



Article

Granular Weighted Fuzzy Approach Applied to Short-Term Load Demand Forecasting

Cesar Vinicius Züge and Leandro dos Santos Coelho *

Graduate Program in Electrical Engineering, Federal University of Parana, Curitiba 81530-00, PR, Brazil; cesarvinicius12@gmail.com

* Correspondence: leandro.coelho@ufpr.br

Abstract: The development of accurate models to forecast load demand across different time horizons is challenging due to demand patterns and endogenous variables that affect short-term and long-term demand. This paper presents two contributions. First, it addresses the problem of the accuracy of the probabilistic forecasting model for short-term time series where endogenous variables interfere by emphasizing a low computational cost and efficient approach such as Granular Weighted Multivariate Fuzzy Time Series (GranularWMFTS) based on the fuzzy information granules method and a univariate form named Probabilistic Fuzzy Time Series. Secondly, it compares time series forecasting models based on algorithms such as Holt-Winters, Auto-Regressive Integrated Moving Average, High Order Fuzzy Time Series, Weighted High Order Fuzzy Time Series, and Multivariate Fuzzy Time Series (MVFTS) where this paper is based on Root Mean Squared Error, Symmetric Mean Absolute Percentage Error, and Theil's U Statistic criteria relying on 5% error criteria. Finally, it presents the concept and nuances of the forecasting approaches evaluated, highlighting the differences between fuzzy algorithms in terms of fuzzy logical relationship, fuzzy logical relationship group, and fuzzification in the training phase. Overall, the GranularWMVFTS and weighted MVFTS outperformed other evaluated forecasting approaches regarding the performance criteria adopted with a low computational cost.

Keywords: short-term load forecasting; fuzzy time series; machine learning; probabilistic forecasting



Citation: Züge, C.V.; Coelho, L.d.S. Granular Weighted Fuzzy Approach Applied to Short-Term Load Demand Forecasting. *Technologies* **2024**, *12*, 182. <https://doi.org/10.3390/technologies12100182>

Academic Editor: Valeri Mladenov

Received: 28 June 2024

Revised: 1 August 2024

Accepted: 23 August 2024

Published: 1 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Energy companies benefit from load demand forecasting once short-term is used for monitoring, operational safety, turbine control, network integration, stability, and resource reallocation; medium-term is used mostly for power system management, economic dispatch, load balancing, minor maintenance, and load shedding; and long-term is commonly used for maintenance planning, operations management, turbine generator coupling or decoupling planning, operation cost optimization, and generation capacity estimation.

Short-term load forecasting is useful and necessary for the economic generation of power due to the grid's inability to store energy without specific equipment to do so, the economic allocation between plants (unit commitment scheduling), maintenance schedules, and for system security such as peak load shaving by power interchange with interconnected utilities [1].

The behavior of the electrical energy demand is directly influenced by other variables such as climate, economic, and characteristic user behavior factors [2–5]. All involved load behavior factors bring difficulty and an extra challenge to the problem, but on the other hand, these factors bring a better and more efficient solution [6]. The classical or stochastic models are commonly used in medium and long-term forecasting. The most used classical methods used to forecast are multiple linear regressions, Auto-Regressive Integrated Moving Average (ARIMA), including Seasonal ARIMA and variants, exponential smoothing, and spectral analysis. These forecasting methods stand out due to the prediction of linear factors efficiency and low cost [4,7].

Nowadays, many different approaches have been applied for short-term load forecasting (STLF) such as Artificial Intelligence (AI) and fuzzy systems. AI-based systems are mainly used due to their symbolic reasoning, flexibility, and explanation capabilities. Based on the problem of distributing and transmission companies to forecast the energy demand, there are many works using machine learning (ML) models, and the most cited studies used artificial neural networks to forecast the load demand and approved the notable accuracy of the models, but the computation time and processing necessity are elevated, much more than many other methods based on its sophisticated structure. A methodology using fuzzy rules to incorporate historical weather and load data was proposed by [8] (1996) to increase the power and efficiency of the load forecasting solution [9].

This paper presents two contributions. First, it addresses the problem of the accuracy of the probabilistic forecasting model for short-term time series where endogenous variables interfere by emphasizing a low computational cost white box approach such as Granular Weighted Multivariate Fuzzy Time Series (GranularWMFTS) based on the Fuzzy Information Granules (FIG) method and a univariate form such as Probabilistic Fuzzy Time Series. Secondly, it compares time series forecasting models based on algorithms such as Holt-Winters, ARIMA, High Order Fuzzy Time Series (HOFTS), Weighted High Order Fuzzy Time Series (WHOFTS) and Multivariate Fuzzy Time Series (MVFTS) where this paper is based on Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (sMAPE) and Theil's U Statistic (U) relying on 5% error criteria.

The remainder of this article is organized as follows. Section 2 presents the foundation of each approach evaluated in this paper and the changes over time that enhanced the forecasting models. Section 3 describes the case of study, nuances founded on each forecasting model as well as the accuracy achieved in terms of RMSE, sMAPE, and U criteria with optimizations on short-term load demand forecasting.

