Journal of Business Research 144 (2022) 844–853

Contents lists available at ScienceDirect

Journal of Business Research

journal homepage: www.elsevier.com/locate/jbusres

Multi-target CNN-LSTM regressor for predicting urban distribution of short-term food delivery demand

Alessandro Crivellari [a,b,*], Euro Beinat [a,c], Sandor Caetano [d], Arnaud Seydoux [d], Thiago Cardoso [d]

[a] *Department of Geoinformatics—Z_GIS, University of Salzburg, 5020 Salzburg, Austria*
[b] *Department of Computer Science and Engineering, Southern University of Science and Technology, 518055 Shenzhen, China*
[c] *Prosus Group, 1082 MD Amsterdam, the Netherlands*
[d] *iFood, 06020-012 Osasco (Sao Paulo), Brazil*

ARTICLE INFO

Keywords:
Food delivery
Customer behavior
Deep learning
Recurrent neural networks
Urban modeling

ABSTRACT

The food delivery market has increased rapidly in the last few years, becoming a well-established reality in the business world and a common feature of urban life. Food delivery platforms provide the end-to-end services that connect restaurants with consumers, including the delivery service to those people ordering food through an online portal. A key component of these platforms is logistics, specifically the logistics of drivers. Ideally, the number of drivers operating in an urban area should be just the right number to serve the demand in that area. Since the demand is extremely dynamic in space and time, the spatial–temporal distribution of drivers remains a challenging problem, partially solved by means of variable incentives in different city areas at different times. In this context, a precise demand prediction would avoid a local lack of drivers in some areas, and an inefficient concentration of drivers in some other areas. For this reason, we propose a deep neural network-based methodology to forecast short-term food delivery demand distribution over urban areas. The study, carried out on a real-world dataset from a food delivery company, focuses on hourly demands and frequent prediction updates. The sequential modeling approach, designed to catch rapid changes and sudden variations beyond the general demand trend, is based on a multi-target CNN-LSTM regressor trained on location-specific time series. The methodology uses a single model for all service areas simultaneously, and a single one-step volume inference for every area at each time update. The results disclose a better performance over baselines (historical estimates for the same time-area) and more traditional statistical approaches (moving averages and univariate time-series forecasting), demonstrating a promising implementation potential within an online delivery platform framework.

1. Introduction

Propelled by developments in information communication technologies, online platforms have become ordinary entities in people's everyday life, by allowing for an instant match between demand and supply (Kenney and Zysman, 2016; Wood et al., 2019; Howcroft and Bergvall-Kåreborn, 2019). Online food delivery services have gained a central role, increasing the related market dramatically in the last few years and becoming a well-established reality in the business world. Characterized as platforms responsible for ordering, paying and monitoring the delivery process (Pigatto et al., 2017), they especially encountered the need for service intermediaries of small and medium restaurants (Yeo et al., 2017). By expanding choice and convenience, the online food delivery market segment has been attracting remarkable investments across the Americas, Asia, Europe, and Middle East; its global revenue amounts to 107.4 billion US$ in 2019 (Statista. "eServices Report, 2020).

Food delivery applications allow customers, by means of a smartphone, to order food items from a wide range of restaurants and have them shortly delivered at the doorstep. The increasing popularity lies in a mutual benefit for both consumers and food service providers. Ease, speed and precision in the ordering and delivery process draws the attention of customers (Cho et al., 2019; Doan Ngoc, 2013; Roh and Park, 2019); an increased revenue, labor expenses reductions and the

* Corresponding author.
E-mail addresses: alessandro.crivellari@sbg.ac.at, crivellari@sustech.edu.cn (A. Crivellari), euro.beinat@sbg.ac.at (E. Beinat), sandor.caetano@ifood.com.br (S. Caetano), arnaud.seydoux@ifood.com.br (A. Seydoux), thiago.cardoso@ifood.com.br (T. Cardoso).

https://doi.org/10.1016/j.jbusres.2022.02.039
Received 11 December 2020; Received in revised form 7 February 2022; Accepted 12 February 2022
Available online 20 February 2022

facilitation of supply activities attracts the providers (See-Kwong et al., 2017). Nowadays, online food delivery is deeply integrated in the urban life, providing a specific delineation to food supply systems and inevitably influencing people's food-related habits.

A key component to the success of food delivery systems is represented by short and predictable waiting times, inserted in a policy of strict time optimization. In platform-to-consumer delivery companies, this aspect particularly relates to the logistics of drivers, primary factor in densely populated regions. Compared to physical shopping, online purchases are generally more concerned about time (Hsiao, 2009). If delivery takes too long, customers' satisfaction will decrease, potentially leading to a loss of clients and sales volume; as part of the service quality, delivery time markedly affects consumers' decision-making (Xiaomin and Yi, 2017; Zhang et al., 2019).

In this context of food delivery optimization, a company's main challenge is the logistics and distribution of drivers across the city. Ideally, the number of drivers located in each urban area should correspond to the local demand, whereby more drivers are needed in the areas that are expected to receive a higher number of orders. Due to the extremely dynamic profile of urban demand in space and time, the spatial–temporal distribution of drivers is guided, in practice, by means of variable incentives in different city areas at different times. A correct demand prediction would avoid a local lack of drivers (longer delivery time and decrease of customers' satisfaction) in some areas and an inefficient concentration of drivers in some other areas. Our research direction is to use predictive analytic tools to analyze the food demand history of a specific delivery platform for predicting the future distribution of its expected demand volume.

To support the planning and logistics of deliveries within the platform, we propose a deep neural network-based methodology to forecast short-term food delivery demand distribution over space. We focus on predicting the future number of orders in each area of the city as a basis for supporting logistics decisions, such as driver logistics. To take into account realistic application scenarios, we target hourly demands and frequent prediction updates. In particular, since demand volume is subjected to very rapid changes in time and space, the challenge lies in grasping abrupt variations beyond the general trend. We therefore present a spatially-oriented short-term forecasting methodology, leveraging sequential modeling and deep learning techniques.

The underlying criteria is based on mining temporal patterns of order volumes, assuming that the current demand exhibits some dependences on past volume quantities. The characteristics of the series are hypothesized to carry essential information for anticipating future demand conditions. While typical time series analyses focus on learning the general trend via purely statistical methods (Aslanargun et al., 2007; Junior et al., 2014; Chmieliauskas and Guršnys, 2019; Samal et al., 2019), the natural characterization of food demand over time is defined by quick volume variations that needs to be properly detected in order to provide a satisfactory service. Those approaches, indeed, tend to lead to ineffective predictions when in presence of intense oscillations and shifts to boundary conditions. Moreover, since multiple city areas are taken into account, multiple predictions are to be performed, whereby each location is associated to different demand volumes and different sequential patterns over time. Our goal is to collectively generate predictions based on a single model (and a single training process) comprising the totality of urban areas, therefore avoiding singularly fitting each area with its corresponding distinctive model. In practice, this way provides a much more efficient strategy, since the model is deployed as a whole, and not in the form of hundreds of unique models. Furthermore, it is intended to leverage a combined global view of the city trend, not only analyzing patterns of single areas, but also detecting inter-location relations in the demand variation.

