Computational Economics
https://doi.org/10.1007/s10614-024-10723-0

Cash Flow Forecasting for Self-employed Workers: Fuzzy Inference Systems or Parametric Models?

Luis Palomero[1,2] · **Vicente García**[3] · **J. Salvador Sánchez**[2]

Accepted: 13 September 2024

Abstract
Cash flow forecasting is an important task for any organization, but it becomes crucial for self-employed workers. In this paper, we model the cash flow of three real self-employed workers as a time series problem and compare the performance of conventional parametric methods against two types of fuzzy inference systems in terms of both prediction error and processing time. Our evaluation demonstrates that there is no winning model, but that each forecasting method's performance depends on the characteristics of the cash flow data. However, experimental results suggest that parametric methods and Mamdani-type fuzzy inference systems outperform Takagi–Sugeno–Kang-type systems.

Keywords Cash flow · Time series · Fuzzy inference system · Self-employed worker

1 Introduction

Cash flow is the broad term representing the net amount of cash and cash equivalents flowing in and out of a business during a given time period, reflecting its operational, investment, and financing activities. Cash received depicts inflows, while money spent depicts outflows. A positive cash flow (i.e., cash inflow is higher than cash outflow) keeps a company afloat. Cash flow forecasting is the process of

✉ J. Salvador Sánchez
sanchez@uji.es

Luis Palomero
al063893@uji.es

Vicente García
vicente.jimenez@uacj.mx

[1] Declarando Asesores 3.0 S.l., Donat 13c, 1, 12001 Castellón de la Plana, Spain

[2] Department Llenguatges i Sistemes Informàtics, Institute of New Imaging Technologies, Universitat Jaume I, Av. Sos Baynat s/n, 12071 Castellón de la Plana, Spain

[3] División Multidisciplinaria en Ciudad Universitaria, Universidad Autónoma de Ciudad Juárez, Av. José de Jesús Macías, 32000 Ciudad Juárez, Mexico

Published online: 26 September 2024

obtaining an estimate of the future financial position of a business. Therefore, this is an crucial element of financial management to prevent a business from failing as cash inflows and outflows outline its liquidity and solvency, understood as the availability and the possession of liquid assets to pay one's debts, respectively (Weytjens et al., 2021).

However, the importance of the cash flow is paired with its inherent complexity. This complexity arises because cash flow depends on discretionary decisions by various economic and legal actors, such as late payments, particularly intrusive regulations, or unexpected events such as the recent COVID-19 pandemic. Consequently, analyzing and forecasting the cash flow involves dealing with imprecise, vague, and diffuse information. This, therefore, requires considerable ongoing effort to carry out thorough monitoring. Due to their limited resources and capacities, this is a challenging task for small and medium-sized enterprises (SMEs).

In macroeconomic terms, the impact of SMEs on the European Union's (EU) economy and on employment creation is supported by the high number of this type of business, which in the EU are enterprises that employ less than 250 persons and have an annual turnover of no more than €50 million and/or an annual balance sheet of no more than €43 million (European Commission, 2003). In 2021, 99.8% of all enterprises operating in the EU-27 non-financial business sector, which includes the industry, construction, distribution trade and services sectors, were SMEs. They generated the equivalent of 64% of total employment and 52% of total value added in the non-financial business sector. In addition, micro SMEs (i.e., enterprises employing less than ten persons) accounted for more than 93.2% of all SMEs, 44.3% of employment and 35.2% of value added (European Commission and Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs (2022)).

A special type of micro SMEs are the self-employees. These are individuals who work for themselves and do not have an employment contract with an employer. In 2016, 30.6 million individuals were self-employed in the EU-28, accounting for 14% of total EU-28 employment (European Commission, Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs, 2017). In the light of the economic importance of self-employment, the issue of cash flow forecasting for this type of business seems important enough to explore further. Unfortunately, the fact that most self-employed workers do not hire staff, and their high administrative burdens, make it too difficult to carry out this task comprehensively.

Given this challenge, it is essential to explore robust methods for accurate cash flow forecasting. An effective approach is to model cash flow forecasting as a time series forecasting problem (i.e., predicting future data based on knowledge of the past) (Rodrigues & de Oliveira Serra, 2022). Time series forecasting has demonstrated to be an effective tool for decision-making processes in a wide range of real-life application domains (De Gooijer & Hyndman, 2006). A time series Z of size m can be formulated as an ordered sequence of observations $Z = (z_1, z_2, \dots, z_m)$ distributed over time, where $z_t \in \mathbb{R}$ denotes the value taken by the series for the *t-th* time period (Parmezan et al., 2019). Based on the prior knowledge about data distribution, the time series forecasting models can be classified into parametric (probabilistic) and non-parametric (computational).

Several comparative studies (Weytjens et al., 2021; Dadteev et al., 2020; Hongjiu et al., 2012) suggest that computational methods perform better than parametric ones, although these results must be taken with caution. One reason may be that suggested by Salas-Molina et al. (2018), who consider that the reason for the worse performance of parametric models may be due to the characteristics of the time series (normality, correlation, stationarity and linearity). Another possibility may be related to experimental biases or limitations, such as a small amount of time series reviewed, short forecasting horizons h, and absence of benchmarks (Makridakis et al., 2018).

In an in-depth review, Makridakis et al. (2018) used 1045 real-time monthly series to compare 33 parametric and computational models over different horizons. They found that computational methods achieved worse results than parametric ones or, in some cases, even than a Random Walk (RW) method. Parmezan et al. (2019) obtained similar results comparing 11 forecasting methods, excluding the RW, over 55 real-domain series. Some parametric methods, such as an autoregressive integrated moving average (ARIMA) or its seasonal version, showed scores similar to those of computational methods, such as Support Vector Machines (SVM). These disparities can be due to several reasons, such as the difficulty of finding time series that meet the aforementioned parametric requirements or that computational methods require much larger training data sets than the parametric ones (Cerqueira et al., 2019).

At this point, the possibility arises of considering a fuzzy inference system (FIS) based on fuzzy set theory (Zadeh, 1965). Fuzzy logic can be viewed as an attempt to formalize decision making in an environment of imperfect information (Zadeh, 2008). In practice, the difference between traditional time series and fuzzy time series is that the observations of the former are real numbers, while those of the latter are fuzzy sets or linguistic values where vagueness is naturally incorporated. Like probabilistic and computational models, fuzzy models are aseptic of the context, so there is no problem in applying them to various financial analysis problems (Gil-Lafuente, 2005) or directly predicting cash flows (Hsu, 2016). While previous research on data from well-established companies (Cheng et al., 2010; Hsu, 2016; Wang & Ning, 2015) suggested promising results, to the best of our knowledge, cash flow data from self-employed workers has not yet been studied, ignoring the potential performance of these methods in this context.

A specific problem where a FIS-based model can be especially appropriate relates to suggesting the Spanish Social Security Contribution tax amount for the self-employed workers, defined according to an estimate of their annual revenue (Boletín Oficial del Estado, 2022). Thus, Palomero et al. (2024) developed a forecasting algorithm based on average monthly revenues, showing promising results. Accordingly, this work aims to conduct a first study to analyze the capabilities of several well-known FISs in the problem of forecasting cash flow information for self-employed workers and gain a better understanding of the benefits and/or limitations of the FISs. To this end, we compare six FISs and three parametric methods in terms of predictive and time performance on the cash flow of three real self-employed workers.

Henceforward, the rest of the paper is organized as follows. Section 2 summarizes a pool of works focused on cash flow prediction and management. Section 3 introduces the time series forecasting models used in this study. Section 4 presents the data, while Sect. 5 describes the experimental set-up. The results are discussed in Sect. 6. Finally, Sect. 7 summarizes the main findings and limitations, and suggests possible directions for further research.

2 Related Work

Numerous studies have been carried out to predict future cash flows, mainly for project management in construction companies. However, the literature related to the particular case of cash flow forecasting for SMEs is limited, especially for the self-employed workers.

An evolutionary fuzzy hybrid neural network was proposed to enhance project cash flow control, where the fuzzy logic is used to put the neural network in between a fuzzification and defuzzification layer (Cheng et al., 2010). Tangsucheeva and Prabhu (2014) developed a stochastic cash flow forecasting technique by integrating a Markov chain model of the aggregate payment behavior across all firm customers and a Bayesian model of individual customer payment behavior. Yao et al. (2016) introduced a method to investigate stochastic cash flows in both wealth and liability dynamic processes using the Markov chain. Hsu (2016) employed the adaptive neuro-fuzzy inference system (ANFIS) model for cash flow prediction in an e-commerce service provider. Fuzzy logic, weighted SVM and a fast messy genetic algorithm were fused into a fuzzy inference model for cash flow prediction (Cheng & Roy, 2011). Similarly, a hybrid inference model that combines least squares SVM and an adaptive time function was proposed to predict the future cash flow demand (Cheng et al., 2015). Boloş and Sabău-Popa (2017) developed an adaptive fuzzy scheme to control the risk of cash flow. Batselier and Vanhoucke (2017) designed a cash flow forecasting method to integrate earned value management metrics into the exponential smoothing technique.

Wang and Ning (2015) implemented a hybrid learning algorithm that combines the adaptive population activity particle swarm optimization algorithm with the least squares method to optimize the parameters of the ANFIS model, which is then used to predict the cash flow time series of a commercial bank. An interval type-2 fuzzy project scheduling was proposed by Mohagheghi et al. (2017) to predict project cash flow. A Mamdani-type fuzzy logic system with stochastic fuzzy variables was developed by Boloş et al. (2019) to detect cash flow deficits in bank loans over a period of time. For cash flow forecasting, Yang et al. (2019) constructed a hybrid learning method based on ARIMA and long short-term memory (LSTM) neural network. Lee and Kim (2019) employed two regression models to predict the future operating cash flow measured through the equations developed by Dechow et al. (1998). Weytjens et al. (2021) evaluated the performance of ARIMA, Prophet, multi-layer perceptron (MLP) and LSTM. Furthermore, they introduced a new cost function to estimate how much money is lost using imperfect predictions compared to the hypothetical case of perfect predictions.

Wang et al. (2015) proposed a combined model based on back propagation neural network and gray prediction method to improve the forecasting accuracy of bank cash flow. Khanzadi et al. (2017) developed a cash flow prediction model based on the Bayesian Belief network to avoid bankruptcy for construction firms. Using an LSTM neural network, Cheng et al. (2020) developed a forecasting model for future cash flow of construction projects based on independent and time-dependent variables. Dadteev et al. (2020) compared the forecasting accuracy of ARIMA, MLP and regression models, showing that MLP performs the best in regions where the cash flow is affected by many factors. In the paper by Zhu et al. (2022), a back propagation neural network with weights and thresholds optimized through a genetic algorithm was proposed to predict the cash flow of enterprises. Talebi et al. (2022) evaluated the performance of regression and MLP neural network models to predict future cash flows, concluding that a structure with 16 hidden neurons is the best approach.

Focused on the particular characteristics of SMEs, Xu et al. (2020) implemented a methodology based on combining social network information and cash flow data to predict bankruptcy. Bondina et al. (2021) employed the Holt-Winters model to predict the cash flows of an agricultural organization. A hybrid transaction classification and cash flow prediction model based on deep recurrent neural networks was proposed by Kotios et al. (2022). Palomero et al. (2024) developed a method to suggest Social Security contributions based on an adjusted Simple Moving Average (SMA) forecasting method. In their study, the authors compared similar methods and found that at least in the scenario analyzed, more straightforward methods, such as SMA and Exponential Time Series, showed the most promising results.

3 Forecasting Models

This section briefly describes the time series forecasting models used in the experiments in this study. As already noted, we applied parametric and FIS models to compare their performance in predicting the cash flow of self-employed workers.

