Econometric versus machine learning methods for time-series forecasting: a case study for the platform economy

Mihaela Simionescu (mihaela.simionescu@faa.unibuc.ro)
Faculty of Business and Administration, University of Bucharest and Institute for Economic Forecasting, Romanian Academy

Bogdan Oancea (bogdan.oancea@faa.unibuc.ro - corresponding author)
Faculty of Business and Administration, University of Bucharest

Oana Simona Hudea (oana.hudea@faa.unibuc.ro)
Faculty of Business and Administration, University of Bucharest

ABSTRACT

Platform economy is an innovative concept that supports the use of digital platforms for business models. In order to take maximum benefit of such platforms, not only the surrounding context is important, but also the ability to precisely determine their destination and to adapt accordingly. With this in mind, we proceeded to the theoretical and applicative analysis of the same, undertaking to underline, based on a comparative analysis of specific econometric versus machine learning methods, how to better forecast short time-series, so as to predict the evolution of the number of participants to such particular platforms. Overall, we identified an important limitation of the Long Short-Term Memory networks, one of the most advanced and effective machine learning techniques for univariate time series forecasting, namely the complexity of computations and the uncertainty regarding the accuracy of results, as compared to the econometric approach, herein mainly represented by SARIMA models. Despite the intensive utilization of machine learning techniques, the current research evidenced the outperformance of the implemented econometric models in some cases. Further research might consider conformal machine learning techniques, to obtain uncertainty quantification too, including a larger number of Long Short-Term Memory networks specific architectures.

Keywords: *Platform Economy; Time Series Forecasting; Machine Learning; Long Short-Term Memory (LSTM); SARIMA Models; Teleconference Platforms; Hybrid Forecasting*

JEL classification: *C53, C45, C32, L86, O33*

1. INTRODUCTION

Platform economy is an innovative concept that supports the use of digital platforms for business models. In this case, business analysts support the digital innovation in a competitive market. The platforms make the interconnection between various stakeholders like entrepreneurs, consumers and businesspersons that can share resources and various products (Chen, 2019). The platform economy is any digital platform that connects various networks of people using the internet. The role of online platforms is to match various groups like customers, producers, advertisers, users etc. and to support their interactions and, maybe, transactions (Codagnone, 2022). Crossed network externalities ensure the creation of value and profit.

The platform economy also influences the labor market due to improvement in connectivity and technological progress. Job opportunities can be found using apps and online outsourcing platforms.

Few examples of platforms in marketplace are related to service provision (Bolt, Airbnb, Uber), goods (e.g. Amazon, AliExpress, Overstock, Thrive Market, Target, Uncommon Goods), payments (e.g. Stripe, Square, Gumroad, PayPal, Amazon Pay), software development (e.g., Huawei, Apple, Samsung, Oppo, Soni, Microsoft, Dell, Lenovo Salesforce) etc.

There are many advantages of platform economy: no trade barriers, fast data circulation, higher participation of users and creation of open economic systems (Chen, 2019). There are three major challenges related to platform economy:

1. Revisions of regulations and laws for platform-based firms are necessary because of issues related to safety, fair competition, rights protection, taxes etc. However, some experts consider that the actual legal framework should be used also for platform economy and others support the idea that the clients are those who rate the platform and ensure self-regulation (Chen, 2019). The connection between people and personal data might be considered an ethical problem in the use of economy platform. The data on behavior should be subject to privacy (Codagnone, 2022);
2. The power structure between platform workers and their platforms deals with different typologies of workers: primary dependent workers that completely count for platform's earnings; partially dependent workers for whom platforms provide just a part-time job; supplemental workers that benefit of additional earnings from platforms (Chen, 2019). The challenges are related to less workers' protection, jobs' marketization, higher competition supported by no

barriers that determine high pressure in terms of working conditions and payment. Fragmentation of the work is a disadvantage for online labor platform and for platform economy in general. Another issue is related to competition and the potential results inducing monopolistic tendencies in a platform economy (Codagnone, 2022);
3. Workforce ecosystem management beyond the company refers to enterprises that make use of open talent economy. However, many firms do not accept to engage in a new workforce ecosystem of open talent while practices and various policies are not regulated (Chen, 2019).

The period of COVID-19 pandemic contributed to intensification of digital transformation and augmented the volume of online services and the number of users. In this context, there are specific advantages and limitations of platform economy brought by the pandemic.

The recent global epidemic has enhanced the lock-ins. There are significant differences between sectors in terms of revenue share and traffic during the pandemic. For example, these parameters have improved for marketplaces at national level, search engines and social media, but have worsen for travel and tourism sectors. New platforms appeared during the medical crisis for education and health sectors. GAFAM incumbents strengthened their position on online markets and remained active in M&A. The companies with traditional business encountered a critical situation during the medical crisis because their capacity to support innovation and adaptation has been affected. This unfavorable situation represented an advantage for platform companies that increased their market share. Bluetooth Low Energy technology for COVID-19 contact tracing has been developed by Google and Apple, but despite their benefit, they raise many concerns on aspects related to privacy (Codagnone, 2022).

In the EU, there is the Observatory on the Online Platform Economy that should support the Commission to monitor and regulate online platforms. This entity also manages problems related to transparency in online transactions, ranking made for search engine results, rights for online intermediates. The Observatory tackles three dimensions: economic significance, power over users and consequences of power. A sample of 56 platforms was analyzed by the Observatory on the Online Platform Economy and some conclusions were drawn in terms of economic significance: few platforms concentrate a large share of revenues; the platforms in the sample attracted a large amount of funding; for collaborative economy and social media platforms, the parent companies attracted more funds than the parent firms of the other platforms. The share of companies that made direct online sales was 21% in 2021, compared to 16%, in 2019. 10% of the companies made sales through marketplaces in 2021, compared to 6%, in 2019. During the pandemic, in 2021, 74% of the people using Internet ordered various products and services using online platforms, compared to 70% in 2019. The dimension related to power over users discusses aspects related to vulnerability of businesses to modifications made in the policies related to platforms, popularity among customers and competitive strategies of online platforms. The consequences of power are seen by both users and companies. The number of working platforms has rapidly grown and it is expected to have almost 45 million people working on platform by 2050 (EU Observatory, 2023).

However, in order to take maximum benefit of such platforms, not only the surrounding context is important, but also the ability to precisely determine the purpose of their use, their destination and to adapt accordingly. Virtual conferences, for instance, involve the use of dedicated platforms and a related specific approach, being more than just basic online meetings, in terms of formality and goal, as conferences' scope is to disseminate information on longer periods of time (Whyman, 2023).

With this in mind, we proceeded to the theoretical and applicative analysis of the same, undertaking to underline, based on a comparative analysis of specific econometric versus machine learning methods, how to better forecast related short time-series, so as to be able to predict the evolution of the number of participants to such particular platforms.

Therefore, this introduction continues with the theoretical background, rendering, conceptually, aspects concerning collaborative/multi-participant decision-making and platforms designed for teleconference, being followed by a case-study revealing the results obtained in the matter in a specific case, namely the one of the Romanian Academy (RA). The last section of the paper provides the conclusions drawn, accompanied by related discussions and future directions of research.

2. THEORETICAL BACKGROUND

A basic model designed for activities concerning decisions consists in more stages: intelligence, proposal of models and alternatives, selection of the optimal decision, assessment of the impact of the decision implementation, that might suggest resuming the process (Filip, 2022). In the field of collaborative decision-making, the process model developed by Simon could be extended for multi-participant area. The stages of this model could include the following elements:

- Preparation involves establishing the main features of the analyzed issue (aim, field, actual context, criteria, potential restrictions) and the empowering decision unit;
- Collective understanding includes agreement on a common perspective on the issue and on the implementation of the process;
- Solution proposal is based on the identification and advancement of the other suitable models to manage the issue;
- Negotiation and debate are needed to stimulate proposals and gain support among parties;
- Decision making is based on general agreement or favorable vote of the majority;
- Control consists in the elaboration of the report with the description of decision-making (Filip, 2022).

The most common collaboration forms are close collaboration (exchange of ideas to make decisions), asymmetric collaboration between people making decisions and their assistants /consultants and soft collaboration with anonymous members of the team (Suduc et al., 2009).

In uncertain times like Covid-19 pandemic, postponing decisions is a usual practice, but sometimes the decisions should be made fast. The involvement of more people in the team might help in generating more ideas and making a relevant decision. This involvement of more people supposes acknowledgement of decisions, selection of few people who will make the decision, involvement of experts who will applicate the decision, creation of forum for discussions. Making the decision to set up a nerve center is also essential for platform economy. Critical small choices could be made by anticipating more scenarios, selecting the most important options from a long list and asking other people to make small choices (Vallarasi, 2022).

Teleconferencing is the manner of interconnecting a varied number of individuals being located into different geographic areas, allowing various collaborative tasks, based on the real time propagation of information using certain software and devices**.**

All sorts of platforms are nowadays available for supporting teleconferencing, each of them having particular features (Suduc et al., 2009) that make it appropriate for different contexts.