To meet these technical and business requirements, our method relies on a multi-target CNN-LSTM regressor trained on location-specific time series. The model is conceived to output the urban distribution of short-term food delivery demand by leveraging a single training process for all urban areas and a single one-step volume inference for every area at each time update. Starting from the original set of food delivery orders received and recorded through a commercial online platform (including time stamp of the order and location of the restaurant), data are first aggregated in space and time, building sequences of numbers of orders whereby each area is represented by a series of values unfolding in a fixed time step. The sequences are then stacked together as a combined input to the recurrent network-based regressor, which is jointly trained on the block of time series to learn the underlying patterns of urban food delivery demand. A multi-target dense layer finally provides a number of output values equal to the number of city areas.

The suggested approach is purely data-driven, capturing variability patterns directly from demand volume sequences, without requiring any manual feature extraction. Each individual location's prediction is therefore based on the collective analysis of urban demand over several geographic areas. Once tested on a real-world dataset, our methodology reveals a prominent feasibility in the context of short-term distributed food delivery demand forecasting, disclosing better performances over baselines and traditional approaches.

2. Methodology

We present a forecasting tool that collectively learns sequential patterns of spatial–temporal demand variation for predicting the number of food orders received in each urban sub-territory at the next time step in the future. This section presents the adopted methodological path, from defining collective time series to the use of deep neural networks in the form of a sequential multi-target regressor.

2.1. Collective time series definition

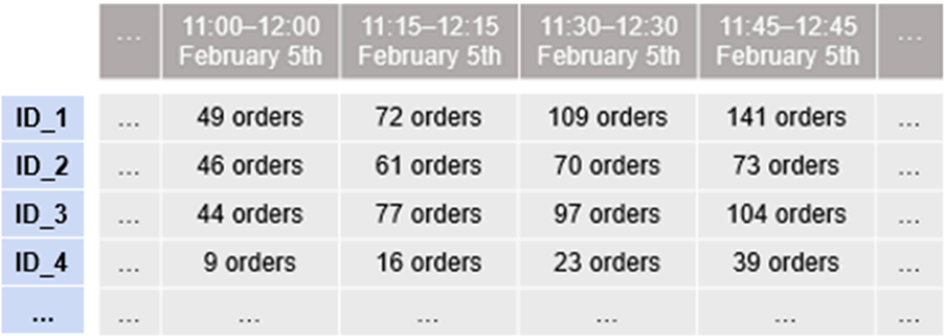
The approach is based on the primary consideration of geographic partitioning, assuming a city divided in multiple local areas. Following a multi-sequential modeling perspective, each individual urban sub-territory is associated to a time series describing the evolution of its food delivery history. Its events are organized into a sequence of order counts falling in a time window and following a time step for updating. Therefore, the time series of a city area a arises as a sequence of chronologically ordered count values, defined according to the fixed time window Δt, and unfolding into the predefined update step t, namely $S_a = \{\#orders([\tau, \tau + \Delta t])_a | \tau = t, 2t, 3t, \cdots\}$.

The global order sequence of the entire city is obtained by stacking together the sequences of all areas, synchronized at the same update step and time window. The input to the prediction model is a multidimensional time series, whose dimensionality refers to the number of city areas. More precisely, given a territory division into N areas, the portion of the sequence identifying a particular update time $\tau + \Delta t$ is made of a vector of N values reporting the number of orders within the time span $[\tau, \tau + \Delta t]$ in each area a, namely $S(\tau + \Delta t) = [\#orders([\tau, \tau + \Delta t])_a | a = ID\_1, ID\_2, \cdots, ID\_N]$.

As illustrated in Fig. 1, the same longitudinal position along the stacked sequence identifies demand volumes referring to the same time span. The values of the two time unit variables Δt and t are arbitrary, and should be set according to the data source and the forecasting problem. The final data configuration consists of a block of longitudinally-stacked time series, unfolding in identical time steps, distinctively reporting the consecutive numbers of received delivery orders in each of the areas within the city territory.

2.2. Multi-target deep learning model for distributed food delivery demand forecasting

The proposed deep neural network model is designed for simultaneously processing the distributed food delivery demand volume over the city. Its structure consists of three building blocks: a multi-dimensional input layer, a recurrent block of CNN and LSTM layers,



	...	11:00–12:00 February 5th	11:15–12:15 February 5th	11:30–12:30 February 5th	11:45–12:45 February 5th	...
ID_1	...	49 orders	72 orders	109 orders	141 orders	...
ID_2	...	46 orders	61 orders	70 orders	73 orders	...
ID_3	...	44 orders	77 orders	97 orders	104 orders	...
ID_4	...	9 orders	16 orders	23 orders	39 orders	...
...	...	...	...	...	...	...

Fig. 1. Overall city representation as a block of longitudinally-stacked sequences of food delivery demand volume (defined with a time window of 1 h and an update step of 15 min), each of them referring to a specific urban area.

and a multi-target output layer. A visual representation of the modeling conceptual design is shown in Fig. 2. Each component is individually treated in the following subsections.

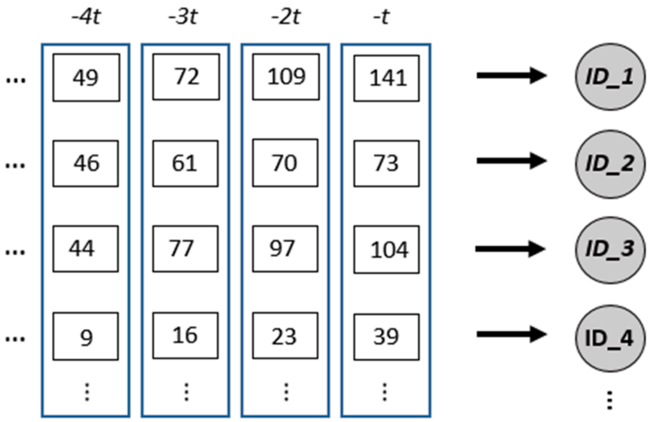


Fig. 3. Input layer representation: vectors across consecutive time steps associate a same area to the same position along the array, always heading towards the same neuron.

2.2.1. *Input layer*

The underlying idea is to collectively include multiple urban areas into a single predictive model that is able to process their corresponding time series without any manual prearranged data blending. We therefore utilize a multi-dimensional input layer, each of whose neurons is selectively targeted to a specific reference area, in such a way the model receives multiple series but through distinctive input channels.

Specifically, suppose that a city territory is divided into N urban areas, whereby each area is described by its characteristic time series. The effective input consists of a number of N time series, which, stacked together, identify a sequence of N dimensions. At each time step, the model receives a vector of N values; vectors across consecutive time steps associate the same area to the same position along the array, always heading towards the same neuron of the input layer, as represented in Fig. 3. Food delivery demands of different urban areas are therefore analyzed simultaneously but acquired through separate entries, hence combining two processing perspectives: the sequential evolution of urban demand over time, and its geographic distribution across multiple areas of the city. The training process is consequently driven by successive chronologically-ordered N-dimensional vectors, leading the model to learn global and local sequential patterns.

2.2.2. *Recurrent block*

The recurrent block is responsible for detecting sequential patterns in the temporal variation of urban demand distribution. Its internal architecture is based on a combination of convolutional and LSTM layers, aiming to effectively mine characteristic behaviors along the time series. The idea is to use a CNN layer to reshape the raw input data into a more convenient representation format, attenuating the noise in the multi-dimensional sequence, and to employ LSTM layers to efficiently capture sequential pattern information. In other words, we leverage the capability of convolutional layers of extracting implicit meaningful series' characteristics and the effectiveness of LSTM layers for exploring short-term and long-term dependencies. The advantages of this combination on handling sequential data have led to recent increasing adoptions in a variety of applications related to time series analysis (Livieris et al., 2020; Livieris et al., 2020; Pintelas et al., 2020). We hereby briefly describe the two layer types constituting the proposed recurrent block.