2. Materials and Methods

This section aims to present some fundamentals of each chosen approach used in this article.

2.1. Data

This article used a public dataset from an electrical company in the United States fo America. PJM (Pennsylvania, New Jersey, and Maryland) is a regional transmission organization (RTO) that coordinates the movement of wholesale electricity in all or parts of Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia. The dataset provided is a time series of sample frequency of 1h for all those regions and there's only the load demand on a specific timestamp so for increasing the model some variables such as seasonality, lags, and week days were created to enhance the model. More details of the statistical form of the data can be seen in Table 1.

Table 1. Dataset statistical metrics.

Metric	Load Demand
count	145,366
mean	32,080.22
standard deviation	6464.01
minimum	14,544.00
25%	27,573.00
50%	31,421.00
75%	35,650.00
maximum	62,009.00

2.2. Holt-Winters

The Triple Exponential Smoothing method was the result of an extension of Double Exponential Smoothing by adding the seasonal component Equation (3). Also, this one is used when the dataset has level, trend, and seasonality, and the equation addition is used to handle the seasonality. This method is named Holt-Winters (HW). There are two main models on HW depending on the type of seasonality: the additive and multiplicative seasonal models.

In HW α represents the weight of the level component, β represents the weight of the trend component, and γ represents the weight of the seasonal component. So, S_t represents the current seasonal component based on current $\hat{y}_t$ observation, previous level, trend, and seasonality component values l_{t-1} , b_{t-1} and S_{t-m} as well. Hence, the next forecast value is the sum of all three components, where h is the number of forecast values, e.g., $\hat{y}_{t+1}$ is the next forecast value once $h = 1$.

The m was used to denote the frequency of the seasonality, so $m = 12$ represents monthly data. Additionally, L was employed to guarantee that the seasonal index estimates utilized in the forecasting process originate from the time series last year, $L = (h - 1)/m$ where

$$l_t = \alpha(\hat{y}_t - S_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (1)$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \quad (2)$$

$$s_t = \gamma(\hat{y}_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (3)$$

$$\hat{y}_{t+h} = l_t + h * b_t + s_{t+h-m(L+1)} \quad (4)$$

The Multiplicative Seasonal Model is recommended when the amplitude of the seasonal pattern is proportional to the average level of the time series, the seasonal variation is changing proportionally to the level of the series.

2.3. ARIMA Model

Box-Jenkins approach applied to the ARIMA model, cited from now on as ARIMA, has three components: Auto-Regressive (AR), Integrated (I), and Moving Average (MA) components. The dependent relationship between an observation and a prior observation is used by the AR component. The I component employs the difference of raw data to make the time series stationary. Finally, the MA component leverages the dependence between an observation and a residual error from a moving average model applied to lag observations. We remove an observation from the preceding time step.

Each term of the ARIMA model has explicit components that function as parameters. ARIMA (p, d, q) where:

- p : number of lag observations included in the model (parameter to AR component)
- d : number of times that the raw observation is differenced (parameter to I component)
- q : the size of the moving average window (parameter to MA component)

AR model is a regression where the current observation only depends on its lags. In the MA model, y_t depends linearly on the current and past errors evaluated. MA model is typically used with the AR model bringing the Auto Regressive Moving Average (ARMA) model. So, ARIMA is nothing more than the joining of the three components leading to Equation (5) given by:

$$y_t = \alpha + \beta_1 y_{t-1} + \dots + \beta_p y_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \dots + \phi_q \epsilon_{t-q} \quad (5)$$

where y_{t-1} is the first lag of the series, β_1 is the coefficient of the first AR term, ϕ_1 is the coefficient of the first MA term, α is the intercept term, also estimated by the model, and ϵ_t is the error of a given point in time t . If the mean of y is zero, α is not be included.

2.4. Fuzzy Systems

Fuzzy logic was introduced by [10] as fuzzy set theory. The term fuzzy logic refers to the observation which individuals frequently base their conclusions on non-numerical or imprecise information. Before fuzzy logic, the mathematical set representation of a value was Boolean logic: true if the value was in the set or false otherwise. The fuzzy logic has duality; an element may belong and simultaneously not belong to the same set at certain levels, like when the membership value is between $[0, 1]$. The fuzzy set has no boundaries and is usually overlapping.

Let X be a numerical variable where $X \in \mathbb{R}$, this variable is called the universe of discourse, or its acronym, U and the highest and lowest value of the numerical variable is the universe of discourse's interval, represented as $U = [\min(X), \max(X)]$. The step to represent numerical values into linguistic values is called fuzzification, where value is a linguistic term, so a fuzzy set is associated with this value. A fuzzy set is represented by $\tilde{a}$ where $\tilde{a} \in \tilde{A}$ where $\tilde{A}$ is the fuzzy universe of discourse.

A Fuzzy Time Series (FTS) theory was laid out by [11] using fuzzy set logic. In contrast to stochastic forecasting methods, FTS does not require large data sets, which makes it more suitable than conventional forecasting systems in some cases. However, if the data exhibit multiple seasonal patterns, a larger data set will be necessary. Otherwise, FTS can process both crisp and fuzzy values.