3.1 Parametric Models

The parametric models assume a priori knowledge about the data distribution, which makes the model rely on a set of parameters that must be determined to optimize the predictions: the more complex the model, the heavier assumptions are made (Jarrett & Plouffe, 2011). In this paper, three parametric models were used in the experiments: simple exponential smoothing (SES), Holt-Winters' seasonal exponential smoothing (HOLT), and ARIMA.

3.1.1 Simple Exponential Smoothing

The SES forecasting method infers the value of the next observation in a time series from the average of the last t observations. Here, each observation is weighted by the smoothing parameter α $(0 < \alpha < 1)$, which exponentially decays over time so that the most recent observations have more influence calculating future predictions. The specific formula for SES (Gardner, 1985) is:

$$L_t = \alpha z_t + \alpha(1-\alpha)z_{t-1} + \cdots + \alpha(1-\alpha)^{m-1}z_1, \tag{1}$$

where L_t denotes the new estimate and z_t corresponds to the last observation.

The high computational cost for each new estimate L_t based on all observations can be avoided by rewriting Eq. (1) in terms of the current value of the time series and the value computed in the previous time (Gardner, 1985):

$$L_t = \alpha z_t + (1-\alpha)L_{t-1} \tag{2}$$

Each smoothed value (prediction) is the weighted average of the previous observations, where the weights decrease exponentially depending on the value of parameter α. If $\alpha = 1$, then the previous observations have no impact on the model (i.e., the forecast values are simply the current value); if $\alpha = 0$, the current observation is ignored, and the smoothed value consists entirely of the previous smoothed value (Pan, 2011).

The popularity of the SES model comes from its mathematical simplicity and good accuracy. The main difficulty of the method stands in finding the most appropriate value for the smoothing parameter α and the initial value of L_t (the smaller the value of α, the more important is the selection of the initial value of L_t).

3.1.2 Holt-Winters' Seasonal Exponential Smoothing

The HOLT model consists of three equations with different smoothing constants related to the primary components of the time series: level (L), trend (T) and seasonality (S). This model can be reformulated by an additive or multiplicative decomposition of its components depending on the seasonal pattern of the time series (Winters, 1960). One can choose the multiplicative model when the amplitude of the seasonal variation increases with the increase of the series average level, and the additive model when the amplitude remains constant over time.

The multiplicative HOLT model employs the following equation (Winters, 1960):

$$z_{t+h} = \left(L_t + hT_t\right)S_{t-s+h}, \tag{3}$$

where s denotes the number of observations that make up a seasonal variation and z_{t+h} indicates the prediction value z for the time period $t+h$. The components L_t, T_t and S_t are calculated as follows (Winters, 1960):

$$L_t = \alpha\frac{z_t}{S_{t-s}} + (1-\alpha)\left(L_{t-1} + T_{t-1}\right) \tag{4}$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} \tag{5}$$

$$S_t = \gamma \frac{z_t}{L_t} + (1 - \gamma) S_{t-s}, \tag{6}$$

where the smoothing constants α, β and γ are in the range [0, 1].

Like in the case of SES, the HOLT model is a recursive process, in which the values of level and trend can be initialized in the same period s and the initial seasonality can be computed by the ratio between the first observations and the average of the first time period (Winters, 1960).

$$L_s = \frac{1}{s}(z_1 + z_2 + \cdots + z_s) \tag{7}$$

$$T_s = \frac{1}{s}\left(\frac{z_{s+1} - z_1}{s} + \frac{z_{s+2} - z_2}{s} + \cdots + \frac{z_{s+s} - z_s}{s}\right) \tag{8}$$

$$S_1 = \frac{z_1}{L_s}, S_2 = \frac{z_2}{L_s}, \ldots, S_s = \frac{z_s}{L_s} \tag{9}$$

For the implementation of the additive HOLT model, we have the following equations:

$$z_{t+h} = L_t + hT_t + S_{t-s+h} \tag{10}$$

$$L_t = \alpha(z_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1}) \tag{11}$$

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1} \tag{12}$$

$$S_t = \gamma(z_t - L_t) + (1 - \gamma)S_{t-s} \tag{13}$$

Note that Eq. (12) is identical to Eq. (5) of the multiplicative model. The difference lies in the use of the other equations, in which the seasonal indexes are summed and subtracted, rather than multiplied and divided as in the multiplicative model.

In the additive model, the values of L and T can be initialized using the same equations of the multiplicative model, while the seasonality values are here calculated using the following equation:

$$S_1 = z_1 - L_S, S_2 = z_2 - L_2, \ldots, \S_s = z_s - L_s \tag{14}$$

3.1.3 Autoregressive Integrated Moving Average

The ARIMA models of order (p, d, q) allow modeling time series from the combination of three statistical procedures: (i) autoregression (AR(p)), (ii) integration, and (iii) moving averages (MA(q)). The integration procedure consists of taking successive differences from the original series Z: the first difference is denoted by $\Delta z_t = z_t - z_{t-1}$, the second difference is $\Delta^2 z_t = \Delta(\Delta z_t) = \Delta(z_t - z_{t-1})$, and the dth difference is defined as $\Delta^d z_t = \Delta(\Delta^{d-1} z_t)$. Combining these components allows complex series to be modeled more easily than using the autoregressive or moving average components separately. Thus, ARIMA(p, 0, 0) corresponds to an autoregressive model (AR(p)), ARIMA(0, 0, q) is equivalent to the moving averages model (MA(q)), and ARIMA(p, 0, q) is the combination of autoregressive and moving averages models (ARMA(p, q)).

ARIMA can adjust its structure to complex stationary time series. If a time series is non-stationary, it can be transformed using a data differentiation procedure to make the time series stationary. This procedure together with the ARMA structure leads to the ARIMA model with order (p, d, q), which is defined by the following equation (Box et al., 2015; Parmezan et al., 2019):

$$I'_t = \delta + \sum_{i=1}^{p} \phi_i I'_{t-i} + \sum_{i=1}^{q} \theta_i e_{t-i} + e_t, \tag{15}$$

where $I'_t = \Delta^d z_t = \Delta(\Delta^{d-1} z_t)$, δ represents the initial level of the model calculated by Eq. (16), d is the degree of the differentiation operator; ϕ_p and θ_q refer to the parameters of the autoregressive procedure with lag length p and the moving average with lag length q, respectively; and e_t is the white noise in a distribution with zero average and constant variance σ_e^2:

$$\delta = \mu(1 - \phi_1 - \phi_2 - \cdots, -\phi_p), \tag{16}$$

where μ reflects the stationary process average.

One of the main problems related to parametric models is to the definition of their parameters (Gardner, 1985; Winters, 1960). In the case of ARIMA models, the most common technique is the Box–Jenkins (2015) method, which is an iterative process based on four steps: selection of model structure, identification of model orders, estimation of model coefficient, and verification of the fitted model. Later, the Hyndman-Khandakar methodology was also proposed as a strategy to parameterize ARIMA through a step-by-step procedure (Hyndman & Khandakar, 2008).

3.2 Fuzzy Inference Systems

A FIS is an inference mechanism that establishes a relationship between a series of input and output sets. To define these relationships, the inference system uses fuzzy set theory to handle uncertainty. A fuzzy set is a class of objects with a continuum of membership grades (Zadeh, 1965). Let U be the universe of discourse (UoD), then a fuzzy set $\tilde{A}$ of U is characterized by a membership function $\mu_{\tilde{A}}(x)$, which associates each element $x \in U$ with a real value between 0 and 1. This function indicates the degree to which element x belongs to the fuzzy set $\tilde{A}$, that is, the membership degree of x to $\tilde{A}$. It means that in fuzzy set theory, the closer the membership degree of x is to 1, the higher the probability that x belongs to $\tilde{A}$.

Definition 3.1 (Fuzzy Time Series) (Song & Chissom, 1993) Let $Y(t)(t = \ldots, 0, 1, 2, \ldots)$ be a subset of R^1 and the universe of discourse on which the fuzzy sets $\mu_i(t)(i = 1, 2, \dots)$ are defined and let $F(t)$ be a collection of $\mu_i(t)(i = 1, 2, \dots)$. Then $F(t)$ is called a fuzzy time series on $Y(t)(t = \dots, 0, 1, 2, \dots)$.

Definition 3.2 (Fuzzy logical relationship) (Singh, 2017). Assume $F(t-1) = \tilde{A}_i$ and $F(t) = \tilde{A}_j$. The relationship between $F(t)$ and $F(t-1)$ is called fuzzy logical relationship (FLR), which can be represented as: $\tilde{A}_i \rightarrow \tilde{A}_j$ where $\tilde{A}_i$ and $\tilde{A}_j$ refer to the left and right sides of the FLR, respectively.

The FLRs are also known as IF-THEN rules, in which the first part is referred to as the rule's antecedent and the second part is called the consequent of the rule. The FISs are rule-based mechanisms that establish a relationship between a series of input and output sets, and employ four common steps to face the forecasting problems of time series: (i) a fuzzification process to translate crisp (real-valued) inputs into fuzzy values, (ii) a knowledge base to define the appropriate membership functions and the IF-THEN rules based on available historical data, (iii) an inference system to perform predictions accordingo to these rules and, (iv) a defuzzification process to transform the fuzzy forecasts back into a crisp value. Figure 1 shows a general scheme of a FIS.

Although building a FIS may seem straightforward, the construction of several of its components, such as the fuzzifier and the knowledge base, can be accomplished using complex and specialized methods. A common approach combines soft computing techniques such as artificial neural networks, rough sets, and evolutionary computing (Singh, 2017). This makes FISs more reliable at the cost of requiring greater computational effort if complex inference systems need to be solved with sufficient data. However, more straightforward approaches can also show competitive results in conditions similar to those described by Makridakis et al. (2018) and Cerqueira et al. (2019).

There are two types of widely-used FISs: the Mamdani model (Mamdani, 1974; Mamdani & Assilian, 1975) and the Takagi–Sugeno–Kang (TSK) model (Takagi & Sugeno, 1985; Sugeno & Kang, 1988). While the fuzzification of input variables and application of fuzzy operators in IF-THEN rules are the same in both types of FIS, they mainly differ in terms of translating the fuzzy outputs inferred from the

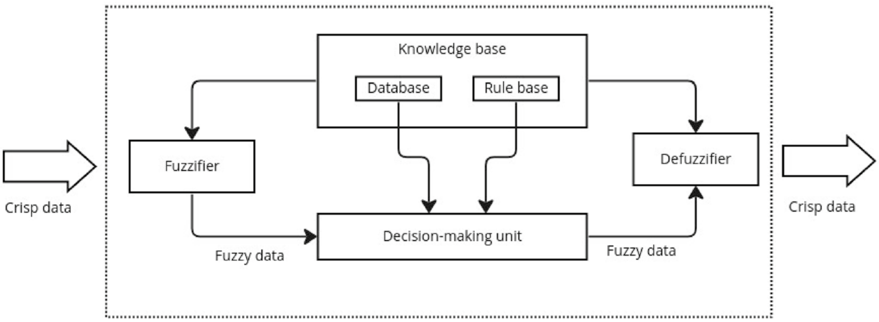


Fig. 1 Schematic view of a FIS (Jang, 1993)

fuzzy rules into crisp values (the defuzzification process). On the other hand, in general, the Mamdani-type systems have a better ability to interpret, whereas the TSK-type models have a better accuracy in approximation.

For Mamdani-type systems, the inference of the ith rule R^i to describe the relationship between the input data $x = (x_1, x_2, \dots, x_n)$ and the output y can be expressed as follows:

$$R^i\ :\ \text{IF}\, x_1 \text{ is}\, \tilde{A}_1^i \text{ AND}\, x_2 \text{ is}\, \tilde{A}_2^i \text{ AND } \dots x_n \text{ is}\, \tilde{A}_n^i \text{ THEN}\, y^i \text{ is}\, B^i,$$

where $\tilde{A}_1^i, \tilde{A}_2^i, \dots, \tilde{A}_n^i$ and B^i are fuzzy sets.

