: 10.1109/TSTE.2017.2747765. URL http://ieeexplore.ieee.org/document/8023858/.

A. Ahmed and M. Khalid. A review on the selected applications of forecasting models in renewable power systems. *Renewable and Sustainable Energy Reviews*, 100:9–21, 2019.

M. A. Alves, P. C. d. L. Silva, C. A. Severiano Junior, G. L. Vieira, F. G. Guimarães, 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*, pages 591–596. Springer LNCS Series, 2018.

ANEEL. ANEEL - Agencia Nacional de Energia Elétrica, Brasil. http://www.aneel.gov.br/, 2018. Accessed: 2018-07-23.

P. Angelov. Outside the box: an alternative data analytics framework. *Journal of Automation Mobile Robotics and Intelligent Systems*, 8(2):29–35, 2014a.

P. Angelov. Anomaly detection based on eccentricity analysis. In *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)*, pages 1–8, Dec 2014b. doi: 10.1109/EALS.2014.7009497.

P. Angelov and X. Zhou. Evolving fuzzy systems from data streams in real-time. In *2006 International symposium on evolving fuzzy systems*, pages 29–35. IEEE, 2006.

J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F. J. Martinez-de Pison, and F. Antonanzas-Torres. Review of photovoltaic power forecasting. *Solar Energy*, 136: 78–111, 2016.

A. W. Aryaputera, D. Yang, L. Zhao, and W. M. Walsh. Very short-term irradiance forecasting at unobserved locations using spatio-temporal kriging. *Solar Energy*, 122: 1266–1278, 2015.

P. Bauer, A. Thorpe, and G. Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525(7567):47–55, 2015.

J. Boland. Spatial-temporal forecasting of solar radiation. *Renewable Energy*, 75:607–616, 2015.

F. Cao, M. Ester, W. Qian, and A. Zhou. Density-based clustering over an evolving data stream with noise. In *In 2006 SIAM Conference on Data Mining*, pages 328–339, 2006.

S.-M. Chen and C.-D. Chen. Handling forecasting problems based on high-order fuzzy logical relationships. *Expert Systems with Applications*, 38(4):3857–3864, 2011.

S.-M. Chen et al. Forecasting enrollments based on fuzzy time series. *Fuzzy sets and systems*, 81(3):311–319, 1996.

B. S. J. Costa, P. P. Angelov, and L. A. Guedes. Fully unsupervised fault detection and identification based on recursive density estimation and self-evolving cloud-based classifier. *Neurocomputing*, 150:289 – 303, 2015. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2014.05.086. Bioinspired and knowledge based techniques and applications The Vitality of Pattern Recognition and Image Analysis Data Stream Classification and Big Data Analytics.

R. Dambreville, P. Blanc, J. Chanussot, and D. Boldo. Very short term forecasting of the global horizontal irradiance using a spatio-temporal autoregressive model. *Renewable Energy*, 72:291–300, 2014. ISSN 18790682. doi: 10.1016/j.renene.2014.07.012. URL http://dx.doi.org/10.1016/j.renene.2014.07.012.

P. C. de Lima Silva, H. J. Sadaei, R. Ballini, and F. G. Guimarães. Probabilistic forecasting with fuzzy time series. *IEEE Transactions on Fuzzy Systems*, 28(8):1771–1784, 2020.

P. C. d. L. e Silva. *Scalable Models for Probabilistic Forecasting with Fuzzy Time Series*. PhD thesis, Universidade Federal de Minas Gerais, 2019.

P. C. d. L. e Silva, C. A. Severiano Jr, M. A. Alves, M. W. Cohen, and F. G. Guimarães. A new granular approach for multivariate forecasting. In *Latin American Workshop on Computational Neuroscience*, pages 41–58. Springer, 2019.

P. C. d. L. e Silva, C. A. Severiano Junior, M. A. Alves, R. Silva, M. W. Cohen, and F. G. Guimarães. Forecasting in non-stationary environments with fuzzy time series. *Applied Soft Computing*, page 106825, 2020.

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, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2009.02.057. URL http://dx.doi.org/10.1016/j.eswa.2009.02.057.

E. Egrioglu, C. H. Aladag, and U. Yolcu. Fuzzy time series forecasting with a novel hybrid approach combining fuzzy c-means and neural networks. *Expert Systems with Applications*, 40(3):854–857, 2013.

F. Fan, K. Bell, D. Hill, and D. Infield. Wind forecasting using kriging and vector autoregressive models for dynamic line rating studies. In *PowerTech, 2015 IEEE Eindhoven*, pages 1–6. IEEE, 2015.

T. Gafurov, J. Usaola, and M. Prodanovic. Incorporating spatial correlation into stochastic generation of solar radiation data. *Solar Energy*, 115:74–84, 2015.

J. Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition, 2010. ISBN 1439826110, 9781439826119.

J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

J. M. Garibaldi. The need for fuzzy ai. *IEEE/CAA Journal of Automatica Sinica*, 6(3): 610–622, 2019.

J. M. Garibaldi, M. Jaroszewski, and S. Musikasuwan. Nonstationary fuzzy sets. *IEEE Transactions on Fuzzy Systems*, 16(4):1072–1086, 2008.

A. Ghaderi, B. M. Sanandaji, and F. Ghaderi. Deep forecast: deep learning-based spatio-temporal forecasting. *arXiv preprint arXiv:1707.08110*, 2017.

N. Golyandina, V. Nekrutkin, and A. A. Zhigljavsky. Analysis of time series structure: Ssa and related techniques. 2001.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

M. Hahsler and M. Bolaños. Clustering data streams based on shared density between micro-clusters. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1449–1461, June 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2522412.

M. Hahsler and J. Forrest. *streamMOA: Interface for MOA Stream Clustering Algorithms*, 2019. URL https://CRAN.R-project.org/package=streamMOA. R package version 1.2-1.

M. Hahsler, M. Bolanos, J. Forrest, et al. Introduction to stream: An extensible framework for data stream clustering research with r. *Journal of Statistical Software*, 76(14):1–50, 2017.

T. Hong, P. Pinson, and S. Fan. Global energy forecasting competition 2012, 2014.

R. Hyde, P. Angelov, and A. MacKenzie. Fully online clustering of evolving data streams into arbitrarily shaped clusters. *Information Sciences*, 382:96–114, 2017.

IEA. Renewables 2019 - market analysis and forecast from 2019 to 2024. Technical report, International Energy Agency, 2019. URL iea.org/reports/renewables-2019.

IEA. Global energy review 2020. Technical report, International Energy Agency, 2020. URL https://www.iea.org/reports/global-energy-review-2020.

R. H. Inman, H. T. Pedro, and C. F. Coimbra. Solar forecasting methods for renewable energy integration. *Progress in Energy and Combustion Science*, 39(6):535 – 576, 2013. ISSN 0360-1285. doi: http://dx.doi.org/10.1016/j.pecs.2013.06.002.

P. Jiang, H. Yang, and J. Heng. A hybrid forecasting system based on fuzzy time series and multi-objective optimization for wind speed forecasting. *Applied energy*, 235:786–801, 2019.

T. A. Jilani, S. M. A. Burney, and C. Ardil. Multivariate High Order Fuzzy Time Series Forecasting for Car Road Accidents. *World Academy of Science, Engineering and Technology*, 2(1):288–293, 2007. ISSN 00014575. doi: 10.1017/CBO9781107415324.004.

J. Jung and R. P. Broadwater. Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews*, 31:762–777, 2014.

G. Kariniotakis. *Renewable Energy Forecasting: From Models to Applications*. Woodhead Publishing, 2017.

M. Khodayar and J. Wang. Spatio-temporal graph deep neural network for short-term wind speed forecasting. *IEEE Transactions on Sustainable Energy*, 10(2):670–681, 2018.

H. Lan, C. Zhang, Y.-Y. Hong, Y. He, and S. Wen. Day-ahead spatiotemporal solar irradiation forecasting using frequency-based hybrid principal component analysis and neural network. *Applied energy*, 247:389–402, 2019.

L. W. Lee, L. H. Wang, S. M. Chen, and Y. H. Leu. Handling forecasting problems based on two-factors high-order fuzzy time series. *IEEE Transactions on Fuzzy Systems*, 14 (3):468–477, 2006. ISSN 10636706. doi: 10.1109/TFUZZ.2006.876367.

D. Leite, R. Ballini, P. Costa, and F. Gomide. Evolving fuzzy granular modeling from nonstationary fuzzy data streams. *Evolving Systems*, 3(2):65–79, 2012.

A. Lemos, F. Gomide, and W. Caminhas. Multivariable gaussian evolving fuzzy modeling system. *Fuzzy Systems, IEEE Transactions on*, 19(1):91–104, 2011.

G. A. Licciardi, R. Dambreville, J. Chanussot, and S. Dubost. Spatiotemporal pattern recognition and nonlinear pca for global horizontal irradiance forecasting. *IEEE Geoscience and Remote Sensing Letters*, 12(2):284–288, 2015.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

J. Maia, C. A. Severiano Junior, F. G. Guimarães, C. L. de Castro, A. P. Lemos, J. C. F. Galindo, and M. W. Cohen. Evolving clustering algorithm based on mixture of typicalities for stream data mining. *Future Generation Computer Systems*, 106:672–684, 2020.

