Transportation Research Part C 138 (2022) 103571

Contents lists available at ScienceDirect

Transportation Research Part C

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

Predictive and prescriptive performance of bike-sharing demand forecasts for inventory management

Daniele Gammelli [a,*], Yihua Wang [b], Dennis Prak [c,d], Filipe Rodrigues [a], Stefan Minner [b,e], Francisco Camara Pereira [a]

[a] Department of Technology, Management and Economics, Technical University of Denmark, Kgs. Lyngby, 2800, Denmark
[b] Logistics and Supply Chain Management, School of Management, Technical University of Munich, 80333 Munich, Germany
[c] Department of Operations, University of Groningen, PO Box 800, 9700 AV Groningen, The Netherlands
[d] Department Industrial Engineering and Business Information Systems, University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
[e] Munich Data Science Institute (MDSI), Technical University of Munich, 85748 Garching, Germany

ARTICLE INFO

Keywords:
Bike-sharing system
Rebalancing problem
Demand forecast
Inventory level
Deep generative model

ABSTRACT

Bike-sharing systems are a rapidly developing mode of transportation and provide an efficient alternative to passive, motorized personal mobility. The asymmetric nature of bike demand causes the need for rebalancing bike stations, which is typically done during nighttime. To determine the optimal starting inventory level of a station for a given day, a User Dissatisfaction Function (UDF) models user pickups and returns as non-homogeneous Poisson processes with piece-wise linear rates. In this paper, we devise a deep generative model directly applicable in the UDF by introducing a *variational Poisson recurrent neural network* model (VP-RNN) to forecast future pickup and return rates. We empirically evaluate our approach against both traditional and learning-based forecasting methods on real trip travel data from the city of New York, USA, and show how our model outperforms benchmarks in terms of system efficiency and demand satisfaction. By explicitly focusing on the combination of decision-making algorithms with learning-based forecasting methods, we highlight a number of shortcomings in literature. Crucially, we show how more accurate predictions do not necessarily translate into better inventory decisions. By providing insights into the interplay between forecasts, model assumptions, and decisions, we point out that forecasts and decision models should be carefully evaluated and harmonized to optimally control shared mobility systems.

1. Introduction

The value of bike-sharing programs as an urban mobility solution is increasingly recognized by several cities around the world. They provide a flexible transport solution that easily connects to other modalities, and mitigates traffic congestion and air pollution. Bike-sharing concepts provide a healthy, cost- and time-efficient alternative to passive, motorized transportation (Sohrabi et al., 2020). Whereas the first pioneering experiments – such as the White Bikes project in Amsterdam (1965) – were completely unregulated, successful later implementations depended heavily on IT to prevent vandalism and theft (DeMaio, 2009). More recently, this IT usage enabled the application of advanced operations research and data science methods to optimize strategic, tactical, and operational decisions. There are currently over 2000 bike-sharing programs active world-wide, covering almost 10 million bikes,

[*] Corresponding author.
E-mail addresses: daga@dtu.dk (D. Gammelli), yihua.wang@tum.de (Y. Wang), d.r.j.prak@utwente.nl (D. Prak), rodr@dtu.dk (F. Rodrigues), stefan.minner@tum.de (S. Minner), camara@dtu.dk (F.C. Pereira).

https://doi.org/10.1016/j.trc.2022.103571
Received 28 July 2021; Received in revised form 13 January 2022; Accepted 18 January 2022
Available online 4 March 2022

a rapid growth compared with 2 million bikes in 2016 and 700,000 bikes in 2013 (Richter, 2018; Meddin et al., 2021). The vast majority of these projects are station-based and consist of networks of fixed-location stations with physical bike slots (Sohrabi et al., 2020; Shaheen et al., 2010). One of the major challenges of bike-sharing systems is that trip origins and destinations are asymmetrically distributed (e.g. reflecting commuting into a downtown in the morning and vice-versa in the evening), making the overall system imbalanced and sensitive to disturbances. To counteract this, the bikes in such networks are usually rebalanced during the night, when demand is low. This is called static rebalancing (Laporte et al., 2015; Tian et al., 2020).