Conversely, the output of a TSK-type system for the ith rule is typically a polynomial function of the input variables:

$$R^i\ :\ \text{IF}\, x_1 \text{ is}\, \tilde{A}_1^i \text{ AND}\, x_2 \text{ is}\, \tilde{A}_2^i \text{ AND } \dots x_n \text{ is}\, \tilde{A}_n^i \text{ THEN}\, y^i = a_0 + a_1^i x_1 + a_2^i x_2 + \dots + a_n^i x_n$$

Table 1 summarizes the four Mamdani-type and the two TSK-type FISs used in the experiments of this work. The column ‘Grouping technique’ indicates the approach used to define the fuzzy sets from the crisp data. The methods used in this study were initially developed several years ago and have been widely used in previous research. Their widespread use and recognition as established methods have facilitated their development under open-source licenses and their distribution through public repositories such as the Comprehensive R Archive Network (CRAN) or GitHub.

3.2.1 Wang–Mendel System

The Wang–Mendel (WM) method (Wang & Mendel, 1992) can be described as a model-free trainable fuzzy system. In other words, as it is trained from the past data, no mathematical model is required to solve the forecasting problems. Given data pairs (x, y), the operation of the WM approach consists of five steps:

Table 1 Fuzzy inference systems included in the experiments

Fuzzy inference system	Grouping technique	Type
Wang and Mendel technique (Wang & Mendel, 1992)	Space partition	Mamdani
Hybrid neural fuzzy inference system (Kim & Kasabov, 1999)	Fuzzy neural network	Mamdani
Genetic fuzzy system based on Thrift’s method (Thrift, 1991)	Genetic algorithm	Mamdani
Genetic fuzzy system based on MOGUL method (Herrera et al., 1998)	Genetic algorithm	Mamdani
Adaptive-network-based fuzzy inference system (Jang, 1993)	Fuzzy neural network	TSK
Dynamic evolving neural-fuzzy inference system (Kasabov & Song, 2002)	Clustering	TSK

- Step 1 (creation of fuzzy input and output regions): For each of the input and output variables, triangular functions are used to divide each domain into $2N + 1$ regions denoted by SN (Small N), $\ldots, S1$ (Small 1), CE (Center), $B1$ (Big 1), $\ldots, BN$ (Big N), and a fuzzy membership function is assigned to each region.
- Step 2 (data fuzzification and rules generation): This includes determining the degrees of membership of input–output data in different fuzzy regions and their assignment to the region with the highest degree. After this, the corresponding rules can be generated from the input–output data.
- Step 3 (assignment of degree to rules): To resolve the possible conflicting rules (rules with the same IF-part, but with different THEN-part) and reduce the number of rules, a degree ($D(R^i)$) is assigned to each rule, and only the rules with the highest degree are kept. The degree value of a rule is defined as the product of the degrees of its components and the degree of the data pair that generated the rule. For instance, for a rule R : IF x_1 is A AND x_2 is B THEN y is C, the degree is defined as $D(R) = \mu_A(x_1) \times \mu_B(x_2) \times \mu_C(y)$.
- Step 4 (creation of combined fuzzy rule base): This is a lookup table whose individual boxes are assigned fuzzy rules, where each AND rule is in only one box and OR rules are assigned to all boxes in the rows and/or columns corresponding to the regions of their IF-parts. If there is more than one rule in a box of the fuzzy rule base, the rule with the highest degree is taken.
- Step 5 (defuzzification): The input space is mapped to the output space based on the combined fuzzy rule base.

This simple one-pass construction procedure allows Mamdani-type models to be built quickly as the space partitioning process is carried out using the 'grid partitioning' technique, which divides it into equally spaced intervals. Although this partitioning method is efficient, it is usually not enough to generate fuzzy rules that are sufficiently representative of the real data of the distribution. A set of solutions is based on optimization processes of this UoD partitioning process through hybrid techniques such as artificial neural networks (Singh, 2017).

3.2.2 Hybrid Neural Fuzzy Inference System

The hybrid neural fuzzy inference system (HYFIS) proposed by Kim and Kasabov (1999) is a two-phase method based on optimizing a WM model through a neural network. In the first stage (the structure learning phase), a set of fuzzy rules is extracted from the input–output pairs by using the WM technique. The second stage (the parameter learning phase) focuses on tuning the membership functions through a multi-layered perceptron network based on a gradient descent learning algorithm. In particular, this connectionist structure includes a total of five layers, where the nodes in the hidden layers are in charge of representing the membership functions and rules:

- Layer 1: The nodes in this layer represent input crisp values. Each node is connected to only those nodes of the next layer that represent the linguistic values of corresponding linguistic variables.

- Layer 2: The nodes in this layer work as Gaussian membership functions to convert the crisp values into linguistic variables. The connection weights are set to unity and the membership functions are spaced equally over the weight space.
- Layer 3: Each node in this layer represents the IF-part of a fuzzy rule and performs the AND operation. As in the previous layer, the connection weights are also set to unity. In this way, all the nodes in this layer form a fuzzy rule base.
- Layer 4: Each node in this layer represents the THEN-part of a fuzzy rule and performs the OR operation. The links define the consequences of the rules. The nodes of layers 3 and 4 are fully connected, and the connection weights between these layers represent certainty factors of the associated rules. The activation of the node represents the degree to which all fuzzy rules together support this membership function.
- Layer 5: The nodes in this layer represent the output variables, thus acting as a defuzzification procedure.

3.2.3 Genetic Fuzzy System Based on Thrift's Method

The optimization method presented by Thrift (GFS.THRIFT) (Thrift, 1991) is based on a genetic algorithm and focused on obtaining the best transition rules without influencing the partition of the UoD. Here, all the rules between the different fuzzy values and the phenotypes of the predicted values are considered as chromosomes. A new population is obtained using standard crossover and mutation operators applied to the chromosomes, generating a new Mamdani-based model whose prediction is compared within the test values. Finally, once all the iterations have been carried out, the candidate solution is identified as the one with the lowest error (Riza et al., 2015).

3.2.4 Genetic Fuzzy System for Rule Learning Based on MOGUL

The model proposed by Herrera et al. (1998) (GFS.MOGUL) is based on MOGUL, which is a methodology to obtain genetic fuzzy rule-based systems under the iterative rule learning approach. This algorithm is developed in three steps:

- Step 1 (genetic generating process): This is a learning procedure that consists of a fuzzy rule generation approach and a covering method of the set of inputs. The fuzzy rule generating approach uses a real coding genetic algorithm that codes a single fuzzy rule in each chromosome, whereas the covering method is an iterative process that runs the generating approach choosing the best chromosome (rule), considers the relative covering value this rule provokes over the set of inputs and removes the inputs with a covering value greater than a predefined threshold.
- Step 2 (genetic simplification process): Since several rules obtained in the first step may be similar, the rule base is now simplified with a binary-coded genetic algorithm, thus selecting the most cooperative set of fuzzy rules.

- Step 3 (genetic tuning process): This step fits the membership functions of the fuzzy rules by minimizing a squared error function that is defined by a set of input–output data for evaluation.

In the first stage, the rules are defined using a competitive genetic process; the second step is the tuning process of the rule and database and finally, the last step is the generation of the fuzzy rule based system (Herrera et al., 1998; Riza et al., 2015).

3.2.5 Adaptive-Network-Based Fuzzy Inference System

ANFIS (Jang, 1993) is a hybrid system that integrates the characteristics of neural networks with TSK fuzzy inference systems. The architecture of ANFIS consists of five layers, each layer with a specific purpose. To describe the structure of the neural network and emphasize the basic ideas of ANFIS, assume that the system has two inputs (x_1, x_2) and one output (y), and contains two IF-THEN rules:

$$R^1 : \text{IF}\, x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } B_1 \text{ THEN } y_1 = p_1x_1 + q_1x_2 + r_1$$
$$R^2 : \text{IF}\, x_1 \text{ is } A_2 \text{ AND } x_2 \text{ is } B_2 \text{ THEN } y_2 = p_2x_1 + q_2x_2 + r_2$$

- Layer 1 (fuzzification layer): This layer takes the crisp inputs that are then transformed into fuzzy values through membership functions $\mu_{A_i}(x)$. For instance, one can use a generalized bell membership function (Eq. 17) or a Gaussian membership function (Eq. 18):

$$\mu_{A_i}(x) = \frac{1}{1 + \left|\frac{x-c_i}{a_i}\right|^{2_{b_i}}} \tag{17}$$

$$\mu_{A_i}(x) = \exp\left[-\left(\frac{x-c_i}{a_i}\right)^2\right] \tag{18}$$

  where x is the input to node i, A_i is the fuzzy set associated with this node function, and a_i, b_i and c_i are premise parameters.
- Layer 2 (rule layer): This layer generates the firing strengths (w_i) for the rules, which are calculated by using the fuzzy AND connective of product of the membership values computed in the first layer

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(x) \quad i = 1, 2 \tag{19}$$

  where $\mu_{A_i}(x)$ and $\mu_{B_i}(x)$ are the membership degrees of fuzzy sets A and B, respectively.
- Layer 3 (normalization layer): This layer calculates normalized firing strengths as the ratio of the firing strength for a rule to the total of the firing strengths for all rules.

$$\overline{w}_i = \frac{w_i}{w_1 + w_2} \tag{20}$$

- Layer 4 (defuzzification layer): In each node of this layer, weighted values of rules are calculated to indicate the contribution of each rule to the overall output:

$$\overline{w}_i y_i = \overline{w}_i (p_i x_1 + q_i x_2 + r_i) \tag{21}$$

where p_i q_i and r_i are consequent parameters.

- Layer 5 (summation layer): The last layer, formed by a single node, computes the overall output by summing up the outputs obtained for each rule in the previous layer.

$$y = \sum_i \overline{w}_i y_i = \frac{\sum_i w_i y_i}{\sum_i w_i} \tag{22}$$

3.2.6 Dynamic Evolving Neural-Fuzzy Inference System

The dynamical evolving fuzzy neural inference system (DENFIS) (Kasabov & Song, 2002) is an extension of the evolving fuzzy neural network (Kasabov, 1998), which allows incremental learning, supervised and unsupervised learning. DENFIS uses an evolving clustering method (ECM) to partition the inputs into clusters, whose centers are then used to form the antecedent part of the fuzzy rules. Based on the weighted recursive least squares estimator, a first-order linear model is developed for the consequent part of each fuzzy rule.

DENFIS can be used for both online and offline learning. Online DENFIS only uses first-order TSK-type fuzzy rules to form a dynamic inference engine, whereas the offline model uses both first-order and high-order TSK fuzzy rules (Kasabov & Song, 2002). In both approaches, triangular membership functions with three parameters are used:

$$\mu_i(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & c \le x \end{cases} \tag{23}$$

where b is the cluster center on the x dimension, $a = b - d \times Dthr$ and $c = b + d \times Dthr$ (d = 1.2–2 and the distance threshold value $Dthr$ is a clustering parameter).

For an input vector, the output of the system is calculated with the weighted average of each rule's output.

