Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 19, Iss. 2, pp. 29-50, 2021

A TUTORIAL ON FUZZY TIME SERIES FORECASTING MODELS: RECENT ADVANCES AND CHALLENGES

Patrícia O. Lucas[1,2,3], **Omid Orang**[1,3], **Petrônio C. L. Silva**[2]
Eduardo M. A. M. Mendes[5], **Frederico G. Guimarães**[1,4]

[1]Machine Intelligence and Data Science (MINDS) Laboratory, Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brazil
[2]Federal Institute of Education Science and Technology of Northern Minas Gerais, Brazil
[3]Graduate Program in Electrical Engineeering, Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil
[4]Department of Electrical Engineering, Universidade Federal de Minas Gerais (UFMG), Belo Horizonte, Brazil
[5]Department of Electronics, Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte, Brazil
patricia.lucas@ifnmg.edu.br, omid.orang2009@gmail.com, petronio.candido@ifnmg.edu.br
emmendes@ufmg.br, fredericoguimaraes@ufmg.br

Abstract – Time series forecasting is a powerful tool in planning and decision making, from traditional statistical models to soft computing and artificial intelligence approaches several methods have been developed to generate increasingly accurate forecasts. Fuzzy Time Series (FTS) methods have been introduced in the early 1990's to handle data uncertainty and to undercome the statistical assumptions of linearity. Many studies have been reporting their good accuracy, simplicity, potential for interpretability and reduced computational complexity. This paper presents a tutorial for FTS methods. First, a review of the relevant literature is made, offering a foundation on the main concepts and FTS-based models for different time series and different types of forecasts. Then, the current challenges and possible solutions, are discussed alongside a timeline of the research developed in this area by the authors that aims at filling some of these gaps. Finally, a tutorial on the pyFTS library is presented. PyFTS is an open and free library coded in Python programming language that was developed by the MINDS Lab (Laboratory of Machine Intelligence and Data Science) and, also provides a set of transformation functions for pre-processing time series and a set of metrics and databases for benchmarking, in addition to implementing several FTS models in the literature.

Keywords – Time series forecasting, Fuzzy Times Series, Soft Computing, PyFTS.

1. Introduction

Properly estimating a future event or phenomenon plays a crucial role in the planning and conscious decision-making of activities in many fields of science, engineering, medicine, economics and etc. Forecasting is therefore a powerful tool for inferences on the conditions and values of any event in the future based on past and present information. Forecasting is one of the main goals of time series analysis and, based on the literature, there is a wide range of time series forecasting algorithms, from statistical methods to soft computing ones. One of the main challenges of conventional methods such as moving average, regression or exponential moving average is the need for a large number of observations and the lack of flexible ways to embed linguistic variables or expert opinions. Also, these methods are limited by the linearity assumption [1–3].

In order to mitigate these drawbacks, fuzzy set theory has attracted the attention of scientists to predict future phenomena while handling linguistic terms. Unlike traditional time series that deal with real numbers, the fuzzy set based time series $F(t)$ is a collection of time-indexed fuzzy linguistic terms $f_1(t), f_2(t), \dots f_n(t)$ with significant ability to deal with the uncertainty and ambiguity inherent in data collection. Fuzzy Time Series (FTS) methods have been gaining more relevance in recent years due to many studies reporting their good accuracy. FTS are soft computing methods that produce data-driven, non-parametric, simple, computationally inexpensive, and readable models for time series analysis and prediction. In addition, they do not require a large dataset as in statistical models and deep learning models [4].

More recently, three review papers have been published on FTS methods. In 2019, Bose and Mali [5] carried out an extensive review of papers published from 1993 to 2018, presenting the main stages of a classic FTS model and the techniques that were incorporated into these stages to improve the accuracy of these models. Panigrahi and Behera [6] present a brief review of works from 1993 to 2017. The most recent review is presented by Palomero et al. [7], showing a bibliometric analysis on 118 articles published between 2017 and 2021. These works show that much research has been carried out to improve the performance of FTS methods, but many issues still need attention.

The goal of this paper is to serve as a tutorial on FTS forecasting models for newcomers in the field, from the basic models, including common training and forecasting procedures, to the more recent advances in the field. Also, for those readers more familiar with FTS research, we highlight some new challenges that we think have not yet been adequately addressed in the literature, such as MIMO (Multiple Input Multiple Output) models, scalability to big data, high-dimensional times series, non-stationary FTS, handling streaming data and hybrid models. Since 2018, some recent advances have been done in these topics, bringing new solutions to these challenges, and these have not been covered in the survey papers available so far. Additionally, this paper describes a FTS Python library that implements the basic FTS models and many of these recent solutions. pyFTS is a free and open framework for the Python programming language developed by the MINDS Lab[1] in Brazil.

The paper is organized as follows. Section 2 presents an overview of FTS methods. Section 3 points out the challenges that FTS methods still face and the recent research that aims at filling some of these gaps. Finally, a brief tutorial on the pyFTS library is given in Section 4.

2 Fuzzy Time Series

The definition of Fuzzy Time Series, from [8], starts with a univariate time series $Y(t) \in \mathbb{R}$, for $t = 0, 1, \ldots, T$, where the Universe of Discourse U is delimited by the known bounds of Y, such that $U = [\min(Y), \max(Y)]$, as illustrated in Figure 1. Upon U, k fuzzy sets A_j, for $j = 1, \ldots, k$, are defined and each one with its own membership function μ_{A_j}.

$F(t)$ is called a Fuzzy Time Series over Y if $f(t) = \mu_{A_j}(y(t))$ is the collection of fuzzyfied values of Y for $j = 1, \ldots, k$ and $t = 0, 1, \ldots, T$. FTS model $\mathcal{M}$ is composed of a base of rules grouped in the form precedent $\rightarrow$ consequent that shows how the fuzzy sets A_j relate to each other over time, that is, how the time series Y behaves. The group of fuzzy sets A_j, for $j = 1, \ldots, k$, can also be understood as the linguistic values of a Linguistic Variable $\tilde{A}$.

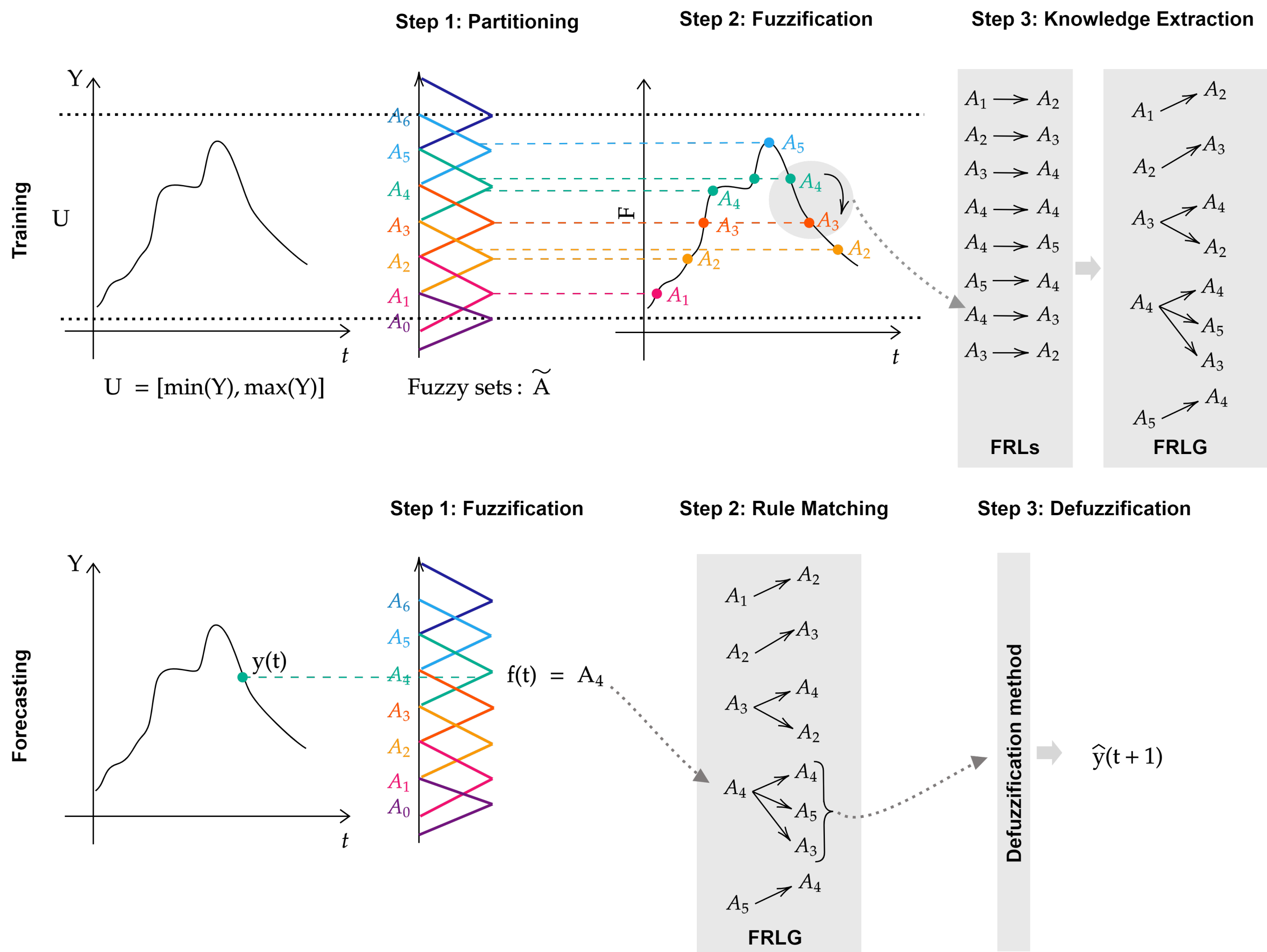

Figura 1: Training and Forecasting Procedures for Time Invariant FTS. FLRs are fuzzy logical rules and FLRGs are FLR grouped by precedents.

There are several categories of FTS methods, varying mainly by their order Ω and time-variance. The number of fuzzy sets in the rules precedent indicate the model order or how much past information is available to the model $\mathcal{M}$ to recognize the possible temporal patterns and make a forecast. Given the time series data Y, First Order models need just $y(t-1)$ data to forecast $y(t)$,

[1] The Machine Intelligence and Data Science (MINDS) Laboratory is a research lab at the Department of Electrical Engineering of Universidade Federal de Minas Gerais (UFMG). For more information: https://minds.eng.ufmg.br/ and https://pyfts.github.io/pyFTS/build/html/index.html

while High Order models require $y(t-1), \ldots, F(t-\Omega)$ data to forecast $y(t)$. The time variance indicates if an FTS model $\mathcal{M}$ changes along the time. If the model is static, it is called Time Invariant, otherwise Time-Variant [8–10].

A generic method can be extracted from the wide range of variations of FTS methods by splitting the FTS approach in two main procedures, the training and forecasting methods [4]. The training phase, illustrated in Figure 1, has the basic objective of creating a knowledge representation of the time series from the historical data. The training phase consists of stages 1 to 3 described below.

Step 1 - **Universe of Discourse partitioning**: partitioning is the most significant part of training phase. It is applied to split the Universe of Discourse U into fuzzy sets by making linguistic values $\tilde{A}$ through various types of available techniques in the literature.

Step 2 - **Fuzzification**: in this step the numeric data Y is converted into linguistic variables $\tilde{A}$ via given partitioning method with regards to their memberships degrees to the fuzzy sets. The obtained linguistic or fuzzified version of $y(t) \in Y$, represents the fuzzy time series F in which $f(t) \in F$ are fuzzy values.

Step 3 - **Knowledge or Temporal Pattern Extraction**: this step focuses mainly on extracting knowledge from F due to the past observations or lags. In other words, this stage analyzes the sequential terms in F, grouping the patterns in rules $(A_i) \rightarrow (A_j), (A_k), \ldots$ and can be said that if $f(t)$ is (A_i) then $f(t+1)$ is $(A_j), (A_k), \ldots$. For example, in Figure 1, A_4 and A_3 generate the fuzzy logical rule (FLR) $A_4 \rightarrow A_3$ which is then grouped into the GFLR $A_4 \rightarrow A_4, A_5, A_3$.

In the forecasting procedure, as can be seen in Figure 1, a new sample $y(t)$ is fed into the trained FTS model $\mathcal{M}$, containing the rules, to predict the future value $\hat{y}(t+1)$ at $t+1$. Given a sample $y(t)$, the model can be used to forecast $y(t+1)$ with the following steps:

Step 1 - **Fuzzification**: the fuzzification step is the same for both forecasting process and training procedure. It means that it translates the numeric sample $y(t) \in Y$ into a fuzzy value $f(t)$, where $f(t) \in \tilde{A}$.