2.1 **Evaluation of existing teleconferencing platforms**

Taking into account the relevant technical characteristics, such as information transparency, precision or response time; the quality of application, including factors like effective transparency scalability and flexibility; and the quality of provision, which involves aspects like price, provider independence, reputation, ease of adaptation, or integration with other applications, among others, along with considerations such as the number of participants supported per session, time limits (particularly for free platforms), user-friendly interfaces, and compatibility with various end devices, the selection process can be made more efficient.

Given the fact that, depending on circumstances, some platforms might fit better in relation to others, we are going to describe, thereafter, the five most frequently used ones (Software Testing Help, 2023; Walsh, 2023; Zoho, 2023; Zoom, 2023; Cisco, 2023; Microsoft, 2023; Google, 2023; Google Workspace, 2023).

Zoho Meeting – platform popular especially among medium and large organizations (more than 280,000 entities using it), available both in free format, allowing for up to 100 participants, with a time limit of 60 minutes, as well as in paid for access format (amounting up to €20/month), allowing for a large number of participants, no more than 250, in the case of Meetings, and no more than 3000, in the case of Webinars, in continuous daily sessions, usable on desktop applications (Mac, Windows, and Linux) and mobile applications for Android and iOS, browser-based, with Firefox and Google Chrome extensions, with virtual background, permit for audio/VoIP, webcam and file sharing, consent for reminder notes, turning to video, adding events to calendar, analytics and reports, settings related to email, authentication in two steps and recording, for all versions (Software Testing Help, 2023; Zoho, 2023)

Zoom – high definition video and voice platform, highly popular among different sized groups of people, at organizational level including, being frequently considered also by educational institutions, with a user-friendly free variant, allowing for up to 100 participants, but imposing a time restriction of just 40 minutes/session, and a paid for one: Pro, Business, Business Plus and Enterprise, ranging from 100 to 1000 participants/meeting, authorizing 30-hour sessions, the price of the same starting from €139.90 /year, featured by easy access, from any type of device, based on desktop applications or mobile applications for iOS and Android, a high level of integration with other application (2000+), with powerful security issues, based on Secure Socket Layer encryption, permitting the writing and share of whiteboards, team chat or mail and calendar benefits, making records and transcribing features, irrespective of the version considered, the cloud storage and Essential Apps, among others, being available, instead, in exchange for payment (Software Testing Help, 2023; Walsh, 2023; Zoom, 2023).

Webex Meeting – platform destined for both few or more persons, even big companies, with Free, Starter, Business and Enterprise versions, the first one allowing for up to 100 participants and setting a time limit of 40 minutes, while the latter ranging from a maximum limit of 150 to 1000 participants and sessions of maximum 24 hours, the prices to be incurred starting from €13.50/month, with high level of application integration (100+), advanced security based meetings, shared content, video with specific layouts, instruments that allow optimization of voice and elimination of noise, traditional messaging, interface sharing, tools designed to make records, cloud storage and file transfer and organization, for all existing versions (Software Testing Help, 2023; Cisco, 2023).

Microsoft Teams – very popular platform designed for seamless efficiency and collaboration, used by physical and legal entities of various dimensions, even educational elements, available both for free, allowing the connection of no more than 100 participants per session, limited to one hour, and in return for a payment of $4/month to $12.50/month, for Teams Essential, 365 Personal, Family and Business, with a one-to-one and group meeting duration of up to 30 hours, for a volume of participants amounting to 300, integrated with Office applications, providing meetings with video and calling options, including meeting joining without accounts, and giving access to specific backgrounds, planned meetings, noise elimination, screen sharing, visualization of default Outlook calendar, activities distribution and file incorporation, and offering phone or online support, without limitation, for all versions considered (Software Testing Help, 2023; Walsh, 2023; Microsoft, 2023).

Google Meet – high-definition video conference platform, mainly recommended for small businesses, with a free format of up to 100 participants and maximum 60 minutes/session, and a paid for one, for Business Starter, Standard, Plus and Enterprise, in exchange for more than 6 dollars each month during one year, unifying between 100 to 500 participants, being a desktop and Android based application, integrated with a full series of Google products, like Google Calendar or Gmail, with high security benefits, the data being encrypted, providing support, including permit for whiteboard or screen sharing, allowing for getting live captions and offering features such as on-meeting hand raise and question and answer type poll, the recording and 30 GB to unlimited storage capacity being allowed, in exchange, just for the paid versions (Software Testing Help, 2023; Walsh, 2023; Google, 2023; Google Workspace, 2023).

The above rendered teleconferencing platforms, selected based on their renown and frequency of use, given the limited space of the paper, represent just a small part of the extensive list with such platforms, herein being included, without limitation: GoTo Meeting, TrueConf Online, Skype, Whereby, BlueJeans, Slack, Jitsi Meet, Blackboard Collaborate, BigBlueButton, Dialpad Meetings (Software Testing Help, 2023), all of them sharing some common elements related to time and cost-saving, as well as to increase in efficiency for both individuals and organizations of any kind.

2.2 **Managing a teleconference**

An effective meeting involves a particular attention, the meeting agenda having to include the description of objectives, topics to be covered, related participants, in terms of number and identity, persons to take the floor and approached specific topics, meeting time and length (Indeed, 2023) and so on. But achieving a high level of efficiency implies much more than that, the consideration of the main managerial functions: making plans, organize elements, leading, evaluating and checking (Richard, 2021) and their thorough implementation while dealing with meetings, becoming a must.

As for teleconferences, there are some particular aspects that should be considered in addition, the former being significantly different from face-to-face meeting in terms of technical issues, hardware and software items needed, related knowledge as for the utilization of applications, devices, functions etc.

Given the above-mentioned functions, we are going to reveal thereafter the most important steps to be taken into account when coming about teleconference management (IEE Computing Soceity, 2023; IEEE Computing Society, 2023; Casamo, 2023; Indeed, 2023; Stack, 2023; Condeco, 2016; FreeConferenceCall, 2022; Free Management Books, 2023; InTheBlack, 2020; and Martin, 2023).

Teleconference planning (3-6 months or more, in advance (IEEE Computing Society, 2023)) requires:

- Scheduling teleconferences in the most convenient moment of the day (most conferences occur between 10 a.m. and 2 p.m., on Tuesdays and Wednesdays), taking also into account the fact that participants to the meeting might be located in different time zones and they should be all gathered at the same time, also determining the probable length of it (most conferences last for about 45-60 minutes) (Casamo, 2023; Stack, 2023; Condeco, 2016; FreeConferenceCall, 2022; Free Managemetn Books, 2023; InTheBlack, 2023);
- Identifying the most appropriate locations in terms of signal, ambience, light and so on and recommend participants, in due time, to do the same (Indeed, 2023; FreeConferenceCall, 2022);
- Preparing the topics to be discussed, under a basic form and an extended version, so as to be able to keep the discussion going and

on track, avoiding idle times, the drawing up of some additional materials, just in case, being highly recommendable (Casamo, 2023; Free Managemetn Books, 2023; InTheBlack, 2023);

- Preliminary structuring of discussions, well determining the order and the intervention time of speakers (Casamo, 2023);
- Setting the budget relating to the host platform, data storage, professional staff support and other related costs, as the case may be (IEEE Computing Society, 2023);
- Predefining volunteer roles and training volunteers, such as chairs, moderators or technical experts, among others (IEEE Computing Society, 2023);
- Defining virtual meeting rooms as networking lounges, designating someone to monitor the same and to answer specific questions, if any (IEEE Computing Society, 2023);
- Selecting the most appropriate platforms, fit for the meeting related needs, given the number of participants, the facilities provided, the available financial funds etc. (IEEE Computing Society, 2023; Condeco, 2016);
- Choosing the proper teleconferencing equipment, consisting in computers, tablets, phones or similar devices (IEEE Computing Society, 2023; Martin, 2023);
- Getting acquainted with the technical aspects relating to the selected platform and equipment, in terms of needed functions, checking in advance the ability of adequately using the same (Casamo, 2023; Martin, 2023);
- Providing meeting related instructions, such as date, time and length, topics to be discussed, documents, notes, preliminary work or reading required, platform to be used and so on (Casamo, 2023; Indeed, 2022; Stack, 2023; FreeConferenceCall, 2022; Free management Books, 2023);
- Testing in advance the functionality of the platform and the internet connection, by conducting a few trials and asking the participants to do the same, to ensure you can hear one another, so as to avoid miscellaneous technological errors that might impede the proper conferencing process (Casamo, 2023; Indeed, 2022; Stack, 2023; Martin, 2023);
- Generating and sending the access codes one week, two days before and the very day of the meeting, to the right recipients (Indeed, 2022; Stack, 2023; FreeConferenceCall, 2022; Free management Books, 2023).