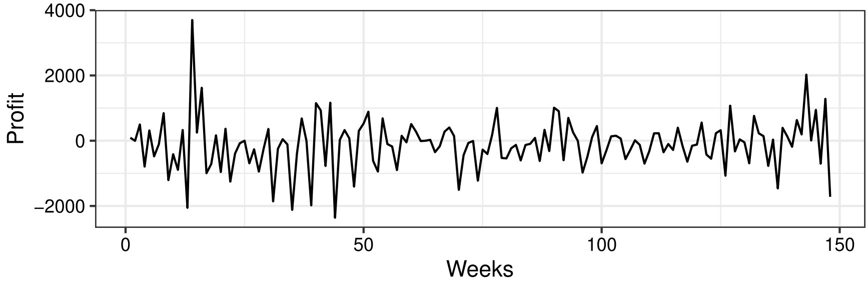
4 Description of the Time Series Data

Declarando, a Spanish online accounting consultancy specialized in self-employed workers, provided the data used in this study. The information on the profit of each self-employed worker was used as a basis for the construction of the time series. Here, the profit corresponds to the difference between the accounting income and expenses recorded during a given time period, mainly used to calculate the self-employed income and taxes payable. The main difference between cash flow and accounting data is that cash flow records actual cash inflows and outflows, while accounting data includes a broader range of financial transactions, recorded on an accrual basis, meaning that they reflect transactions as they occur, regardless of whether cash has been exchanged. The reasons for not considering explicit data on cash inflows and cash outflows were: (i) This information is validated by the Spanish Tax Agency, exposing self-employed workers to fines in case of errors, (ii) the accounting of these workers is relatively simple, so income and expenses are relatively similar to receipts and payments and, (iii) cash inflows and outflows are not entirely reliable since they tend to combine professional activity with personal activity, which may cause distortions.

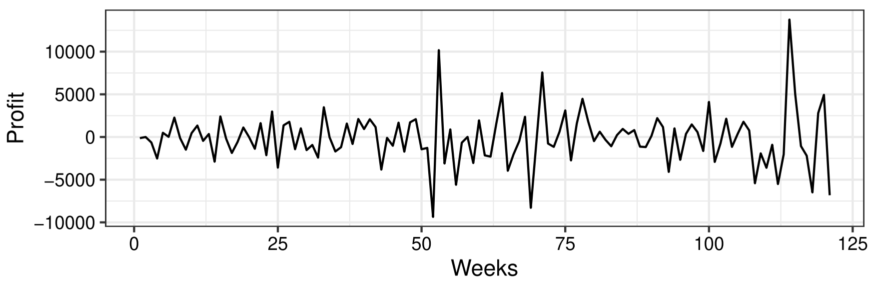
In this study, we used the weekly profit of three self-employed workers who had with at least 90 weeks of data and less than 10% missing values. The profile of the three workers corresponds to a photographer (Self-employed A) and two retailers (Self-employed B and Self-employed C). Figure 2 shows the three time series of the weekly evolution of the profit for each self-employed worker.

Table 2 summarizes some statistics of the data for each self-employed worker. The average weekly incomes were €−112.71, €−132.63 and €440.04 for Self-employed A, Self-employed B and Self-employed C, respectively. Regardless of the averages, a wide variability in income was observed, both considering the standard deviation of the values and their different ranges. Positive kurtosis and skewness were observed for all three self-employed workers, suggesting a high concentration of values and a significant right-tail. These values can be interpreted in the sense that during most weeks the profits were in a concentrated range, with peaks of large punctual profits that can be statistically understood as outliers: one in the case of Self-Employed A, two in the case of Self-Employed B, and eight in the case of Self-Employed C.

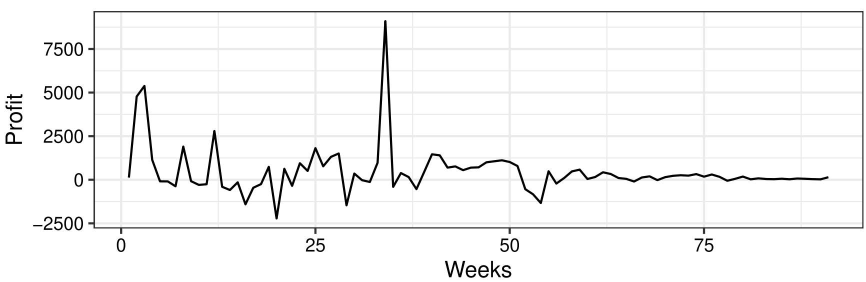
The Durbin–Watson test aims to describe the autocorrelation of a time series. The value of the statistic ranges from 0 (perfect positive autocorrelation) to 4 (perfect negative autocorrelation). Positive autocorrelations (p-value < 0.05) were observed in the case of Self-employed C, while no significant autocorrelation was observed for Self-employed A and Self-employed B. The Augmented Dickey-Fuller test was applied to determine whether or not the time series were stationary, assuming stationarity if the p-value was less than 0.05. In this case, it was possible to conclude that the three time series were stationary since their p-values were < 0.05.



(a) Self-employed A



(b) Self-employed B



(c) Self-employed C

Fig. 2 Time series included in the experiments

5 Experimental Design

Figure 3 summarizes graphically the experimental design. At a glance, the experiments aim to compare the performance of the selected models under nine different contexts. These contexts are based on the combination of three time series and a series of forecasting horizons $h = (1, 3, 6)$. A common framework, called rolling origin, is then used to perform the context-model comparisons. Having nine models of interest (ten if we include the baseline RW), a set of 90 groups of

Table 2 Some statistics of the experimental data (mean, standard deviation, median, min, max in €)

	Self-employed A	Self-employed B	Self-employed C
Length	148	121	91
% Null	1.35	0.83	0
Mean	−112.72	−132.63	440.04
Std	780.78	3066.54	1365.75
Median	−72.75	−129.27	152.59
Min	−2360.11	−9369.91	−2221.11
Max	3698.12	13757.77	9097.80
Kurtosis	3.70	4.02	18.18
Skewness	0.42	0.66	3.56
Number of outliers	1	2	8
t-test p-value ($\mu = 0$)	0.081	0.635	0.003
Durbin–Watson test value	2.44	2.29	1.65
Durbin–Watson test p-value	0.997	0.945	0.048
Augmented Dickey–Fuller test p-value	0.01	0.01	0.01

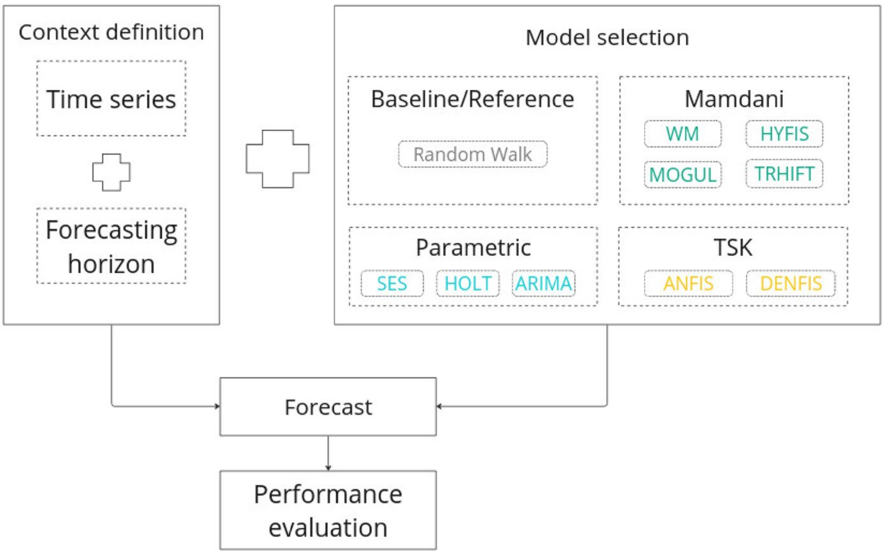


Fig. 3 Overview of the experimental set-up. Note that the Random Walk method will be used as a baseline model in the experiments

forecasts is generated. Finally, the generated forecasts and their derived metrics generated are evaluated at Sect. 6.

While the aforementioned design aimed to offer a global overview of the model, the following lines detail the forecasting procedure and the error measurement metrics. First, the rolling origin framework is explained in Subsect. 5.1. The forecasting process is detailed in-depth at Subsect. 5.2, leaving the explanation of the error metrics to the final Subsect. 5.3.

5.1 The Rolling Origin Framework

The comparison of a forecasting model M under each context is based on the rolling origin framework, which has been explicitly defined for time series-related contexts.

The rolling origin is an iterative procedure that maintains the same forecasting horizon and, on each iteration, the forecast origin advances over the time series, thus effectively creating multiple test periods for evaluation. Subsequently, with every new forecast origin, new data becomes available, which can be used for re-fitting of the model (Hewamalage et al., 2022).

We have complemented this framework using an expanding window strategy. This strategy considers all past observations as the period used during the model training process, and it is suitable when dealing with short time series, such as the ones used in this study (Hewamalage et al., 2022). Consider whether a time series could be short or long for a predictive task is difficult, as it depends on many factors, such as the complexity of the learning algorithm. In this case, under the terms described by Cerqueira et al. (2019), the lengths of our time series could be considered small.

The objective of using the rolling origin over an expanding window strategy is the generation of 30 predictions for each pair of context and model. Subsequently, as one part of the context is the h, the number of rolling origin iterations will vary from 30 loops when $h = 1$ up to five when $h = 6$, as is depicted in Fig. 4. Since we have executed 90 iterations to predict 30 weeks for each loop, we have performed a total of 1350 forecasts, predicting 2700 weeks.

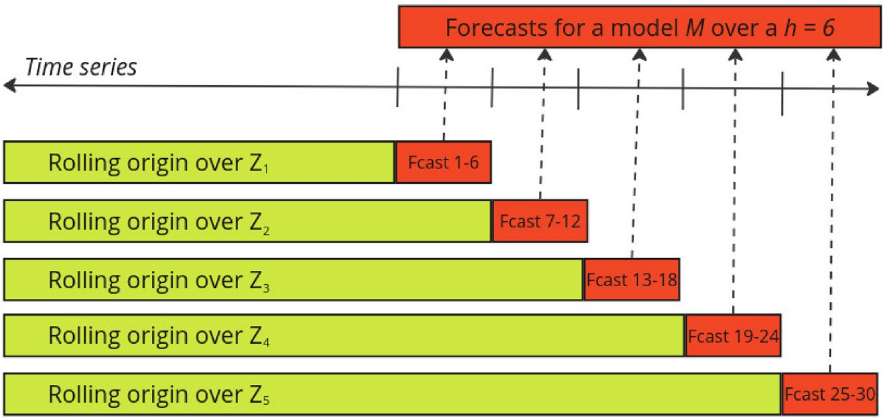


Fig. 4 Illustrative example of rolling origin over an expanding window to extract the forecasts of a model M under a scope based on a particular time series with $h = 6$

5.2 The Model Fitting Procedure

The aforementioned rolling origin framework does not cover the specific process of making predictions, which is detailed as follows. Considering an original time series Z_i of size m, a time series X, which is a subset of Z_i, is used to train a Candidate Model (CM) and generate the subsequent $\hat{h}$ predictions. This process is depicted in Fig. 5, and detailed as follows:

First, the input X is transformed into $X' \in [0-1]$ by applying the *minmax* method

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{24}$$

The *minmax* normalization has been done as is required for all fuzzy-based models except for DENFIS. Then, X' is used to train a CM using a strategy which aims to maintain simplicity and is detailed below. Once the CM is defined, the next $\hat{h}'$ forecasts, which are still in the range $[0-1]$, are obtained and finally denormalized using

$$X = X' \times (\max(X) - \min(X)) + \min(X) \tag{25}$$

to obtain the $\hat{h}$ predictions from the ith iteration of the rolling origin loop.

5.2.1 The model training process

The model training process has been designed to be deliberately straightforward to reduce errors, being, in fact, important due to the numerous predictions performed. To this end, all models were extracted from only two different software libraries, sharing similar interfaces that can be easily integrated into the framework. Concretely, parametric models were extracted from the *Forecast* library, developed by Hyndman and Khandakar (2008), while we used the *FRBS* library as reference for the fuzzy-based methods (Riza et al., 2015).