R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, et al. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293–312. Elsevier, 2019.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

C. Molnar. *Interpretable Machine Learning*. 2019. `https://christophm.github.io/interpretable-ml-book/`.

D. C. Montgomery, C. L. Jennings, and M. Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

S. R. Moreno and L. dos Santos Coelho. Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy inference system. *Renewable energy*, 126:736–754, 2018.

A. M. Nobre. *Short-term solar irradiance forecasting and photovoltaic systems performance in a tropical climate in Singapore*. PhD thesis, Universidade Federal de Santa Catarina, 2015.

NREL. Oahu solar measurement grid. `https://midcdmz.nrel.gov/oahu_archive/`, 2018. Accessed: 2020-08-01.

H. T. C. Pedro and C. F. M. Coimbra. Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy*, 86(7):2017–2028, 2012.

I. M. Peters, C. Brabec, T. Buonassisi, J. Hauch, and A. M. Nobre. The impact of covid-19-related measures on the solar resource in areas with high levels of air pollution. *Joule*, 4(8):1681–1687, 2020.

M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

H. J. Sadaei, P. C. d. L. 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*, 175:365–377, 2019.

S. Samanta, M. Pratama, and S. Sundaram. A novel spatio-temporal fuzzy inference system (spatfis) and its stability analysis. *Information Sciences*, 505:84–99, 2019.

C. A. Severiano Junior, F. G. Guimarães, and M. W. Cohen. Very short-term solar forecasting using multi-agent system based on extreme learning machines and data clustering. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

C. A. Severiano Junior, P. C. Silva, H. J. Sadaei, and F. G. Guimarães. Very short-term solar forecasting using fuzzy time series. In *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2017.

C. A. Severiano Junior, P. C. d. L. e Silva, M. W. Cohen, and F. G. Guimarães. Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, 171:764–783, 2021.

J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. P. L. F. d. Carvalho, and J. a. Gama. Data stream clustering: A survey. *ACM Comput. Surv.*, 46(1):13:1–13:31, July 2013. ISSN 0360-0300. doi: 10.1145/2522968.2522981.

P. Silva, M. A. Alves, C. Severiano Jr, G. Vieira, F. Guimarães, and H. Sadaei. Probabilistic forecasting with seasonal ensemble fuzzy time-series. In *XIII Brazilian Congress on Computational Intelligence, Rio de Janeiro*, 2017.

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*, 27:370–385, 2018.

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)*, pages 1–8. IEEE, 2018.

Q. Song and B. S. Chissom. Fuzzy time series and its models. *Fuzzy sets and systems*, 54 (3):269–277, 1993.

L. Suganthi, S. Iniyan, and A. A. Samuel. Applications of fuzzy logic in renewable energy systems–a review. *Renewable and sustainable energy reviews*, 48:585–607, 2015.

Y. Sun, G. G. Yen, and Z. Yi. Evolving unsupervised deep neural networks for learning meaningful representations. *IEEE Transactions on Evolutionary Computation*, 23(1): 89–103, 2018.

C. Sweeney, R. J. Bessa, J. Browell, and P. Pinson. The future of forecasting for renewable energy. *Wiley Interdisciplinary Reviews: Energy and Environment*, 9(2):e365, 2020.

L. J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437 – 450, 2000. ISSN 0169-2070. doi: https://doi.org/10.1016/S0169-2070(00)00065-0. URL http://www.sciencedirect.com/science/article/pii/S0169207000000650. The M3- Competition.

C. Vincent, G. Giebel, P. Pinson, and H. Madsen. Resolving nonstationary spectral information in wind speed time series using the hilbert–huang transform. *Journal of Applied Meteorology and Climatology*, 49(2):253–267, 2010.

D. Yang, C. Gu, Z. Dong, P. Jirutitijaroen, N. Chen, and W. M. Walsh. Solar irradiance forecasting using spatial-temporal covariance structures and time-forward kriging. *Renewable Energy*, 60:235–245, 2013.

D. Yang, Z. Dong, T. Reindl, P. Jirutitijaroen, and W. M. Walsh. Solar irradiance forecasting using spatio-temporal empirical kriging and vector autoregressive models with parameter shrinkage. *Solar Energy*, 103:550–562, 2014. ISSN 0038092X. doi: 10.1016/j.solener.2014.01.024.

R. Yu, Z. Liu, X. Li, W. Lu, D. Ma, M. Yu, J. Wang, and B. Li. Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space. *Applied energy*, 238:249–257, 2019.