Teleconference organizing involves:

- Keeping the process simple, the presentation brief and the schedule short, as long as this does not affect the quality of the meeting (Indeed, 2022; Stack, 2023);
- Structuring the discussion during the process, as people have to be moderated in order to express their point of view about the discussed topics or in case of Q&A sessions, a well-established process in the matter being extremely important (Casamo, 2023);
- Sharing the whiteboard and all necessary materials, during the meeting;
- Setting breaks, mainly if the conference lasts for longer time intervals (for instance about or more than two hours) (IEEE Computing Society, 2023; FreeConferenceCall, 2022);
- Creating meeting minutes and recording the session, as long as this second facility is available, for their subsequent distribution to the target audience (IEEE Computing Society, 2023; Indeed, 2022; Free management Books, 2023).

Teleconference leading implies:

- Guiding the process, not allowing for unauthorized participants, for the ones being late, manifesting inappropriate behavior, yelling or speaking all at once and so on, being ready to mute or even to disconnect the same, depending on circumstances (Indeed, 2022);
- Welcoming and introducing the persons joining the meeting or at least presenting the speakers in brief (InTheBlack, 2020);
- Muting hosts and participants, whenever it is necessary, while not taking the floor, in order not to disturb discussions (Stack, 2023);
- Avoiding idle times, by taking the floor for expressing the own point of view about the topics discussed (one reason for having two hosts available) or by inviting the ones having been silent until then to express their opinion (IEEE Computing Society, 2023; InTheBlack, 2020);
- Ending the discussion in a positive note, making participants feel that their contribution was appreciated (InTheBlack, 2020).

Teleconference evaluating and controlling includes:

- Downloading and saving all materials recorded/posted/disseminated, to archive them for further use, if any (IEEE Computing Society, 2023);
- Getting feedback from participants so as to get their point of view about the teleconference strengths and weaknesses (IEEE Computing Society, 2023, Indeed, 2022);

- Learning lessons, in order to understand the steps to take for teleconferences to come in order to improve the entire process, both in terms of content and in terms of technical aspects(IEEE Computing Society, 2023; Indeed, 2022; Martin, 2023).

A proper teleconference management sets the grounds not just for an efficient fulfilment of the pre-established goals for a particular case, with unexpected events or less errors, even for acquiring a good notoriety of these services and for improving their trust in relation thereto.

3. MATERIALS AND METHODS

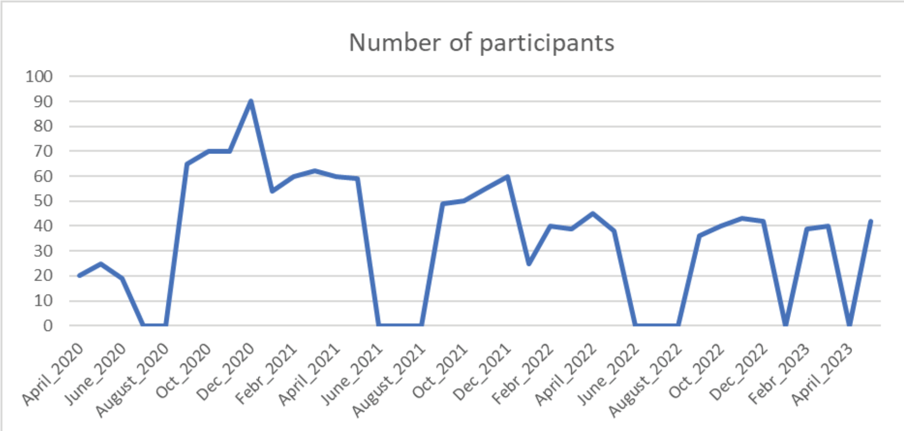
Time series forecasting is a vital task in data analysis and predictive modelling, involving the prediction of future values based on the past values of the same variable or the past values of other variables too. In the former case we have at our disposal a univariate time series, i.e., a single sequential dataset. This characteristic makes the forecasting of the future values more difficult because we have only the previous values of the same indicator and, based on its past behavior, we have to predict the future. In this context, machine learning methods proved to be essential for capturing intricate patterns and relationships within time series data, enabling accurate predictions and informed decision-making.

Considering the monthly volume of the participants in online meetings organized by the prestigious Romanian Academy in the period 2020:01-2023:05, a SARMA model is proposed to make forecasts for the next three months. Figure 1 suggests the evolution of the number of participants before and after the use of teleconferencing platform in the period April 2020-May 2023. Summer months are usually characterized by no participants, while ascending trends are observed in the first five months of each year and in the last four months of the year.

In the period April-August 2020, the platform was not used. In the period September-December 2020, the platform was intensively used with a maximum number of participants of 90 people. Since January 2021, the hybrid meetings have started, and the number of participants decreased. In the summer periods (June-August), no participants were registered in 2021 and 2022.

The evolution of the volume of participants to online meeting in the RA before and after the implementation of the teleconferencing platform

Figure 1



Source: Authors based on own data

SARIMA models extend the time series ARIMA models by considering repetition in the evolution of an indicator with a certain lag denoted by s. The seasonal element of the time series might be composed by: 1. an autoregressive seasonal part of order P, where P is the number of autoregressive seasonal components; 2. a moving average seasonal part of order Q, where Q is the number of moving average seasonal components; 3. components identified after a number of seasonal differencing D, where D is the number of differencing to achieve a stationary time series. The autocorrelation and partial autocorrelation functions (ACF and PACF) should be analyzed to identify a SARIMA process. If time series presents seasonal factors, then the values of PACF and ACF are significantly different from zero for moments showing seasonal and non-seasonal components. The stationary SARMA (p,q) x $(P,Q)_s$ with D=0 is written in its general form as:

$$\phi(L)\varphi(L^s)(y_t - \mu) = \theta(L)\Theta(L^s)\varepsilon_t. \quad [1]$$

The seasonal components ensure the correlations decomposition by seasons and are represented by SAR(P) and SMA(Q):

$$\varphi(L^s) = 1 - \varphi_1 L^s - \varphi_2 L^{2s} - \cdots - \varphi_P L^{P_s} \quad [2]$$

$$\Theta(L^s) = 1 - \Theta_1 L^s - \Theta_2 L^{2s} - \cdots - \Theta_Q L^{Q_s} \quad [3]$$

The non-seasonal components are represented by AR(p) and MA(q):

$$\phi(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \quad [4]$$

$$\phi(L) = 1 - \theta_1 L - \theta_2 L^2 - \cdots - \theta_q L^q \quad [5]$$

The short-run impact is controlled by non-seasonal components. In practice, p and q are selected to have checked the following inequalities: $p < 0.5s$ and $q < 0.5s$.

We may have different periodicity for autoregressive and moving average seasonal components. In this case, (p,q) x $(P,Q)_{s,s'}$ is written as:

$$\phi(L)\varphi(L^s)(y_t - \mu) = \theta(L)\Theta(L^{s'})\varepsilon_t \quad [6]$$

s- autoregressive component seasonality; s'-moving average component seasonality.

Before estimations, unit root tests or stationarity tests should be applied. In this case, Augmented-Dickey Fuller test (ADF), Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test, Phillips-Perron test (PP) are employed. The ADF test is based on three regression models (with tendency and constant, with constant and no trend and no constant) that include lagged variable to avoid errors' autocorrelation, knowing the process y_t:

$$\Delta y_t = \alpha y_{t-1} + \beta + \gamma t + \sum_{j=1}^{p} \delta_j \Delta y_{t-j} + \varepsilon_t \quad [7]$$

$$\Delta y_t = \alpha y_{t-1} + \beta + \sum_{j=1}^{p} \delta_j \Delta y_{t-j} + \varepsilon_t \quad [8]$$

$$\Delta y_t = \alpha y_{t-1} + \sum_{j=1}^{p} \delta_j \Delta y_{t-j} + \varepsilon_t \quad [9]$$

$\alpha, \beta, \gamma, \delta_j$ – parameters, t-index for time and ε_t- white noise (null average, constant variance, uncorrelated with y_{t-j} for any j=1,2,..,p)

The null assumption considers non-stationary process, H_0: $\alpha = 0$. The null hypothesis of PP also states the existence of unit root, while in the case of KPSS, the null hypothesis assumes stationarity. The sequential testing procedure proposed by Dickey and Pantula is used to check for unit root starting with the time series double differenced, model with trend and intercept.

One of the most advanced and effective machine learning techniques for univariate time series forecasting is Long Short-Term Memory (LSTM) networks, introduced by (Hochreiter, & Schmidhuber , 1997). LSTMs are a specialized type of recurrent neural network (RNN) designed to handle the challenges posed by sequential data, such as long-range dependencies and vanishing gradient issues. These networks excel at capturing complex temporal patterns, making them particularly well-suited for modelling and predicting time-evolving phenomena (Zhang, 2003; Lippton et al., 2015; Lim and Zohren, 2021). Unlike traditional RNNs, LSTMs are designed to alleviate the vanishing gradient problem (Hochreiter, 1998; Rehmer and Kroll, 2020) and capture long-range dependencies within sequential data.