Step 2 - **Rule Matching**: find the rules that contain the fuzzy set $f(t)$ in its precedent. The activation of the rules are given by the membership values in $f(t)$.

Step 3 - **Defuzzification**: the aim of this process is transforming the fuzzy value $f(t+1)$ to a crisp value $\hat{y}(t+1)$. In additions point forecasting methods, other types of defuzzification have been proposed for intervals and probabilistic distributions forecasting in the literature, see for instance [4].

In the remaining part of this section, each process is discussed in more details, while referring to the most relevant studies and models in the FTS literature. It should be mentioned that there exist design choices and effective hyperparameters, such as number of partitioning, partitioning method, membership function, knowledge extraction methods, fuzzification and defuzzification methods on the FTS processes. Accordingly most of FTS methods depend on the proper selection of these hyperparameter values, considering the accuracy and parsimony of the model.

2.1 Defining the Universe of Discourse

The natural definition of the Universe of Discourse is $U = [\min(Y), \max(Y)]$, but it is common that the upper and lower bounds be exceeded by a confidence margin. Therefore the modified version of Universe of Discourse is $U = [l_b, u_b]$ for time series $Y(t)$, where $l_b = \min(Y) - D_1$ and $u_b = \max(Y) + D_2$, with $D_1 = \min(Y) \cdot r$, $D_2 = \max(Y) \cdot r$ and $0 < r < 1$. Although $r = 0.2$ is a typical value, it can be determined differently based on the problem [4, 11]. The authors in [12] outlined the other modification of U as $U = [l_b, u_b] = [\min(Y) - D, \max(Y) + D]$, where $D = 0.2 \cdot (\max(Y) - \min(Y))$. The boundary extension is helpful in both stationary and non-stationary time series to adapt to fluctuations near the bounds of U.

2.2 Data Partitioning

Now it is the time to split the defined Universe of Discourse U to create overlapping intervals and associate them with values of the linguistic variable $\tilde{A}$. Creation of $\tilde{A}$ strongly depends on three hyperparameters: the number of partitions, the partitioning techniques and membership function μ.

2.2.1 Partitioning technique

The partitioning methods are commonly divided into two groups according to the length or size of intervals: fixed (equal) intervals and unequal interval techniques as illustrated in Figure 2.

In an equal-sized partitioning technique the length of generated fuzzy sets are equal. Due to its main advantages, which are simplicity and not being time consuming, it has been adopted by numerous scientists [13–27]. Grid partitioning proposed in [8] is known as the simplest partitioning scheme. Considering k as the number of intervals, then in Grid partitioning method, k fuzzy sets with the same size will be created by splitting U into k overlapping fuzzy sets. Some works in the literature used simple heuristic partitioning methods, including [28–32].

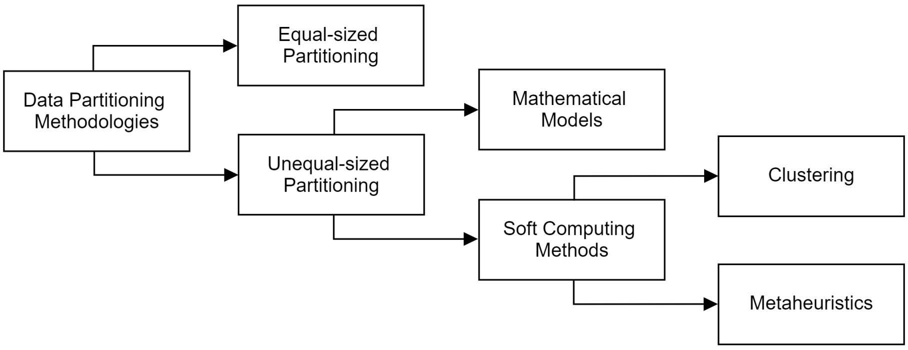

Figura 2: Classifications of Data Partitioning Methods

Despite the mentioned advantages for equal length partitioning schemes, their performance are poor when the data is not distributed uniformly [33]. Thus various types of unequal-sized partitioning methods have been introduced in the literature. These can be categorized into two groups: mathematical and soft-computing techniques. Different types of mathematical partitioning methods exist such as Distribution-based and average-based partitioning [34], Heuristic information [28] based on Chen's model, frequency-density-based partitions [35], automatic clustering developed in [36] due to the maximum data difference and initializing the number of cluster by one. Deciding if the data belongs to the current cluster or if it is needed to add new cluster strictly depends on the maximum data difference. They also proposed another modification in [37] in which the number of initial clusters is the same as the size of data (agglomerate clustering). In [38] a new technique named Mean-Based Discretization is proposed. The mean values of data are employed to partitioning U into two subsets U_A and U_B. Then each subset is further divided into intervals based upon some guiding criterion.

Entropy partitioning methodology was encouraged by [39], with a fixed number of unequal length partitions. The authors used trapezoidal membership function, as also employed in [40] and [41]. Entropy discretization has been employed by other researchers in [2, 37, 42–45]. The Fuzzy Information Granule (FIG) is another partitioning technique used by many researchers [46–50]. It is based on the granular computing concept introduced and developed by [51–53]. In [54] a new form of granule named linear Gaussian FIG (LFIG) relies on linear regression, introduced to cope with some disadvantages of traditional FIG such as triangular FIG and interval FIG. A new fast, adoptable and flexible FIG named as FIG-FTS was introduced in [55] with high accuracy and competitive results.

The other group of partitioning techniques is on the basis of clustering. One of the most famous fuzzy-based clustering method used in many forecasting models is fuzzy C-Means (FCM). FCM partitioning is exploited in [56–63]. Although these studies adopt the FCM partitioning method, the FLR are generated differently. For instance Cheng used ANFIS, Rule based and Interpolation techniques in his publications. Li et al. had adopted Rule based approach, rough sets, Hidden Markov model, Vector quantization technique. Meanwhile Egrioglu et al. had employed back-propagation Neural Network (BPNN), Fuzzy relation matrix and Particle Swarm Optimization (PSO). Other clustering methods such as subtractive clustering [58, 64], K-Means [65], Fuzzy K-Medoids [66], Fuzzy C-Medoid [67], Self-Organizing Maps [68] and rough-fuzzy c-means [69] have been applied for FTS partitioning.

During these years the intelligent optimization techniques have been widely utilized, and Particle Swarm Optimization (PSO) was a popular one [61, 70–72]. Modifications of PSO involving MTPSO algorithm [73], HPSO [74], NPSO [75] as types of rule based techniques used by other researchers to create fuzzy logic relation by which the forecasting accuracy will be improved. Other authors such as Chen and Kao [71], Lu et al. [47] and Singh and Borah [72] also considered PSO partitioning by implementing Support Vector Machine (SVM), Granular computing and Type-2 fuzzy sets to build FLR in their forecasting models. In addition, other optimization techniques have been used, including Genetic Algorithm (GA) in [76–78], Ant Colony Optimization in [79], Harmony Search in [80], Imperialist Competitive in [81] and Simulated Annealing in [82].

Although optimization and clustering based partitioning methods are more time consuming than equal length partitioning methods, they tend to improve forecasting accuracy in general. Using optimization techniques are beneficial to reach near optimal positions of split points between two intervals, while the application of clustering methods improves the rate of accuracy indirectly by positioning the intervals according to the data distribution.

2.2.2 The Number of Partitions k

Another parameter that has a strong effect on the accuracy of the model is the number of fuzzy sets. As mentioned in [4] there is no linear relation among the accuracy of the model and the number of partitions, that is, the optimal number of partitions should be determined to obtain the best accuracy. Choosing a large value for k increases the complexity of the model because too many fuzzy sets and rules are generated, perhaps more than the required number. As a result, the risk of over-fitting from data and consequently reproducing noise will be raised as well as the cost of computation. On the other hand, small values of k lead to simpler models and the possibility of under-fitting is increased. Therefore, there is a clear balance between the accuracy (bias) and capacity (variance). This factor has effect on the readability and explainability of the model [83].

2.2.3 The Membership Function

The membership function specifies the degree in $[0, 1]$ to which a crisp value belongs to a fuzzy set. Although the influence on the accuracy is less important than other parameters, it is recommended to test different fuzzy sets in the model. Trapezoidal, Triangular and Gaussian are the most common ones. Based on the literature, triangular membership function is applied by default in many researches, but there are some references that employed Trapezoidal function such as [8, 17, 31, 84].

A combination of trapezoidal and triangular functions has been also considered [85]. However, since the mentioned membership functions are composed of straight line segments, they are not smooth at the corner points specified by the parameters. Therefore smooth and non-linear Gaussian membership functions have been adopted as well, by considering midpoint and width as its parameters. It is worth notions that in comparison to the other factors, membership function has less effect on the forecasting accuracy but the proper selection of μ can directly impact on the readability and explainability of the model.

In addition to the mentioned fuzzy sets, which are more common, some references developed different FTS methods applying other types of fuzzy sets, including picture fuzzy sets and picture fuzzy clustering [86], Type-2 fuzzy sets [87–90], hesitant differential fuzzy sets [91], non-stationary fuzzy sets [11, 92] and so on.

2.3 Fuzzification

Fuzzification is a process of transforming crisp values to the linguistic terms or fuzzy quantities. In Fuzzy Time Series model, each interval contains significant information occurred in the past. So, for preventing the important information to be lost, the fuzzy sets need to be managed in an efficient way. Therefore some fuzzification processes were provided by researchers in the literature. Initial studies in [8, 13, 30] proposed the maximum membership degree as the main fuzzification method, that is, just the fuzzy set with the greatest membership value is considered. It is computationally simple but the likelihood of losing information will be high by ignoring various neighbouring membership degrees as well as raising the risk of under-fitting.

For solving this issue, the holistic fuzzification approach is proposed as an alternative. In this process instead of considering only the maximum membership degree, all members are taken into account. In the first method $f(t) = \max(\mu_{A_i}(y(t)))$, while in the holistic method: $f(t) = \mu_{A_i}(y(t))$ for all fuzzy sets. However, by applying the holistic method, over-fitting is still remaining as a main trap. In the α-cut fuzzification method, unlike the holistic method, only the membership degrees above the defined α value are acceptable. Mathematically speaking for each $A_i \in \tilde{A}$, $f(t) = A_i$, if $\mu_{A_i}(y(t)) \geq \alpha$. Therefore it makes a good balance between over-fitting and under-fitting.

In this step, the fuzzy logic relation (FLR) requires to be established after the time series data has been completely fuzzified. As mentioned earlier, FLRs are built to identify the relationship among the previous and current states (antecedent and consequence). The next process is knowledge extraction from fuzzified values, which is explained in more details in the following section.

2.4 Knowledge Extraction

Knowledge extraction from fuzzy time series F is carried out based on the historical observations. Various methodologies including rule based models, matrix models, weighted rule methods, smart techniques, statistical methods, fuzzy cognitive map, neural network and statistical techniques have been explored in the literature to discover temporal pattern on F. Thus, in the following an overview of these methods is focused on.

2.4.1 Matrix and Rule-based Models

As we wave discussed earlier, FLR represent the transitions to relate previous and current fuzzified values. More specifically, an FLR is given by $F(t-1) \to F(t)$ or $A_i \to A_j$, which is a description of fuzzy rules. A_i is the fuzzified value at time $t-1$ as Left Hand Side (LHS) of the rule (precedent) and A_j is fuzzified value at time t, as Right Hand Side (RHS) of the rule (consequent). According to this definition, Song and Chissom [93, 94] proposed matrix models in which fuzzy logical matrix R is defined as $\mathbf{R_{ij}} = ({A_i}^T) \cdot A_j$ using Max-Min composition. In other words, $f(t) = f(t-1) \circ R(t, t-1)$, where $R(t, t-1)$ is an operational matrix consisting of all FLR for all fuzzy sets and $\circ$ plays the role of Max-Min operator. Other researchers also used this method, see for instance [40, 76, 95–105].

The rule based method was proposed based on the Fuzzy Logical Relation Group (FLRG) by Chen in [13]. In a sense, the rule base is composed of FLRGs, since the temporal patterns are grouped by their precedents forming the extracted knowledge. Ease of computation and reduction of time of computation are counted as the main advantages of the rule based method compared with the matrix method. FLRGs groups the rules with the same antecedent and different consequent. All FLRs with the same left hand side are grouped into an FLRG. For instance suppose that $A_1 \to A_1$, $A_1 \to A_2$, $A_1 \to A_3$ are the FLRs, these fuzzy logical relationships can be grouped into a fuzzy logical relationship group as $A_1 \to A_1, A_2, A_3$. Besides improving interpretability and readability of Chen's model, the performance of rule based model is superior to the matrix model one. The mentioned technique is exploited in many references, such as [28, 34, 36–38, 43, 63, 72, 106–111], using diverse defuzzification approaches. Some works applied Heuristic FLRG such as [112, 113]. The idea of using differential fuzzy sets and differential fuzzy relationships (DFLR) is put forth in [114] by subtracting successive indexes of ordinary FLRs.