The FTS has basic components to represent the forecasting process using this approach [12]. These components are: Define universe of discourse; Data partitioning; Fuzzification; Formulation of fuzzy logical relationship (FLR) and fuzzy logical relationship group (FLRG); Defuzzification.

With J being the number of fuzzy sets, the universe of discourse partitioning technique seeks to divide the universe of discourse U and produce linguistics variables $\tilde{A}$ made up of the fuzzified datasets A_j . The lower and upper bounds $[l, u]$ to the discourse universe are typically given a confidence margin when it comes to fuzzy set theory. Thus, Equation (6) may be used to represent the universe of discourse, and as a rule, l and u have the same value. The purpose of these margins is to aid in the forecasting process' fuzzification to account for variations in the discourse universe's limits.

$$U = [\min(X) - l, \max(X) + u] \quad (6)$$

Three key model hyperparameters, namely the membership function μ , the partitioning strategy, and the number of needed partitions k , determine the $\tilde{A}$ throughout the partitioning process. Depending on the discourse universe, the number of partitions can be any integer that establishes the number of fuzzy sets. In the interval $[0, 1]$, the membership function determines the extent to which a crisp value belongs to a fuzzy set. Figure 1 illustrates the partitioning scheme approaches from the point of view of the ML engineer and his options, depending on each type of problem. Model accuracy is directly correlated with the k value; however, this correlation becomes non-linear for small values of k , which results in a few fuzzy sets representing the discourse universe, underfitting the model by generating a crude generalization with oversimplified patterns. On the other hand, a high value of k results in minor noisy fluctuations and excessive specificity, which overfits the model and creates a large number of fuzzy sets.

The degree to which a value belongs to a fuzzy set in the interval $[0, 1]$ is defined by the membership function $\mu : U \rightarrow [0, 1]$. There are several methods to map a value's membership on fuzzy sets; the most often employed ones are the generalized bell, triangular, trapezoidal, and Gaussian mappings. Although this phase has little bearing on the accuracy, it is crucial for improving the model's readability and explainability.

The procedure utilized in the partition phase to divide the discourse universe itself and determine the boundaries, middle point, and length of the fuzzy set is referred to as the partitioning scheme. An equal-size Partitioning is a logical way to solve the partitioning problem but is not always the best approach. The partitioning scheme directly affects the accuracy of the model and this partitioning approach gives the same importance for

all fuzzy sets, that might not be the case for all datasets. The unequal-sized partitioning scheme has multiple subtopics that will not be discussed here but include entropy and clustering approaches. Entropy is an alternative to equal-sized partitioning, where there is a weight on fuzzy sets to give more importance to some and a high level of coverage to others.

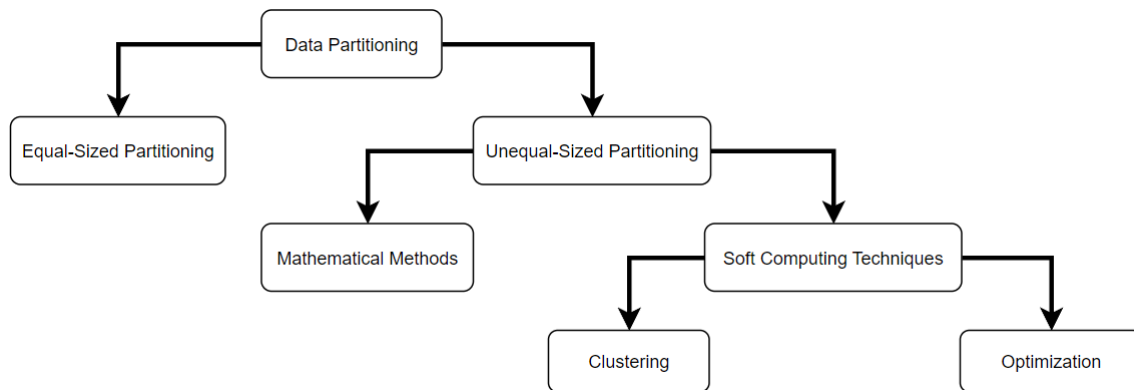


Figure 1. Data Partitioning approaches.

Entropy based methods begins by assuming an equal-size fuzzy set split and applying the weight to transform the fuzzy set length and numbers according to necessity. The clustering partitioning scheme, on the other hand, is based on discovering the number of clusters according to the distance of the centroids. In response to the previously mentioned optimization issue, this approach is far more costly and difficult to fit, but the outcome may be much more accurate. To exemplify the partitioning schemes discussed above, Figure 2 shows a visual reference.