The main operational-level decision problems of bike-sharing systems are demand forecasting, inventory decision-making, and rebalancing. These three problems are typically considered sequentially, with inventory targets being constraints for the rebalancing (routing) problem, and demand forecasts in turn serving as inputs to decide on these inventory targets. The target inventory level at the beginning of a day results from minimizing the so-called "user dissatisfaction", a penalty cost arising from arriving customers that do not find an available bike, and returning customers that do not find an empty slot to return their bike. Theoretical papers on bike-sharing rebalancing either assume given (ranges of) starting inventory levels or given functions for the cost resulting from these starting inventory levels. Empirical applications typically use a so-called User Dissatisfaction Function (UDF) that models pickups and returns as non-homogeneous Poisson processes. Such a UDF was first proposed by Raviv and Kolka (2013). The rates of these processes are assumed to be piece-wise constant (e.g. per hour) and typically obtained by taking historical averages of the same day and hour (O'Mahony and Shmoys, 2015; Schuijbroek et al., 2017; Freund et al., 2019).

A vast literature stream on demand forecasting for bike-sharing systems has rapidly emerged during the last decade. The topic attracts attention as it constitutes an exemplary case where historical demand records together with data on explanatory variables are abundantly available. Bike-sharing demand is known to be heavily dependent on temporal information (intra- and inter-day), but also on the weather (Eren and Uz, 2020), with several machine learning approaches being applied to model these relationships. However, the resulting forecasts are typically studied in isolation from inventory decisions, where authors judge the quality of their forecasts on standard accuracy metrics, such as MAE, (R)MSE, and R^2, but not on their eventual performance in the UDF. In this work, we argue that predictive and prescriptive performance goals should be carefully aligned when designing new predictive models, so to understand the relations between different methods and avoid unconscious overfitting to practically irrelevant forecasting metrics.

The contribution of this paper is threefold. First, we propose a neural architecture capable of modeling the pickups and returns as Poisson processes, thus being directly applicable in the UDF through a predict-then-optimize framework, whereby assumptions of the optimization algorithm are effectively encoded within upstream predictive model. Specifically, we propose a deep generative model whereby we represent the unknown Poisson rates as latent variables and where the time-dependent dynamics are captured by a Recurrent Neural Network (RNN). Second, we empirically evaluate our model against both traditional and learning-based approaches on real trip data from the 30 most active stations of New York Citi Bike, and show how our model outperforms benchmarks in terms of predictive and prescriptive performance. Third, we study existing mismatches between forecasting accuracy and decision performance. Specifically, the user dissatisfaction cost corresponding to a certain starting inventory level is a complex function of all pickup and return rates during the day and the hourly differences between them, thus creating a misalignment between the prediction and decision objectives. We propose to measure the error in the daily cumulative difference between pickups and return rates, and find that this better predicts inventory performance than MAE, MSE, and R^2.

The remainder of this paper is structured as follows. We first summarize relevant research directions in Section 2. We then introduce the main theoretical foundations and formally present the proposed approach in Section 3. Section 4 discusses empirical results on real world trip data and an extensive comparison with other forecasting methods. Section 5 concludes the paper.

2. Literature review

We review the literature that is relevant to our study in three main streams: static rebalancing, inventory modeling, and demand forecasting. We discuss these streams top-down, indicating how rebalancing problems depends on inventory models, that in turn depend on demand forecasts. Finally, we touch upon the disconnect between demand forecasts and inventory models for bike-sharing, and its parallels in the wider demand forecasting and inventory control literature. For a recent, general literature review on bike-sharing problems, we refer to Shui and Szeto (2020).

2.1. Static rebalancing

The vast majority of bike-sharing systems are statically rebalanced during nighttime, when demand is low and the impact of rebalancing is highest (Laporte et al., 2015). As a result, the majority of existing literature focuses on static rebalancing (Tian et al., 2020). Examples of work on dynamic rebalancing during the day are Angelopoulos et al. (2018), Caggiani et al. (2018), and Warrington and Ruchti (2019). Most literature on static rebalancing assumes a given target inventory level for each station as input to the system-level routing problem (e.g. Chemla et al., 2013; Dell'Amico et al., 2014; Wang and Szeto, 2021). Erdoğan et al. (2014) define a range of allowable inventory levels for each station, leading to more cost-efficient routing. This approach is also adopted by Kadri et al. (2016) and Schuijbroek et al. (2017).