CNN layer. Originally intended for automatically extracting features from images (Rawat and Wang, 2017; Krizhevsky et al., 2012), traditional convolutional layers work by applying several small sliding kernels across a 2D matrix, producing multiple 2D feature maps. By applying different convolution kernels on each subregion of the input image, multiple convolved features are generated, enhancing more meaningful characteristics than the initial input data. The same concept can be translated in a sequential processing domain through 1D

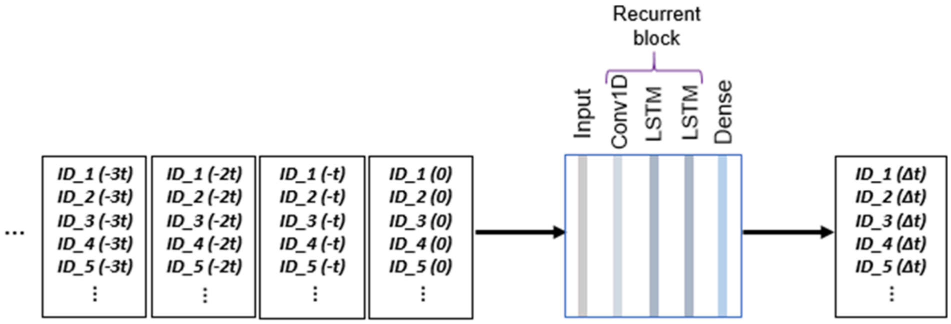


Fig. 2. High-level overview of the modeling conceptual structure.

convolutions. A 1D convolutional layer slides kernels across a sequence, providing a 1D feature map per kernel, where each map represents a very short learned sequential pattern. The number of kernels defines the layer output dimensionality: K kernels produce K 1-dimensional sequences or, equivalently, a K-dimensional sequence. Convolving with a stride greater than 1 allows shortening the original length, helping the subsequent LSTM layers detect longer patterns; and since the kernel size is typically chosen equal or larger than the stride, all input elements are used to compute the CNN output, leading the layer learning process to preserve the useful information by only dropping unimportant details. The use of a convolution process preceding recurrent architectures is particularly beneficial in cases of high oscillations and long dependencies along the series. The output of a given neuron located in the position j in the feature map k of a 1D CNN layer is summarized in Equation (1), whereby L is the length of the kernel, s is the stride, $j^{'}$ is the index identifying an element of the input sequence x, N is the input dimensionality (i.e., the number of urban areas), w and b are the corresponding internal weights and bias connections.

$$z_{j,k} = \sum_{v=0}^{L-1} \sum_{k^{'}=0}^{N-1} x_{j^{'},k^{'}} . w_{v,k^{'},k} + b_k \text{with} j^{'} = j \times s + v \tag{1}$$

LSTM layer. LSTM (Hochreiter and Schmidhuber, 1997) belongs to the family of recurrent networks, a specific type of artificial neural network specialized in the processing of sequential data. Analyzing sequences one element at a time, it repeatedly feeds itself with the output it produced at the previous step, along with the new element in the sequence. Its unit structure consists of a state vector and four distinctive neural networks, which are responsible for the vector updates. The task-related information is indeed encoded in the state vector, and can be selectively deleted or increased at each training step. Specifically, as reported in the Equations (2)-(7): a forget gate f defines which past information to erase from the state vector; an input gate i determines which state values to update; a tanh network provides a vector C of new values to store; the updated state vector C_t is therefore obtained after the action of f, i and C on C_{t-1}; and finally, an output gate o is inserted to selectively control the LSTM outcome h, which derives from the multiplication of o with the tanh of the updated state C_t. The list of operations refers to a certain time step t, with x_t denoting the corresponding input element, and the various W and b indicating the internal weights and biases of each distinctive neural network. In the last time step preceding prediction, the LSTM output vector carries the overall compressed characterization of the original sequence, then used for generating the explicit forecasting. When multiple LSTM layers are stacked together, the following layer is fed with the output of the previous layer at the same time step, and the final output characterization refers to the vector at the last step of the last layer.

$$f_t = \sigma\left(W_f \bullet [h_{t-1}, x_t] + b_f\right) \tag{2}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{3}$$

$$C_t = \tanh(W_C \bullet [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * C_t \tag{5}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

In the context of our analysis, the sequential input element to the recurrent block is represented, at each time step, by the current demand volume distribution in the form of a vector representation obtained by slicing the multi-dimensional sequence on the temporal axis. As a result of the network architecture, the information on singular urban areas, initially targeting different entry channels, is then collectively processed within the CNN-LSTM layers, ending up in mixed evolving vector characterizations. The city areas' individual peculiarities are therefore blended together during the recurrent learning procedure and spit out in a shared encoded representation as a single output vector. The totality of the CNN and LSTM internal parameters W and b are repeatedly updated during the training phase, optimizing the final characterization. Fig. 4 visually reports an exemplifying conceptual framework of the whole recurrent block.

2.2.3. Output layer

The output layer is the model structure addressing the translation of the LSTM final vector representation into the explicit food delivery demand distribution across the urban territory. The layer outcome involves multiple output values, one for each city area, according to the pre-defined territory division. Specifically, each output neuron emits the future estimated number of orders for its particular target area, representing a piece of the global combined outcome in the form of a multi-target regressive prediction.

The layer is structured as a multi-output fully-connected neural network on top of the recurrent block, behaving as a transition point from the compressed encoded information in a single LSTM output vector to the multiple simultaneous predictions constituting the geographically-distributed final outcome. The reshaping of the implicit vector representation into the explicit forecasted values is depicted in Fig. 5.

The formal description of the output layer is reported in Equation (8), where N refers to the total number of urban sub-divisions, W and b indicate the weights and biases of the network layer, and h_{last} identifies the final vector representation of the recurrent block.

$$\#orders_j = W_{(out)j} h_{last} + b_{(out)j} \forall j \in (0, N] \tag{8}$$

2.3. Model training

The data feeding process for the neural network model is performed by scanning the multi-dimensional sequence of demand distribution with a sliding window, identifying the training features and the target variable at each input step. The window, consecutively moving forward by one step until the end of the sequence, defines multiple input segments of a fixed length, whereby the segment length represents the extent of the continuous pattern mining activity for sequentially forecasting the future demand; its choice is a hyperparameter to tune, strongly dependent on the dataset characteristics and the time resolution of the sequence.

During the training phase, the deep learning model receives such collection of segments, organized as sequential input values together with their corresponding desired target variable, and aims to minimize the mean squared error between the predicted demand volumes and the real registered amounts. Through backpropagation and mini-batch stochastic training, the weights of every network layer are tweaked in the direction of the gradient, attempting to generate forecasting outcomes that gradually becomes closer and closer to the real targets.

In the testing phase, the prediction of the future demand distribution relies on the model's parameter configuration that was set up by learning historical patterns during training. The most likely demand volume estimation is therefore obtained through the evaluation of the recent measurements preceding the forecasting time stamp, according to the past automatically-learned sequential patterns of food delivery demand variations over space and time.