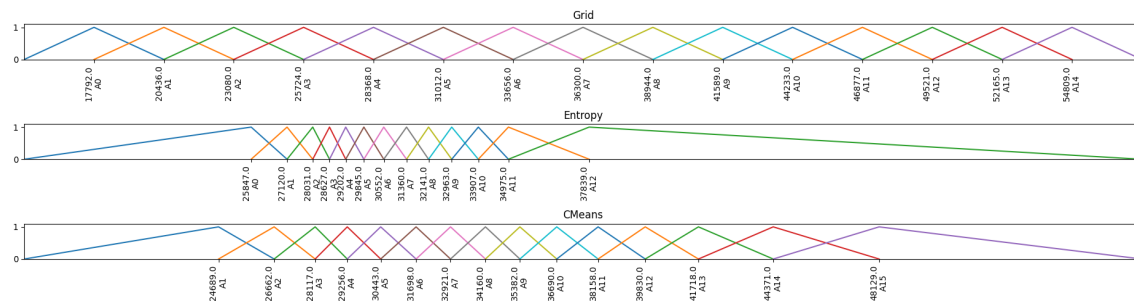


Figure 2. Data Partitioning Scheme.

The fuzzification aims to determine the degree of allegiance between the input and the fuzzy set. The fuzzification process maps every $x \in U$ into a fuzzy set $A \in U$ based on the membership function chosen. The fuzzification process began during the partitioning explanation phase. However, in order to extract its meaning, the next step is to build a relationship and divide the fuzzy sets into a group.

The FLR is a fuzzy rule regarding *precedence* $\rightarrow$ *consequence* or even $A_i \rightarrow A_j$. This rule describes the temporal relationship pattern $F(t-1) \rightarrow F(t)$, thus the dataset D generates $T-1$ fuzzy logical relationships [13]. Therefore, FLR generates a matrix *precedence* $\times$ *precedence* fulfilled with the *consequence* of each combination; this matrix is also called a fuzzy rule matrix.

Aiming to decrease the complexity of processing a high-dimensional matrix, FLRG, proposed by Chen in [14], extracts the relationship of precedence combinations. FLRG is based on grouping $A_i \rightarrow A_j$. For example, Figure 3 shows a fuzzified set extracted based on a grid partitioning scheme of Figure 2 where FLRG brings the relationship between two sets to a group of relationship with that set and helps a one-step-ahead forecasting finding $A_i \rightarrow A_j$.

Given a $F(t)$, FLRG finds this set on the antecedent part of the rules and finds out all possibilities on consequence to perform the one-step-ahead forecasting bringing a better reliability and readability to the model separating the fuzzy sets into groups and consequences, improving performance while making more easier.

```

A1 -> A2
A2 -> A2, A3, A4, A5
A3 -> A1, A2, A3, A4, A5, A6, A7
A4 -> A2, A3, A4, A5, A6, A7, A8
A5 -> A3, A4, A5, A6, A7, A8, A9
A6 -> A3, A4, A5, A6, A7, A8, A9
A7 -> A10, A11, A4, A5, A6, A7, A8, A9
A8 -> A10, A11, A4, A5, A6, A7, A8, A9
A9 -> A10, A11, A5, A6, A7, A8, A9
A10 -> A10, A11, A12, A6, A7, A8, A9
A11 -> A10, A11, A12, A8, A9
A12 -> A10, A11, A13, A8
A13 -> A11, A13

```

Figure 3. Fuzzy Logic Relationship Group.

2.5. High Order Fuzzy Time Series (HOFTS)

HOFTS model, proposed by [15], uses high order Ω parameters to enhance the prediction with lagged values. This order (Ω) parameter is the memory size of the model, but even with multiple lagged values, it's still a univariate approach. Also, an order $\Omega > 3$ has a negative impact in this model accuracy [14].

FLR is the basis of HOFTS once many antecedents Ω imply a consequence $F(t)$, e.g., if precedence is a load demand on lag $t - 1, t - 2, t - 3$ the consequence corresponds to the outputs that were experienced in the estimation period $F(t)$. Accordingly, the high-order FLR was introduced in [14] as illustrated in Equation (7) where each precedence item's weight must add up to 1.

$$F(t - 1), F(t - 2) \dots F(t - \Omega) \rightarrow F(t) \quad (7)$$

2.6. Weighted High Order Fuzzy Time Series (WHOFTS)

WHOFTS unlike HOFTS assumes that all precedence do not have the same importance in the defuzzification and FLRG steps; however, FLR is the same as HOFTS. There are some ways to calculate the weights, one of them, demonstrated in [16], is shown on Equation (8). Another way, like in [14,17], is to set a higher constant to older lags.

$$\omega_i = \frac{\#A_i}{\#RHS} \forall A_i \in RHS \quad (8)$$

The entire number of temporal patterns with the same predecessor, such as $F(t - 1), \dots, \text{and } F(t - \Omega)$, is represented by $\#RHS$, or Right Hand Side, in this instance $F(t)$. The number of occurrences of A_i on temporal patterns with the same antecedent is represented by $\#A_i$ under this scenario.

2.7. Multivariate Fuzzy Time Series

Indeed, managing multivariate time series is not a simple operation, but a common approach used is a clustering model to reduce multivariate data to multiple univariate data, like Fuzzy C-Means in [18–20].

Multivariate Fuzzy Time Series (MVFTS) is a point forecasting model of type Multiple Input/Single Output (MISO) with order $\Omega = 1$. MVFTS starts to differ on the partition step once each endogenous variable can have different membership functions and partitions, like grid and seasonal time grid in the case of cyclic variables such as month and hour.