LSTMs achieve this by introducing memory cells that can store information over extended periods. An LSTM network consists of repeating units called cells, each of them having three main components: an input gate, a forget gate, and an output gate. These gates regulate the flow of information, allowing the network to selectively remember or forget past observations and decide what to pass on to the next time step. The structure of a LSTM cell is shown in Figure 2. Here, the symbols ⊗ and ⊕ are element-wise operations, c_t is the cell state vector, h_t is the hidden state vector (the output vector of the LSTM unit) and X_t is the input vector of the LSTM unit.

The input gate takes input from the current time step and decides which information to add to the memory cell. It involves a sigmoid activation function that transforms the input and decides how much of it should be added.

The forget gate determines what information to discard from the memory cell. It considers the previous memory cell output value and the current input, and it involves a sigmoid activation function to determine how much information to forget. The memory cell is updated based on the input gate and forget gate decisions. The input gate's output is element-wise multiplied (Hadamard product) with a candidate new value (tanh activation) and added to the previous memory cell value, as determined by the forget gate.

The output gate decides what to output from the memory cell. It considers the current input and the updated memory cell value. The output gate output is then passed through a sigmoid activation and elementwise multiplied with the tanh of the memory cell value to produce the final output.

All the LSTM cells are connected in a chain, allowing them to process sequential data one step at a time while maintaining a memory of past information. This type of architecture enables LSTMs to effectively capture and utilize long-range dependencies in time series data.

The equations describing the behavior of an LSTM cell are (Hochreiter and Schmidhuber,1997):

$$f_t = \sigma_g(W_f X_t + U_f h_{t-1} + b_f) \qquad [10]$$

$$i_t = \sigma_g(W_i X_t + U_i h_{t-1} + b_i) \quad [11]$$

$$o_t = \sigma_g(W_o X_t + U_o h_{t-1} + b_o) \quad [12]$$

$$\tilde{c}_t = \sigma_c(W_c X_t + U_c h_{t-1} + b_c) \quad [13]$$

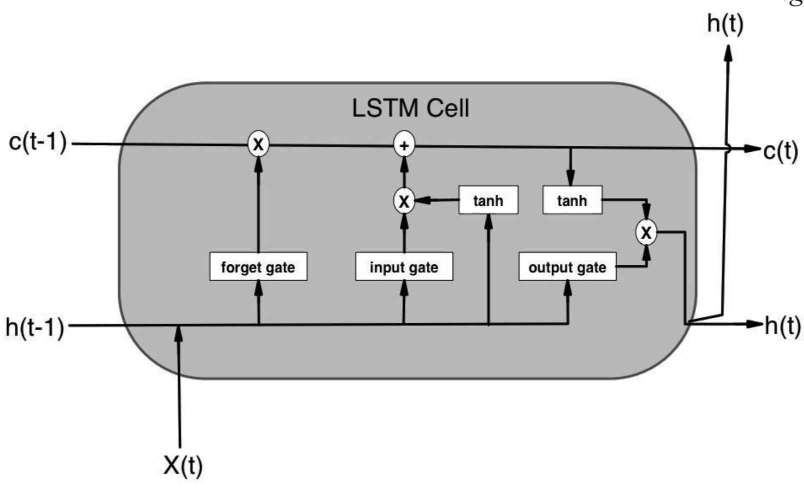
$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad [14]$$

$$h_t = o_t \otimes \sigma_h \, c_t \quad [15]$$

where W_f, W_i, W_o, W_c are the weight matrices of the input for the forget, input, output and the memory cell connections, U_f, U_i, U_o, U_c are the corresponding matrices for the recurrent connections, $\otimes$ is used for the Hadamard product, $X_t \in \mathbb{R}^p$ is the input vector, $f_t \in (0,1)^h$, $i_t \in (0,1)^h$, $o_t \in (0,1)^h$ are the activation vectors for the forget, input and output gates, $h_t \in (-1,1)^h$ is the hidden state vector, $\tilde{c}_t \in (-1,1)^h$ is the cell input activation, $c_t \in \mathbb{R}^h$ is the cell state vector, b_* are the biases vectors for each gate. The superscript *p* stands for the dimension of the input space and for the number of hidden and the superscript *h*, for the number of hidden units. The activation functions for the forget, input and output gates, σ_g, is the sigmoid function, while σ_c and σ_h are both the hyperbolic tangent function.

The structure of an LSTM cell

Figure 2



Source: own representation

The key features of LSTM networks for time series forecasting can be summarized as follows (Chen et al., 2022):

- Long-Term Dependencies: LSTMs are capable of capturing relationships between distant time steps, enabling them to learn and exploit complex patterns that may span across the entire data sequence;
- Solving the Vanishing Gradient Problem: LSTMs use gating mechanisms to control the gradient flow during training, addressing the vanishing gradient issue commonly encountered in traditional RNNs. This enables more stable and effective training;
- Flexible Input Handling: LSTMs can handle varying lengths of input sequences, making them adaptable to time series data with irregular and/or missing data points;
- Multiple Memory Cells: LSTMs consist of multiple memory cells that allow the network to remember different aspects of the input sequence independently;
- Nonlinear Transformations: LSTMs apply nonlinear transformations to input data at each time step, allowing them to model intricate temporal patterns and fluctuations;

The effectiveness of LSTM networks in capturing intricate time series patterns has led to their widespread use and success in various real-world forecasting scenarios.

In the context of time series forecasting, LSTM networks can be trained to predict future values based on historical observations. They take a sequence of past data points as input and produce corresponding forecasts as output. By learning from past patterns, LSTMs can capture seasonality, trends, and other complex temporal relationships, enabling accurate predictions for various applications such as finance (Murat et al., 2020; Rundo et al., 2019), stock market forecasting (Jaydip and Mehtab, 2022; Sreelekshmy et al., 2017), weather prediction (Zahra and Suykens, 2020; Miao et aal., 2020; Yong et al., 2019), speech recognition (Oruh et al., 2022), handwriting recognition (Carbone et al., 2020), and more. For a review of LSTMs and their application see for example Yong et al. (2019) and the references therein.

To implement LSTM networks for time series forecasting, we proceeded as follows:

- pre-processed the data, structuring it into suitable input-output pairs for supervised learning;
- defined the LSTM architecture;
- trained the model using historical data, performing a grid search for the best hyperparameter values;

- used the trained model to make predictions on unseen future data points and assessed the prediction accuracy.

LSTMs has different architectures each with hyperparameters that can greatly influence the performance of the predictions. In our experiments we considered three LSTM models for one step ahead predictions and two models for three steps ahead predictions.

Stateless and stateful are two variations of recurrent neural networks (LSTM included) that have two different ways of handling sequential information and managing the hidden states across time steps. Stateless LSTM networks do not retain any information about previous time steps i.e., each input sequence is treated independently, and the network resets its hidden state after processing each sequence. Since stateless LSTMs do not consider temporal dependencies between sequences, they may struggle with capturing long-range patterns and dependencies in data. On the other hand, stateful LSTM networks maintain memory of previous time steps and carry forward their hidden states. This enables them to capture long-term dependencies and relationships between sequential data points. By preserving the context across time steps, stateful LSTMs are more effective in modelling complex patterns that span multiple sequences. While for univariate time series forecasting stateful networks seemed to be a better choice, in our case the data set is small, and time sequences that the network may learn are short, thus making stateless networks also appropriate. We employed both types of networks in our experiments, but due to the limited computational resources, we tried all combinations of hyperparameters for stateless networks and only a limited set for stateful versions, these models having higher demands for memory and computing power.

A parameter that we've also took into consideration when searching for the best network setup was related to shuffling or not the input sequences when training the network. Shuffling the data involves randomly rearranging the order of the input sequences before feeding them into the LSTM network during training. This practice is common in many machine learning tasks to prevent the model from learning spurious correlations that may arise from the sequential order of the data. However, when it comes to univariate time series forecasting, shuffling may not always be the best choice because shuffling can disrupt the inherent order of the sequences and hinder the model's ability to learn meaningful patterns. Time series data often follows trends, seasonality, and other sequential patterns that should be preserved to make accurate forecasts. Shuffling the data could result in the LSTM losing the ability to capture these crucial temporal relationships, leading to suboptimal performance. Conversely, there are situations where shuffling the data can be beneficial. For instance, if the time series data lacks strong temporal dependencies and each data point is relatively independent of its neighbors, shuffling could help the model generalize better and reduce overfitting. Additionally, shuffling might be advantageous when using stateful LSTMs, as it can mitigate the risk of the model memorizing the sequential order of the training data and improve its ability to generalize to unseen sequences. We experimented with both shuffling and non-shuffling approaches.

Another hyperparameter considered in our approach was the dropout rate. Dropout is a regularization technique commonly used in deep learning, including Long Short-Term Memory (LSTM) networks. Dropout is employed in LSTMs to mitigate overfitting, which occurs when a model becomes too specialized to the training data and performs poorly on new, unseen data. In the context of LSTM networks, dropout involves randomly setting a fraction of the output values of LSTM units to zero during both training and inference. This has the effect of temporarily "dropping out" certain connections within the network, forcing the model to become more robust and preventing it from relying too heavily on specific features or patterns in the training data. By introducing dropout, LSTMs become less likely to overfit and are better able to generalize to new sequences. The dropout rate should be carefully chosen through experimentation, as too high a dropout rate can lead to underfitting and reduced model capacity.