3. Experiment

This section presents the food delivery dataset and reports the experimental setup and the achieved results, conveying our findings on predicting demand distribution in a real-world setting. The evaluation touches upon multiple viewpoints, opening to comparisons with baseline approaches. The model implementation and training were carried

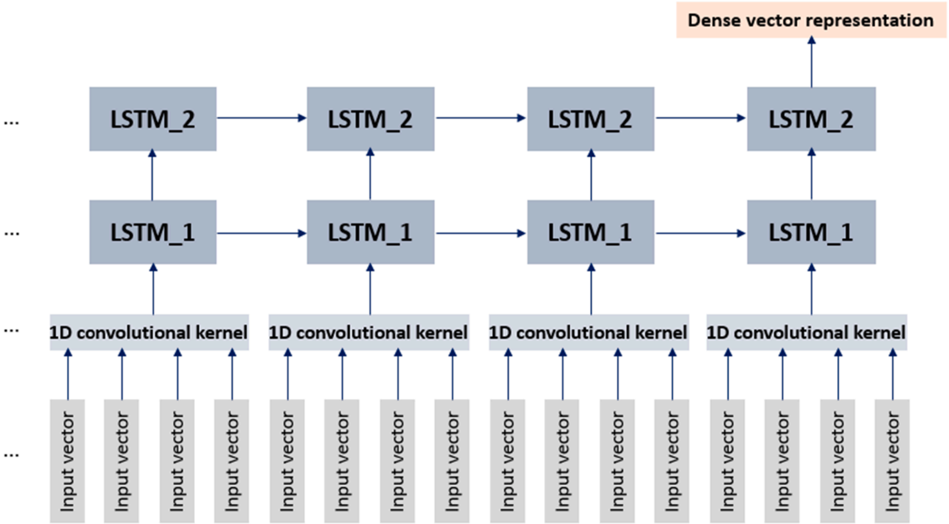


Fig. 4. Exemplifying conceptual framework of the recurrent block, represented as including two LSTM layers and one CNN layer (in the picture involving a kernel size of 4 and a stride of 4).

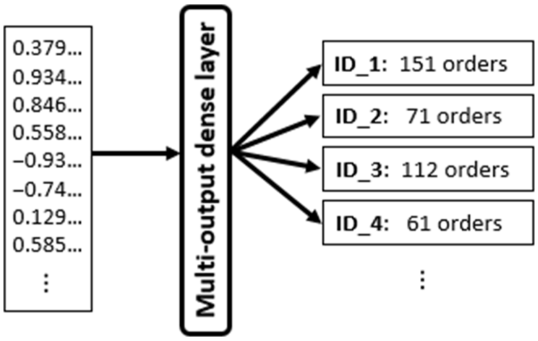


Fig. 5. Output layer processing role as translation point from the LSTM final vector representation to the explicit food delivery demand distribution across the city areas.

out on TensorFlow (Tensorflow, 2022).

3.1. Dataset

The study of food delivery demand trends was approached by analyzing a real-world data sample of delivery orders' information, associated to a major online food delivery company. Specifically, we leveraged five weeks of historical user-anonymized delivery request events, recorded in early 2020, in a high-density metropolitan area. The overall demand across the city was therefore embodied in a dataset including every food delivery order received and recorded through the company's online platform, whereby each delivery request comprises information on the time stamp of the order and the location of the restaurant or commercial venue to which the request was directed. This highlights the spatial–temporal characteristic of the data, identifying each observation with a date and time attribute (with a level of granularity even up to seconds) and the geographic coordinates locating the targeted restaurant.

The data pre-processing focused on transforming these single observations into a spatial–temporal map of aggregated demand distribution, following a discretization process in space and time. The sparse events need indeed to be grouped according to a certain spatial resolution, and the continuity of time discretized in fixed time steps. Motivated by a realistic application scenario, we formulated the problem as predicting the demand distribution in the next hour, with a prediction update every 15 min. Moreover, we opted for a space discretization based on the *H3* grid system (H3: Uber's Hexagonal Hierarchical Spatial Index), setting up an *H3* resolution equal to 7, which identified hexagonal grid cells having an edge length of 1.22 km. This resulted in 290 portions of urban territory covering the metropolitan area.

The final input to the model therefore consisted of a 290-dimensional sequence unfolding in 15-minute time steps, whose elements reported the aggregated number of orders received in the past hour in each of the grid cells. A single input time step was shaped as a vector of 290 values, representing the distribution of the current hourly demand volume over the whole city.

3.2. Experimental settings

The specification of the neural network model relied on a recurrent block made of one CNN layer and two LSTM layers. The CNN layer was characterized by 512 filters, a kernel size of 8, and a stride of 8; the two LSTM layers featured a hidden size of 512 neurons each. The sliding window identifying the input features was set to comprise the 24 h before prediction, therefore a total of 96 scaled values. The output was instead defined as a vector of 290 dimensions, representing the forecasted number of orders in the next hour for each of the reference areas. The training process leveraged a mean squared error loss function, minibatch stochastic training, and Adam optimizer (Kingma and Ba, 2014). The evaluation phase was required to focus on a data portion previously unseen during training; we therefore split the dataset in two parts, namely a training set of 4 consecutive weeks, and a testing portion comprising the successive week.

The definition of an overall measure of global performance was based on the typical metrics of mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE). A simple approach consisted of a plain average of the totality of prediction updates over the testing week. However, since the received food delivery demand was unevenly distributed across different areas, a weighted score based on its geographic distribution was considered to be primarily important. The underlying idea was to weight the prediction error on the basis of the general local demand, whereby the weighting factor was represented by the fraction of global weekly demand volume contained in each separate area. Those areas receiving a high number of requests were therefore supposed to determine a prominent influence on the overall score. Moreover, we further adjusted the metrics based on the hour of the day, weighting each single prediction on the basis of the expected target volume (weighting factor represented by the fraction of global weekly demand volume contained in each area at each updating prediction step), giving more importance to those forecasting updates that were intended to deal with larger amounts of orders.

Additional analyses also focused on single geographic areas, even combined into spatial representation perspectives that aimed to highlight the prediction error's geographic distribution and sparsity.

To provide a proper understanding of the quality of our model, the performance results were compared to baseline methods leveraging cyclical historical recurrence extractions and regressive statistical models, common approaches for time series forecasting tasks. Cyclical history-based predictions assume that the expected demand in the future is equal to the volume registered at the same hour of the same day in the past weeks. Statistical sequential processes, instead, fit the historical series to predict the next step of its trend; ARIMA and FBprophet are widely used methodologies in this sense.

The next subsection organizes the experimental findings, organically examining the outcomes resulting from the forecasting analysis of the expected future food delivery demand.

3.3. Results

The overall measures of MSE, RSME and MAE are reported in Table 1, disclosing a comparison of the CNN-LSTM model with the cyclical recurrence-based approaches leveraging different historical spans, namely 1 week, 2 weeks, and the whole month. The first baseline therefore assumes that the predicted number of orders is equal to the number of orders of the previous week, at the same day of the week and at the same time. The other two baselines are grounded in the same principle but, instead of only focusing on the previous week, they target the previous two weeks and the previous month respectively, averaging the corresponding daytime-specific values (e.g., the number of orders on the next Wednesday at 1 pm is predicted as the average of the orders in the past Wednesdays at 1 pm). By comparing the baselines, the best results are obtained in correspondence of a historical time span of 2 weeks. On the other hand, our model is shown to outperform the three approaches, registering a MSE of 115 versus a best baseline's outcome of 161.