Tabela 1: Some examples of neural network based methods for creating FLRs.

Neural Network based methods	Authors
BPNN including multilayers perceptron (MLP)	[2, 3, 9, 15–18, 61, 115, 129, 130]
Single Multiplicative Neuron (SMN)	[130–132]
Pi-Sigma NN (PSNN)	[133]
Adaptive Neuro Fuzzy Inference System (ANFIS)	[58, 63, 134–136]
Generalized regression neural network (GRNN)	[137]

2.4.2 Hybrid Method with Smart Techniques

According to the literature, in addition to various types of intelligent techniques applied to FTS prediction, the application of Artificial Neural Networks (ANN) is more common than others due to benefits such as good handling of nonlinear problems, increased accuracy and complexity reduction especially when the number of variables is large. Backpropagation Neural Networks (BPNN) are used widely by many researchers to construct FLR, which was suggested for the first time in [115]. Different types of Neural Networks (NN) based methods used in FTS forecasting have been organized in Table 1 [12, 116].

In recent years deep learning has attracted the attention of researchers to apply its in the FTS models. For instance in [12] new Deep Belief Network (DBN), Long-Short Term Memory (LSTM) and Support Vector Machine(SVM) have been tested for modeling the FLRs. Additionally, a modified average-based method is proposed to determine the effective length of intervals. Tran et al. [117] proposed a method that uses LSTM networks as knowledge model, while Backpropagation algorithm is considered as the learning method. A hybrid method which combines convolutional neural networks and FTS is introduced in [118] for short-term load forecasting. The authors proposed Image-FTS, using a binary image created by stacking the fuzzified values of several past lags.

Type-2 Fuzzy logic [72,88,107,119], fuzzy inference system [120], rough set theory [42,121–123], c-fuzzy-decision tree [27], fuzzy decision tree (FDT) based on C4.5 [124] and Fuzzy Cognitive Maps (FCMs) [125, 126] are other types of intelligent techniques that have been used in conjunction with FTS.

Particle Swarm Optimization (PSO) has been used by other researchers to establish FLR [62,127]. The Hybrid Particle Swarm FTS (HPSO-FTS) was introduced in [74] to build optimal rules. Refined High-order Weighted FTS with Imperialist Competitive Algorithm (RHWFTS-ICA) is employed in [77] to optimize and find best value for lags. The authors in [63,79,108,121] focused on GA to find the best interval or to extract weight rule matrix, as in [128].

2.4.3 Hybrid Method with Statistical Models

Various statistical techniques have been proposed for time series forecasting, including Box-Jenkis method (ARIMA), Moving Average (MA), Autoregressive model (AR) and ARMA as a mixture of AR and MA, which are the most popular ones. Some advanced statistical techniques are mixed to improve the efficiency of FTS forecasting as hybrid methods, such as Fuzzy ARIMA (FARIMA) model [138], AutoRegressive Fractional Integrated Moving Average (ARFIMA) model [139]. A combination of seasonal time series ARIMA (SARIMA) model and the fuzzy regression model named as fuzzy seasonal ARIMA (FSARIMA) is used in [140]. A new Fuzzy ARIMA model is proposed in [139] to tackle some drawbacks of conventional ones. In [80] a hybrid model of exponential time series is introduced to predict stock market. In [64] ANFIS is combined with MA as the hybrid forecasting methodology. A new hybrid method which is a mixture of FTS with Seasonal Auto Regressive Fractionally Integrated Moving Average (SARFIMA) is proposed in [81]. A hybrid model combining ARFIMA with FTS for forecasting long-memory time series is presented in [127].

2.4.4 Weighted Rule Models

The occurrence of temporal patterns in the historical data when building the FLRG is one of the issues which has been ignored by researchers [13, 28, 93, 94, 99], which means that the importance of recurrent and unique patterns of data in forecasting is the same. Furthermore the weights of recent and older patterns are the same in the basic FTS methods. Thus Yu [30] proposed the Weighted Fuzzy Time Series (WFTS), which adds weights to FLRG considering rule's recurrence and their chronological order. In this way different weights are given to the individual fuzzy relations. According to the monotonically structure of weight matrix, the higher weights are allocated to the recent FLR. Thus in this model, weights are assigned to each fuzzy set and the Weighted Fuzzy Logical Relation Group (WFLRG) is created. Chen [123] incorporated the concept of the Fibonacci sequence in the Song and Chissom's model and the weighted method of Yu's model to forecast stock price. A fluctuation type matrix is used to calculate weights of FLR in some models [2, 58, 141, 142].

There exist other weighted rule based models as well in the literature. Ismail and Efendi [143] presented Improved Weighted FTS (IWFTS), with different weighted rule assignment which considers only the recurrence of every rule. Other authors exploited Exponential growth of FTS instead of linear growth named as Exponential Weighted FTS (EWFTS) [144] and [80]. Trend Weighted Fuzzy Time Series (TWFTS) is the other weighted method presented by [145] and [146].

Silva [4] proposed the recent version of weighted FTS, the so-called Weighted High Order FTS (WHOFTS), with weights added to the proposed High Order FTS. In Weighted Interval FTS or W[I]FTS, the extension form of WHOFTS, the prediction intervals are based on the weighted intervals of the RHS fuzzy sets on each FLRG, weighted by their fuzzy membership in relation to the input value. Probabilistic Weighted FTS (PWFTS), a time invariant and heuristic method to generate probabilistic weighted rule model, is proposed in [147]. PWFTS consists of weights in both sides of the rule, that is, in LHS and RHS of the rules. In this model the weights represent conditional probabilities, as a more effective approach to embrace the uncertainty. The unique feature of the mentioned method is its high ability of forecasting point, intervals and probability for high order models with multiple-step ahead.

2.5 Defuzzification Process

The last step of FTS forecasting process is defuzzification which converts fuzzy sets in the RHS of the rules to a numerical value. In this process the obtained fuzzified value $f(t+1)$ is mapped into the single crisp number $\hat{y}(t+1)$. Various defuzzification methods exist in the literature. The initial method presented by Song and Chissom [8, 9] follows the principles bellow:

1. If there is only one membership with the greatest value in the output vector, then the forecasted value $\hat{y}(t+1)$ is the midpoint of the fuzzy set with this maximum membership.
2. If there is more than one consecutive maximum membership, $\hat{y}(t+1)$ will be calculated by averaging the midpoints of the corresponding fuzzy sets.
3. Otherwise, $\hat{y}(t+1)$ will be the mean value of midpoints of each fuzzy set. In other words we have $\hat{y}(t+1) = \sum_i \mu_i \cdot c_j$ for $j \in f(t)$. μ_i is the membership degree and c_j is the midpoint of each fuzzy set.

In Chen's model [13], the defuzzification process is based on the following regulations:

1. If the consequence or RHS of the rule is empty, then the $\hat{y}(t+1)$ is the midpoint of the fuzzy set $f(t)$, corresponding to a naïve forecast.
2. If the RHS in FLR consists of only one fuzzy set, $\hat{y}(t+1)$ is the midpoint of the corresponding fuzzy set.
3. Otherwise, the value of $\hat{y}(t+1)$ is the mean value of midpoints of those fuzzy sets in the RHS.

With the goal of accuracy enhancement, different types of FTS models have been proposed in the literature, with variations of defuzzification rules in Chen's model. In these models, a weighted average of mid points of the fuzzy sets in the RHS of the rules can be considered as the forecasted value. These weights are determined through different methods based on the literature, for instance fuzzy sets index numbers or number of occurrences of a transition in the data.

In weighted rule models [148] the weight of each fuzzy rule with respect to the input observations are determined and the predicted value obtained through these weights. In [149] the defuzzifized value obtained via Euclidean distances of the subscripts of FLR instead of exact matching of FLRs. In [70] the predicted value is computed by a combination of global information related to relevant FLR and the local information about latest fuzzy fluctuation. Duru and Bulut [150] proposed a model in which the latest value factor and previous error patterns are exploited to predict the next value via fuzzy integrated logical forecasting (FILF) and extended FILF (E-FILF) algorithm. Teoh et al. [122] introduced a combination of center of gravity method and adaptive expectation model to calculate the forecasted output value. In [149] the defuzzified result is obtained through Euclidean distances of the subscripts of FLR. It is worth mentioning that various studies focused on different strategies to calculate the weights in defuzzification process. For instance trend-weighted recurrence of FLRs in the FLRG [145], ordered weighted aggregation (OWA) [20], computing membership degrees in the fuzzy sets [56], employment of weight of each fuzzy logical relationship such based on the recurrence and the chronological order [31] or trend-order [84], optimal weighting vector [151] are samples of proposed methods.

In Sadaei et al. [144], the forecasted output is adjusted by considering the number of active rules and expected mean point of each rule (multiplication of weighted mean of the midpoints of their RHS consequents and the weights). In this method the importance of all the patterns are the same. In order to overcome this problem, [152] proposed high order FTS (HOFTS) model as an updated and weighted sum method in which each pattern is weighted by its activation by considering expected mean point of each rule and the activation of the rule. A time invariant, rule based and high-order method known as Weighted Interval FTS model (W[I]FTS) designed by Silva [4] uses a new defuzzification technique based on intervals. In this model, the final forecast interval is computed based on the sum of the rules intervals weighted by the membership value of each rule. In the Probabilistic Weighted Fuzzy Time Series (PWFTS) method presented in [147], the output of training process is the Probabilistic Weighted FTPG (PWFTPG), which adds weights on the LHS and the RHS to calculate their fuzzy empirical probabilities. The generated PWFTPG model is applied to produce the output in the forecasting process. This stage performed in four steps with three types of defuzzification strategies including probabilistic forecasting, interval forecasting and point forecasting.

3 Recent Advances and Challenges

In this section, we discuss the challenges and advances that we consider valuable in the area of FTS. Although many techniques to improve the performance of these models have already been presented so far, we can point out issues that still need attention. The main challenges raised in this research are: multivariate methods, MIMO (Multiple Input Multiple Output) forecasting, scalability to big data, high-dimensional times series, non-stationary FTS, streaming data and hybrid models (presented in Sections 2.4.2 and 2.4.3). Recent works have already advanced on these challenges and also bring new approaches with probabilistic and interval forecasting in FTS. We divide the next sections into Multivariate FTS, Non-Stationary FTS and Probabilistic and Interval FTS, where the works that have already advanced on the aforementioned challenges are presented.

3.1 Multivariate Fuzzy Time Series

Multivariate FTS consists of using multiple Fuzzy Time Series to solve a problem. Thus, by considering a model $\mathcal{M}$, single step ahead multivariate forecasting is denoted as $\mathbf{y_1}^{t+1} = \mathcal{M}(\mathbf{Y}^t)$. Generally, the multivariate FTS can be grouped into Multiple Input Single Output (MISO) and Multiple Input Multiple Output (MIMO) [4].

It is worth mentioning that some multivariate FTS models exploited clustering technique widely to reduce multivariate data to univariate ones, see for instance [59, 148]. This approach is based on FCM clustering or using other automatic clustering methods proposed in [37, 153]. In [154] the authors proposed High-Order Multi-Variable FTS (HMV-FTS) algorithm based on FCM clustering. Chen [155] introduced fuzzy variation group to predict TAIEX in which each FVG groups the FLRG's of each variable by their co-occurrence. Some models of first and high order multivariate FTS forecasting techniques are accessible in the literature, including first order multivariate FTS [28, 37, 67, 73] as well as existing high order multivariate FTS in [1, 18, 36, 50, 56, 72, 73, 82, 88, 108, 121, 149, 156–158].

In contrast to conventional multivariate FTS models, a novel MISO method termed as Weighted Multivariate FTS (WMVFTS) is the presented in [159], which is a first order method with some modifications in the training and forecasting procedures compared to conventional one. The Distributed Evolutionary Hyperparameter Optimization (DEHO) is also proposed for training the WMVFTS model. WMVFTS is able to handle scalability to big data and automated optimization of hyperparameters.

One of the main concerning issues of dealing with multivariate FTS is the increasing complexity of rules, as the dimension grows. Implementing Fuzzy Information Granules (FIG) to map multivariate time series into univariate time series is counted as the solution, which is based on the concept of information granule proposed in [160]. FIG represents a subset of a wider domain used in some researches [46–50]. FIGs have fuzzy sets (or linguistic terms) that are sufficiently interpretable to achieve a high level of human cognitive abstraction.