The logic of FLR and FLRG stays the same despite the multivariate nature of the data, and only the fuzzyfication process handles this uncoupled. Weighted Multivariate Fuzzy Time Series (WMVFTS) differ from MVFTS once it assumes that not all precedence have equal the same importance in the defuzzification and FLRG steps; however, FLR is the same as MVFTS and the equation of weights can be one of many possible ways to reach a proper weight for the data, like Equation (8).

2.8. GranularWMFTS Model

The foundation of GranularWMFTS is Fuzzy Information Granules (FIG), a method of defining entities that represent subsets of a broader domain that was introduced by [21]. Every FIG functions as a multivariate fuzzy set, or a composite of distinct fuzzy sets from various variables, enabling the replacement of a vector with a scalar containing the data point's greatest membership value. When the variables are treated as target and endogenous, FIG-FTS serves as a Multiple Input/Multiple Output (MIMO) model, enabling multivariate forecasting.

Given a multivariate time series $Y = (y_1(t), \dots, y_n(t)), t = 0, \dots, T$. Each $y_i(t)$ have a corresponding variable V_i . Data points $f(t) \in F$ that represent a series of fuzzy information granules $g_i \in g$ are then assembled to generate the resultant FTS F . The global linguistic variable FIG is the union of all g_i . In GranularWMVFTS, each variable $V_i \in V$ is independent and has its linguistic variables $\tilde{V}_i$. These granules are the combination of a fuzzy set for each variable, such as $g_i = A_j^{V_i}, \forall V_i \in V$. Each variable's membership function adheres to the minimal triangular norm (T-norm), and the midpoints (c_j) of its internal fuzzy sets serve as the FIG set's index. That is, once $f(t) = g_i$, each multivariate data point $y(t) \in Y$ may be transformed by the linguistic variable FIG in the fuzzyfication stage into $g_i \in FIG$.

After the fuzzification process, the fuzzified data can feed a Probabilistic Weighted Fuzzy Time Series (PWFTS) model to continue the process (FLR, FLRG, and so on) with point, interval, or probabilistic forecasting. This PWFTS model creates a Fuzzy Temporal Pattern (FTP) that is quite similar to FLR, in terms of representation: *precedence* $\rightarrow$ *consequence*, despite the temporal dependency that involves the consideration of time intervals and the relationships between events over time. Moreover, despite the time dependence of FTP, a FLRG-like Fuzzy time Pattern Group (FTPG). When a certain set A_i is recognized at time t (the antecedent), each FTPG may be interpreted as the set of possible outcomes that could occur at a particular time $t + 1$ (the consequent) [16].

Finally, the empirical probability is computed using Probabilistic Weighted FTPG (PWFTPG), in which the precedence π_j and consequence w_{ji} sides are weighted to measure their fuzzy empirical probabilities. Given that the fuzzy set A_j is identified at time t , the weights w_{ij} can be interpreted as the empirical conditional probability of the fuzzy set A_i on time $t + 1$.

3. Results and Discussion

Regarding the methodologies provided in the preceding part, each method behaves differently with the same data set; hence, a comparison is made to demonstrate forecasting accuracy. For this study, the PyFTS library was used and 48 steps ahead in the forecasting. Since the data set is univariate, the best multivariate model was used. Hour and month variables were introduced using the grid partitioning schema, and for demand variables, a Gaussian partitioning scheme was used, as shown in Figure 4.

Regarding STLD, it's common to expect a value as a result, or a point prediction, on the data set at $t + 1$, but a probability prediction tells a lot more and follows the path to be usable to any application, carrying the interval expected the true value to be and the error based on the distribution of the data. The last model, GranularWMVFTS, supports a forecasting distribution probability and returns a graphical resource with probabilistic information carrying the uncertainty of each $\hat{y}$ as shown on Figure 5. Despite that, all results from all models tested are presented in Table 2.

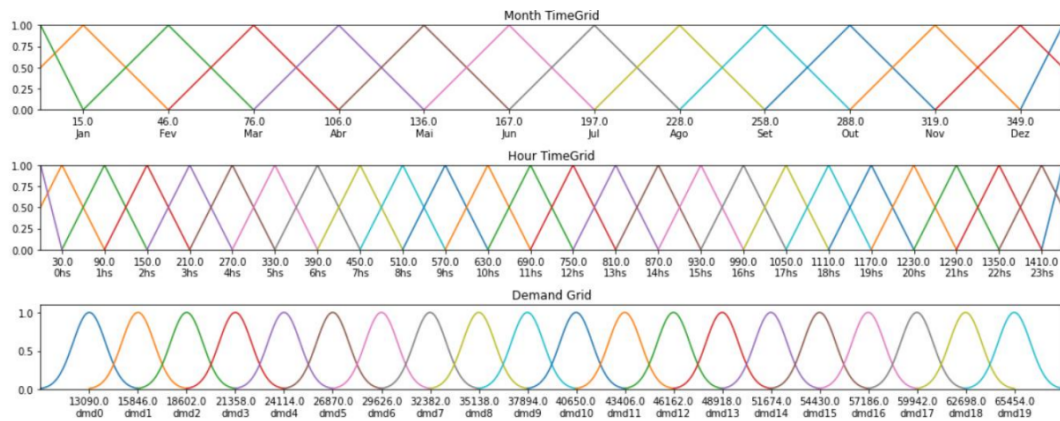


Figure 4. Multivariate Fuzzy Partitioning Short Term Load Demand.