Another main decision relates to the model training process itself, which has been automated as much as possible. We have relied on the Akaike Information Criterion (AIC) to fit the parametric methods. It is a standard and widely used process that penalizes the fit of the model with the number of parameters that need to be estimated (Hyndman & Athanasopoulos, 2018).

In contrast, the fuzzy-based models are considered non-parametric algorithms that must be explicitly hyperparameterized. It is necessary to define several variables such as the model order, the membership functions or the number of fuzzy sets, among others. Therefore, generating a small, fast, and accurate model becomes challenging (Silva, 2019). Having in mind the simplicity, we have used a simple validation process. First,

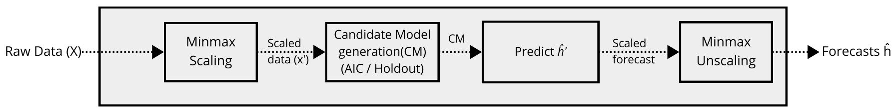


Fig. 5 Flowchart of the model fitting procedure

we have divided the time series X' into a training T' and validation V' subsets, having the V' the last h observations and leaving the remaining ones for T'.

The number of fuzzy sets $\tilde{A}$ has been defined as $\log_2$ of the training series length, similarly as used by Efendi et al. (2015); Ismail et al. (2015); Ucar et al. (2018). The only exception is the DENFIS model, where $\tilde{A}$ is defined internally.

The other parameter defined relates to model order, which ranges to a training matrix from 2 ($t' - 2$ and $t' - 1$) to 12 ($t' - 12, t' - 11, \dots, t' - 1$) previous observations. We have used a holdout validation process to select the most promising model, understanding it as a procedure based on training all the model candidates over T' and selecting the one that presents the lowest Mean Average Error (MAE) over V' (Parmezan et al., 2019). We have deliberately avoided the parameterization of other values to minimize both the algorithmic complexity and computational effort and let the models compete under similar conditions, as most of the extra parameters are specific to each model.

In the results section, we have included a subsection where we have hypothesized about the effect of including extra context variables. To this end, we have repeated the same experimental process, including this context for all models, except the univariate SES and HOLT, while excluding GFS.THRIFT model due to computational limitations. For the ARIMA, the context was included as external regressors, while it was included as an extra set of input parameters together with the model order at the fuzzy-based models.

Independently of using an AIC-based process or a hold-out validation, the model fitting procedure generates a CM model parameterized using the entire X time series and subsequently predicts the cash flow of the following h weeks, depicted as *Fcast* $a - b$ at Fig. 4.

5.3 Evaluation Criteria

For cash flow forecasting, which is modeled as a regression problem, various regression metrics, such as the mean squared error, the mean absolute error (MAE), the root mean squared relative error, the mean absolute scaled error or the coefficient of determination, can be applied to measure the model performance (Tang et al., 2022). In our experiments, we used MAE and the unscaled mean bounded relative absolute error (UMBRAE) (Chen et al., 2017). The MAE, which is the most commonly used scale-dependent measure, is defined as follows:

$$\text{MAE} = \frac{\sum_{t=1}^{n} |e_t|}{n} \tag{26}$$

where e_t denotes the forecasting error.

MAE calculates the mean of the absolute errors, so it is very easy to compute and understand (Chen et al., 2017). However, since MAE is a scale-dependent measure, we have used the relative UMBRAE metric when comparing forecasts generated under different experimental conditions. The UMBRAE is a modification of the mean bounded relative absolute error (MBRAE), which can be defined as:

$$\text{MBRAE} = mean(BRAE) = mean\left(\frac{|e_t|}{|e_t| + |e_t^*|}\right) = \frac{1}{n}\sum_{t=1}^{n}\frac{|e_t|}{|e_t| + |e_t^*|}, \tag{27}$$

where e_t represents the forecasting error at time t, e_t^* denotes the forecasting error at time t obtained by some benchmark method, and n is the total number of predictions.

Though MBRAE is adequate for comparing forecasting models, it is a scaled error that cannot be directly interpreted as a normal error ratio to reflect the forecasting error size. UMBRAE was proposed to obtain a more interpretable measure:

$$UMBRAE = \frac{MBRAE}{1 - MBRAE} \tag{28}$$

This performance measure overcomes some challenges common to other existing metrics for time series forecasting (Tang et al., 2022; Makridakis et al., 2020; Ly, 2021), as it is resistant to outliers, symmetric, scale-independent, and easily interpretable: when UMBRAE = 1, the proposed model performs equal to the benchmark method; when UMBRAE < 1, its performance is (1-UMBRAE)*100% higher than that of the benchmark method and; when UMBRAE > 1, its performance is (UMBRAE-1)*100% lower than that of the benchmark method (Chen et al., 2017).

Apart from measuring the forecasting error with MAE and UMBRAE, we also evaluated the model performance regarding the overall processing time. Since the goal of Declarando is to implement a real-time cash flow forecasting tool, the time needed to make the predictions becomes a key objective that should also be considered. The processing time was recorded from the set of functions offered by the *tictoc* library[1].

6 Results

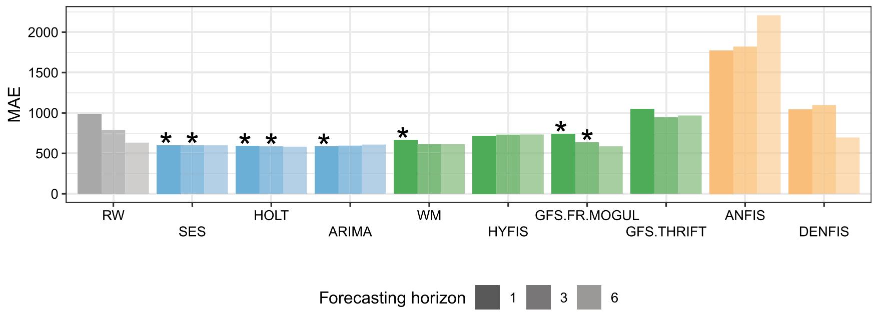
For greater clarity and conciseness, the discussion of the experimental results is presented in four blocks: (i) the errors (measured with MAE) obtained for the time series of each self-employed worker are analyzed; (ii) the prediction performance of the methods when considering the time series of the three self-employed workers is evaluated by using the UMBRAE score as reference; (iii) the effect of including context variables on the forecasting performance is studied; and (iv) the overall processing time of each forecasting model is evaluated.

Note that the baseline method used in the calculation of UMBRAE was the RW model, which has become the benchmark statistical technique for financial time series forecasting (Adhikari & Agrawal, 2014; Hewamalage et al., 2022). All the results will be made available on request.

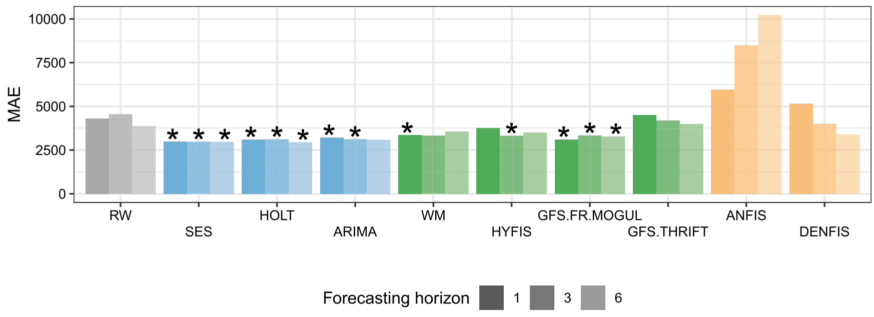
[1] The *tictoc* library is available on CRAN at https://CRAN.R-project.org/package=tictoc.

6.1 Individual Analysis of the Time Series of Each Self-employed Worker

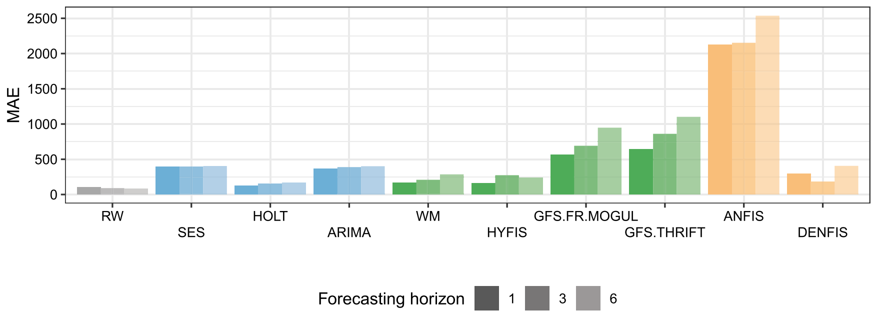
As shown in Table 2, the characteristics of the three self-employed workers were quite different; therefore, it seems appropriate to analyze the results for each worker individually. Figure 6 summarizes the MAE scores for each method and for each



(a) Self-employed A



(b) Self-employed B



(c) Self-employed C

Fig. 6 MAE scores for the three self-employed workers. Stars above the columns indicate that the MAE score was significantly lower than the baseline method based on a DM test self-employed worker. Columns highlighted with a star indicate statistically significant differences (p-value < 0.05) between the forecasting model and the baseline (RW) method using Diebold–Mariano (DM) tests (Diebold & Mariano, 2002; Diebold, 2015) where the H_A had a lower error than the benchmark. In these tests, we have specifically used the modified version proposed by Harvey et al. (1997).

Analyzing Fig. 6, a similar behavior was observed for all methods when applied to the time series of Self-employed A and Self-employed B. The three parametric models and all the Mamdani-type models except GFS.THRIFT gave the best results. On the other hand, several comparisons showed significantly better scores than the RW model. Thus, for example, it was found that the best models were SES, HOLT and GFS.FR.MOGUL, since they showed the best results for $h = (1, 3)$ in the simulation of Self-employed A (Fig. 6a) and for all values of h in the case of Self-employed B (Fig. 6b).

Although the behavior of the prediction models on Self-employed C (Fig. 6c) was similar to the other two, some significant differences must be considered. First, the best model by far was the RW baseline method. Secondly, the best model was HOLT, followed by WM, HYFIS and, to a lesser extent, DENFIS. Unlike the other two cases (Self-employed A and Self-employed B), the GFS.FR.MOGUL model showed poorer results in the simulation of Self-employed C.

Due to the great diversity of results between the three groups of techniques and the nine models, we ran a set of pairwise DM tests, whose statistical p-values were adjusted by using the Benjamini-Hochberg method (Benjamini & Hochberg, 1995). Summaries of these scores are reported in Tables 3, 4 and 5 for each self-employed worker, respectively. Each cell in these tables shows the three comparisons made, one for each h, as follows: The symbol (+) indicates that the method in the corresponding row (left) exhibits a lower error than the one in the column (top), whereas the symbol (−) represents the opposite. Cases labeled with a dot (·) denote non-statistically significant differences.