The only observation of the standard mean scores can effectively be misleading because of the presence of a vast portion of areas receiving a very low number of orders, therefore influencing the global average score and pushing it towards low numbers. This is the reason why the MAEs are all very similar, being a plain average over a lot of zero and almost-zero values (some areas normally receives very few daily orders and most of merchants are closed overnight). We therefore additionally report the weighted mean scores based on the local demand volume each area receives, emphasizing the influence of very active regions. Table 2 shows indeed a marked tendency of increasing the performance difference between the model and the baselines.

Finally, Table 3 reports the scores further weighted on the demand of single prediction updates, highlighting the influence, within each area, of the time spans in which large amounts of orders are delivered.

Since different areas have different demand volumes and therefore different degrees of prediction error, we provide a glimpse of error distribution analysis on the multitude of grid cells. Specifically, Fig. 6 compares the distributed performance of the CNN-LSTM model with respect to the best baseline. Considering each bar in the plots as representing a specific reference grid cell area, the lower plot shows the weekly delivery demand of the top 40 areas by demand volume (as a percentage of the average demand), and the upper and the middle plots report the difference between the error (MSE and MAE respectively) of our model and the one of the "previous 2 weeks" baseline. Positive values indicate a better performance of the baseline; negative values are in favor of the CNN-LSTM. As observable in the graphs, the model performs better in the large majority of areas, and does not seem to be particularly influenced by different volumes of orders.

Table 1
Comparison results (in terms of MSE, RMSE, and MAE) of the CNN-LSTM model with respect to the cyclical recurrence-based baselines referring to the previous week, previous 2 weeks and previous month.

	CNN-LSTM	Previous week	Previous 2 weeks	Previous month
MSE	115.8	183.5	161.2	171.3
RMSE	10.7	13.5	12.7	13.1
MAE	4.1	4.6	4.2	4.2

Table 2
Comparison results (in terms of MSE, RMSE, and MAE, weighted on the area-specific weekly demand volume) of the CNN-LSTM model with respect to the cyclical recurrence-based baselines referring to the previous week, previous 2 weeks and previous month.

Weighted by area	CNN-LSTM	Previous week	Previous 2 weeks	Previous month
MSE	783.3	1284.1	1174.6	1346.7
RMSE	27.9	35.8	34.2	36.6
MAE	13.9	16.6	16.0	16.2

Table 3
Comparison results (in terms of MSE, RMSE, and MAE, weighted on the demand of area-specific single prediction updates) of the CNN-LSTM model with respect to the cyclical recurrence-based baselines referring to the previous week, previous 2 weeks and previous month.

Weighted by upd. step	CNN-LSTM	Previous week	Previous 2 weeks	Previous month
MSE	1441.3	2084.5	1919.7	2319.0
RMSE	37.9	45.6	43.8	48.1
MAE	22.6	25.8	24.9	25.4

For a further exploration, we tested the statistical models of FBprophet (Prophet Project) and ARIMA (Pmdarima Project) on selected reference areas. In particular, Fig. 7 reports the prediction errors of the top 10 areas by demand volume. The two models follow the general trend in the series but are not able to catch the rapid variations, ending up in poor performances in terms of overall errors, when compared to the CNN-LSTM model and the baselines. Moreover, they require a separate fit for each area, not handling multi-output procedures, leading to inefficient solutions.

Even considering area by area, the prediction error is also largely affected by different hours of the day, potentially concentrating only in specific time spans. For example, the reference sample area in Fig. 9 discloses a volume of orders per hour (averaged over the testing week) that reveals two clear peaks in correspondence to lunch and dinner time; the related MAE per hour shows that the model outperforms the baseline particularly in those lunch and dinner time spans, whereas the error values are overlapping during night time and early afternoon.

By targeting both temporal and geographic factors, an interesting analysis relies on visual maps representing the spatial–temporal distribution of prediction errors, highlighting when and where major mistakes occurred. A characteristic example is reported in Fig. 10, depicting the geographic distribution of the average prediction errors in the weekly time window 7 pm-9 pm, whereby a darker color indicates higher numbers of wrongly predicted orders. The figure clearly exhibits an error profile that is generally less spread in space for CNN-LSTM, compared to the baselines.

A proper quantification of the prediction error's geographical spreading is illustrated in the plots of Fig. 11, counting the number of areas with an error exceeding a certain threshold (i.e., 10, 20, and 30 wrongly predicted orders) with respect to each hour of the day. The comparison graph, again averaged over the testing week, shows a substantially better performance of the CNN-LSTM model. If we focus, for example, on the time span of 7 pm, our model involves about 70 areas with an error over 10 orders, whereas the best baseline elicits around 90

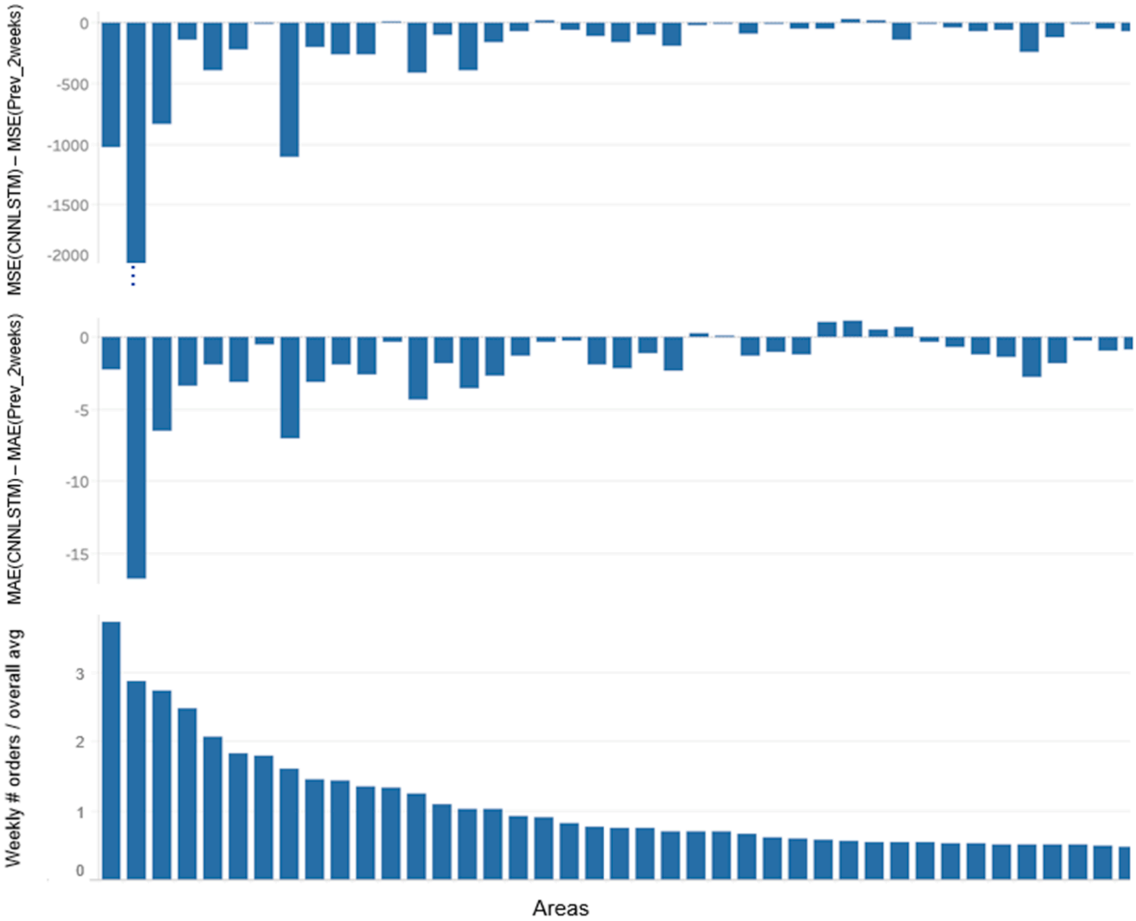


Fig. 6. The lower plot reports the weekly delivery demand of the top 40 areas by demand volume (as a percentage of the average demand); the upper and the middle plots displays the difference between the error (MSE and MAE respectively) of the CNN-LSTM model and the one of the "previous two weeks" baseline.

areas. Observing the number of areas with an error over 20 orders, we even obtain 15 versus 30 areas, therefore halving the geographic spreading of the error. Moreover, further extreme behaviors are also present, such as at 9am, where the count says 3 versus 16 areas with an error over 30 orders. These findings are particularly useful for analyzing spatial and temporal predictability; an error that is much less spread over the territory means that the wrong predictions are delimited in a much smaller region of the city.