The other hyperparameters that we used to search for the network configuration with the best accuracy prediction was the number of neurons in the LSTM layer(s) and the number of training epochs.

In the pre-processing step we built a dataset suitable for a supervised learning method, as pairs of the form *(X, Y)* where *X* is the input vector and *Y* is the output value. In our specific case X is a vector consisting of a number of past values and Y is the next value in the timeseries that we want to predict.

If we denote by $X = (X_0, X_1, X_2, \ldots, X_T)$ our timeseries, the training and test sets are of the form:

$$\left(X_{t-nlags-1}, X_{t-nlags-2}, \ldots X_{t-1}, X_t\right), (X_{t+1}) \qquad [16]$$

$$(X_{t-nlags-2}, X_{t-nlags-3}, \ldots X_t, X_{t+1}), (X_{t+2}) \qquad [17]$$

$$(X_{t-nlags-3}, X_{t-nlags-4}, \ldots X_{t+1}, X_{t+2}), (X_{t+3}) \qquad [18]$$

where *nlags* is the number of lags (past values) used to build sequences of data points to predict the value at $t+1$, $t+2$, $t+3, etc.$ (one point ahead forecasting) and of the form:

$$(X_{t-nlags-1}, X_{t-nlags-2}, \dots X_{t-1}, X_t) , (X_{t+1}, X_{t+2}, X_{t+3}) \qquad [19]$$

$$(X_{t-nlags-2}, X_{t-nlags-3}, \dots X_t, X_{t+1}) , (X_{t+2}, X_{t+3}, X_{t+4}) \qquad [20]$$

$$(X_{t-nlags-3}, X_{t-nlags-4}, \dots X_{t+1}, X_{t+2}) , (X_{t+3}, X_{t+4}, X_{t+5}) \qquad [21]$$

for three points ahead forecasting (i.e., to predict the values at ($t+1$, $t+2$, $t+3$), ($t+2$, $t+3$, $t+4$), ($t+2$, $t+4$ $t+5$) etc.).

4. RESULTS AND DISCUSSION

First, any potential detection of the unit root in data for the volume of participants to online meeting in the RA is verified. The ADF, PP and KPSS tests are applied using the sequential testing procedure previously mentioned, the results being seen in Table 1.

The results of tests to check for stationarity for the volume of participants to online meeting in the RA (2020:04-2023:05)

Table 1

Time series in:	**Type of model for ADF test:**	**ADF stat.**	**Type of model for PP test:**	**PP stat.**	**Type of model for KPSS test:**	**KPSS stat. (critical value at 1% level: 0.739)**
the second difference	No Constant and no Linear Trend	-6.838304 (<0.01)	No Constant and no Linear Trend	-30.37906 (<0.01)	Constant	0.079664
the first difference	No Constant and no Linear Trend	-6.85819 (<0.01)	No Constant and no Linear Trend	-7.788636 (<0.01)	Constant	0.096643
Level	Constant and Linear Trend	-5.145959 (0.0014)	Constant	-3.423024 (0.0164)	Constant	0.155896

Source: own calculations in EViews 9

Note: p-values in brackets for ADF and PP tests

The results suggest that the time series for the volume of participants to online conferences is stationary at 5% significance level. According to ADF and KPSS tests, the stationarity is verified at 1% significance level, while PP test suggests stationary data series at 5% significance level.

More SARMA models were run on this time series and only a valid model was identified that is presented in Table 2: SARMA (1,0) $(1,0)_{12}$ model. The parameters are significant at 5% level. DW statistic is approximately equal to 2, supporting errors non-serial correlation that might be observed also from the correlogram of residuals represented in the Appendix 1.

The results of estimation for SARMA (1,0) $(1,0)_{12}$ model

Table 2

Variable	Coef.	Std. dev.	t calc.	p-value
Constant	33.50398	12.81867	2.613687	0.0132
AR(1)	0.529696	0.128859	4.110656	0.0002
SAR(12)	0.678021	0.166548	4.071026	0.0003
Sigma squared	259.5667	58.12348	4.465781	0.0001
R-squared: 0.589801	Adjusted R-squared: 0.553607	F-statistic (p-value in brackets): 16.29551 (0.000001)	Durbin-Watson statistic: 1.949173	Jarque-Bera stat. (p-value in brackets): 1.266491 (0.530866)

Source: own calculations in EViews 9

According to Jarque-Bera test, we do not have proof to reject the normal distribution of errors at 1% significance level. Dynamic and static forecasts are provided in Appendix 1. The static forecasts perform better than the dynamic ones according to forecast accuracy measures. Accuracy measures like mean absolute error (MAE), root mean squared error (RMSE), mean absolute percentage error (MAPE) and Theil inequality coefficient present lower values for static predictions compared to dynamic ones. Own forecasts are provided for the horizon June 2023-August 2023. The short-run dynamic forecasts for the volume of participants to online conferences in the RA using SARMA (1,0) $(1,0)_{12}$ model (horizon: June 2023-August 2023) looks as follows: 34 (June 2023), 25 (July 2023) and 24 (August 2023), while the static ones indicate 14 (June 2023) and 0 (July and August 2023).

In our experiments with LSTM networks, we used the last 14 data points from the original timeseries to build the test set, to assess the quality of predictions, and the rest of data points for the training purposes.

We used 3 types of LSTM networks for one step ahead predictions, namely:

- A very simple structure with one LSTM layer and one output layer;
- A stacked approach where we use two consecutive LSTM layers and one output layer;
- A Bi-LSTM network: it consists of two LSTMs, which makes the input to flow in both directions, forward and backwards. This

approach was reported to substantially improve the forecasting accuracy [36].

For three steps ahead prediction we used two types of networks:

- A Stacked model with two LSTM layers outputting a vector that can be interpreted as a multistep forecasting;
- An Encoder-Decoder architecture, a special type of network design introduced by (Sutskever, 2014) for Seq2Seq problems with good performances for multi-step predictions (Chandra et al., 2021). Encoder-Decoder LSTM networks manage variable-length input and output sequences through a two-step process. Initially, they encode individual input sequences one by one, utilizing a latent vector representation. Subsequently, these sequences are decoded from the said representation.

Table 3 gives all the hyperparameters and their values that we considered when experimenting with LSTMs for our time series forecasting. We employed a grid search by varying all these parameters to find the best model in terms of minimum MSE.

The parameters of the networks and their values

Table 3

Parameter	Values	
Number of neurons in the LSTM layer(s)	50,100,150	
Number of lags	One step ahead prediction	Three steps ahead prediction
	1,2,3,4,5,6,7,8,9	4,5,6,7,8,9
Droupout rate	0, 0.2, 0.4	
Number of training epochs	100, 500, 1000	
Shuffling	True/False	

Source: own representation

We run the experiments using Python ver. 3.9 with Pandas 2.0.3, Keras 2.13.1, TensorFlow 2.13.0, Scikit-learn 1.3.0 and NumPy 1.24.3 libraries under the Windows 11 operating system. For all tests we used the Mean Squared Error as the loss function and the ADAM optimizer.

The activation function used for the LSTM units was the Rectified Linear Unit function (ReLU) (Nair and Hinton, 2010) which is more computational efficient than tanh or sigmoid functions and also helps mitigate the vanishing gradient problem, which can occur with sigmoid and tanh activation functions.

Due to the stochastic nature of the training process of the LSTMs, as well as in order to provide some uncertainty estimations, for each type of networks and for each value of hyperparameters, we repeated the experiments 30 times and reported the mean value of the RMSE together with its standard error.

In table 4 we present the results for the best model according to the RMSE on the test set, for each of networks used, in the case of one step ahead prediction.