Instead of assuming a given set of allowable starting inventory levels in the rebalancing problem, a more integrated approach is to include a UDF. This function maps the starting inventory level to a penalty cost, taking into account the stochastic transactions that occur throughout the day. This approach is used by e.g. Raviv et al. (2013), Szeto et al. (2016), and Ho and Szeto (2017), although these authors do not further describe the UDF. They solely assume its existence.

2.2. Inventory modeling

In rebalancing models, a number of studies approach the question of how to define either the range of allowable starting inventory levels, or the UDF. Nair and Miller-Hooks (2011), Nair et al. (2013), and Maggioni et al. (2019) introduce two-sided constraints for failed pickups at empty stations and failed returns at full stations. However, they only consider net demand and not the evolution of pickups and returns throughout the day.

Raviv and Kolka (2013) propose to model pickups and returns as independent, non-homogeneous Poisson processes with piecewise constant rates. This approach has subsequently been used by many authors in their applied work. O'Mahony and Shmoys (2015) determine rebalancing decisions, Çelebi et al. (2018) find the best locations for bike-sharing stations, and Freund et al. (2019) determine slot allocations and devise incentives for crowdsourcing rebalancing.

Also in large-scale applications, where inventory levels and routing decisions are taken for multiple stations simultaneously, a station-by-station UDF approach is used to determine optimal inventory levels for each station. The system-level decision problem may include additional constraints, for example on the total number of available bikes. Several authors report that solving such integrated problems is computationally infeasible for realistically-sized problems. Jian et al. (2016) and Alvarez-Valdes et al. (2016) apply simulation approximations. Contrarily, Vogel et al. (2014), Frade and Ribeiro (2015), and Datner et al. (2019) assume that demand for trips between station pairs is deterministic. Schuijbroek et al. (2017) combine the UDF pickup and return model with the logic of Nair and Miller-Hooks (2011) to find a range of inventory levels that satisfy constraints with respect to the number of failed pickups and returns.

Almost all authors that empirically apply the UDF use some form of historical averaging to estimate the pickup and return rates. Raviv and Kolka (2013) use time intervals of 1, 5 and 30 min. Alvarez-Valdes et al. (2016) and Çelebi et al. (2018) use an hourly interval, whereas Schuijbroek et al. (2017) and Jian et al. (2016) define the time interval as 15 and 30 min, respectively. O'Mahony and Shmoys (2015) and Freund et al. (2019) use 20-min intervals.

In line with the reviewed literature, we also use the UDF approach as originally proposed by Raviv and Kolka (2013) to determine target inventory levels. We deem comparing single target inventory levels more illustrative than comparing ranges of inventory levels and their implied service levels. Nevertheless, we remark that the procedure can be applied analogously to a service-based model such as that of Schuijbroek et al. (2017).

2.3. Demand forecasting

Whereas empirical contributions to bike-sharing rebalancing and inventory decision making mainly use simple historical averaging to estimate the pickup and return rates, there exists a vast literature on bike-sharing demand forecasting. Attempts are made to derive explanatory power from exogenous variables, such as weather and temporal information, using both classical and machine learning prediction techniques.

Rixey (2013) uses multivariate regression with data gathered from multiple bike-sharing systems, identifying a number of variables that have statistically significant correlations with station-level demand. More recently, the focus has shifted to machine learning approaches. System-level demand is forecasted by Xu et al. (2018) using long short-term memory neural networks, and by Guo et al. (2019) using graph neural networks.