These statistical tests confirm the observations observed in Fig. 6, where the three parametric methods and the Mamdani-type models, except for GFS.THRIFT, have demonstrated similar behavior in both the case of Self-employed A (Table 3) and Self-employed B (Table 4). On the other hand, the worst method

Table 3 DM test for self-employed A

	(2)	(3)	(4)	(5)	(6)	(7)	(8)	DENFIS (9)
SES (1)	···	···	···	···	···	++·	+++	···
HOLT (2)		···	···	···	···	++·	+++	···
ARIMA (3)			···	···	···	++·	+++	···
WM (4)				···	···	+··	+++	···
HYFIS (5)					···	···	+++	···
GFS.FR.MOGUL (6)						++·	+++	···
GFS.THRIFT (7)							+··	···
ANFIS (8)								−·−

Table 4 DM test for Self-employed B

	(2)	(3)	(4)	(5)	(6)	(7)	(8)	DENFIS (9)
SES (1)	· · ·	· · ·	· · ·	· · ·	· · ·	· · +	+ + +	+ · ·
HOLT (2)		· · ·	· · ·	· · ·	· · ·	· · +	+ + +	· · ·
ARIMA (3)			· · ·	· · ·	· · ·	· · +	+ + +	+ · ·
WM (4)				· · ·	· · ·	· · ·	+ + +	· · ·
HYFIS (5)					· · ·	· · ·	· + +	· · ·
GFS.FR.MOGUL (6)						· · ·	+ + +	· · ·
GFS.THRIFT (7)							· · +	· · ·
ANFIS (8)								· − −

Table 5 DM test for Self-employed C

	(2)	(3)	(4)	(5)	(6)	(7)	(8)	DENFIS (9)
SES (1)	− − −	− · ·	− − ·	− · ·	· · ·	· · ·	+ + +	· · ·
HOLT (2)		+ + +	· · ·	· · ·	· · ·	+ · ·	+ + +	· · ·
ARIMA (3)			− − ·	− · ·	· · ·	· · ·	+ + +	· · ·
WM (4)				· · ·	· · ·	+ · ·	+ + +	· · ·
HYFIS (5)					· · ·	+ · ·	+ + +	· · ·
GFS.FR.MOGUL (6)						· · ·	+ + ·	· · ·
GFS.THRIFT (7)							+ + ·	· · ·
ANFIS (8)								− − −

in both scenarios was ANFIS, followed by GFS.THRIFT in Self-employed A. In the case of Self-employed B, only the ANFIS model was consistently worse than the other methods, while the performance of DENFIS ($h = 1$) and GFS.HRIFT ($h = 6$) was worse than almost all parametric methods.

The comparison of the Self-employed C scores presented in Table 5 also reflects the lower performance of ANFIS. Based on these findings, it has been observed that the performance of HOLTs was superior to that of the other parametric models for the three cases of h, although it remains comparable to WM, HYFIS, GFS.FR.MOGUL and DENFIS, especially in the one week-ahead forecast with WM and HYFIS, as depicted in Fig. 6c.

The results of Tables 3, 4 and 5 emphasize the observations extracted from Fig. 6, highlighting that the different parametric methods and all Mamdani-based methods (except GFS.THRIFT) performed similarly and reasonably well, while HOLT was the best model for Self-employed C. However, if we focus on the performance of the baseline, the most outstanding result is that of the case of Self-employed C, where was much better than all other forecasting methods.

After analyzing these results, it can be stated that the behavior of the nine models was quite variable depending on the time series of each self-employed worker.

This diversity of results could be related to the statistical characteristics of each time series. Thus, as reported in Table 2, the time series of Self-employed A and Self-employed B, were characterized by low values of kurtosis and skewness and no significant autocorrelation. On the other hand, the results indicated that the experimental models were worse than the benchmark method when applied to the time series of Self-employed C, which was characterized by very high values of kurtosis and skewness and a very significant positive autocorrelation.

6.2 Aggregate Analysis for all Time Series

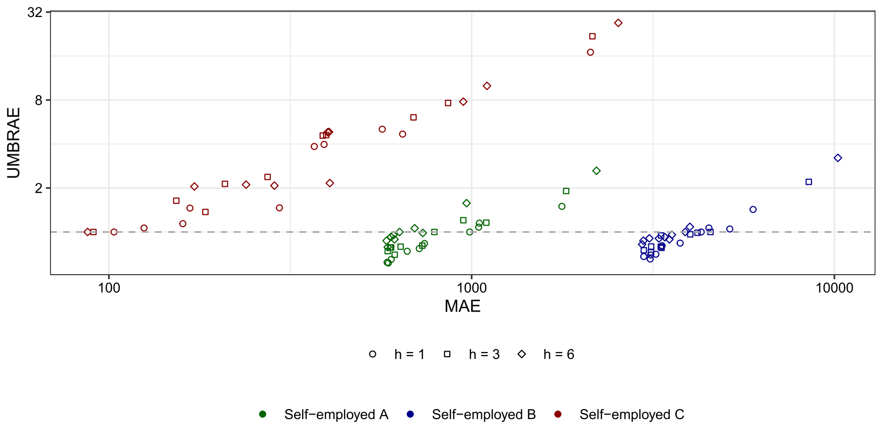
Even after verifying that the prediction performance of the models depends on the characteristics of each time series, we intended to establish an order of the methods by pooling the results of all the time series. To this end, we used the UMBRAE metric instead of the MAE, since this metric focuses on the relative difference concerning the baseline, thus simplifying the comparison process. The median UMBRAE scores are reported in Table 6, while Fig. 7 provides a comparative overview of the scores.

Figure 7a compares UMBRAE with MAE. It shows the correlation between both metrics in each self-employed scenario, especially when $UMBRAE > 1$. Specifically, the Spearman's ρ were 0.79, 0.85 and 0.96 for the Self-employed A, B and C, respectively, with the p-values $\ll 0.05$. Furthermore, Fig. 7b emphasizes the difference in UMBRAE scores according to each method and scenario. Both figures highlight the disparity in the magnitude of the errors observed in the cases of Self-employed A and Self-employed B, where almost all results indicated $UMBRAE < 1$, unlike those of Self-employed C.

In order to compare the different results, we ran a Friedman test followed by a Nemenyi post hoc test (see Fig. 8). To conduct a comprehensive evaluation using a substantial number of observations, we considered the h of each simulation, thus comparing a set of 30 one-step-ahead predictions, 10 three-step-ahead predictions, and 5 six-step-ahead predictions ($n = 45$ observations). The Friedman test p-value was $\ll 0.000001$, indicating that the null-hypothesis of equality could be rejected (p-value < 0.05).

Table 6 Median of UMBRAE values (Bold values indicate the best score)

Model	Approach	Self-employed A	Self-employed B	Self-employed C
SES	Parametric	0.784	0.750	4.609
HOLT	Parametric	0.740	**0.726**	1.637
ARIMA	Parametric	0.782	0.796	4.568
WM	Mamdani	**0.738**	0.946	2.077
HYFIS	Mamdani	0.807	0.839	2.108
GFS.FR.MOGUL	Mamdani	0.794	0.779	6.087
GFS.THRIFT	Mamdani	1.203	1.065	7.636
ANFIS	TSK	1.908	2.204	21.840
DENFIS	TSK	1.079	0.964	**1.463**



(a) The MAE (X-axis) and UMBRAE (Y-axis) scores obtained from the different simulations

(b) The UMBRAE values obtained by each method

Fig. 7 Comparison of the forecasting performances measured by using the MAE and UMBRAE metrics

The aggregate analysis, depicted in Fig. 8, reveals that the best method was HOLT, followed by two Mamdani-type fuzzy methods, HYFIS and WM. The graph shows that the HOLT model performed significantly better than the remaining methods, while GFS.THRIFT and ANFIS performed significantly worse than the others. However, the RW baseline method obtained the fourth best result without being significantly worse than the best forecasting models tested, at least from a statistical point of view.

Finally, as several algorithms showed similar performance, for a complete understanding of the behavior of these models, we ran a Wilcoxon signed-rank test between each pair of methods and adjusted *p*-values with Benjamini–Hochberg false

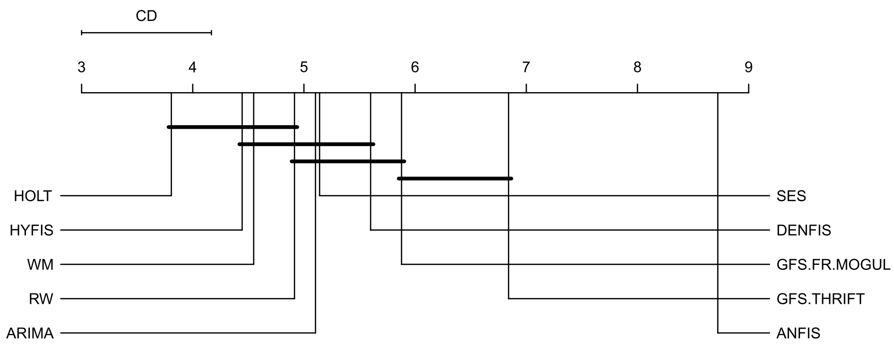


Fig. 8 Comparison of all models against each other using the Nemenyi post hoc test for the aggregate analysis

discovery rate at a significance level of 0.05. Figure 9 shows the median of differences between each pair of methods on the Y-axis. Each point represents a model if there were statistically significant differences for the reference method (the one on the X-axis). Thus, the points that appear below the dashed line (i.e., the origin located at 0.0) indicate that the corresponding model was significantly worse than the reference method, while those that are above the origin indicate that they were better than the reference method (adjusted p-value < 0.05). The best method (HOLT) is highlighted in blue. As can be seen, GFS.FR.THRIFT and ANFIS were clearly the worst prediction models. On the other hand, this figure demonstrates that

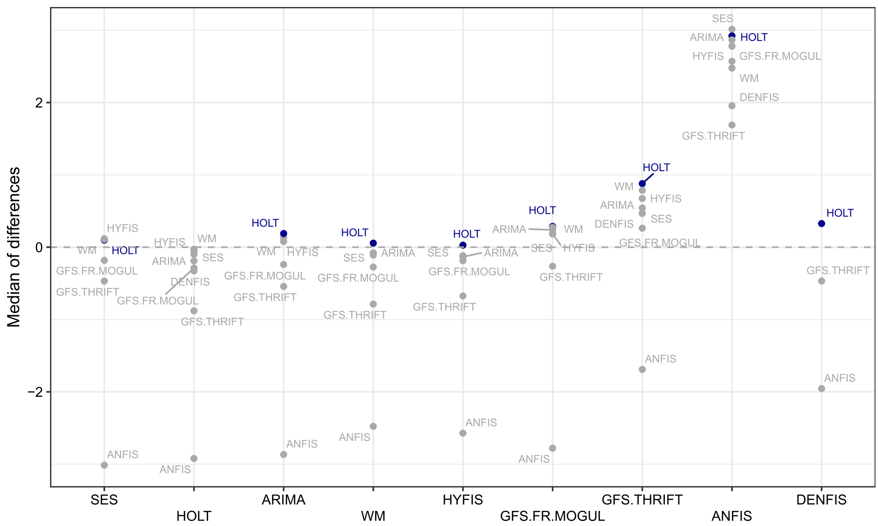


Fig. 9 Median of UMBRAE differences between models with significant differences (adjusted p-value < 0.05)

the HOLT model significantly outperformed all other methods to a greater or lesser extent, thus corroborating the findings observed with the Nemenyi post hoc test.

6.3 Performance Analysis by Including Several Context Covariates

The purpose of this experiment was to investigate the impact of including context variables in the forecasting models. We added information about the year and month of week, the presence of non-working days, and about the status of the COVID-19 restrictions, which were extracted from the paper by Lorenzo-Sáez et al. (2022). These covariates were included without any prior analysis to assess their overall effect on a systematic forecasting system.

The overall effect of adding context covariates has yet to be shown to be beneficial. To test for differences between the two versions, we conducted a Wilcoxon paired signed rank test, which yielded a *p*-value of 0.99. The alternative hypothesis was that the version with context variables would have a lower UMBRAE value. The high *p*-value indicated insufficient evidence to conclude a causal relationship between the addition of the context variables and an overall improvement in performance. However, not observing a significant overall improvement could hide local improvements. Table 7 reports the median of the UMBRAE values of each method for each of the three self-employed workers when including the context covariates. Note that UMBRAE values for SES, HOLT and GFS.THRIFT were taken from Table 6 since their inclusion in the context analysis was not feasible, as elucidated in Sect. 5.