The Fuzzy Information Granule Fuzzy Time Series (FIG-FTS) method, as an extension version of univariate PWFTS model, was introduced in [161] as an adaptable and flexible model to perform high order multivariate forecasting for many steps ahead. FIG-FTS works by translating a multivariate time series into an univariate FTS data applying Fuzzy Information Granules (FIG). The FIG-FTS is a MIMO method with the capability of point, interval and probability forecasting. Moreover, in [89] researchers outlined a new long term forecasting approach to generate generalized zonary time variant fuzzy information granule (GTZ-FIG) by means of step-wise linear division (SLD) method as variable-length division method, which contain both the trend and range of fluctuations of time series. This study used self-evolving interval type-2 LSTM fuzzy neural network which is a novel recurrent neural network based on the LSTM mechanism. According to the GZT-FIGs properties, a double-network prediction model is adopted, where FIGs with different trends would be trained and predicted in two independent inference systems, respectively. This method was explored to solve the problem of long dependency of time series through implementing zonary time-variant fuzzy information granule.

In [162] a method that focuses on the problem of high-dimensional time series. The authors tachled this issue by projecting the original high-dimensional data into a low dimensional embedding space using self-organizing Kohonnen maps and later using the Weighted Multivariate FTS method (WMVFTS) for rule discovery and forecasting. In [163], an embedding-based method is used for high-dimensional non-stationary time series.

To handle streaming data, the Evolving Multivariate FTS (e-MVFTS) is proposed in [164]. e-MVFTS provides an adaptive spatio-temporal forecasting method to deal with changes in the data distribution or concept drifts in data streams.

3.2 Non-Stationary Fuzzy Time Series

In non-stationary data the statistical properties, e.g., the mean and standard deviation are not constant over time. Song and Chissom [94] proposed the initial representation of time variant FTS model. The proposed time variant approach defines a sliding window of observation W, the length of memory window and refreshing intervals. Thus the model is rebuilt from scratch along the sliding window. Some studies employed different knowledge models and weighting schemes [22, 35, 165, 166].

In order to tackle the high cost of computation related to model retraining, Alves et al. [92] implemented a further technique with incremental changes and without the requirement of total model reconstruction. The basic idea is to adopt Non-Stationary Fuzzy sets (NSFS), as introduced by [167]. NSFS extends the typical fuzzy sets by changing the membership function μ over time through changes in location, width or adding noise. The non-stationary membership function (NSMF) and the perturbation function are two ways of implementing NSFS. Despite the desirable performance of the developed model, some extreme changes in the time series data such as abrupt concept drifts or conditional variance limit its performance.

Silva et al. [11] proposed the Non-Stationary FTS (NSFTS) method to handle the issue and rectify the model adaptability. In proposed NSFTS method, the intervals adaptation against changes in the data after training process is performed by transferring sets using the mean and variance of the residuals in the forecasting process. It means that in high residual variance, a reduction in range of fuzzy sets is needed to reduce granularity. Therefore, if the bounds of U change, the sets are adapted to respond to this change as well. Thus, NSFTS method is composed of three procedures including training procedure, parameters adaptation

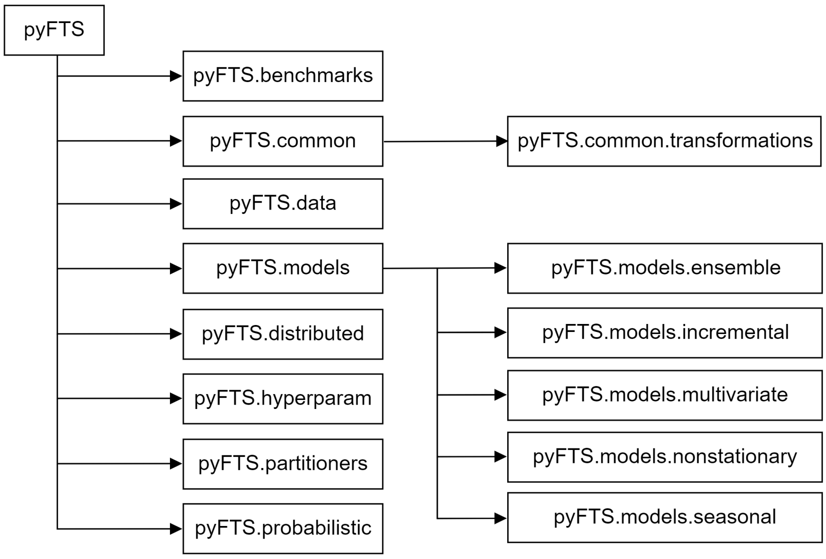

Figura 3: Architecture of the pyFTS library packages.

procedure and forecasting procedure. According to the membership function adaptability of NSFTS as the statistical properties of time series vary, NSFTS can respond to the bias and variance shifts by implementing the changes in parameters of the fuzzy sets. Furthermore, this model is able to adapt to changes in the time series just by adjusting the fuzzy sets, without modifications in the model structure and the knowledge base. As a result, the proposed method is cheaper compared to other ones involving model retraining.

3.3 Probabilistic and Interval Fuzzy Time Series

The concept of probabilistic forecasting approaches was put forth by [168], defining two major categories of forecasting including intervals and probability distributions to deal with uncertainties that limit the predictability of forecasting models [169–171], specially by increasing the horizon of forecasting in the case of point forecasting.

Although the probabilistic forecasting methods provide the uncertainty appraisal with the whole Universe of Discourse, they are limited by expensive cost of computation and time consuming tasks. In addition, the available models are insufficient in case of forecasting intervals or probability distributions in FTS domain. In order to eliminate this gap, novel FTS approaches were proposed in [4], namely [$\mathbb{I}$]FTS and W[$\mathbb{I}$]FTS as fast, scalable, time invariant, rule based and high-order method to bind the fuzzy uncertainty of the FTS models providing interval forecasting. Ensemble FTS method is also proposed in [4] for probabilistic forecasting by employing the partitioning and ordering uncertainty.

Although these methodologies are flexible to easily apply for interval and probabilistic FTS, the intervals of [$\mathbb{I}$]FTS and W[$\mathbb{I}$]FTS do not provide probability distributions and the intervals do not consist of probabilistic uncertainty. Therefore, in order to handle the mentioned drawback, a recent new integrated model known as Probabilistic weighted FTS (PWFTS) was proposed for point, interval and probabilistic forecasting in [147], which is a cheaper and parsimonious model compared to others.

4 A Fuzzy Time Series Library for Python

The pyFTS is an open source library for building Fuzzy Time Series models in python, developed by the Laboratory of Machine Intelligence and Data Science (MINDS) at the Federal University of Minas Gerais (UFMG).

The architecture of the pyFTS library is shown in Figure 3. The models available in pyFTS can be accessed by the *pyFTS.models* package, extending from conventional models as published in [13], to more advanced models presented in Section 3.

In addition to the FTS models, pyFTS also provides through the *pyFTS.benchmarks* package other time series forecasting models, such as ARIMA and Naive, to facilitate benchmarking. With the pyFTS.data package, you can upload several common time series databases such as TAIEX, NASDAQ, S&P 500, passengers, etc. It is also possible to make transformations on this data that can be used for pre-processing and/or post-processing through the pyFTS.common.transformations package.

Two packages, pyFTS.distributed and pyFTS.hyperparam, are still under development. pyFTS.distributed trains and predicts models in a distributed fashion and is now available for the MVFTS model. pyFTS.hyperparam does the hyperparameter optimization of the models and is now available for the WHOFTS model. In order to extend these packages to other models and automate the entire fuzzy time series prediction pipeline, an Automated Machine Learning (AutoML) project for pyFTS is underway.

In this section, a two-part pyFTS tutorial is presented. The first one (A) aims to present the training and prediction procedures step by step, as shown in Figure 1. The second part (B) aims to demonstrate the use of several univariate and multivariate models and compare their results. Other tutorials can be accessed at https://github.com/PYFTS/notebooks.

To get started, one needs to install pyFTS and upload the database:

```
pip install pyFTS
```

In this tutorial, the average daily index of NASDAQ stocks will be predicted. For this, the data will be divided into training and testing and into univariate and multivariate time series. An example of the training data can be seen in Figure 4.

```
#Univariate time series
from pyFTS.data import NASDAQ
data = NASDAQ.get_dataframe()
train = data['avg'].values[:2000]
test = data['avg'].values[2000:]

#Multivariate time series
import pandas as pd
data['Date'] = pd.to_datetime(data["Date"], format='%Y-%m-%d')
train_mv = data.iloc[:2000]
test_mv = data.iloc[2000:]

#Data visualization
import matplotlib.pyplot as plt
fig = plt.subplots(figsize=[20,5])
plt.plot(train)
pass
```

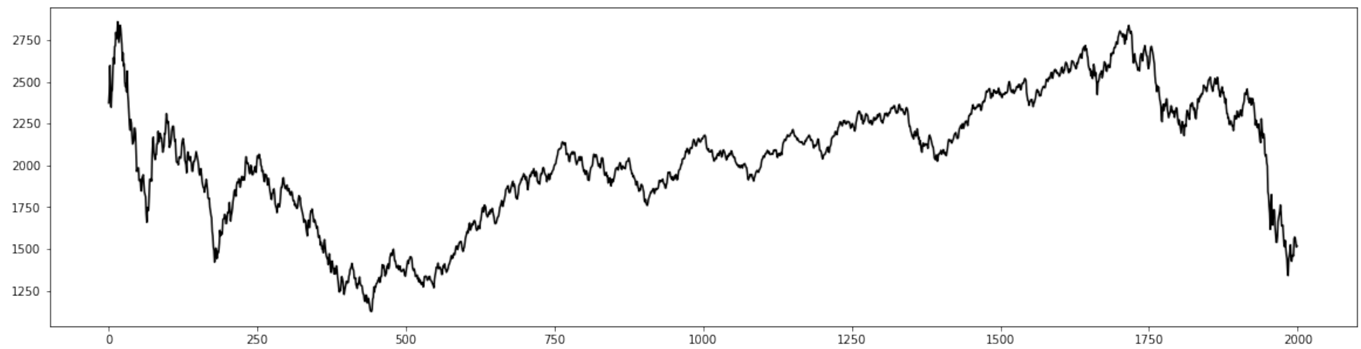

Figura 4: Training sample from the NASDAQ database.

The transformed data will not be used in this tutorial, but here is an example of how to use the Differentiation transformation. The result can be seen in Figure 5.

```
from pyFTS.common import Transformations
diff = Transformations.Differential(1)
fig = plt.subplots(figsize=[20,5])
plt.plot(diff.apply(train), label="glo_avg")
pass
```

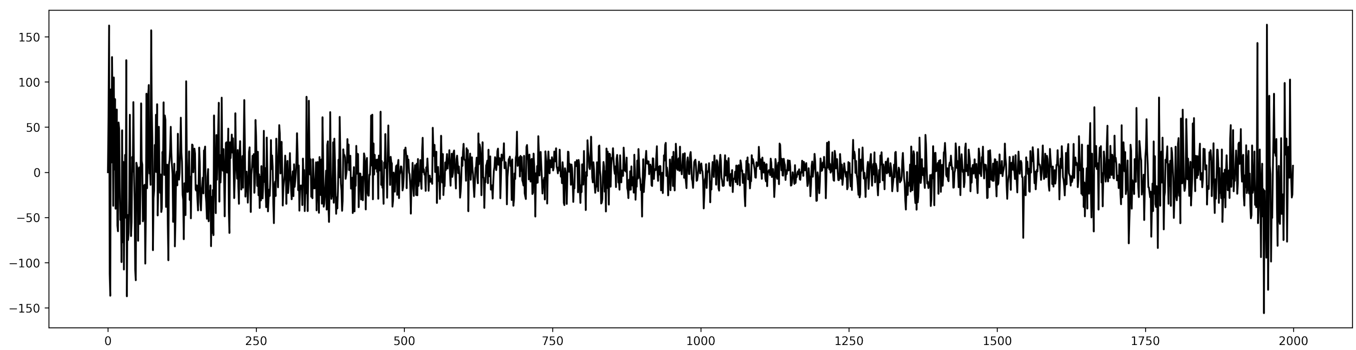

Figura 5: Result of applying differentiation in the NASDAQ database training sample.

4.1 Tutorial A: FTS model training and prediction process

Training procedures

1. Universe of Discourse partitioning and creation of Fuzzy Sets. To partition U, *Grid partitioning* will be used, but other methods are also available in the *pyFTS.partitioners* package. Parameters:
   * data: the time series training data;
   * npart: the minimum number of partitions/fuzzy sets;
   * mf: the membership function that will be used, which by default is triangular (trimf). The various membership functions can be found in *pyFTS.common.Membership*;
   * transformation: if any transformation is used in the series, it must be informed.