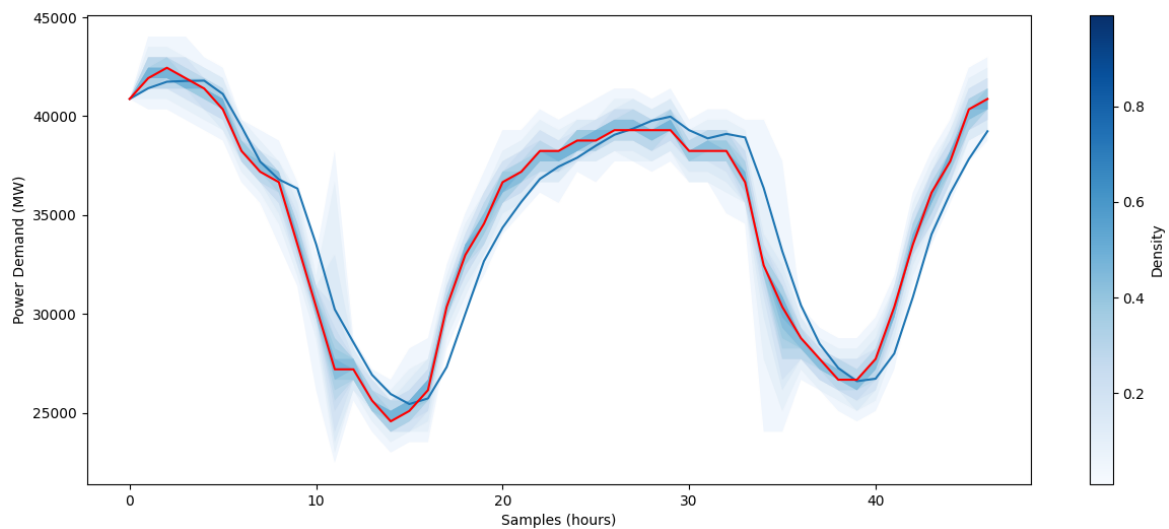


Figure 5. Granular WMVFTS Load Demand Result.

Table 2. Load Demand Forecasting Results.

Model	RMSE	sMAPE (%)	U	Time Taken (s)
AR 2	9572.42	21.92	7.68	5
AR 5	9533.13	21.85	7.64	9
ARMA 5,2	9540.72	21.84	7.65	278
ARIMA 14,1,4	2177.52	4.53	1.75	464
Holt-Winters	3808.50	6.73	3.06	36
Conventional FTS	4135.09	9.52	3.30	84
Trend Weighted FTS	1434.53	2.44	1.13	533
HOFTS	2170.40	4.42	1.68	139
WHOFTS	1121.37	2.01	0.88	141
Probabilistic FTS	1025.76	1.80	0.81	114
Multivariate FTS	784.89	1.72	0.49	136
Weighted MVFTS	483.25	1.15	0.30	138
Granular WMVFTS	538.54	1.33	0.32	1049

Root mean square error is one of the most used error evaluation metrics of a model in forecasting quantitative data. Formally RMSE is defined by Equation (9) where $\hat{y}_i$ are predicted values, y_i are observed values, and n is the number of observations. The meaning of RMSE is a Euclidian distance of these points, ignoring the number of observations (n). Now considering the whole rooted variables, RMSE can be thought of as the normalized distance of the forecasted values from the observed ones. Once RMSE is a statistical

evaluation based on variance and mean of the error, it expects that the observed data can be decomposed as Equation (9) where error represents distributed random noise with mean zero.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}{n}} \quad (9)$$

RMSE has a double purpose in data science, such as serving as a heuristic for training models or evaluating trained models for usefulness. In case of use to evaluate the model's accuracy, RMSE metrics output is strongly dependent on unity. RMSE reliability should be rated, an RMSE too small can represent an overfitting issue due to the number of parameters exceeding the number of data points. So, the result should be evaluated, and think if the small result is an overfitting or a good forecasting.

sMAPE is also a well-known error evaluation metric because the output is a percentage error, which does not depend on the unity itself. The sMAPE is computed as the average of the absolute % errors between the actual values, y , and the anticipated values, $\hat{y}$. Each error is weighted by the total of the absolute values of the actual and predicted values. A score of 0 denotes an exact match between the actual and anticipated values, while a score of 1 denotes no match at all. The final score falls between 0 and 1. A lower sMAPE number is preferable, and the percentage error is often calculated by multiplying it by 100%. The optimal score is 0; lower values are preferable.