4. Discussion and conclusion

We presented a deep learning methodology for predicting the distributed food delivery demand volume over space. The approach focused on a very short-term forecasting process and a distributed geographic profile, relying on sequential modeling. The underlying idea consisted of building location-specific sequences made of order counts unfolding in fixed updating steps and referring to the consecutive registered amounts of delivery volume within a selected time frame. The proposed method involved a recurrent neural network-based model, in the form of a multi-target regression problem. The model receives multiple input sequences (each related to a unique urban area) concatenated on the time axis. Each input step is a vector containing the values of each of the areas at the same time stamp. The output, on the other hand, is a vector estimating the future demand distribution over space, namely the expected number of orders in each of the reference urban areas. The network architecture is based on a CNN-LSTM framework, combining a first 1D-convolutional layer, subsequent LSTM layers, and a final multi-output fully-connected layer. We assessed the feasibility of the methodology on a real-world dataset of food delivery orders.

In the paper, we highlighted the advantages of our approach when it comes to its implementation into production. The proposed neural network handles the demand in each area collectively, relying on one single global training process, whereas most traditional statistical models require a different fit for each different area. The prediction consists of a single one-step inference of demand volume distribution at each time update, automatically releasing the full block of new location-specific output values at once. Moreover, the network does not require a refit for each prediction iteration, but needs only to follow a periodical retraining based on the evolving trends over time.

Our method properly detects rapid changes and variations beyond the general trend (crucial for a correct short-term demand prediction), in contrast to regressive statistical methods, which tend to perform poorly in this predictive regime. The model was also compared to cyclical historical baselines, built on the assumption that patterns are repeated following strict weekly, daily and hourly patterns. The experiment particularly highlighted a 2-week historical time span as the most reliable weekly-averaging window, among the baselines, for our specific case study. The CNN-LSTM model, however, demonstrated its capability of detecting sequential patterns and generally outperformed the baselines, providing a substantial improvement for correctly predicting food delivery demand distribution. This implied an implicit more sophisticated hidden pattern arrangement along the series, carrying information on future abrupt demand changes and variations, beyond a simple cyclical repetitiveness. The overall idea was therefore based on mining the historical evolution of demand trend over the territory divisions, as a possible meaningful hint for depicting the expected distribution in the future.

Furthermore, we assessed the results from a geographic perspective,

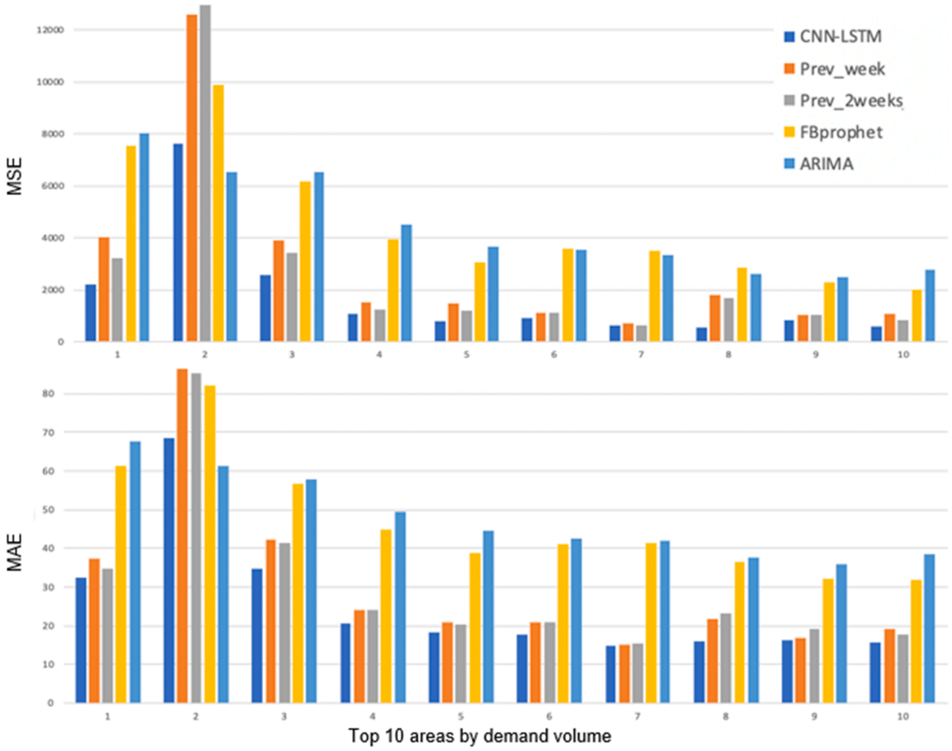


Fig. 7. Prediction errors of the top 10 areas by demand volume. Fig. 8 enriches the previous results in terms of percentage of predicted volume error with respect to the real volume. The main tendency ranges between 10 and 15 percent of prediction error.

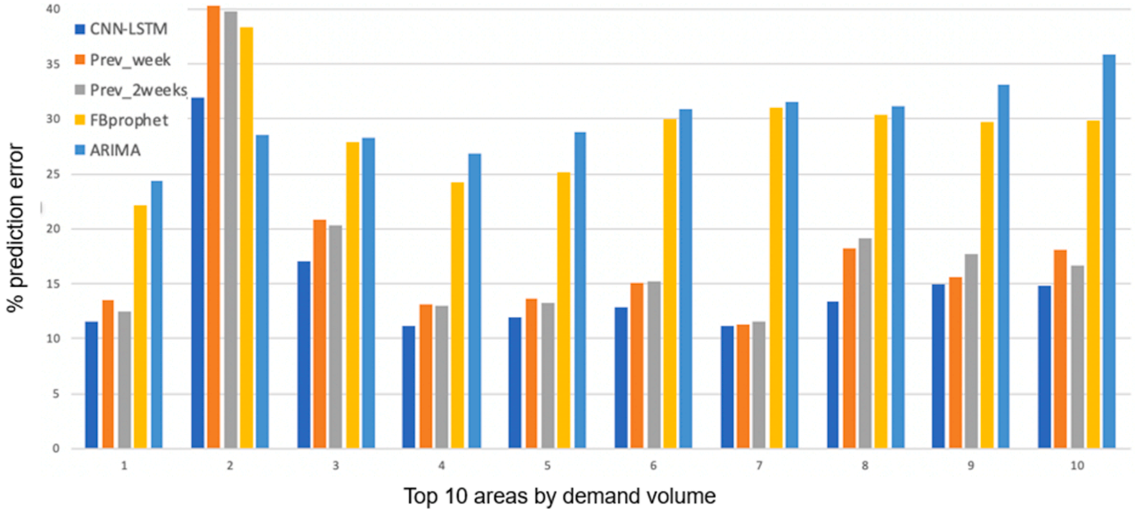


Fig. 8. Percentage of predicted volume error with respect to the real volume.

analyzing the spreading of prediction errors across the city. We observed that the CNN-LSTM model provided a substantially smaller error spreading over space, concentrating the errors in a smaller number of urban areas. This is especially favorable for predictability assessments, since it allows labeling only a small limited number of areas as “non-reliable”, instead of facing a widespread uncertainty in large urban regions. We can therefore conclude that the model offers a proper effectiveness in terms of both quantitative global performance and spatial distribution of prediction errors. It is worth mentioning, however, that each specific time span and geographic area can be potentially considered as a separate micro-environment that can be singularly studied in further details. Whereas delving into minute spatial and temporal peculiarities is beyond the scope of this work, it is important to keep in mind the inherent distinctiveness of different space–time intersections.