Some comments can be drawn from Table 7. As can be seen, the results for Self-employed A and Self-employed B were quite similar to those in Table 6. The median value for all cases was similar to the UMBRAE value of 1, except for ANFIS and DENFIS whose values were significantly higher. The ARIMA model obtained a statistically significant improvement of 2.986 for Self-employed C, going from an UMBRAE value of 4.568–1.582. On the other hand, the performance of some fuzzy-based models was significantly worse. For example, in the case of GFS.FR.MOGUL, it was found to degrade by 6.113 times.

Table 7 Median of UMBRAE values (Bold values indicate the best (lowest) UMBRAE score) using covariates; values written using italic characters were taken from Table 6

Model	Self-employed A	Self-employed B	Self-employed C
SES	*0.784*	*0.750*	*4.609*
HOLT	***0.740***	*0.726*	*1.637*
ARIMA	0.793	0.743	***1.582***
WM	1.104	0.905	4.413
HYFIS	0.881	0.815	2.641
GFS.FR.MOGUL	1.146	***0.679***	37.207
GFS.THRIFT	*1.289*	*1.065*	*7.636*
ANFIS	3.602	2.645	28.885
DENFIS	3.516	2.189	7.126

After this first analysis, we carried out a more detailed comparison between the performance of the models for each time series. Figure 10 shows the median of UMBRAE obtained without using the context variables (X-axis) versus those corresponding to context variables (Y-axis). The only case with a relative improvement is seen in ARIMA for Self-employed C, while the UMBRAE score with GFS.FR.MOGUL for this same worker increased drastically according to the version that does not include these covariates.

It should be pointed out that although several local improvements were observed, adding this contextual information led to an overall decrease in performance, suggesting that incorporating them might not be considered beneficial when using these types of forecasting models. In this particular set of scenarios, including context variables have even proven to be detrimental. This emphasizes the importance of including truly correlated context variables rather than trying to include as many as possible, at the risk of model overfitting.

6.4 Processing Time

This block of results focuses on computational resources, which constitute a factor that plays an important role in many decision-making problems. In particular, Fig. 11 shows the average processing time for each forecasting method across the three self-employed workers. To evaluate the execution time of each model under the same conditions, only the execution time of the subset of the first group of experiments over $h = 1$ was measured. It is important to note that all the experiments were carried out on Intel Core i7 CPU at 2.60 GHz with 16 GB of RAM.

As expected, HOLT and SES (i.e., the most straightforward models) were the least time-consuming methods, with an average delay of less than 0.01 s, followed by the ARIMA method and WM. The more complex FIS appeared to be the most

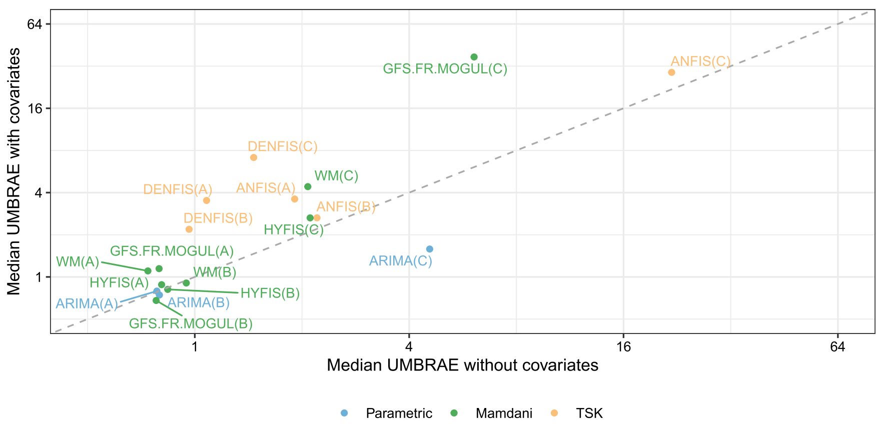


Fig. 10 Comparison of the UMBRAE values with (Y-axis) and without (X-axis) context variables

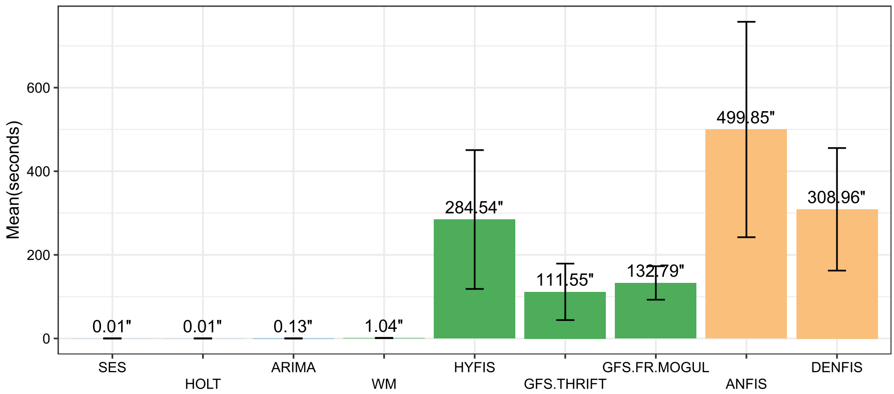


Fig. 11 Average processing times for each method (error bars represent the standard deviations)

time-consuming models, mainly due to the hyperparameterization process carried out.

It is worth mentioning that the results presented in this study were obtained using the default set of parameters for each model, except for the dynamic definition of the number of lags used in each forecast. However, in practice, it is expected to empirically tune model parameters on a training data set to achieve optimal performance. Therefore, the reported processing times may not accurately reflect the computational effort required to fit these parameters, which could even be significantly higher. While this may not be a concern for offline forecasting applications, it is important to consider the computational cost of each method when implementing real-time forecasting systems.

7 Conclusions and Future Work

Cash flow data is derived from various actors' activities, events, and decisions, making it complex to predict, especially for self-employed workers. This complexity highlights the potential use of fuzzy logic-based computational models to address the cash flow forecasting problem. To this end, a study has been carried out on three time series of real-life data comparing three conventional algorithms and six fuzzy-based methods (four Mamdani-type and two TSK-type methods) regarding predictive performance and processing time.

The experiments suggested that the straightforward Mamdani-type WM and its neural network-optimized version (HYFIS) achieved the best results together with the exponentially smoothed HOLT parametric method, and significantly outperformed the TSK-type inference systems (especially ANFIS). Adding more complexity to the models has not automatically improved the prediction results as was observed in the general context of the time series forecasting by Makridakis et al. (2018). Including context covariates has been shown to be counterproductive, as errors generally increased, mainly in the case of autocorrelated time series of Self-employed C.

In addition, simple parametric methods are also less time-consuming. These conclusions have been corroborated by several statistical tests that have been carried out to check for any significant difference between the forecasting methods. Among the tested models, it has been observed that the TSK-type systems generally exhibit the lowest performance together with the HYFIS model and, to a lesser extent, the two GFS approaches. WM is the exception, as its average processing time is more similar to the parametric models rather than the other fuzzy-based models.

A fundamental aspect beyond the models based on fuzzy logic is the simplicity of the generated models, as they work by summarizing the input crisp data into fuzzy sets. Although forecasts produced by fuzzy models such as those reviewed in this study are unlikely to be superior to forecasts produced by more complex methods, their simplicity would make them a practical alternative for use in simple infrastructures or environments where the time and computational resources are important factors, such as a scenario similar to the one provided in Palomero et al. (2024). These theoretical advantages would be diluted by including complex pre-processing steps, making them potentially difficult to execute on lightweight infrastructures or even in real-time. Even so, the inherent complexity of the cash flow of this type of company does mean that simple methods based on fuzzy logic can be considered as an alternative or complement to similar parametric methods, as long as few resources are consumed.

We have hypothesized that differences in performance can be due to the statistical characteristics of the time series. However, this statement needs further investigation to be able to draw more reliable conclusions. Therefore, this question constitutes our most immediate avenue for future research. Another issue that deserves our attention is the application of deep neural networks and deep random forests to time series forecasting.

Despite its contributions, the results of the present study should not be interpreted without accounting for some limitations that could be addressed in future work. First, the research has focused on exploring cash flow data from only three self-employed workers, so any generalizations are limited to this particular context, without aiming to generalize prior of performing deeper analyses. Secondly, this study could be extended by including other methods that have not been investigated here, as well as expanding the tuning process both in the parameterization of the model and in the inclusion of context variables. A third valuable avenue for future research is the application of the method analyzed in this work to firm-level data, which could provide a challenging opportunity to delve into a broader context and generate insights that complement the findings of this study.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The authors have not disclosed any funding.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Adhikari, R., & Agrawal, R. K. (2014). A combination of artificial neural network and random walk models for financial time series forecasting. *Neural Computing & Applications, 24*, 1441–1449. https://doi.org/10.1007/s00521-013-1386-y

Batselier, J., & Vanhoucke, M. (2017). Improving project forecast accuracy by integrating earned value management with exponential smoothing and reference class forecasting. *International Journal of Project Management, 35*(1), 28–43. https://doi.org/10.1016/j.ijproman.2016.10.003

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological), 57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Boletín Oficial del Estado (2022). *Real decreto-ley 13/2022, de 26 de julio, por el que se establece un nuevo sistema de cotización para los trabajadores por cuenta propia o autónomos y se mejora la protección por cese de actividad.* Accessed 09 May 2024. https://www.boe.es/eli/es/rdl/2022/07/26/13.

Boloş, M.-I., Bradea, I.-A., & Delcea, C. (2019). The development of a fuzzy logic system in a stochastic environment with normal distribution variables for cash flow deficit detection in corporate loan policy. *Symmetry, 11*(4), 548. https://doi.org/10.3390/sym11040548

Boloş, M.-I., & Sabău-Popa, D. C. (2017). Developing an adaptive fuzzy controller for risk management of company cash flow. *International Journal of Fuzzy Systems, 19*, 414–422. https://doi.org/10.1007/s40815-016-0159-z

Bondina, N., Bondin, I., & Palova, I. (2021). Methodological justification and analytical support for cash flow forecasting. *Scientific Papers Series Management, Economic Engineering in Agriculture and Rural Development, 21*(2), 111–118.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control* (5th ed.). Hoboken: Wiley.

Cerqueira, V., Torgo, L. & Soares, C. (2019). *Machine Learning vs Statistical Methods for Time Series Forecasting: Size Matters.* Retrieved from arXiv:1909.13316

Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PLoS ONE, 12*(3), 1–23. https://doi.org/10.1371/journal.pone.0174202

Cheng, M.-Y., Cao, M.-T., & Herianto, J. G. (2020). Symbiotic organisms search-optimized deep learning technique for mapping construction cash flow considering complexity of project. *Chaos, Solitons & Fractals, 138*, 109869. https://doi.org/10.1016/j.chaos.2020.109869

Cheng, M.-Y., Hoang, N.-D., & Wu, Y.-W. (2015). Cash flow prediction for construction project using a novel adaptive time-dependent least squares support vector machine inference model. *Journal of Civil Engineering and Management, 21*(6), 679–688. https://doi.org/10.3846/13923730.2014.893906

Cheng, M.-Y., & Roy, A. F. (2011). Evolutionary fuzzy decision model for cash flow prediction using time-dependent support vector machines. *International Journal of Project Management, 29*(1), 56–65. https://doi.org/10.1016/j.ijproman.2010.01.004

Cheng, M.-Y., Tsai, H.-C., & Sudjono, E. (2010). Evolutionary fuzzy hybrid neural network for project cash flow control. *Engineering Applications of Artificial Intelligence, 23*(4), 604–613. https://doi.org/10.1016/j.engappai.2009.10.003

Dadteev, K., Shchukin, B., & Nemeshaev, S. (2020). Using artificial intelligence technologies to predict cash flow. *Procedia Computer Science, 169*, 264–268. https://doi.org/10.1016/j.procs.2020.02.163

De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting, 22*(3), 443–473. https://doi.org/10.1016/j.ijforecast.2006.01.001

Dechow, P. M., Kothari, S., & Watts, R. L. (1998). The relation between earnings and cash flows. *Journal of Accounting and Economics, 25*(2), 133–168. https://doi.org/10.1016/S0165-4101(98)00020-2

Diebold, F. X. (2015). Comparing predictive accuracy, twenty years later: A personal perspective on the use and abuse of diebold-mariano tests. *Journal of Business & Economic Statistics, 33*(1), 1–1. https://doi.org/10.1080/07350015.2014.983236

Diebold, F. X., & Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & Economic Statistics, 20*(1), 134–144. https://doi.org/10.1198/073500102753410444

Efendi, R., Ismail, Z., & Deris, M. M. (2015). A new linguistic out-sample approach of fuzzy time series for daily forecasting of Malaysian electricity load demand. *Applied Soft Computing, 28*, 422–430. https://doi.org/10.1016/j.asoc.2014.11.043

European Commission. (2003). *Commission Recommendation of 6 May 2003 Concerning the Definition of Micro, Small and Medium-Sized Enterprises*. Accessed 04 May 2023. http://data.europa.eu/eli/reco/2003/361/oj.