More closely resembling the rebalancing decisions that are to be made, several authors have also applied machine learning to forecast station-level demand. Wang and Kim (2018) and Chen et al. (2020) employ RNNs, Lin et al. (2018) propose graph neural networks, and Sohrabi et al. (2020) use a generalized extreme value model. Fournier et al. (2017) use a sinusoidal model to deal with seasonalities. Random forests have been adopted by Yang et al. (2016), Du et al. (2019), and Sathishkumar and Cho (2020). Gammelli et al. (2020a,b) employ probabilistic techniques to estimate true demand using Tobit regression combined with Gaussian processes to mitigate the bias caused by censored demand observations, in both single and multi-output settings. In the presence of demand censoring, Negahban (2019) estimates real demand with a combination of simulation and bootstrapping, whereas Albiński et al. (2018) present a data-driven approach to estimate achieved service levels. Boufidis et al. (2020) compare various machine learning models in predicting station-level hourly pickups and returns. Zhang et al. (2021) iteratively update demand forecasts using a neural network and optimize a static rebalancing problem, but also separate both tasks.

Whereas several above-mentioned authors do focus on predicting station-level pickups and returns, they solely judge their forecasts using traditional loss measures for separate pickup and return demands. It is not explored how (pickup and return) demand forecasts perform if they are used to optimize stations' starting inventory levels. This existing disconnect between prediction and optimization has been addressed in general terms by Elmachtoub and Grigas (2021). Particularly, the lacking interface between demand forecasting and inventory control has been pointed out by Tratar (2010), Prak et al. (2017), and Kourentzes et al. (2020). Syntetos et al. (2010) state that the orders of magnitude of forecasting accuracy and inventory performance may differ wildly. Babai et al. (2014) find that positively biased forecasts can actually be beneficial if the demand distribution is misspecified.

In the field of bike-sharing, where (forecasts of) pickups and returns together determine the inventory trajectory of a station, the interface between predictive and prescriptive performance remains unstudied, despite the abundance of forecasting methods applied. This paper sheds light on how predictions affect decisions and, ultimately, system performance.

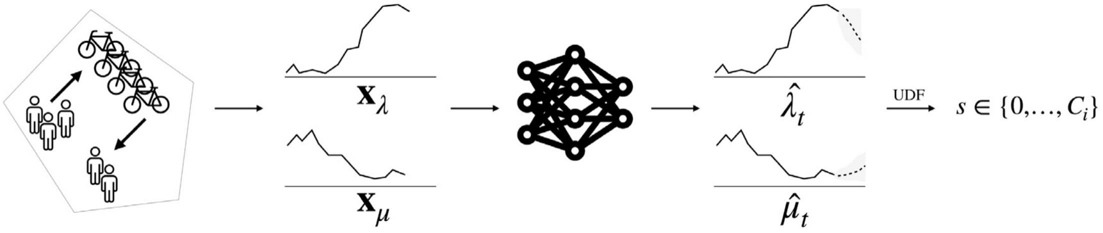


Fig. 1. An illustration of the framework determining the proposed inventory decision strategy. Given historical observations of pickup ($\mathbf{x}_\mu$) and return ($\mathbf{x}_\lambda$) processes, the VP-RNN computes predictions ($\hat{\mu}_t$ and $\hat{\lambda}_t$). With these, we estimate and minimize the UDF to find the optimal daily starting inventory level (s).

2.4. Recurrent latent variable models

In recent years, strong evidence has been gathered in favor of combinations bringing together the representative power of RNNs with the consistent handling of uncertainties given by probabilistic approaches, such as in Fabius and van Amersfoort (2015), Chung et al. (2016), Krishnan et al. (2016), Rangapuram et al. (2018), Ahmadi and Tani (2019), Salinas et al. (2020). The core concept underlying recent developments is the idea that, in current RNNs, the only source of variability is found in the conditional emission distribution (i.e. typically a unimodal distribution or a mixture of unimodal distributions), making these models inappropriate when modeling highly structured data.

One line of research particularly relevant for our work focuses on the definition of recurrent latent variable models for time-series forecasting. Despite the common probabilistic interpretation, existing approaches broadly differ on the way in which these handle uncertainty over the observed data. For example, the architectures introduced by Rangapuram et al. (2018) and Salinas et al. (2020) can both adapt to different likelihood functions. However, in order to enable for a tractable (and exact) maximization of the marginal likelihood at training time, these models involve the computation of *point estimates* of either likelihood function parameters (Salinas et al., 2020), or a linear state space model (Rangapuram et al., 2018). As we further elaborate in successive sections of this paper, a core element of this work is represented by the estimation of a full posterior distribution over likelihood parameters.