Our contribution demonstrates the feasibility of mining the rapid flow of order data to infer the future demand distribution. Specifically, in the context of short-term demand forecasting, where demand is subjected to abrupt changes and variations, the prediction needs to be very sensitive to small and sudden swings. Moreover, a real-world setup implies frequent prediction updates (e.g., every 15 min) for a multitude of urban areas, which calls for an efficient automatic solution to continuously provide forecasts. Our multi-target CNN-LSTM approach offers a possible answer to these challenges, standing out as a promising architecture to deal with spatially-distributed short-term predictions.

In wider terms, this contribution benefits the general online delivery domain, a relatively new form of consumption that is rapidly developing

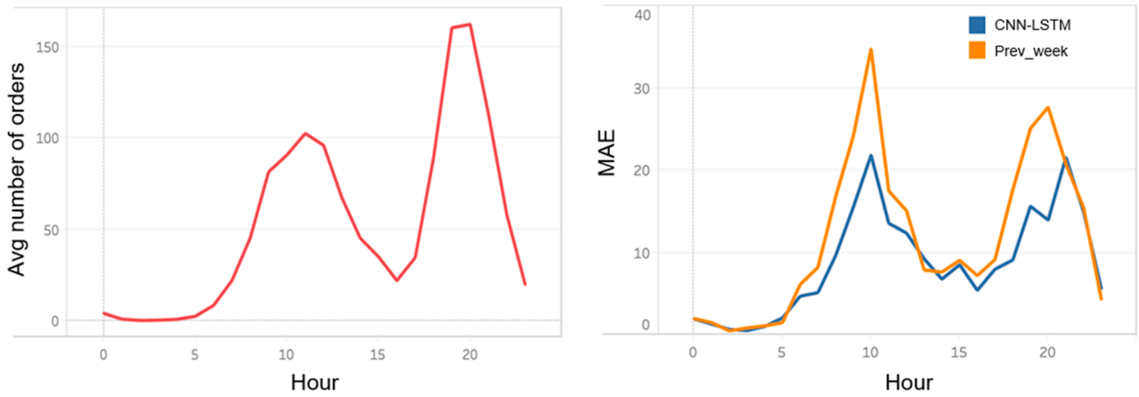


Fig. 9. Exemplifying description of a sample area in terms of order distribution over time and corresponding prediction error.

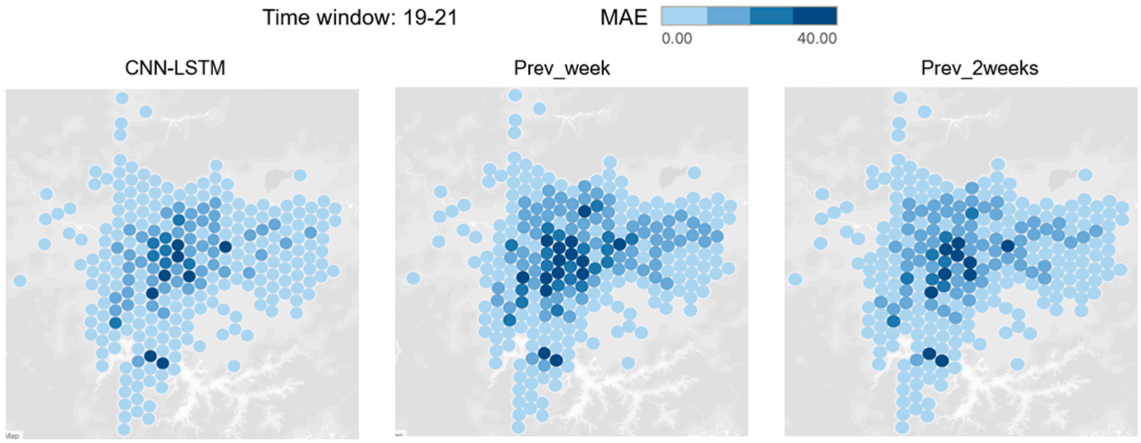


Fig. 10. Geographic distribution of the average prediction errors in the weekly time window 7 pm-9 pm.

across the world. In addition to the more established restaurant delivery, which is the focus of this paper, the market is expanding towards any type of on-demand delivery, from groceries to medicines. Traditional planning methods based on predefined schedules, pre-set delivery capacity and delivery policies are well understood and widely applied. They are however sub-optimal because of an insufficient ability to adapt to demand dynamically and in near real-time. Short-term prediction of demand is a key input to optimize the system, estimating, for instance, the delivery capacity needed to achieve a specific degree of customer satisfaction (e.g., the percentage of orders within a certain delivery time), as well the time and space allocation of this delivery capacity.

Possible future developments can be considered along three directions. The first one goes towards a deeper intuition of the overall prediction quality from a purely applied business domain point of view, even in terms of economic outcomes. Rather than quantifying statistical errors, a specific analysis on the practical consequences of underestimations and overestimations is the next step to take, assessing the prediction outputs from a business perspective. Therefore, a special focus would consist of exploring and isolating those cases when the predicted numbers of orders are less than the real ones (not enough drivers in the area, and consequently delays and unsatisfactory service), and the cases where the forecasted demand is higher than the actual one (wrong incentives leading to monetary loss). A second research direction should concentrate on adapting and evaluating the model for multiple prediction steps in the future, forecasting several hours ahead and assessing the maximum horizon that still allows to outperform the baselines. Finally, the last direction revolves around the design of further comparative models and approaches, potentially including additional input information (such as environmental and event-specific knowledge, besides the historical sequential demand trend) that can possibly contribute to a better refinement of the forecasting outcomes. More in general, the proposed methodology can be tested for different use cases, not limited to food delivery demand analysis, involving distributed spatial–temporal phenomena characterized by rapid variations and abrupt distinctive patterns over a grid-based territory.

In conclusion, the use of a multi-target recurrent neural network-based model arises as a promising methodology for geographically-distributed sequence forecasting, collectively processing multiple inputs and producing simultaneous multiple outputs. Its proposed effective introduction in the background of food delivery demand prediction contributes to highlighting the potential of deep learning for online delivery platforms, serving as a basis for business decisions and economic strategies aimed to an improved quality of customer-oriented services.

CRediT authorship contribution statement

Alessandro Crivellari: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Euro Beinat:** Conceptualization, Funding acquisition, Investigation, Project administration, Supervision, Writing – review & editing. **Sandor Caetano:** Writing – review & editing, Supervision, Resources, Investigation, Conceptualization. **Arnaud Seydoux:** Conceptualization, Investigation, Supervision, Writing – review & editing. **Thiago Cardoso:** Writing – review & editing, Supervision, Investigation.