   Figure 6 shows the fuzzy sets created for the variable avg.

```
from pyFTS.partitioners import Grid
fig, ax = plt.subplots(nrows=1, ncols=1,figsize=[22,3])
part = Grid.GridPartitioner(data=train,npart=35)
part.plot(ax)
```

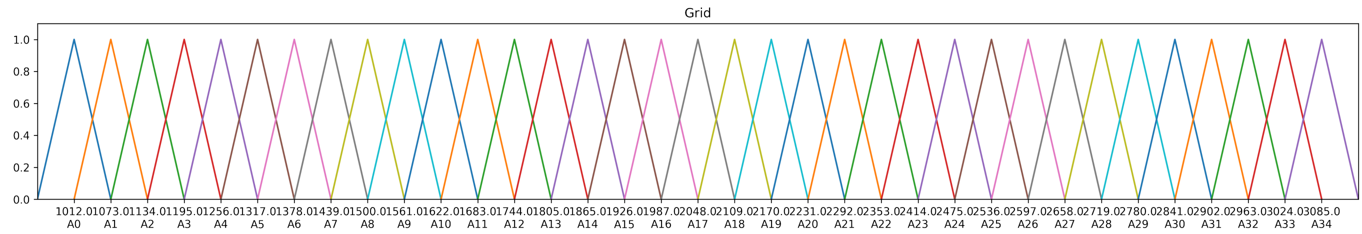

Figura 6: Caption

2. Fuzzification: in this step the numerical values of the time series Y will be converted into fuzzy values of the linguistic variable $\tilde{A}$, giving rise to the fuzzy time series F.

```
from pyFTS.common import FuzzySet as fz
F = fz.fuzzyfy_series(train, part.sets)
print(F)
```

   $\cdots$ ['A19', 'A19', 'A19', 'A19', 'A19', 'A18', 'A18 ', 'A18', 'A18', 'A18', 'A18', 'A18', 'A18', 'A18', 'A18'] $\cdots$

3. Knowledge Extraction:

   3.1. Generation of fuzzy logical rules (FLRs):

```
from pyFTS.common import FLR
rules = FLR.generate_non_recurrent_flrs(F)
print([str(k) for k in rules[0:10]])
```

   [$\cdots$,'A22 $\rightarrow$ A23', 'A23 $\rightarrow$ A26', 'A26 $\rightarrow$ A24', 'A24 $\rightarrow$ A22', 'A23 $\rightarrow$ A24', 'A24 $\rightarrow$ A26', 'A26 $\rightarrow$ A27', 'A27 $\rightarrow$ A26', 'A26 $\rightarrow$ A28', 'A28 $\rightarrow$ A28', $\cdots$]

   3.2. Group fuzzy logical rule (FLRGs): now, the grouping of the FLRs to generate the FTS model will be done. Models implemented in pyFTS are available in the *pyFTS.models* package. Here the *ConventionalFTS* model, published in [13], will be used. The fit method trains the model from the training data and the linguistic variable already built by the partitioner.

```
from pyFTS.models import chen
model = chen.ConventionalFTS(partitioner=part)
model.fit(train)
print(model)
```

   Conventional FTS:
   $A2 \rightarrow A2, A3$
   $A3 \rightarrow A2, A3, A4$
   $A4 \rightarrow A3, A4, A5$
   $A5 \rightarrow A4, A5, A6, A7$
   $A6 \rightarrow A5, A6, A7$
   $\vdots$
   $A28 \rightarrow A26, A27, A28, A29$
   $A29 \rightarrow A28, A29, A30$
   $A30 \rightarrow A29, A30$

Forecasting procedures

1. Fuzzification and Rule Matching: the input value $y(t)$ will be converted into fuzzy values of the linguistic variable $\tilde{A}$, generating the value $f(t)$. For this, method *get_maximum_membership_fuzzyset* will be used so that the set with the highest relevance is chosen.

```
from pyFTS.common import FuzzySet as fz
f = fz.get_maximum_membership_fuzzyset(18876, part.sets)
print(f)

```

A_0 : trimf([950.5879428571428, 1011.582, 1072.576057142857])

2. Defuzzification: now it is necessary to convert $f(t+1)$ to a numerical value. The *predict* method uses the already trained model to make predictions. There are three possible forecast types, indicated by the 'type' parameter: 'point' (default), 'interval' and 'distribution'. Read about the model used, as not all of them work with all forecast types. Finally, the 'steps_ahead' parameter indicates the forecast horizon, or how many steps ahead you want to forecast.

```
model.predict([18876], type='point', steps_ahead=1)

```

[2810.9066857142834]

4.2 Tutorial B: forecast with different univariate and multivariate models

Univariate Models: HighOrderFTS, WeightedHighOrderFTS and ProbabilisticWeightedFTS models will be used with order 1, 2 and 3 using the same partitioning as in Tutorial A. Figure 7 shows the prediction of 100 test data values.

```
from pyFTS.common import Util
from pyFTS.models import hofts, pwfts
import numpy as np

models = []
fig, ax = plt.subplots(nrows=1, ncols=1, figsize=[20,3])
ax.plot(test[:100], label='Original')
rows = []
for method in [hofts.HighOrderFTS, hofts.WeightedHighOrderFTS, pwfts.ProbabilisticWeightedFTS]:
  for order in [1, 2, 3]:
    model = method(partitioner=part, order=order)
    model.shortname += str(order)
    model.fit(train)
    forecasts = model.predict(test)
    for k in np.arange(order):
      forecasts.insert(0,None)
    ax.plot(forecasts[:100], label=model.shortname)
    models.append(model.shortname)
    Util.persist_obj(model, model.shortname)
    del(model)
handles, labels = ax.get_legend_handles_labels()
lgd = ax.legend(handles, labels, loc=2, bbox_to_anchor=(1, 1))
```

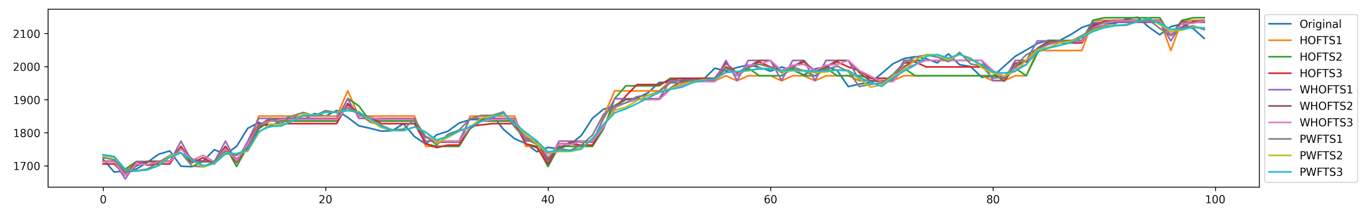

Figura 7: Performance of the HighOrderFTS, WeightedHighOrderFTS and ProbabilisticWeightedFTS for point forecast.

Multivariate Models: For the multivariate models we will use two more variables besides avg, Open and High. Figure 8 shows the fuzzy sets created for each variable.

```
from pyFTS.partitioners import Grid, Util as pUtil
from pyFTS.models.multivariate import common, variable, mvfts
from pyFTS.models.seasonal import partitioner as seasonal
from pyFTS.models.seasonal.common import DateTime

vop = variable.Variable("OPEN", data_label="Open", alias='Open', partitioner=Grid.GridPartitioner,
npart=35, data=train_mv)

```

```python
vhi = variable.Variable("HIGH", data_label="High", alias='High', partitioner=Grid.GridPartitioner,
npart=35, data=train_mv)

vavg = variable.Variable("AVG", data_label="avg", alias='avg', partitioner=Grid.GridPartitioner, npart
=35, data=train_mv)

fig, ax = plt.subplots(nrows=3, ncols=1, figsize=[20,6], dpi=300)
vop.partitioner.plot(ax[0])
vhi.partitioner.plot(ax[1])
vavg.partitioner.plot(ax[2])
plt.tight_layout()
```

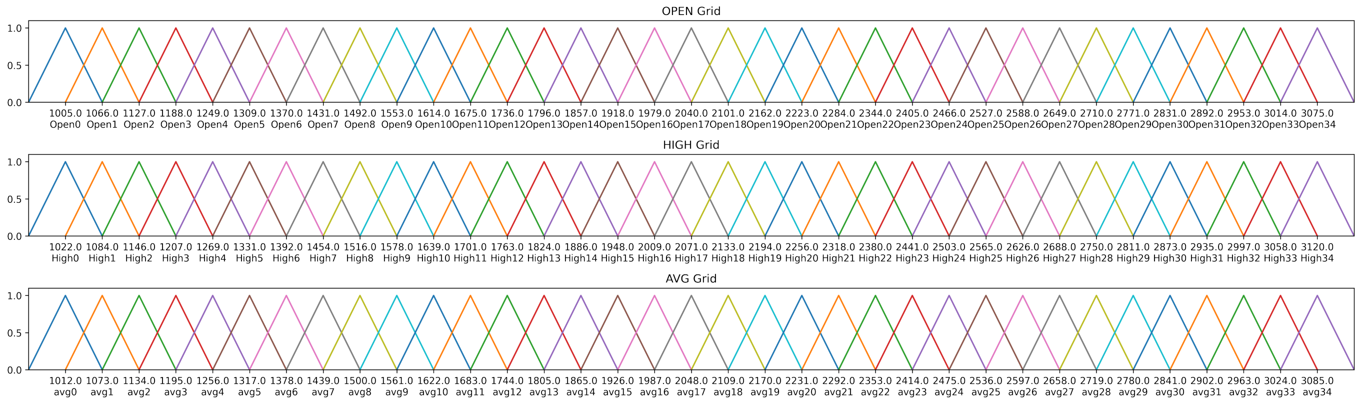

Figura 8: Variables and partitioning.

The models used are: WeightedMVFTS, MVFTS and ClusteredMVFTS. Figure 9 shows the prediction of 100 test data values.

```python
from pyFTS.models.multivariate import mvfts, wmvfts, cmvfts, grid

fig, ax = plt.subplots(nrows=1, ncols=1, figsize=[20,3], dpi=300)

parameters = [
    {},{},
    {'order':2, 'knn': 1},
]

ax.plot(test[100:200],label='Original')

for ct, method in enumerate([mvfts.MVFTS, wmvfts.WeightedMVFTS,
                             cmvfts.ClusteredMVFTS]):
  try:
    if method != cmvfts.ClusteredMVFTS:
      model = method(explanatory_variables=[vop, vhi, vavg], target_variable=vavg, **parameters[ct])
    else:
      fs = grid.GridCluster(explanatory_variables=[vop, vhi, vavg], target_variable=vavg)
      model = method(explanatory_variables=[vop, vhi, vavg], target_variable=vavg, partitioner=fs, **
parameters[ct])

    model.shortname += str(ct)
    model.fit(train_mv)

    models.append(model.shortname)

    forecasts = model.predict(test_mv)

    ax.plot(forecasts[100:200], label=model.shortname)

    Util.persist_obj(model, model.shortname)

    del(model)
  except Exception as ex:
    print(method, parameters[ct])
    print(ex)

handles, labels = ax.get_legend_handles_labels()
lgd = ax.legend(handles, labels, loc=2, bbox_to_anchor=(1, 1))
pass
```

Finally, Table 4.2 presents the forecast nRMSE of the univariate and multivariate models.

Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 19, Iss. 2, pp. 29-50, 2021

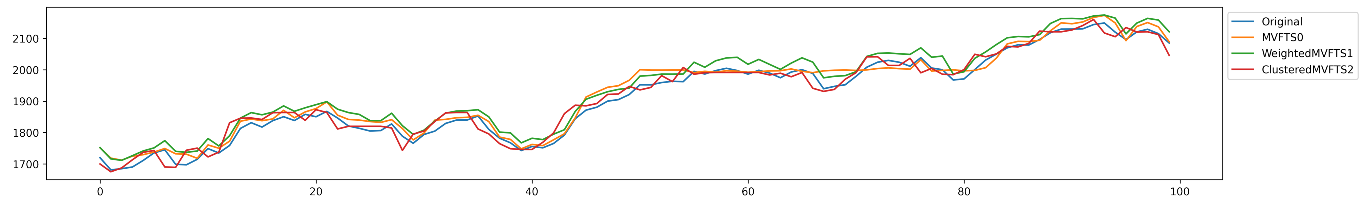

Figura 9: Performance of the WeightedMVFTS, MVFTS and ClusteredMVFTS for point forecast.

```
from pyFTS.benchmarks import Measures

rows = []
for file in models:
  try:
    model = Util.load_obj(file)

    row = [model.shortname, model.order,len(model)]

    if model.is_multivariate:
      rmse,_,_ = Measures.get_point_statistics(test_mv, model)
      row.append(rmse/np.mean(test_mv['avg']))
    else:
      rmse,_,_ = Measures.get_point_statistics(test, model)
      row.append(rmse/np.mean(test))

    rows.append(row)
  except:
    pass
pd.DataFrame(rows,columns=["Model","Order","Size","RMSE"]).sort_values(["RMSE","Size"])
```

Tabela 2: nRMSE of the prediction of univariate and multivariate models.