On a different, but related line of research, Fabius and van Amersfoort (2015), Chung et al. (2016), Ahmadi and Tani (2019) combine deterministic and stochastic (i.e., latent) variables in the definition of recurrent neural architectures. What unites these approaches is the idea that, through potentially high-dimensional latent variables, recurrent models can effectively handle multi-modality in the output distribution. On the other hand, in our work latent variables are not intended to be potentially high-dimensional vectors for modeling of multi-modality (as in the majority of VAE-based literature), but rather, as a way to encode epistemic uncertainty over the unknown, 1-dimensional Poisson rate parameter.

Fundamentally, the modeling choices outlined in this work (e.g. neural architectures, choice of likelihood function, dimensionality and support of latent variables, etc.) are strictly motivated by the assumptions characterizing the downstream decision-making routines, in an attempt to align predictive and prescriptive goals as much as possible.

3. Methodology

In this section, we introduce a framework[1] for inventory decision-making in bike-sharing systems. As illustrated in Fig. 1, this framework consists of a novel probabilistic neural architecture to estimate future pickup and return rates. These estimates are then used in a single-station inventory optimization model which defines pickups and returns as independent, non-homogeneous Poisson processes, and calculates the expected penalty due to failed pickups and returns as a function (the UDF) of the starting inventory level. Lastly, we decide on the optimal inventory level by minimizing the UDF. Either the UDF or its resulting target inventory levels can be used for solving system-level rebalancing problems as well. However, as reviewed in Section 2, the rebalancing problem imposes additional complexity caused by system-level constraints (e.g. the total number of available bikes), which may distract from our core subject of study: predictive and prescriptive performance of bike-sharing demand forecasting methods. Therefore, we zoom in on the connection between the quality of station-level demand predictions and the resulting (station-level) inventory prescriptions.

In this section, we introduce the theoretical fundamentals of our proposed approach. Specifically, we will first review an inventory model that utilizes the predicted pickup and return rates to determine a station's target inventory level at the beginning of a day (Section 3.1). We then outline the proposed VP-RNN (Section 3.2) by first reviewing the theory and notation describing latent variable models, recurrent neural networks and approximate inference (Sections 3.2.1–3.2.3), on which we will build to introduce the proposed generative model (Section 3.2.4).

[1] Code available at: https://github.com/DanieleGammelli/variational-Poisson-rnn.

3.1. Inventory decision model

In this section, we present an inventory model that uses the predicted pickup and return rates to determine a single station's target inventory level at the beginning of a day. The inventory model uses the User Dissatisfaction Function (UDF) proposed in Raviv and Kolka (2013). A bike station is modeled as a double-ended $M_t/M_t/1/C$ queuing system, with the number of customers in the queue representing the number of bikes in the station. The customer inter-arrival times (for bike returns) and service times (i.e., inter-arrival times for bike pickups) are assumed to be exponentially distributed with time-dependent rates λ_t and μ_t, respectively. Similar to Raviv and Kolka (2013), we assume that these rates are piece-wise constant. The capacity of the system C represents the capacity of the bike station, i.e., the total number of docks in the station.

We consider a static rebalancing problem where bikes are rebalanced overnight. Our goal is to determine the target inventory level for rebalancing operations in order to minimize the number of unsatisfied users for both bike pickup and return during the day. According to Raviv and Kolka (2013), the procedure of determining the target inventory level with predicted pickup rates $\hat{\mu}_t$ and return rates $\hat{\lambda}_t$ can be divided into two steps as follows.