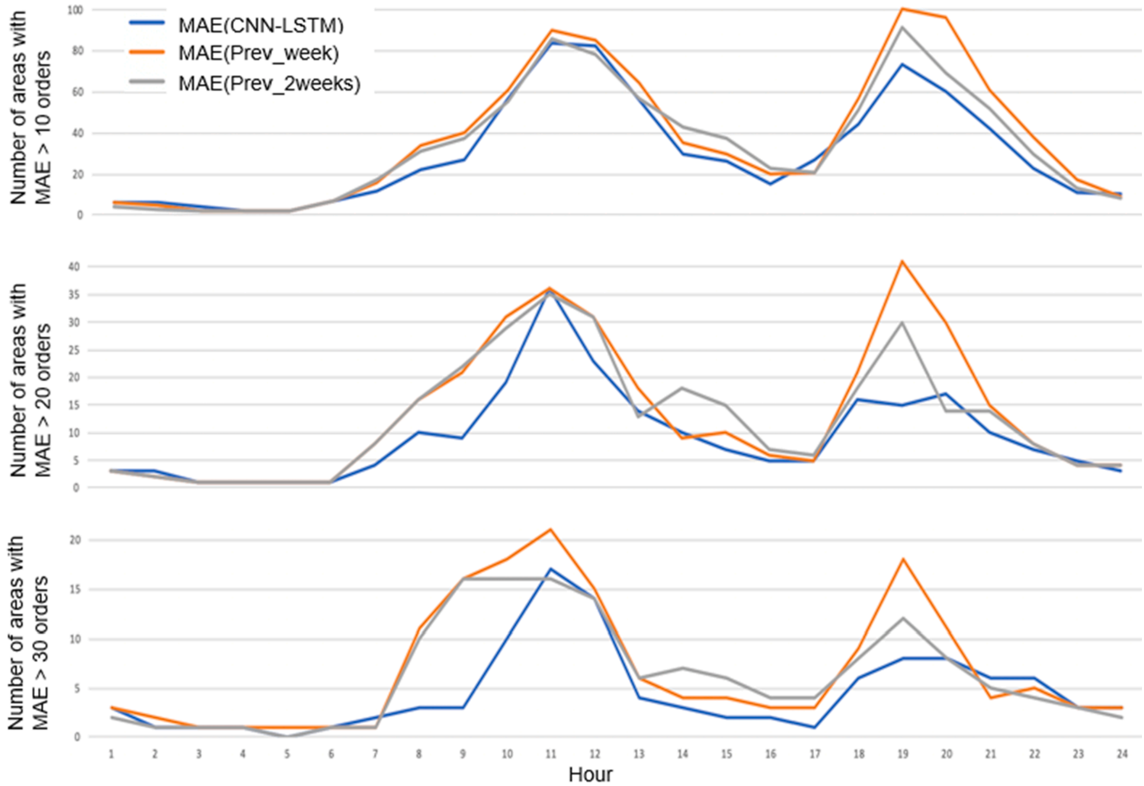


Fig. 11. Number of areas with an error exceeding 10, 20, and 30 wrongly predicted orders, with respect to each hour of the day.

Declaration of Competing 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.

Acknowledgements

This research was funded by the Austrian Science Fund (FWF) through the Doctoral College GIScience at the University of Salzburg (DK W 1237-N23).

The work was carried out as part of a research collaboration with Prosus and iFood. We would like to thank the team at iFood, in particular the Data Science team, for providing access to the data sample, for continuous support and advice, and for creating a collaborative and open work environment. We would also like to thank the colleagues at Prosus for their support and feedback.

References

Aslanargun, A., et al. (2007). Comparison of ARIMA, neural networks and hybrid models in time series: Tourist arrival forecasting. *Journal of Statistical Computation and Simulation, 77*(1), 29–53.

Chmieliauskas, Darius, Guršnys, Darius. (2019). LTE Cell Traffic Grow and Congestion Forecasting. *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE.

Cho, M., Bonn, M. A., & Li, J. J. (2019). Differences in perceptions about food delivery apps between single-person and multi-person households. *International Journal of Hospitality Management, 77*, 108–116.

Doan Ngoc, Ha. *Demand creation of online services for B2B and consumer market-Food delivery in Vietnam*. MS thesis. 2013.

H3: Uber's Hexagonal Hierarchical Spatial Index. https://eng.uber.com/h3/.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Howcroft, D., & Bergvall-Kåreborn, B. (2019). A typology of crowdwork platforms. *Work, Employment and Society, 33*(1), 21–38.

Hsiao, M.-H. (2009). Shopping mode choice: Physical store shopping versus e-shopping. *Transportation Research Part E: Logistics and Transportation Review, 45*(1), 86–95.

Junior, Paulo Rotela, Salomon, Fernando Luiz Riêra, de Oliveira Pamplona, Edson. (2014). ARIMA: An applied time series forecasting model for the Bovespa stock index. Applied Mathematics 5.21, 3383.

Kenney, M., & Zysman, J. (2016). The rise of the platform economy. *Issues in Science and Technology, 32*(3), 61.

Kingma, Diederik P., Jimmy Ba. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*.

Livieris, Ioannis E., et al. (2020). An advanced deep learning model for short-term forecasting US natural gas price and movement. IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham.

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications*, 1–10.

Pigatto, G., et al. (2017). Have you chosen your request? Analysis of online food delivery companies in Brazil. *British Food Journal*.

Pintelas, Emmanuel, et al. (2020). Investigating the problem of cryptocurrency price prediction: A deep learning approach. IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham.

Pmdarima Project. http://alkaline-ml.com/pmdarima/#.

Prophet Project. https://facebook.github.io/prophet/.

Rawat, Waseem, Wang, Zenghui. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural Computing.

Roh, M., & Park, K. (2019). Adoption of O2O food delivery services in South Korea: The moderating role of moral obligation in meal preparation. *International Journal of Information Management, 47*, 262–273.

Samal, K. Krishna Rani, et al. (2019). Time Series based Air Pollution Forecasting using SARIMA and Prophet Model. Proceedings of the 2019 International Conference on Information Technology and Computer Communications.

See-Kwong, G., et al. (2017). Outsourcing to online food delivery services: Perspective of F&B business owners. *The Journal of Internet Banking and Commerce, 22*(2), 1–18.

Statista. "eServices Report 2020 - Online Food Delivery." https://www.statista.com/study/40457/food-delivery/.

TensorFlow open source platform. https://www.tensorflow.org/.

Wood, A. J., et al. (2019). Good gig, bad gig: Autonomy and algorithmic control in the global gig economy. *Work, Employment and Society, 33*(1), 56–75.

Xiaomin, Xu, & Yi, Liu (2017). Customer satisfaction of the third-party logistics enterprise based on AHP: A case study. *International Journal of Information Systems and Supply Chain Management (IJISSCM), 10*(1), 68–81.

Yeo, V. C., Sern, S.-K., & Rezaei, S. (2017). Consumer experiences, attitude and behavioral intention toward online food delivery (OFD) services. *Journal of Retailing and Consumer Services, 35*, 150–162.

Zhang, Siyu, Liu, Luning, Feng, Yuqiang, 2019. A study of factors influencing restaurants sales in online-to-offline food delivery platforms: differences between high-sales restaurants and low-sales restaurants. PACIS.