Model	Order	Size	nRMSE
ClusteredMVFTS [4]	1	1682	0.009658
MVFTS [4]	1	528	0.010021
WeightedMVFTS [161]	2	528	0.010966
WHOFTS3 [4]	3	760	0.335044
HOFTS3 [152]	3	760	0.335252
WHOFTS2 [4]	2	176	0.342641
HOFTS1 [152]	1	31	0.343276
HOFTS2 [152]	2	176	0.343348
WHOFTS1 [4]	1	31	0.344671
PWFTS1 [147]	1	31	0.345139
PWFTS3 [147]	3	760	0.346099
PWFTS2 [147]	2	176	0.346585

5. CONCLUSION

In this paper we presented a tutorial on FTS methods. A review of the relevant literature is carried out, offering a basis on the main concepts and models based on FTS for different time series and different types of forecasts. We also direct readers to contribute to this area of research, pointing out the challenges and recent research aimed at filling some of these gaps. Finally, we illustrated how to apply these concepts in practice by using the pyFTS library to predict the average daily index of NASDAQ stocks with univariate and multivariate FTS methods.

6. Acknowledgements

Patrícia de Oliveira e Lucas would like to thank the support given by the PBQS program of IFNMG. This work has been supported by the Brazilian agencies (i) National Council for Scientific and Technological Development (CNPq), Grant no. 312991/2020-7; (ii) Coordination for the Improvement of Higher Education Personnel (CAPES) and (iii) Foundation for Research of the State of Minas Gerais (FAPEMIG, in Portuguese), Grant no. APQ-01779-21. MINDS Laboratory – https://minds.eng.ufmg.br/.

REFERENCES

[1] M. Shah. “Fuzzy based trend mapping and forecasting for time series data”. *Expert Systems with Applications*, vol. 39, no. 7, pp. 6351–6358, 2012.

[2] M.-Y. Chen. “A high-order fuzzy time series forecasting model for internet stock trading”. *Future Generation Computer Systems*, vol. 37, pp. 461–467, 2014.

[3] O. C. Yolcu, U. Yolcu, E. Egrioglu and C. H. Aladag. “High order fuzzy time series forecasting method based on an intersection operation”. *Applied Mathematical Modelling*, vol. 40, no. 19-20, pp. 8750–8765, 2016.

[4] P. C. L. Silva. “Scalable Models for Probabilistic Forecasting with Fuzzy Time Series”. Ph.D. thesis, UFMG, 09 2019.

[5] M. Bose and K. Mali. “Designing fuzzy time series forecasting models: A survey”. *International Journal of Approximate Reasoning*, vol. 111, pp. 78–99, 8 2019.

[6] S. Panigrahi and H. S. Behera. “Fuzzy Time Series Forecasting: A Survey”. *Advances in Intelligent Systems and Computing*, vol. 990, pp. 641–651, 2020.

[7] L. Palomero, V. García and J. S. Sánchez. “Fuzzy-Based Time Series Forecasting and Modelling: A Bibliometric Analysis”. *Applied Sciences 2022, Vol. 12, Page 6894*, vol. 12, pp. 6894, 7 2022.

[8] Q. Song and B. S. Chissom. “Fuzzy time series and its models”. *Fuzzy sets and systems*, vol. 54, no. 3, pp. 269–277, 1993.

[9] Q. Song and B. S. Chissom. “Forecasting enrollments with fuzzy time series—Part I”. *Fuzzy sets and systems*, vol. 54, no. 1, pp. 1–9, 1993.

[10] Q. Song and B. Chissom. “Forecasting enrollments with fuzzy time series—Part II”. *Fuzzy Sets and Systems*, vol. 62, pp. 1–8, 02 1994.

[11] P. C. de Lima e Silva, C. A. Severiano, M. A. Alves, R. Silva, M. W. Cohen and F. G. Guimarães. “Forecasting in non-stationary environments with fuzzy time series”. *Applied Soft Computing*, vol. 97, pp. 106825, 2020.

[12] S. Panigrahi and H. S. Behera. “A study on leading machine learning techniques for high order fuzzy time series forecasting”. *Engineering Applications of Artificial Intelligence*, vol. 87, pp. 103245, 2020.

[13] S.-M. Chen *et al.*. “Forecasting enrollments based on fuzzy time series”. *Fuzzy sets and systems*, vol. 81, no. 3, pp. 311–319, 1996.

[14] S.-T. Li and Y.-C. Cheng. “Deterministic fuzzy time series model for forecasting enrollments”. *Computers & Mathematics with Applications*, vol. 53, no. 12, pp. 1904–1920, 2007.

[15] C. H. Aladag, M. A. Basaran, E. Egrioglu, U. Yolcu and V. R. Uslu. “Forecasting in high order fuzzy times series by using neural networks to define fuzzy relations”. *Expert Systems with Applications*, vol. 36, no. 3, pp. 4228–4231, 2009.

[16] C. H. Aladag, U. Yolcu and E. Egrioglu. “A high order fuzzy time series forecasting model based on adaptive expectation and artificial neural networks”. *Mathematics and Computers in Simulation*, vol. 81, no. 4, pp. 875–882, 2010.

[17] T. H.-K. Yu and K.-H. Huarng. “A neural network-based fuzzy time series model to improve forecasting”. *Expert Systems with Applications*, vol. 37, no. 4, pp. 3366–3372, 2010.

[18] T. H.-K. Yu and K.-H. Huarng. “A bivariate fuzzy time series model to forecast the TAIEX”. *Expert Systems with Applications*, vol. 34, no. 4, pp. 2945–2952, 2008.

[19] H. J. Teoh, T.-L. Chen, C.-H. Cheng and H.-H. Chu. “A hybrid multi-order fuzzy time series for forecasting stock markets”. *Expert Systems with Applications*, vol. 36, no. 4, pp. 7888–7897, 2009.

[20] S. S. Gangwar and S. Kumar. “Partitions based computational method for high-order fuzzy time series forecasting”. *Expert Systems with Applications*, vol. 39, no. 15, pp. 12158–12164, 2012.

[21] K. Bisht and S. Kumar. “Fuzzy time series forecasting method based on hesitant fuzzy sets”. *Expert Systems with Applications*, vol. 64, pp. 557–568, 2016.

[22] H.-T. Liu and M.-L. Wei. “An improved fuzzy forecasting method for seasonal time series”. *Expert Systems with Applications*, vol. 37, no. 9, pp. 6310–6318, 2010.

[23] S. R. Singh. “A robust method of forecasting based on fuzzy time series”. *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 472–484, 2007.

[24] S. R. Singh. "A simple method of forecasting based on fuzzy time series". *Applied mathematics and computation*, vol. 186, no. 1, pp. 330–339, 2007.

[25] S. R. Singh. "A computational method of forecasting based on high-order fuzzy time series". *Expert Systems with Applications*, vol. 36, no. 7, pp. 10551–10559, 2009.

[26] R. Efendi, Z. Ismail and M. M. Deris. "A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand". *Applied Soft Computing*, vol. 28, pp. 422–430, 2015.

[27] W. Qiu, X. Liu and L. Wang. "Forecasting Shanghai Composite Index Based on Fuzzy Time Series and Improved C-Fuzzy Decision Trees". *Expert Syst. Appl.*, vol. 39, no. 9, pp. 7680–7689, July 2012.

[28] K. Huarng. "Heuristic models of fuzzy time series for forecasting". *Fuzzy sets and systems*, vol. 123, no. 3, pp. 369–386, 2001.

[29] P.-T. Chang. "Fuzzy seasonality forecasting". *Fuzzy sets and systems*, vol. 90, no. 1, pp. 1–10, 1997.

[30] H.-K. Yu. "Weighted fuzzy time series models for TAIEX forecasting". *Physica A: Statistical Mechanics and its Applications*, vol. 349, no. 3-4, pp. 609–624, 2005.

[31] A. Rubio, J. D. Bermúdez and E. Vercher. "Forecasting portfolio returns using weighted fuzzy time series methods". *International Journal of Approximate Reasoning*, vol. 75, pp. 1–12, 2016.

[32] C.-H. Cheng and C.-H. Chen. "Fuzzy time series model based on weighted association rule for financial market forecasting". *Expert Systems*, vol. 35, no. 4, pp. e12271, 2018.

[33] H. Liu, F. Hussain, C. L. Tan and M. Dash. "Discretization: An enabling technique". *Data mining and knowledge discovery*, vol. 6, no. 4, pp. 393–423, 2002.

[34] K. Huarng. "Effective lengths of intervals to improve forecasting in fuzzy time series". *Fuzzy Sets Syst.*, vol. 123, pp. 387–394, 2001.

[35] T. A. Jilani and S. M. A. Burney. "A refined fuzzy time series model for stock market forecasting". *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 12, pp. 2857–2862, 2008.

[36] S.-M. Chen and K. Tanuwijaya. "Multivariate fuzzy forecasting based on fuzzy time series and automatic clustering techniques". *Expert Systems with Applications*, vol. 38, no. 8, pp. 10594–10605, 2011.

[37] S.-M. Chen and K. Tanuwijaya. "Fuzzy forecasting based on high-order fuzzy logical relationships and automatic clustering techniques". *Expert Systems with Applications*, vol. 38, no. 12, pp. 15425–15437, 2011.

[38] P. Singh and B. Borah. "An efficient time series forecasting model based on fuzzy time series". *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2443–2457, 2013.

[39] C.-H. Cheng, J.-R. Chang and C.-A. Yeh. "Entropy-based and trapezoid fuzzification-based fuzzy time series approaches for forecasting IT project cost". *Technological Forecasting and Social Change*, vol. 73, no. 5, pp. 524–542, 2006.

[40] C.-H. Cheng, J.-W. Wang and C.-H. Li. "Forecasting the number of outpatient visits using a new fuzzy time series based on weighted-transitional matrix". *Expert Systems with Applications*, vol. 34, no. 4, pp. 2568–2575, 2008.

[41] Y.-S. Chen, C.-H. Cheng and W.-L. Tsai. "Modeling fitting-function-based fuzzy time series patterns for evolving stock index forecasting". *Applied intelligence*, vol. 41, no. 2, pp. 327–347, 2014.

[42] C.-H. Cheng, T.-L. Chen and L.-Y. Wei. "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting". *Information Sciences*, vol. 180, no. 9, pp. 1610–1629, 2010.

[43] M.-Y. Chen and B.-T. Chen. "Online fuzzy time series analysis based on entropy discretization and a Fast Fourier Transform". *Applied Soft Computing*, vol. 14, pp. 156–166, 2014.

[44] M.-Y. Chen and B.-T. Chen. "A hybrid fuzzy time series model based on granular computing for stock price forecasting". *Information Sciences*, vol. 294, pp. 227–241, 2015.

[45] J. Wang, H. Li and H. Lu. "Application of a novel early warning system based on fuzzy time series in urban air quality forecasting in China". *Applied Soft Computing*, vol. 71, pp. 783–799, 2018.

[46] W. Lu, W. Pedrycz, X. Liu, J. Yang and P. Li. "The modeling of time series based on fuzzy information granules". *Expert Systems with Applications*, vol. 41, no. 8, pp. 3799–3808, 2014.

[47] W. Lu, X. Chen, W. Pedrycz, X. Liu and J. Yang. "Using interval information granules to improve forecasting in fuzzy time series". *International Journal of Approximate Reasoning*, vol. 57, pp. 1–18, 2015.

[48] L. Wang, X. Liu, W. Pedrycz and Y. Shao. "Determination of temporal information granules to improve forecasting in fuzzy time series". *Expert Systems with Applications*, vol. 41, no. 6, pp. 3134–3142, 2014.

[49] W. Wang, W. Pedrycz and X. Liu. "Time series long-term forecasting model based on information granules and fuzzy clustering". *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 17–24, 2015.

[50] W. Deng, G. Wang, X. Zhang, J. Xu and G. Li. "A multi-granularity combined prediction model based on fuzzy trend forecasting and particle swarm techniques". *Neurocomputing*, vol. 173, pp. 1671–1682, 2016.

[51] R. Dong and W. Pedrycz. "A granular time series approach to long-term forecasting and trend forecasting". *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 13, pp. 3253–3270, 2008.

[52] R. Al-Hmouz, W. Pedrycz and A. Balamash. "Description and prediction of time series: A general framework of Granular Computing". *Expert Systems with Applications*, vol. 42, no. 10, pp. 4830–4839, 2015.

[53] J. T. Yao, A. V. Vasilakos and W. Pedrycz. "Granular computing: perspectives and challenges". *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1977–1989, 2013.

[54] X. Yang, F. Yu and W. Pedrycz. "Long-term forecasting of time series based on linear fuzzy information granules and fuzzy inference system". *International Journal of Approximate Reasoning*, vol. 81, pp. 1–27, 2017.