European Commission and Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs (2022). *Annual report on european smes 2021/2022: SMEs and environmental sustainability*. Publications Office of the European Union.

European Commission, Directorate-General for Internal Market, Industry, Entrepreneurship and SMEs (2017). *Annual report on European SMEs 2016/2017: Focus on self-employment*. Publications Office of the European Union.

Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting, 4*(1), 1–28. https://doi.org/10.1002/for.3980040103

Gil-Lafuente, A. M. (2005). *Fuzzy logic in financial analysis* (Vol. 175). Berlin: Springer-Verlag.

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting, 13*(2), 281–291. https://doi.org/10.1016/S0169-2070(96)00719-4

Herrera, F., Lozano, M., & Verdegay, J. L. (1998). A learning process for fuzzy control rules using genetic algorithms. *Fuzzy Sets and Systems, 100*(1–3), 143–158. https://doi.org/10.1016/S0165-0114(97)00043-2

Hewamalage, H., Ackermann, K., & Bergmeir, C. (2022). Forecast evaluation for data scientists: Common pitfalls and best practices. *Data Mining and Knowledge Discovery, 37*(2), 788–832. https://doi.org/10.1007/s10618-022-00894-5

Hongjiu, L., Rieg, R., & Yanrong, H. (2012). Performance comparison of artificial intelligence methods for predicting cash flow. *Neural Network World, 22*(6), 549–564. https://doi.org/10.14311/NNW.2012.22.034

Hsu, S. C. (2016). Fuzzy time series customers prediction: Case study of an e-commerce cash flow service provider. *International Journal of Computational Intelligence and Applications, 15*(4), 1–12. https://doi.org/10.1142/S1469026816500243

Hyndman, R., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). Australia: OTexts.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software, 26*(3), 1–22. https://doi.org/10.18637/jss.v027.i03

Ismail, Z., Efendi, R., & Deris, M. M. (2015). Application of fuzzy time series approach in electric load forecasting. *New Mathematics and Natural Computation, 11*(3), 229–248. https://doi.org/10.1142/S1793005715500076

Jang, J.-S. (1993). ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Transactions on Systems, Man, and Cybernetics, 23*(3), 665–685. https://doi.org/10.1109/21.256541

Jarrett, J. E., & Plouffe, J. S. (2011). The fuzzy logic method for simpler forecasting. *International Journal of Engineering Business Management, 3*(3), 25–52. https://doi.org/10.5772/50939

Kasabov, N. (1998). *Evolving fuzzy neural networks—algorithms, applications and biological motivation* (pp. 271–274). Design and Application of Soft Computing: Methodologies for the Conception.
Kasabov, N., & Song, Qun. (2002). DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems, 10*(2), 144–154. https://doi.org/10.1109/91.995117
Khanzadi, M., Eshtehardian, E., & Esfahani, M. M. (2017). Cash flow forecasting with risk consideration using Bayesian belief networks (BBNS). *Journal of Civil Engineering and Management, 23*(8), 1045–1059. https://doi.org/10.3846/13923730.2017.1374303
Kim, J., & Kasabov, N. (1999). HyFIS: Adaptive neuro-fuzzy inference systems and their application to nonlinear dynamical systems. *Neural Networks, 12*(9), 1301–1319. https://doi.org/10.1016/S0893-6080(99)00067-2
Kotios, D., Makridis, G., Fatouros, G., & Kyriazis, D. (2022). Deep learning enhancing banking services: A hybrid transaction classification and cash flow prediction approach. *Journal of Big Data, 9*, 100. https://doi.org/10.1186/s40537-022-00651-x
Lee, J., & Kim, E. (2019). Foreign monitoring and predictability of future cash flow. *Sustainability, 11*(18), 4832. https://doi.org/10.3390/su11184832
Lorenzo-Sáez, E., Coll-Aliaga, E., Oliver-Villanueva, J.-V., Prieto del Campo, F., & Lerma-Arce, V. (2022). Analysis of the covid-19 lockdown's impact on air quality in the larger cities of Spain. *Sustainability, 14*(9), 5613. https://doi.org/10.3390/su14095613
Ly, K. T. (2021). A COVID-19 forecasting system using adaptive neuro-fuzzy inference. *Finance Research Letters, 41*, 101844. https://doi.org/10.1016/j.frl.2020.101844
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *Plos One, 13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889
Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting, 36*(1), 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014
Mamdani, E. H. (1974). Application of fuzzy algorithms for control of simple dynamic plant. *Proceedings of the Institution of Electrical Engineers, 121*(12), 1585–1588. https://doi.org/10.1049/piee.1974.0328
Mamdani, E. H., & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies, 7*(1), 1–13. https://doi.org/10.1016/S0020-7373(75)80002-2
Mohagheghi, V., Mousavi, S. M., & Vahdani, B. (2017). Analyzing project cash flow by a new interval type-2 fuzzy model with an application to construction industry. *Neural Computing & Applications, 28*, 3393–3411. https://doi.org/10.1007/s00521-016-2235-6
Palomero, L., García, V., Traver, J. & Sánchez, J.S. (2024). *Improving the calculation of social security contributions for spanish self-employed workers through an adjusted revenue estimate.* Zenodo.
Pan, R. (2011). Holt–winters exponential smoothing. In *Wiley encyclopedia of operations research and management science* (pp. 1–8). John Wiley & Sons, Ltd.
Parmezan, A. R. S., Souza, V. M. A., & Batista, G. E. A. P. A. (2019). Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information Sciences, 484*, 302–337. https://doi.org/10.1016/j.ins.2019.01.076
Riza, L. S., Bergmeir, C., Herrera, F., & Benítez, J. M. (2015). frbs: Fuzzy rule-based systems for classification and regression in R. *Journal of Statistical Software, 65*(6), 1–30. https://doi.org/10.18637/jss.v065.i06
Rodrigues, S. E., & de Oliveira Serra, G. L. (2022). An approach for evolving neuro-fuzzy forecasting of time series based on parallel recursive singular spectrum analysis. *Fuzzy Sets and Systems, 443*, 1–29. https://doi.org/10.1016/j.fss.2021.09.009
Rodrigues, S. E., & de Oliveira Serra, G. L. (2022). An approach for evolving neuro-fuzzy forecasting of time series based on parallel recursive singular spectrum analysis. *Fuzzy Sets and Systems, 443*, 1–29. https://doi.org/10.1016/j.fss.2021.09.009
Silva, P.C.d.L.e. (2019). *Scalable models for probabilistic forecasting with fuzzy time series* (Doctoral dissertation, Federal University of Minas Gerais - UFMG). https://doi.org/10.5281/zenodo.3374641
Singh, P. (2017). A brief review of modeling approaches based on fuzzy time series. *International Journal of Machine Learning and Cybernetics, 8*(2), 397–420. https://doi.org/10.1007/s13042-015-0332-y

Song, Q., & Chissom, B. S. (1993). Fuzzy time series and its models. *Fuzzy Sets and Systems, 54*(3), 269–277. https://doi.org/10.1016/0165-0114(93)90372-O

Sugeno, M., & Kang, G. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems, 28*(1), 15–33. https://doi.org/10.1016/0165-0114(88)90113-3

Takagi, T., & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man and Cybernetics, SMC, 15*(1), 116–132. https://doi.org/10.1109/TSMC.1985.6313399

Talebi, B., Abdi, R., Hajiha, Z., & Rezaei, N. (2022). The evaluation of the capability of the regression and neural network models in predicting future cash flows. *Advances in Mathematical Finance and Applications, 7*(2), 327–343. https://doi.org/10.22034/AMFA.2020.1876840.1277

Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., & Li, J. (2022). A survey on machine learning models for financial time series forecasting. *Neurocomputing, 512*, 363–380. https://doi.org/10.1016/j.neucom.2022.09.003

Tangsucheeva, R., & Prabhu, V. (2014). Stochastic financial analytics for cash flow forecasting. *International Journal of Production Economics, 158*, 65–76. https://doi.org/10.1016/j.ijpe.2014.07.019

Thrift, P.R. (1991). Fuzzy logic synthesis with genetic algorithms. in *San Diego, CA: Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 509–513.

Ucar, F., Alcin, O. F., Dandil, B., & Ata, F. (2018). Power quality event detection using a fast extreme learning machine. *Energies, 11*(1), 1–14. https://doi.org/10.3390/en11010145

Wang, J. S., & Ning, C. X. (2015). ANFIS based time series prediction method of bank cash flow optimized by adaptive population activity PSO algorithm. *Information, 6*(3), 300–313. https://doi.org/10.3390/info6030300

Wang, J.S., Ning, C.X. & Cui, W.H. (2015). Time series prediction of bank cash flow based on grey neural network algorithm. in *Proceedings of the international conference on estimation, detection and information fusion*, Harbin, China, pp. 272–277.

Wang, L. X., & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man and Cybernetics, 22*(6), 1414–1427. https://doi.org/10.1109/21.199466

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2021). Cash flow prediction: MLP and LSTM compared to ARIMA and prophet. *Electronic Commerce Research, 21*(2), 371–391. https://doi.org/10.1007/s10660-019-09362-7

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science, 6*(3), 324–342. https://doi.org/10.1287/mnsc.6.3.324

Xu, Y., Kou, G., Peng, Y., & Alsaadi, F. E., et al. (2020). Bankruptcy forecasting for small and medium-sized enterprises using cash flow data. In J. He (Ed.), *Communications in computer and information science* (Vol. 1179, pp. 477–487). Singapore: Springer.

Yang, X., Mao, S., Gao, H., Duan, Y., & Zou, Q. (2019). Novel financial capital flow forecast framework using time series theory and deep learning: A case study analysis of yu'e bao transaction data. *IEEE Access, 7*, 70662–70672. https://doi.org/10.1109/ACCESS.2019.2919189

Yao, H., Li, X., Hao, Z., & Li, Y. (2016). Dynamic asset-liability management in a Markov market with stochastic cash flows. *Quantitative Finance, 16*(10), 1575–1597. https://doi.org/10.1080/14697688.2016.1151070

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control, 8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences, 178*(13), 2751–2779. https://doi.org/10.1016/j.ins.2008.02.012

Zhu, L., Yan, M., & Bai, L. (2022). Prediction of enterprise free cash flow based on a backpropagation neural network model of the improved genetic algorithm. *Information, 13*(4), 172. https://doi.org/10.3390/info13040172