L. A. Zadeh. Fuzzy sets and information granularity. *Advances in fuzzy set theory and applications*, 11:3–18, 1979.

J. Zhang, R. Verschae, S. Nobuhara, and J.-F. Lalonde. Deep photovoltaic nowcasting. *Solar Energy*, 176:267–276, 2018.

Appendix A

Singular Spectrum Analysis

A.1 Singular Spectrum Analysis

The Singular Spectrum Analysis (SSA) is a technique of time series analysis which basically makes a decomposition of the original series into the sum of a small number of independent and interpretable components such as a slowly varying trend, oscillatory components and a structureless noise [Golyandina et al., 2001]. Among the capabilities of SSA for time series analysis, it can find trends at different resolutions, perform smoothing, detect seasonality components and cycles with small or large periods. And for all these tasks, no parametric model of the time series is required. Because of its decomposition properties, the method was applied to separate the original data into trend and residual components, as discussed in Chapter 2.

A.1.1 Methodology

SSA consists of two complementary stages: decomposition and reconstruction. Both stages include two separate steps. A brief discussion on the methodology of the technique presented below.

A.1.1.1 Decomposition

Step 1: Embedding

Embedding is a mapping that transfers a one-dimensional time series $Y_t = (y_1, y_2, \dots, y_t)$ to a multi-dimensional series $X_1, \dots, X_k$ where $X_i = (y_i, \dots, y_{i+L-1}) \in \mathbb{R}^L$.with $k = t - L + 1$. X_i is a *L-lagged vector.*

The embedding process takes one parameter L, called window length, where $2 < L < T$. After this step a trajectory matrix $\mathbf{X} = [X_1, \dots, X_K]$ is returned. $\mathbf{X}$ is a Hankel matrix.

Step 2: Singular value decomposition (SVD)

In this step, a Singular value decomposition of the trajectory matrix is performed. $\lambda_1, \ldots, \lambda_L$ are the eigenvalues of **XX'** in decreasing order of magnitude and $U_1, \ldots, U_L$ are the orthonormal system of the eigenvectors of the matrix **XX'** corresponding to these eigenvalues. Given d the rank of $\mathbf{X}$, where $d = max(i, suchthat \lambda_i > 0)$. Denote $V_i = \mathbf{X'} U_i / \sqrt{\lambda_i}$ and the SVD of the trajectory matrix $\mathbf{X}$ can be written as:

$$\mathbf{X} = X_1 + \ldots + X_d \tag{A.1}$$

where $\mathbf{X_i} = U_i \sqrt{\lambda_i} V_i'$. An SVD decomposition can be useful for measuring the contribution of each component $\mathbf{X_i}$ in the reconstruction of the original time series. Given that among all the matrices $\mathbf{X^{(r)}}$ of rank r, with $r < d$, the best approximation of matrix $\mathbf{X}$ is provided by $\sum_{i=1}^{r} \mathbf{X_i}$. It means that the ratio of contribution of a matrix $\mathbf{X_i}$ of rank r can be calculated by A.2:

$$\% of contribution = \frac{\lambda_i}{\sum_{i=1}^{d} \lambda_i} \tag{A.2}$$

A.1.1.2 Reconstruction

Step 1: Grouping

The grouping step consists of organizing the matrices $\mathbf{X_i}$ into disjoint sets. For each set, the belonging matrices are summed. Let $I = \{i_1, \ldots, i_p\}$ be a set of indexes $i_1, \ldots, i_p$. The matrix $\mathbf{X_I}$ corresponds to a sum of matrices of a set I, $\mathbf{X_I} = \mathbf{X_{i1}} + \ldots + \mathbf{X_{ip}}$. Thus, the grouping of the indexes $J = \{1, \ldots, d\}$ into disjoint subsets $I_1, \ldots, I_m$ refers to the representation:

$$\mathbf{X} = X_{I1} + \ldots + X_{Im} \tag{A.3}$$

Step 2: Diagonal Averaging

Diagonal averaging converts each matrix $\mathbf{X_I}$ to a time series, which is an additive component of the initial series Y_t. Let z_{ij} be an element of a matrix $\mathbf{Z}$. The *k-th* term of the reconstructed series is obtained by averaging z_{ij} over all i and j where $i + j = k + 2$. This procedure is called diagonal averaging or *Hankelization* of the matrix Z. It means that by applying the Hankelization procedure to each matrix component of A.3, produces series $\widetilde{X}^{(m)} = (\widetilde{x}_1^{(m)}), \ldots, \widetilde{x}_t^{(m)})$. This is equivalent to decomposing the original time series $Y_t = (y_1, y_2, \ldots, y_t)$ to a sum of m series reconstructed by diagonal averaging:

$$Y_t = \sum_{i=1}^{m} \widetilde{x}_i^{(m)}) \tag{A.4}$$