[55] M. Hirota and R. Ballini. *Granular Models for Time-Series Forecasting*, pp. 949 – 967. 07 2008.

[56] S. Askari, N. Montazerin and M. F. Zarandi. "A clustering based forecasting algorithm for multivariable fuzzy time series using linear combinations of independent variables". *Applied Soft Computing*, vol. 35, pp. 151–160, 2015.

[57] C.-H. Cheng, G.-W. Cheng and J.-W. Wang. "Multi-attribute fuzzy time series method based on fuzzy clustering". *Expert systems with applications*, vol. 34, no. 2, pp. 1235–1242, 2008.

[58] C.-H. Cheng, L.-Y. Wei and Y.-S. Chen. "Fusion ANFIS models based on multi-stock volatility causality for TAIEX forecasting". *Neurocomputing*, vol. 72, no. 16-18, pp. 3462–3468, 2009.

[59] S.-T. Li, Y.-C. Cheng and S.-Y. Lin. "A FCM-based deterministic forecasting model for fuzzy time series". *Computers & Mathematics with Applications*, vol. 56, no. 12, pp. 3052–3063, 2008.

[60] S.-T. Li, S.-C. Kuo, Y.-C. Cheng and C.-C. Chen. "Deterministic vector long-term forecasting for fuzzy time series". *Fuzzy Sets and Systems*, vol. 161, no. 13, pp. 1852–1870, 2010.

[61] 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*, vol. 40, no. 3, pp. 854–857, 2013.

[62] E. Egrioglu. "PSO-based high order time invariant fuzzy time series method: Application to stock exchange data". *Economic Modelling*, vol. 38, pp. 633–639, 2014.

[63] V. R. Uslu, E. Bas, U. Yolcu and E. Egrioglu. "A fuzzy time series approach based on weights determined by the number of recurrences of fuzzy relations". *Swarm and Evolutionary Computation*, vol. 15, pp. 19–26, 2014.

[64] L.-Y. Wei, C.-H. Cheng and H.-H. Wu. "A hybrid ANFIS based on n-period moving average model to forecast TAIEX stock". *Applied Soft Computing*, vol. 19, pp. 86–92, 2014.

[65] S.-H. Cheng, S.-M. Chen and W.-S. Jian. "Fuzzy time series forecasting based on fuzzy logical relationships and similarity measures". *Information Sciences*, vol. 327, pp. 272–287, 2016.

[66] N. G. Dincer and Ö. Akkus. "A new fuzzy time series model based on robust clustering for forecasting of air pollution". *Ecol. Informatics*, vol. 43, pp. 157–164, 2018.

[67] H. Izakian, W. Pedrycz and I. Jamal. "Fuzzy clustering of time series data using dynamic time warping distance". *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 235–244, 2015.

[68] M. Bahrepour, M.-R. Akbarzadeh-T, M. Yaghoobi and M.-B. Naghibi-S. "An adaptive ordered fuzzy time series with application to FOREX". *Expert Systems with Applications*, vol. 38, no. 1, pp. 475–485, 2011.

[69] P. Maji and S. K. Pal. "Rough set based generalized fuzzy C-means algorithm and quantitative indices". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 6, pp. 1529–1540, 2007.

[70] Y.-L. Huang, S.-J. Horng, M. He, P. Fan, T.-W. Kao, M. K. Khan, J.-L. Lai and I.-H. Kuo. "A hybrid forecasting model for enrollments based on aggregated fuzzy time series and particle swarm optimization". *Expert Systems with Applications*, vol. 38, no. 7, pp. 8014–8023, 2011.

[71] S.-M. Chen and P.-Y. Kao. "TAIEX forecasting based on fuzzy time series, particle swarm optimization techniques and support vector machines". *Information Sciences*, vol. 247, pp. 62–71, 2013.

[72] P. Singh and B. Borah. "Forecasting stock index price based on M-factors fuzzy time series and particle swarm optimization". *International Journal of Approximate Reasoning*, vol. 55, no. 3, pp. 812–833, 2014.

[73] L.-Y. Hsu, S.-J. Horng, T.-W. Kao, Y.-H. Chen, R.-S. Run, R.-J. Chen, J.-L. Lai and I.-H. Kuo. "Temperature prediction and TAIFEX forecasting based on fuzzy relationships and MTPSO techniques". *Expert Systems with Applications*, vol. 37, no. 4, pp. 2756–2770, 2010.

[74] I.-H. Kuo, S.-J. Horng, T.-W. Kao, T.-L. Lin, C.-L. Lee and Y. Pan. "An improved method for forecasting enrollments based on fuzzy time series and particle swarm optimization". *Expert Systems with applications*, vol. 36, no. 3, pp. 6108–6117, 2009.

[75] I.-H. Kuo, S.-J. Horng, Y.-H. Chen, R.-S. Run, T.-W. Kao, R.-J. Chen, J.-L. Lai and T.-L. Lin. "Forecasting TAIFEX based on fuzzy time series and particle swarm optimization". *Expert Systems with Applications*, vol. 37, no. 2, pp. 1494–1502, 2010.

[76] S.-M. Chen and N.-Y. Chung. "Forecasting enrollments using high-order fuzzy time series and genetic algorithms". *International Journal of intelligent systems*, vol. 21, no. 5, pp. 485–501, 2006.

[77] R. Enayatifar, H. J. Sadaei, A. H. Abdullah and A. Gani. "Imperialist competitive algorithm combined with refined high-order weighted fuzzy time series (RHWFTS–ICA) for short term load forecasting". *Energy Conversion and Management*, vol. 76, pp. 1104–1116, 2013.

[78] W. Zhang, S. Zhang, S. Zhang, D. Yu and N. Huang. "A novel method based on FTS with both GA-FCM and multifactor BPNN for stock forecasting". *Soft Computing*, vol. 23, no. 16, pp. 6979–6994, 2019.

[79] Q. Cai, D. Zhang, W. Zheng and S. C. Leung. "A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression". *Knowledge-Based Systems*, vol. 74, pp. 61–68, 2015.

[80] F. Mirzaei Talarposhti, H. Javedani Sadaei, R. Enayatifar, F. Gadelha Guimarães, M. Mahmud and T. Eslami. "Stock market forecasting by using a hybrid model of exponential fuzzy time series". *International Journal of Approximate Reasoning*, vol. 70, pp. 79–98, 2016.

[81] H. J. Sadaei, F. G. Guimarães, C. J. da Silva, M. H. Lee and T. Eslami. "Short-term load forecasting method based on fuzzy time series, seasonality and long memory process". *International Journal of Approximate Reasoning*, vol. 83, pp. 196–217, 2017.

[82] L.-W. Lee, L.-H. Wang and S.-M. Chen. "Temperature prediction and TAIFEX forecasting based on high-order fuzzy logical relationships and genetic simulated annealing techniques". *Expert Systems with Applications*, vol. 34, no. 1, pp. 328–336, 2008.

[83] G. A. Miller. "The magical number seven plus or minus two: some limits on our capacity for processing information." *Psychological review*, vol. 63 2, pp. 81–97, 1956.

[84] A. Rubio, J. D. Bermúdez and E. Vercher. "Improving stock index forecasts by using a new weighted fuzzy-trend time series method". *Expert Systems with Applications*, vol. 76, pp. 12–20, 2017.

[85] M. Bose and K. Mali. "A novel data partitioning and rule selection technique for modeling high-order fuzzy time series". *Applied Soft Computing*, vol. 63, pp. 87–96, 2018.

[86] E. Egrioglu, E. Bas, U. Yolcu and M. Y. Chen. "Picture fuzzy time series: Defining, modeling and creating a new forecasting method". *Engineering Applications of Artificial Intelligence*, vol. 88, pp. 103367, 2020.

[87] A. Pinto, P. Silva, F. Guimarães and E. P. de Aguiar. "Séries Temporais Fuzzy do Tipo-2". 11 2020.

[88] N. Bajestani and A. Zare. "Forecasting TAIEX using improved type 2 fuzzy time series". *Expert Syst. Appl.*, vol. 38, pp. 5816–5821, 05 2011.

[89] C. Luo and H. Wang. "Fuzzy forecasting for long-term time series based on time-variant fuzzy information granules". *Applied Soft Computing*, vol. 88, pp. 106046, 2020.

[90] A. C. V. Pinto, T. E. Fernandes, P. C. L. Silva, F. G. Guimarães, C. Wagner and E. Pestana de Aguiar. "Interval type-2 fuzzy set based time series forecasting using a data-driven partitioning approach". *Evolving Systems*, vol. 13, pp. 703–721, 2022.

[91] Q. Dong and X. Ma. "Enhanced fuzzy time series forecasting model based on hesitant differential fuzzy sets and error learning". *Expert Systems with Applications*, vol. 166, pp. 114056, 2021.

[92] M. Alves, P. Silva, C. Jr, G. Vieira, F. Guimarães and H. Sadaei. "An extension of nonstationary fuzzy sets to heteroskedastic fuzzy time series". 04 2018.

[93] Q. Song and B. S. Chissom. "Forecasting Enrollments with Fuzzy Time Series." 1991.

[94] Q. Song and B. S. Chissom. "Forecasting enrollments with fuzzy time series-Part II". *Fuzzy sets and systems*, vol. 62, no. 1, pp. 1–8, 1994.

[95] Q. Song, R. P. Leland and B. S. Chissom. "Fuzzy stochastic fuzzy time series and its models". *Fuzzy Sets and Systems*, vol. 88, no. 3, pp. 333 – 341, 1997.

[96] C. H. Aladag, U. Yolcu, E. Egrioglu and A. Z. Dalar. "A new time invariant fuzzy time series forecasting method based on particle swarm optimization". *Applied Soft Computing*, vol. 12, no. 10, pp. 3291–3299, 2012.

[97] H.-L. Wong, Y.-H. Tu and C.-C. Wang. "Application of fuzzy time series models for forecasting the amount of Taiwan export". *Expert Systems with Applications*, vol. 37, no. 2, pp. 1465–1470, 2010.

[98] J. Sullivan and W. H. Woodall. "A comparison of fuzzy forecasting and Markov modeling". *Fuzzy Sets and Systems*, vol. 64, no. 3, pp. 279–293, 1994.

[99] J.-R. Hwang, S.-M. Chen and C.-H. Lee. "Handling forecasting problems using fuzzy time series". *Fuzzy sets and systems*, vol. 100, no. 1-3, pp. 217–228, 1998.

[100] S.-M. Chen and J.-R. Hwang. "Temperature prediction using fuzzy time series". *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 2, pp. 263–275, 2000.

[101] T. A. Jilani and S. M. A. Burney. "Multivariate stochastic fuzzy forecasting models". *Expert Systems with Applications*, vol. 35, no. 3, pp. 691–700, 2008.

[102] S. Davari, M. H. F. Zarandi and I. B. Turksen. "An improved Fuzzy Time Series forecasting model based on Particle Swarm intervalization". In *NAFIPS 2009 - 2009 Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 1–5, 2009.

[103] W. Qiu, X. Liu and H. Li. "A generalized method for forecasting based on fuzzy time series". *Expert Systems with Applications*, vol. 38, no. 8, pp. 10446–10453, 2011.

[104] Y. Cheng and S. Li. "Fuzzy Time Series Forecasting With a Probabilistic Smoothing Hidden Markov Model". *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 291–304, 2012.

[105] H.-C. Chuang, W.-S. Chang and S.-T. Li. "A Multi-Factor HMM-Based Forecasting Model for Fuzzy Time Series". 2014.

[106] M. Bose and K. Mali. "A novel data partitioning and rule selection technique for modeling high-order fuzzy time series". *Applied Soft Computing*, vol. 63, pp. 87–96, 2018.

[107] K. Huarng and H.-K. Yu. "A Type 2 fuzzy time series model for stock index forecasting". *Physica A: Statistical Mechanics and its Applications*, vol. 353, pp. 445–462, 2005.

[108] L.-W. Lee, L.-H. Wang and S.-M. Chen. "Temperature prediction and TAIFEX forecasting based on fuzzy logical relationships and genetic algorithms". *Expert Systems with Applications*, vol. 33, no. 3, pp. 539–550, 2007.

[109] E. Bulut. "Modeling seasonality using the fuzzy integrated logical forecasting (FILF) approach". *Expert Systems with Applications*, vol. 41, no. 4, Part 2, pp. 1806–1812, 2014.

[110] H. Yang, Z. Jiang and H. Lu. "A Hybrid Wind Speed Forecasting System Based on a 'Decomposition and Ensemble' Strategy and Fuzzy Time Series". *Energies*, vol. 10, pp. 1422, 09 2017.

[111] W.-J. Lee, H.-Y. Jung, J. Yoon and S. Choi. "A Novel Forecasting Method Based on F-Transform and Fuzzy Time Series". *International Journal of Fuzzy Systems*, vol. 19, 07 2017.