$$\text{sMAPE} = \frac{100\%}{N} \sum_{i=0}^{N-1} \frac{2 * |y_i - \hat{y}_i|}{|y| + |\hat{y}|} \quad (10)$$

The Auto-Regressive method performed better as the p parameter grew but when the q parameter was added, turning into an ARMA model, the accuracy did not grow as expected, the ARIMA model brings up a better accuracy, in comparison of the best AR model. Probably an AutoARIMA can result in a better accuracy next time, but analytically the best result found was for ARIMA ($p, q, d = 14, 1, 4$). HW did not perform well, regarding ARIMA, but could help analyze the trend component. Entering fuzzy-based paradigms, the univariate performs better than any stochastic method tested and was enhanced using a weighted form, and when a multivariate strategy was added, the accuracy grew much more, showing the usability of the multiple seasonality as an input variable to the forecasting problem. Other important variables, such as weather, temperature, and so forth, could enhance the results more. We optimized fuzzy hyperparameters such as partitions k , partitioning schemes, membership functions, and order Ω as follows:

RMSE and sMAPE grew from $k = 5$ to $k = 50$ as k increased as the time of fit and prediction increased, according to an analysis of partition k . The other hyperparameters were fixed on order as $\Omega = 2$, grid partitioning, and the triangular membership function. Regarding the partitioning scheme, shown in Figure 2, three partitioning schemes were analyzed: grid (equal-sized partitioning), entropy (mathematical method), and CMeans (clustering algorithms), where other hyperparameters were fixed on order as $\Omega = 2$, triangular membership function, and partition as $k = 25$.

Grid partitioning had the best result, followed by CMeans and entropy, which had a significant jump compared to grid partitioning. In terms of RMSE CMeans was 63% worse and Entropy was 149% worse than grid partitioning on this dataset with these hyperparameters fixed. Regarding the membership function, three functions were analyzed, such as triangular, trapezoidal, and Gaussian where all other hyperparameters were fixed on order as $\Omega = 2$, grid partitioning, and partition as $k = 25$. There was no difference between triangular and trapezoidal membership functions regarding RMSE and sMAPE but Gaussian had a 7% worse performance. Analyzing order Ω has few options once [14] demonstrate that $\Omega > 3$ affects performance, but on Table 3 what happens to RMSE and sMAPE on orders above 3 and the time taken to fit and predict the data. Thus, the hyperparameters chosen for this paper to FTS models are $\Omega = 2, k = 45$, *GridPartitioningScheme*

and *TriangularMembershipFunction*, used to achieve the results on Table 2. To achieve these times, it was used a 32 GB RAM (random access memory) computer with Intel i7-12700 (Intel, Santa Clara, CA, USA). No GPU (Graphics Processing Unit) was used during the tests.

Table 3. Order Effect on Model Performance.

<i>Order</i>	<i>RMSE</i>	<i>sMAPE (%)</i>	<i>U</i>	<i>Time Taken (s)</i>
1	2085.36	4.00	1.66	114.9095
2	2031.96	3.78	1.61	200.8028
3	2043.58	3.82	1.60	322.2634
4	2183.39	4.22	1.70	517.2139
5	2354.70	4.64	1.87	887.1398
6	2426.31	4.72	2.05	1690.0626

Using the criteria of 5% to choose forecasting methods for short-term Term Load Demand Forecasting, the approved approaches can be seen in Table 2. The 5% criteria were based on how much a company can miss without changing the resultant action regarding the forecasting; the information was provided directly by the technical team. Hence, these approaches could be used to enhance a load-shedding process and even a grid reconfiguration starter, ensuring the availability and reliability of the network. The case of load shedding is quite critical, so the lowest RMSE is required. These approaches could be an exogenous variable in the energy stock to evaluate the price of the energy due to the power demand.

The Granular WMVFTS was chosen as the favorite rather than Weighted MVFTS, despite the best performance of this version, due to ability of GWMVFTS to forecast a set of values inside a universe of possible values, instead of the point prediction of WVMFTS. This decision was made because in short time forecasting, in cases of operational safety, turbine control, network integration, stability, or resource reallocation, a set of statistical proven values gives a confidence to act properly when needed.

4. Conclusions

The study indicates that employing FTS and its variations enhances the accuracy of short-term load demand forecasting compared to other existing approaches. The use of probabilistic forecasting not only provides insights into model uncertainty but also establishes a robust foundation for informed decision-making in the short term. Although GranularWMVFTS did not outperform Weighted MVFTS potentially due to the low hour and monthly seasonality of the endogenous variable, overall, FTS methods demonstrated superior performance over other models in the literature, meeting a 5% criteria with low computational costs in the prediction process. Here were tested various modern fuzzy time series models, and the study suggests comparing results with Adaptive Network-Based Fuzzy Inference System with conformal prediction, enabling distribution-free predictions using FTS univariate or multivariate point predictors. Additionally, the study presented an empirical demonstration of the impact of each hyperparameter on FTS performance, following the approach proposed by Chen [14] and Silva [16].