Network configurations for the minimum value of the RMSE for one step ahead predictions

Table 4

Network type	**Simple**	**Stacked**	**BiLSTM**
Min. RMSE	20.25	20.23	19.06
Std. Err. of RMSE	3.95	1.05	0.38
Number of Lags	5	7	9
Dropout rate	0	0.2	0.4
Number of neurons	150	150	150
Number of training epochs	1000	500	1000
Shuffle input data	False	True	True

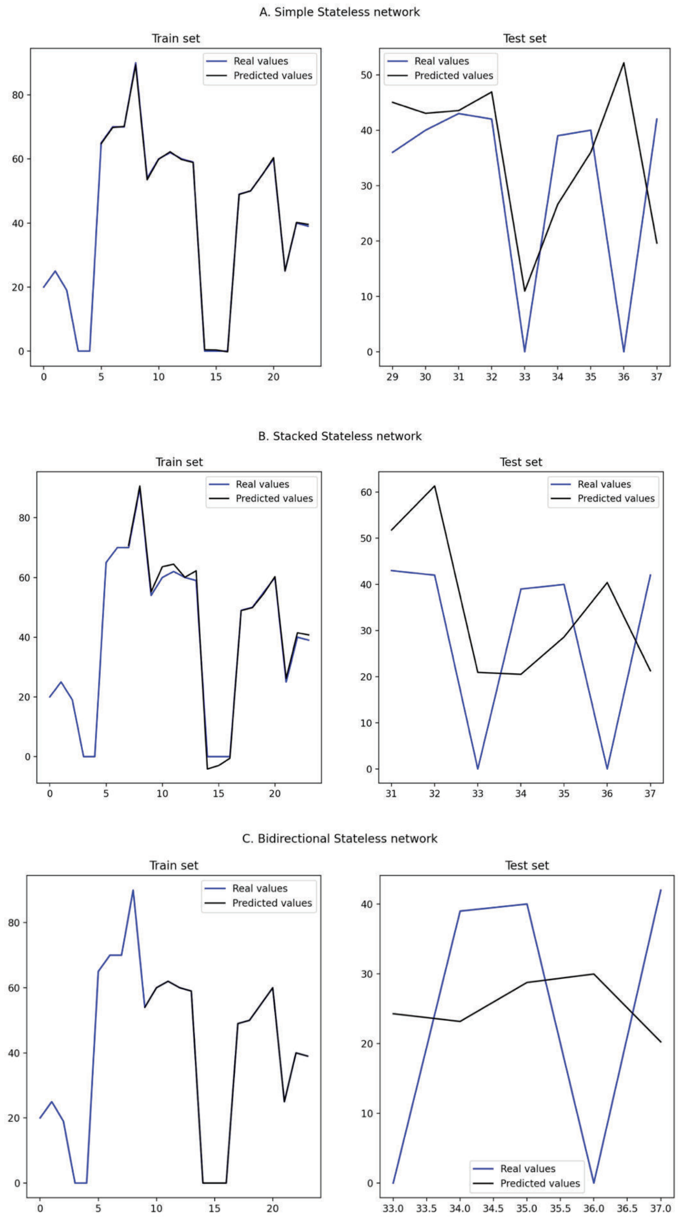
Source: own representation

The minimum RMSE was obtained by BiLSTM, which is in line with previous research (Siami-Namini et al., 2019), but the differences between the three networks are not very consistent. As expected, increasing the number of neurons and the number of training epochs results in a better prediction and shuffling the input data gave better results in two cases. Increasing the number of lags used to predict the next value also decreases the RMSE, which means that longer sequences used for training purposes give better predictions.

In Figure 3 we show, in the case of the three LSTM network architectures (Simple, Stacked and BiLSTM), the real and predicted values for the training and testing sets.

The real values versus the predicted values for the three types of networks

Figure 3



Source: own representation

Table 5 shows the results obtained for three steps ahead predictions. Encoder-Decoder type of network outperforms the Stacked model, which is in line with other results (Du et al., 2019). However, this improved prediction comes with the cost of a longer training time. Again, a higher number of neurons leads to a better performance and a longer series of past values used to predict the future gives better results.

Figures 4 and 5 show sequences of real values versus predicted values for the Encoder-Decoder network described in Table 5, for the training set and the testing set.

Network configurations for the minimum value of the RMSE for three steps ahead predictions

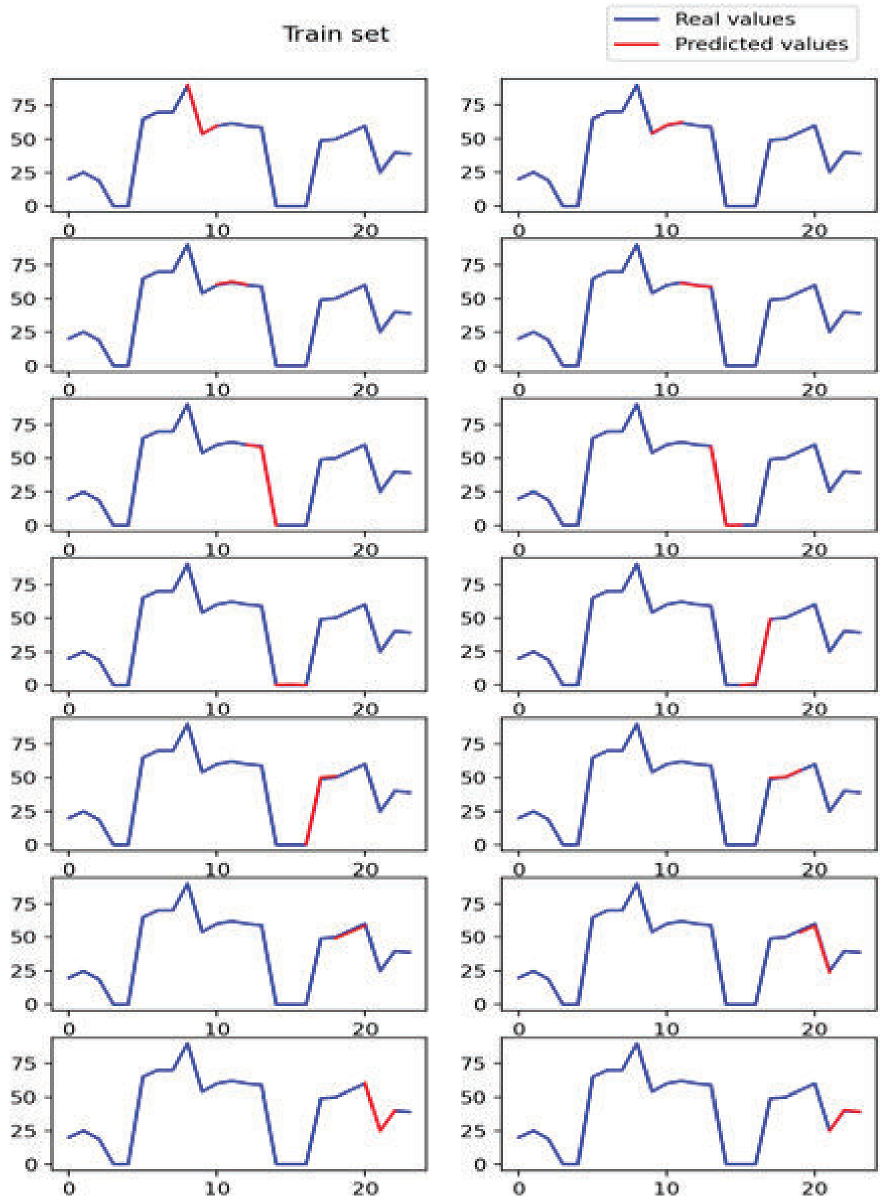
Table 5

Network type	**Simple**	**Stacked**
Min. RMSE	21.09	17.93
Std. Err. of RMSE	2.07	0.98
Number of Lags	9	8
Dropout rate	0	0.2
Number of neurons	150	100
Number of training epochs	100	500
Shuffle input data	True	False

Source: own representation

Train set: real values versus predicted values – three steps ahead prediction for the Encoder-Decoder model

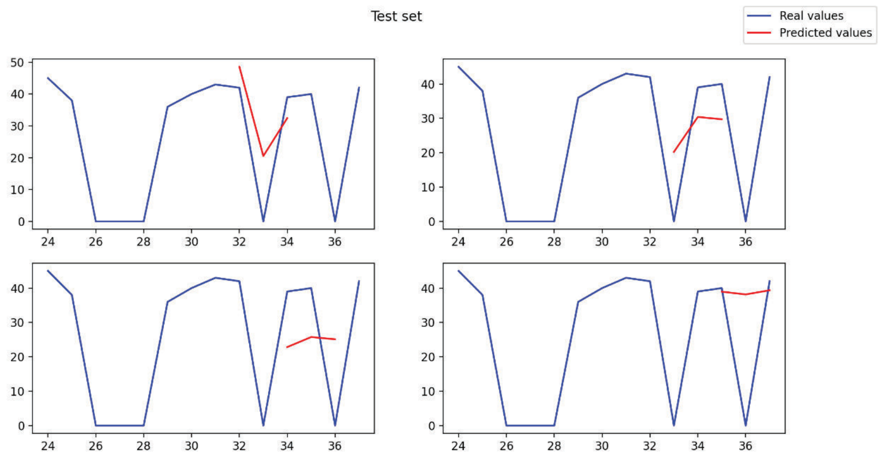
Figure 4



Source: own representation

Test set: real values versus predicted values – three steps ahead prediction for the Encoder-Decoder model

Figure 5



Source: own representation

In table 6 we compare the stateless and stateful approaches for the network configurations given in tables 3 and 5. The stateful configuration used a batch size of one, which is equivalent to online training. It can be easily observed that, in our case, stateful drastically degrades the prediction performance (Table 6).