[112] C.-H. Wang and L.-C. Hsu. "Constructing and applying an improved fuzzy time series model: Taking the tourism industry for example". *Expert Systems with Applications*, vol. 34, pp. 2732–2738, 05 2008.

[113] E. Bai, W. K. Wong, W. C. Chu, M. Xia and F. Pan. "A Heuristic Time-Invariant Model for Fuzzy Time Series Forecasting". *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2701–2707, March 2011.

[114] H. J. Sadaei, R. Enayatifar, M. H. Lee and M. Mahmud. "A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting". *Applied Soft Computing*, vol. 40, pp. 132–149, 2016.

[115] K.-H. Huarng and T. Yu. "The application of neural networks to forecast fuzzy time series". *Physica A: Statistical Mechanics and its Applications*, vol. 363, pp. 481–491, 05 2006.

[116] M. Bose and K. Mali. "Designing fuzzy time series forecasting models: A survey". *International Journal of Approximate Reasoning*, vol. 111, pp. 78–99, 2019.

[117] N. Tran, T. Nguyen, B. M. Nguyen and G. Nguyen. "A Multivariate Fuzzy Time Series Resource Forecast Model for Clouds using LSTM and Data Correlation Analysis". *Procedia Computer Science*, vol. 126, pp. 636–645, 2018. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 22nd International Conference, KES-2018, Belgrade, Serbia.

[118] H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimarães and M. H. Lee. "Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series". *Energy*, vol. 175, pp. 365–377, 2019.

[119] F. Gaxiola, P. Melin, F. Valdez and O. Castillo. "Interval Type-2 Fuzzy Weight Adjustment for Backpropagation Neural Networks with Application in Time Series Prediction". *Inf. Sci.*, vol. 260, pp. 1–14, March 2014.

[120] M. Avazbeigi, S. Doulabi and B. Karimi. "Choosing the appropriate order in fuzzy time series: A new N-factor fuzzy time series for prediction of the auto industry production". *Expert Syst. Appl.*, vol. 37, pp. 5630–5639, 08 2010.

[121] B. Sun, H. Guo, H. Karimi, Y. Ge and S. Xiong. "Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series". *Neurocomputing*, vol. 151, pp. 1528–1536, 03 2015.

[122] H.-J. Teoh, C.-H. Cheng, H.-H. Chu and J.-S. Chen. "Fuzzy time series model based on probabilistic approach and rough set rule induction for empirical research in stock markets". *Data & Knowledge Engineering*, vol. 67, pp. 103–117, 10 2008.

[123] T.-L. Chen, C.-H. Cheng and H.-J. Teoh. "Fuzzy time-series based on Fibonacci sequence for stock price forecasting". *Physica A: Statistical Mechanics and its Applications*, vol. 380, pp. 377–390, 07 2007.

[124] R. R. C. Silva, W. M. Caminhas, P. C. de Lima e Silva and F. G. Guimaraes. "A C4.5 Fuzzy Decision Tree Method for Multivariate Time Series Forecasting". In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, July 2021.

[125] O. Orang, P. C. de Lima e Silva, R. Silva and F. G. Guimarães. "Randomized high order fuzzy cognitive maps as reservoir computing models: A first introduction and applications". *Neurocomputing*, vol. 512, pp. 153–177, November 2022.

[126] O. Orang, R. Silva, P. C. de Lima e Silva and F. G. Guimaraes. "Solar Energy Forecasting With Fuzzy Time Series Using High-Order Fuzzy Cognitive Maps". In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, July 2020.

[127] H. Javedani Sadaei, R. Enayatifar, F. G. Guimarães, M. Mahmud and Z. A. Alzamil. "Combining ARFIMA Models and Fuzzy Time Series for the Forecast of Long Memory Time Series". *Neurocomput.*, vol. 175, no. PA, pp. 782–796, January 2016.

[128] Y. Yandong, S. Li, W. Li and M. Qu. "Power load probability density forecasting using Gaussian process quantile regression". *Applied Energy*, 11 2017.

[129] M. Khashei, S. Hejazi and M. Bijari. "A new hybrid artificial neural networks and fuzzy regression model for time series forecasting". *Fuzzy Sets and Systems*, vol. 159, pp. 769–786, 04 2008.

[130] O. Cagcag Yolcu and H.-K. Lam. "A Combined Robust Fuzzy Time Series Method for Prediction of Time Series". *Neurocomput.*, vol. 247, no. C, pp. 87–101, July 2017.

[131] Çagdas Hakan Aladag. "Using multiplicative neuron model to establish fuzzy logic relationships". *Expert Syst. Appl.*, vol. 40, pp. 850–853, 2013.

[132] O. Yolca and F. Alpaslan. "Prediction of TAIEX based on Hybrid Fuzzy Time Series Model with Single Optimization Process". *Applied Soft Computing*, vol. 66, 02 2018.

[133] E. Bas, C. Grosan, E. Egrioglu and U. Yolcu. "High order fuzzy time series method based on pi-sigma neural network". *Engineering Applications of Artificial Intelligence*, vol. 72, pp. 350–356, 06 2018.

[134] L.-Y. Wei. "A GA-weighted ANFIS model based on multiple stock market volatility causality for TAIEX forecasting". *Applied Soft Computing*, vol. 13, pp. 911–920, 02 2013.

[135] C.-H. Su and C.-H. Cheng. "A Hybrid Fuzzy Time Series Model Based on ANFIS and Integrated Nonlinear Feature Selection Method for Forecasting Stock". *Neurocomput.*, vol. 205, no. C, pp. 264–273, September 2016.

[136] C. Stefanakos. "Fuzzy time series forecasting of nonstationary wind and wave data". *Ocean Engineering*, vol. 121, pp. 1–12, 07 2016.

[137] S. Panigrahi and D. H. Behera. "A computationally efficient method for high order fuzzy time series forecasting". *Journal of Theoretical and Applied Information Technology*, vol. 96, pp. 7215–7226, 11 2018.

[138] F.-M. Tseng, G.-H. Tzeng, H.-C. Yu and B. J. Yuan. "Fuzzy ARIMA model for forecasting the foreign exchange market". *Fuzzy sets and systems*, vol. 118, no. 1, pp. 9–19, 2001.

[139] S. Torbat, M. Khashei and M. Bijari. "A hybrid probabilistic fuzzy ARIMA model for consumption forecasting in commodity markets". *Economic Analysis and Policy*, vol. 58, pp. 22–31, 2018.

[140] F.-M. Tseng, G.-H. Tzeng *et al.*. "A fuzzy seasonal ARIMA model for forecasting". *Fuzzy Sets and Systems*, vol. 126, no. 3, pp. 367–376, 2002.

[141] C.-H. Cheng, S.-F. Huang and H.-J. Teoh. "Predicting daily ozone concentration maxima using fuzzy time series based on a two-stage linguistic partition method". *Computers & Mathematics with Applications*, vol. 62, no. 4, pp. 2016–2028, 2011.

[142] C.-H. Cheng, L.-Y. Wei, J.-W. Liu and T.-L. Chen. "OWA-based ANFIS model for TAIEX forecasting". *Economic Modelling*, vol. 30, pp. 442–448, 2013.

[143] Z. Ismail and R. Efendi. "Enrollment forecasting based on modified weight fuzzy time series". *Journal of Artificial Intelligence*, vol. 4, no. 1, pp. 110–118, 2011.

[144] H. Sadaei, R. Enayatifar, H. Abdullah and A. Gani. "Short-term load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search". *International Journal of Electrical Power & Energy Systems*, vol. 62, pp. 118–129, 11 2014.

[145] C.-H. Cheng, T.-L. Chen, H. J. Teoh and C.-H. Chiang. "Fuzzy time-series based on adaptive expectation model for TAIEX forecasting". *Expert systems with applications*, vol. 34, no. 2, pp. 1126–1132, 2008.

[146] C.-H. Cheng, Y.-S. Chen and Y.-L. Wu. "Forecasting innovation diffusion of products using trend-weighted fuzzy time-series model". *Expert Systems with Applications*, vol. 36, no. 2, pp. 1826–1832, 2009.

[147] 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*, vol. 28, no. 8, pp. 1771–1784, 2020.

[148] S.-M. Chen and Y.-C. Chang. "Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques". *Information sciences*, vol. 180, no. 24, pp. 4772–4783, 2010.

[149] Y. Leu, C.-P. Lee and Y.-Z. Jou. "A distance-based fuzzy time series model for exchange rates forecasting". *Expert Systems with Applications*, vol. 36, no. 4, pp. 8107–8114, 2009.

[150] O. Duru and E. Bulut. "A non-linear clustering method for fuzzy time series: Histogram damping partition under the optimized cluster paradox". *Applied Soft Computing*, vol. 24, pp. 742–748, 11 2014.

[151] S.-M. Chen and B. Phuong. "Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors". *Knowledge-Based Systems*, vol. 118, 11 2016.

[152] C. A. Severiano, P. C. L. 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)*, pp. 1–6, 2017.

[153] W. L. Tung and C. Quek. "eFSM—A Novel Online Neural-Fuzzy Semantic Memory Model". *Neural Networks, IEEE Transactions on*, vol. 21, pp. 136 – 157, 02 2010.

[154] S. Askari and N. Montazerin. "A high-order multi-variable Fuzzy Time Series forecasting algorithm based on fuzzy clustering". *Expert Systems with Applications*, vol. 42, no. 4, pp. 2121–2135, 2015.

[155] S.-M. Chen and C.-D. Chen. "TAIEX forecasting based on fuzzy time series and fuzzy variation groups". *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 1–12, 2010.

[156] E. Egrioglu, C. Aladag, U. Yolcu, V. Uslu and M. Basaran. “A new approach based on artificial neural networks for high order multivariate fuzzy time series”. *Expert Syst. Appl.*, vol. 36, pp. 10589–10594, 09 2009.

[157] E. Egrioglu, C. Aladag, U. Yolcu, M. Basaran and V. Uslu. “A new hybrid approach based on SARIMA and partial high order bivariate fuzzy time series forecasting model”. *Expert Syst. Appl.*, vol. 36, pp. 7424–7434, 05 2009.

[158] P. Singh and B. Borah. “High-Order Fuzzy-Neuro Expert System for Time Series Forecasting”. *Know.-Based Syst.*, vol. 46, pp. 12–21, July 2013.

[159] P. C. L. Silva, P. de Oliveira e Lucas, H. J. Sadaei and F. G. Guimaraes. “Distributed Evolutionary Hyperparameter Optimization for Fuzzy Time Series”. *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1309–1321, September 2020.

[160] L. A. Zadeh. *Fuzzy Sets and Information Granularity*, p. 433–448. World Scientific Publishing Co., Inc., USA, 1996.

[161] P. Silva, C. Severiano, M. Alves, M. Weiss and F. Guimarães. *A New Granular Approach for Multivariate Forecasting*, pp. 41–58. 11 2019.

[162] M. C. dos Santos, F. G. Guimaraes and P. C. de Lima e Silva. “High-dimensional Multivariate Time Series Forecasting using Self-Organizing Maps and Fuzzy Time Series”. In *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, July 2021.

[163] H. V. Bitencourt and F. G. Guimarães. “High-dimensional Multivariate Time Series Forecasting in IoT Applications using Embedding Non-stationary Fuzzy Time Series”. *CoRR*, vol. abs/2107.09785, 2021.

[164] C. A. Severiano, P. C. de Lima e Silva, M. W. Cohen and F. G. Guimarães. “Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems”. *Renewable Energy*, vol. 171, pp. 764–783, June 2021.

[165] S. Singh. “A Simple Time Variant Method for Fuzzy Time Series Forecasting.” *Cybernetics and Systems*, vol. 38, pp. 305–321, 03 2007.

[166] T. Vovan. “An improved fuzzy time series forecasting model using variations of data”. *Fuzzy Optimization and Decision Making*, vol. 18, no. 2, pp. 151–173, 2019.

[167] J. M. Garibaldi, M. Jaroszewski and S. Musikasuwan. “Nonstationary fuzzy sets”. *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 4, pp. 1072–1086, 2008.

[168] T. Gneiting, F. Balabdaoui and A. E. Raftery. “Probabilistic forecasts, calibration and sharpness”. *Journal of the Royal Statistical Society Series B*, vol. 69, no. 2, pp. 243–268, 2007.

[169] S. Makridakis, R. Hogarth and A. Gaba. “Why forecasts fail. What to do instead”. *MIT Sloan Management Review*, vol. 51, no. 2, pp. 83–90, December 2010.

[170] S. Makridakis and N. Bakas. “Forecasting and uncertainty: A survey”. *Risk and Decision Analysis*, vol. 6, pp. 37–64, 01 2016.

[171] R. Krzysztofowicz. “The case of probabilistic forecasting in hydrology”. *Journal of Hydrology*, vol. 249, pp. 2–9, 08 2001.