For future research, the study recommends exploring Bayesian optimization or other specific methods of tuning to identify the optimal combination of hyperparameters for multivariate time series on high-order fuzzy systems with exogenous variables, which may require more computing resources and GPU (Graphics processing unit). Also, conformal prediction is gaining a lot of space on model-agnostic probabilistic forecasting, which could be tested with Weighted MVFTS to overcome the choice made in this article for a probabilistic forecasting method, despite the better performance of this point predictor result.

Author Contributions: C.V.Z.: Writing—review and editing, Writing—original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. L.d.S.C.: Writing—review and editing, Writing—original draft, Supervision, Conceptualization. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Kandari, A.M.; Soliman, S.A.; El-Hawary, M.E. Fuzzy short-term electric load forecasting. *Int. J. Electr. Power Energy Syst.* **2004**, *26*, 111–122. [\[CrossRef\]](#)
2. Barreto, M.N. *Análise e Previsão de Carga Crítica Ativa e Reativa do Sistema Elétrico Brasileiro*; State University of Campinas, School of Electrical and Computer Engineering: Campinas, SP, Brazil, 2017. (In Portuguese)
3. Eljazzar, M.M.; Hemayed, E.E. Feature selection and optimization of artificial neural network for short term load forecasting. In Proceedings of the 18th International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 19–21 December 2017.
4. Guirelli, C.R. *Previsão no Curto Prazo de Áreas Elétricas Através de Técnicas de Inteligência Artificial*; State University of Campinas, School of Electrical and Computer Engineering: Campinas, SP, Brazil, 2016. (In Portuguese)
5. González-Romera, E.; Jaramillo-Morán, M.Á.; Carmona-Fernández, D. Monthly electric energy demand forecasting based on trend extraction. *IEEE Trans. Power Syst.* **2006**, *21*, 1946–1953. [\[CrossRef\]](#)
6. Warrior, K.P.; Shrenik, M.; Soni, N. Short-term electrical load forecasting using predictive machine learning models. In Proceedings of the IEEE Annual India Conference (INDICON), Bangalore, India, 15–17 December 2017.
7. Rejc, M.; Pantoš, M. Short-term transmission-loss forecast for the Slovenian transmission power system based on a fuzzy-logic decision approach. *IEEE Trans. Power Syst.* **2011**, *26*, 1511–1521. [\[CrossRef\]](#)
8. Ranaweera, D.K.; Hubele, N.F.; Karady, G.G. Fuzzy logic for short term load forecasting. *Int. J. Electr. Power Energy Syst.* **1996**, *18*, 215–222. [\[CrossRef\]](#)
9. Sadaei, H.J.; e Silva, P.C.d.L.; Guimarães, F.G.; Lee, M.H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* **2019**, *175*, 365–377. [\[CrossRef\]](#)
10. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353. [\[CrossRef\]](#)
11. Song, Q.; Chissom, B.S. Fuzzy time series and its models. *Fuzzy Sets Syst.* **1993**, *54*, 269–277. [\[CrossRef\]](#)
12. Chen, S.M. Forecasting enrollments based on fuzzy time series. *Fuzzy Sets Syst.* **1996**, *81*, 311–319. [\[CrossRef\]](#)
13. Song, Q.; Chissom, B.S. Forecasting enrollments with fuzzy time series—Part II. *Fuzzy Sets Syst.* **1994**, *62*, 1–8. [\[CrossRef\]](#)
14. Chen, S.M. Forecasting enrollments based on high-order fuzzy time series. *Cybern. Syst.* **2002**, *33*, 1–16. [\[CrossRef\]](#)
15. Severiano, C.A.; Silva, P.C.L.; Sadaei, H.J.; Guimarães, F.G. Very short-term solar forecasting using fuzzy time series. In Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Naples, Italy, 9–12 July 2017.
16. e Silva, P.C.d.L. *Scalable Models for Probabilistic Forecasting with Fuzzy Time Series*; Federal University of Minas Gerais, Graduate Program in Electrical Engineering: Belo Horizonte, MG, Brazil, 2019.
17. Yu, H.K. Weighted fuzzy time series models for TAIEX forecasting. *Phys. A Stat. Mech. Its Appl.* **2005**, *349*, 609–624. [\[CrossRef\]](#)
18. Li, S.T.; Cheng, Y.C.; Lin, S.Y. A FCM-based deterministic forecasting model for fuzzy time series. *Comput. Math. Appl.* **2008**, *56*, 3052–3063. [\[CrossRef\]](#)
19. Chen, S.M.; Chang, Y.C. Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation approaches. *Inf. Sci.* **2010**, *180*, 4772–4783. [\[CrossRef\]](#)
20. Sun, B.Q.; Guo, H.; Karimi, H.R.; Ge, Y.; Xiong, S. Prediction of stock index futures prices based on fuzzy sets and multivariate fuzzy time series. *Neurocomputing* **2015**, *151*, 1528–1536. [\[CrossRef\]](#)
21. Zadeh, L.A. Fuzzy sets and information granularity. In *Fuzzy Systems—Applications and Theory*; Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems; World Science Publishing: Amsterdam, The Netherlands, 1996; pp. 433–448.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.