A comparison between the stateless and stateful LSTMs

Table 6

Network type	Simple	Stacked	BiLSTM	Stacked	Encoder-Decoder
Min. RMSE – stateless	20.25	20.23	19.06	21.09	17.93
Min. RMSE – stateful	31.12	29.24	27.21	25.34	23.89
Number of Lags	5	7	9	9	8
Dropout rate	0	0.2	0.4	0	0.2
Number of neurons	150	150	150	150	100
Number of training epochs	1000	500	1000	100	500
Shuffle input data	False	True	True	True	False

Source: own representation

The minimum RMSE for stateless is registered for encoder-decoder, while the lowest value for stateful is observed in the same case. SARIMA approach has the advantage of much lower computational complexity than ML methods and the possibility of uncertainty quantification. In the case of LSTM, we have lower RMSE in one case, but the higher computational complexity is the main limitation of the method.

5. CONCLUSIONS

Quantifying platform work remains problematic, especially when seeking to generalize about the entire population. Surveys, administrative data, and big data all face limitations in this regard. The most reliable data sources recognize these shortcomings and utilize techniques such as triangulation, population-based weighting, and others to mitigate them (Pesole, 2021). In the overall EU and for each Member States, policymakers should explore mandating platforms to share administrative data in exchange for preferential tax treatment (as in Belgium) or as a condition for operating within their jurisdictions (as with AirBnB in Amsterdam) (Baselgia and Martinez, 2023). This approach can help ensure regulatory compliance and facilitate better socio-economic analysis.

According to Eurostat, in 2022, approximately 3.0% of individuals aged 15-64 participated in digital platform work for at least one hour in the preceding year, based on a pilot survey conducted in 16 EU countries and one EFTA country. The highest prevalence of platform work was observed among those with tertiary education (4.3%), while the lowest rates were found among individuals with lower secondary education (1.8%). Men were more likely to engage in platform work than women (3.2% vs. 2.8%). Eurostat initiated data collection on the emerging phenomenon of digital platform employment in 2022 through a pilot survey within the European Union Labour Force Survey (EU-LFS) (Eurostat, 2024).

Lafuente et al. (2024) assessed the performance of the global digital platform economy using a nonparametric network model (data envelopment analysis) applied to a sample of 116 states in 2019. The designed model accounts for geographic diversity and the complex interactions among governance entities, digital platforms, companies using platforms, and final users. Key findings reveal significant heterogeneity in countries' platform economy configurations, suggesting that a tailored policy approach could yield more effective results (Lafuente et al.,2024). To achieve qualitative improvements in the system, policies focused on accelerating the digital platform economy should be informed by an analysis of its key factors.

The comparative analysis of specific econometric versus machine learning methods, meant for identifying the most appropriate instruments useful for forecasting short time-series, particularly applied to the analysis the evolution of the number of participants to dedicated virtual platforms, in the case of the RA generated interesting results.

Concerning the econometric methods, the unit root ADF and KPSS tests applied revealed that all series are stationary, while the subsequent SARMA models used resulted in only one statistically validated variant. The static and dynamic forecasts indicated a superior performance of the former in relation to the latter in terms of accuracy, while the MAE, RMSE, MAPE and Theil inequality coefficient presented lower values for static predictions as against the dynamic ones. The forecasts provided for the horizon June 2023-August 2023 revealed an expected volume of participants to the online meetings of the RA of 34, in June 2023, 25, in July 2023, and 24, in August 2023, in the dynamic case, respectively of 14, in June 2023, and 0, in July and August 2023, in the static one.

As for the alternative approach, we resorted to LSTM networks, using the last 14 data points from the original timeseries to build the test set, to assess the quality of predictions and the remainder of them for the training purposes. Out of thev3 types of such networks, considered for one step ahead predictions, according to the RMSE on the test, the minimum value was obtained by BiLSTM, however the differences between the three networks not being significant. The increase of the number of neurons and the number of training epochs resulted in a more accurate prediction and shuffling the input data gave better results in two cases. while the increase of the number of lags used to predict the next value decreased RMSE, suggesting that longer sequences used for training purposes provide better predictions. The results obtained for three steps ahead predictions indicated that the Encoder-Decoder type of network outperforms the Stacked model, such improved prediction involving, unfortunately, a longer training time. The same as for the previous rage of models, a higher number of neurons led to a better performance and a longer series of past values used to predict the future generated better results. As for the stateless and stateful approaches for the network configurations, it was ascertained that the latter drastically degrades the prediction performance.

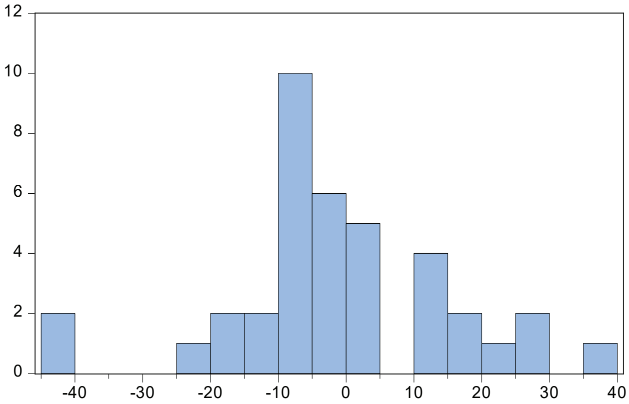
Overall, we observed an important limitation of the LSTM, namely the complexity of computations and the uncertainty regarding the accuracy of results, as compared to the econometric approach. Further research might consider the implementation of conformal machine learning techniques, so as to obtain uncertainty quantification too, including a larger number of LSTM architectures. All in all, despite the intensive utilization of machine learning techniques in the actual research, they did not generate better results than econometric models in all the cases.

The ability to anticipate fluctuations in platform participation, as mentioned earlier, can have practical applications for platform providers, especially when considering flexible paid subscription plans. For example, accurate forecasting models can guide platforms in offering dynamic subscription options, such as allowing users to temporarily suspend paid subscriptions during off-peak months like June through August, when platform activity may decrease due to vacations. This would not only improve customer satisfaction but also optimize the platform's revenue management. Such flexibility can contribute to better resource allocation and a more efficient user experience, reflecting the practical importance of forecasting in the platform economy.

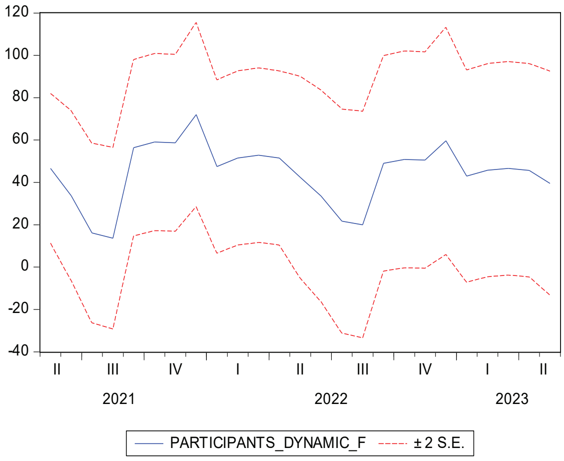
Appendix 1

Correlogram of residuals for SARMA (1,0) $(1,0)_{12}$ model

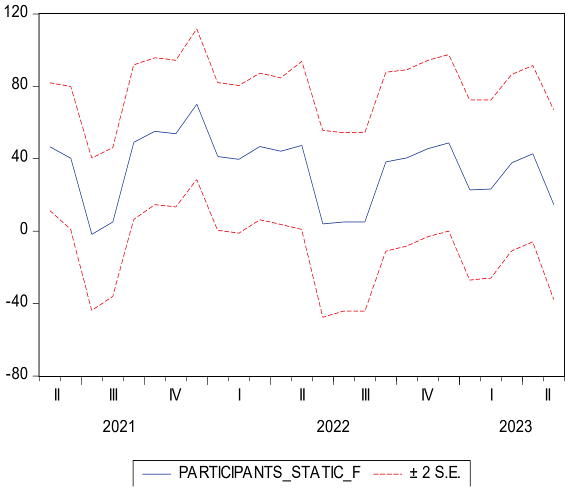
Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob
. | . |	. | . |	1	-0.021	-0.021	0.0182	
. | . |	. | . |	2	-0.022	-0.022	0.0380	
. |*. |	. |*. |	3	0.202	0.201	1.8027	0.179
. | . |	. | . |	4	-0.031	-0.024	1.8447	0.398
. |*. |	. |*. |	5	0.094	0.106	2.2546	0.521
. | . |	. | . |	6	-0.011	-0.053	2.2605	0.688
. |*. |	. |*. |	7	0.090	0.113	2.6613	0.752
.*| . |	.*| . |	8	-0.129	-0.185	3.5025	0.744
**| . |	**| . |	9	-0.235	-0.225	6.3962	0.494
. | . |	.*| . |	10	-0.050	-0.142	6.5319	0.588
. | . |	. |*. |	11	0.063	0.138	6.7579	0.662
. | . |	. | . |	12	-0.009	0.069	6.7628	0.748
. | . |	. | . |	13	-0.062	0.016	7.0002	0.799
. | . |	. | . |	14	-0.041	-0.065	7.1056	0.851
. | . |	. | . |	15	-0.023	0.014	7.1415	0.895
.*| . |	.*| . |	16	-0.079	-0.085	7.5761	0.910



Series: Residuals	
Sample 2020M04 2023M05	
Observations 38	
Mean	-0.905280
Median	-2.445840
Maximum	36.25251
Minimum	-42.76558
Std. Dev.	16.30154
Skewness	-0.169743
Kurtosis	3.827429
Jarque-Bera	1.266491
Probability	0.530866



Forecast: PARTICIPANTS_DYNAMIC_F	
Actual: MEETINGS	
Forecast sample: 2020M04 2023M05	
Adjusted sample: 2021M05 2023M05	
Included observations: 25	
Root Mean Squared Error	19.60209
Mean Absolute Error	15.81796
Mean Abs. Percent Error	16.94235
Theil Inequality Coefficient	0.235684
Bias Proportion	0.557119
Variance Proportion	0.143681
Covariance Proportion	0.299200



Forecast: PARTICIPANTS_STATIC_F	
Actual: MEETINGS	
Forecast sample: 2020M04 2023M05	
Adjusted sample: 2021M05 2023M05	
Included observations: 25	
Root Mean Squared Error	15.24557
Mean Absolute Error	9.900077
Mean Abs. Percent Error	12.12304
Theil Inequality Coefficient	0.200310
Bias Proportion	0.105687
Variance Proportion	0.030659
Covariance Proportion	0.863654

References

1. Chen, D. (2019). *The rise of the platform economy*. https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/humancapital/deloitte-nl-hc-the-rise-of-the-platform-economy-report.pdf
2. Codagnone, C. (2022). *The platform economy after COVID-19: Regulation and the precautionary principle*. https://library.oapen.org/viewer/web/viewer.html?file=/bitstream/handle/20.500.12657/51945/978-3-030-86144-5.pdf?sequence=1&isAllowed=y
3. EU Observatory. (2023). *Research – Observatory on the online platform economy*. http://platformobservatory.eu
4. Whyman, N. (2023). *What is the difference between a conference and a meeting?*https://www.kdmevents.co.uk/help/difference-between-conference-and-meeting/
5. Filip, F. G. (2022). Collaborative decision-making: Concepts and supporting information and communication technology tools and systems. *International Journal of Computers, Communications & Control*, 17(4), 4732. https://univagora.ro/jour/index.php/ijccc/article/view/4732
6. Suduc, A. M., Bizoi, M., & Filip, F. G. (2009). Exploring multimedia web conferencing. *Informatica Economica*, 13(3), 5–17.
7. Vallarasi, A. (2022). *Effective virtual platforms for online meeting*.
8. Boyarsky, K. (2020). *The 10 best free video conferencing platforms*. https://resources.owllabs.com/blog/video-conferencing-tools
9. Software Testing Help. (2023). *15 best online/virtual meeting platforms software in 2022*. https://www.softwaretestinghelp.com/best-virtual-meeting-platforms/
10. IEEE Computing Society. (2023). *Making the decision to host a virtual event*. https://www.computer.org/conferences/organize-a-conference/organizer-resources/hosting-a-virtual-event/cs-virtual-event-resource-guide
11. Suduc, A. M., & Bizoi, M. (2022). AI shapes the future of web conferencing platforms. *Procedia Computer Science*, 192, 2566–2573.
12. Walsh, M. (2023). *The 7 best video conferencing software platforms for 2023*. https://www.dgicommunications.com/video-conferencing-software/
13. Zoho. (2023). *Compare Zoho meetings editions*. https://www.zoho.com/meeting/plan-comparison.html
14. Zoom. (2023). *Plans & pricing for business. Optional Zoom plan add-ons*. https://zoom.us/pricing
15. Cisco. (2023). *Webex meetings*. https://pricing.webex.com/ro/en/
16. Microsoft. (2023). *Find the right Microsoft Teams for your needs*. https://www.microsoft.com/en-us/microsoft-teams/compare-microsoft-teams-options?activetab=pivot%3aprimaryr1
17. Google. (2023). *Video calls and meetings for everyone*. https://apps.google.com/meet/
18. Google Workspace. (2023). *Choose your Google Workspace pricing plan. Try it free for 14 days*. https://workspace.google.com/intl/ro/pricing.html
19. Indeed. (2023). *How to manage effective meetings*. https://www.indeed.com/hire/c/info/effective-work-meetings
20. Richard, D. (2021). *Management* (14th ed.). Cengage Learning Publishing House.
21. Casamo & Associates. (2023). *Planning a time strategy for an effective teleconference*. https://www.casamo.com/planning-a-time-strategy-for-an-effective-teleconference/
22. Indeed. (2022). *What is teleconferencing? 10 steps to a successful meeting*. https://ca.indeed.com/career-advice/career-development/what-is-teleconferencing

23. Stack, L. (2023). *Productive teleconferencing: Planning, process, and protocol*. https://www.theproductivitypro.com/FeaturedArticles/article00070.htm
24. Condeco. (2016). *How to plan a successful video conference in 5 simple steps*. https://www.condecosoftware.com/blog/how-to-plan-a-successful-video-conference-in-five-simple-steps/
25. FreeConferenceCall. (2022). *Conference call countdown: 5 things to consider before your next teleconference*. https://www.freeconferencecall.com/resource-center/conference-call-countdown-5-things-to-consider-before-your-next-teleconference
26. Free Management Books. (2023). *Advance planning a conference call*. http://www.free-management-ebooks.com/faqmt/confcall-07.htm
27. InTheBlack. (2020). *7 ways to run a teleconference – and 6 ways not to*. https://intheblack.cpaaustralia.com.au/careers-and-workplace/7-ways-to-run-a-teleconference
28. Martin, J. (2023). *What is the future of teleconferencing and how to plan for it now*. https://www.coxblue.com/what-is-the-future-of-teleconferencing-and-how-to-plan-for-it-now/
29. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
30. Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50, 159–175.
31. Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint* arXiv:1506.00019.
32. Lim, B., & Zohren, S. (2021). Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194), 20200209.
33. Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6, 107–116.
34. Rehmer, A., & Kroll, A. (2020). On the vanishing and exploding gradient problem in gated recurrent units. *IFAC-PapersOnLine*, 53(2), 1243–1248.
35. Chen, Y., Li, L., Li, W., Guo, Q., Du, Z., & Xu, Z. (2022). Deep learning. In *AI computing systems* (pp. 153-180). Morgan Kaufmann.
36. Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 3285–3292). IEEE.
37. Murat, O. H., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing*, 93, 106384.
38. Rundo, F., Trenta, F., Luigi di Stallo, A., & Battiato, S. (2019). Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24), 5574.
39. Jaydip, S., & Mehtab, S. (2022). Long-and short-term memory (LSTM) networks architectures and applications in stock price prediction. In *Emerging computing paradigms: Principles, advances and applications* (pp. 143–160).
40. Sreelekshmy, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1643–1647). IEEE.
41. Zahra, K., & Suykens, J. A. K. (2020). Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks*, 132, 1–9.
42. Miao, K. C., Han, T. T., Yao, Y. Q., Lu, H., Chen, P., Wang, B., & Zhang, J. (2020). Application of LSTM for short term fog forecasting based on meteorological elements
43. .Yong, R., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, 31(7), 1235-1270.

44. Oruh, J., Viriri, S., & Adegun, A. (2022). Long short-term memory recurrent neural network for automatic speech recognition. *IEEE Access*, 10, 30069-30079.
45. Carbune, V., Gonnet, P., Deselaers, T., Rowley, H., Daryin, A., Calvo, M., Wang, L. L., Keysers, D., Feuz, S., & Gervais, P. (2020). Fast multi-language LSTM-based online handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(2), 89-102.
46. Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems* (pp. 3104-3112). Red Hook, NY: Curran Associates.
47. Chandra, E., Goyal, S., & Gupta, R. (2021). Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access*, 9, 83105-83123.
48. . Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 807-814). Haifa, Israel.
49. . Du, S., Li, T., Yang, Y., Gong, X., & Horng, S. J. (2019). An LSTM based encoder-decoder model for multi-step traffic flow prediction. In *International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary (pp. 1-8).
50. Pesole, A. (2021). Understanding the prevalence and nature of platform work: The measurement case in the COLLEEM survey study. In *Platform Economy Puzzles* (pp. 19-45). Edward Elgar Publishing.
51. Baselgia, E., & Martínez, I. Z. (2023). Behavioral responses to special tax regimes for the super-rich: Insights from Swiss rich lists. *EU Tax Observatory Working Paper*, 12.
52. Venerandi, A., Feliciotti, A., Fleischmann, M., Kourtit, K., & Porta, S. (2023). Urban form character and Airbnb in Amsterdam (NL): A morphometric approach. *Environment and Planning B: Urban Analytics and City Science*, 50(2), 386-400.
53. Eurostat. (2024). Experimental statistics on digital platform employment. Available online at https://ec.europa.eu/eurostat/web/products-eurostat-news/w/ddn-20240718-1
54. Lafuente, E., Ács, Z. J., & Szerb, L. (2024). Analysis of the digital platform economy around the world: A network DEA model for identifying policy priorities. *Journal of Small Business Management*, 62(2), 847-891.