1. Calculate the transient probability $p(s, \delta, t) \equiv \Pr(S(t) = \delta \mid S(0) = s)$, which is the probability of the station being at inventory level $\delta \in \{0, \dots, C\}$ at time $t \geq 0$ given that its starting inventory at time 0 was s. In a non-stationary queue, the transient probabilities are solutions to the Kolmogorov forward Eq. (1), which can be solved efficiently using the fourth-order Runge–Kutta method (Ross, 2014).

$$\begin{aligned}
&\dot{p}(s,0,t) = \hat{\mu}_t \cdot p(s,1,t) - \hat{\lambda}_t \cdot p(s,0,t) \\
&\dot{p}(s,\sigma,t) = \hat{\mu}_t \cdot p(s,\sigma+1,t) + \hat{\lambda}_t \cdot p(s,\sigma-1,t) - (\hat{\mu}_t + \hat{\lambda}_t) \cdot p(s,\sigma,t) \qquad \sigma = 1,\dots,C-1 \\
&\dot{p}(s,C,t) = \hat{\lambda}_t \cdot p(s,C-1,t) - \hat{\mu}_t \cdot p(s,C,t)
\end{aligned} \tag{1}$$

2. Calculate the expected penalty (UDF) due to failed pickups and returns over the observation period $[0, T]$ for all possible starting inventories $s \in \{0, \dots, C\}$. The optimal starting inventory s^* minimizes the UDF as shown in (2).

$$\begin{aligned}
UDF(s) &= \int_0^T l_p \cdot \hat{\mu}_t p(s,0,t) + l_r \cdot \hat{\lambda}_t p(s,C,t)\,\mathrm{d}t \\
s^* &= \arg\min_s UDF(s)
\end{aligned} \tag{2}$$

Here, l_p and l_r denote the unit penalty for each lost pickup and lost return, respectively. The first term in the integral represents the expected user dissatisfaction accumulated when the station is empty, and the second term represents the expected user dissatisfaction accumulated when the station is full.

3.2. Variational Poisson RNN

In this section, we first review and summarize key concepts on latent variable models (Section 3.2.1), approximate inference (Section 3.2.2), and recurrent neural networks (Section 3.2.3). We then build on these concepts to introduce the neural architecture for the proposed VP-RNN (Section 3.2.4). Specifically, through Sections 3.2.1–3.2.3, we aim to give a high-level overview of the key methodological concepts used in this work. Because of this, we choose to keep the narration fairly general and non-application specific. In Section 3.2.4, we then concretely bring these concepts together in the neural architecture defined by the proposed VP-RNN: (i) we encode model uncertainty over the unknown Poisson rates though latent variables (Section 3.2.1), (ii) we approximate the unknown posterior over the latent variables through variational inference (Section 3.2.2), and (iii) we model the time-dependent dynamics of bike-sharing demand through RNNs (Section 3.2.3).

3.2.1. Latent variable models (LVMs)

One of the central problems in the statistical sciences and machine learning is that of density estimation, i.e., the construction of a model of a probability distribution $p(\mathbf{x})$ given a finite sample of N data points $\mathcal{D} : \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ drawn from that distribution. A traditional approach to the problem of density estimation involves a parametric model $p_\theta(\mathbf{x})$, in which a specific form for the density is proposed which contains a set of learnable parameters θ. The parametric model of interest will be a Poisson distribution given by

$$p_\theta(\mathbf{x}) = \mathrm{Pois}(\mathbf{x} \mid \lambda), \tag{3}$$

where $\theta : \{\lambda\}$ is the set of learnable parameters containing the rate of the Poisson distribution. Learning, or parameter estimation, is then achieved by maximizing the (log) likelihood of the observed dataset as a function of the parameters, where it is assumed that the data points $\mathbf{x}_i$ are drawn independently from $p(\mathbf{x})$.

Of particular interest for this paper is the concept of *latent variables*. Specifically, rather than modeling $p(\mathbf{x})$ directly, we introduce a set of unobserved latent variables $\mathbf{z}$ by expressing a model for the joint probability distribution $p(\mathbf{x}, \mathbf{z})$. In practice, this is done by defining the joint probability as a product of two densities: the *prior distribution* $p(\mathbf{z})$ and the *likelihood* $p(\mathbf{x} \mid \mathbf{z})$ (sometimes referred to as the *sampling* or *data distribution*), $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$. In this context, parameter estimation, or *inference*, is achieved by using Bayes' rule, yielding the following *posterior* density:

$$p(\mathbf{z} \mid \mathbf{x}) = \frac{p(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})} = \frac{p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})}{p(\mathbf{x})}, \tag{4}$$

where $p(\mathbf{x}) = \int p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})d\mathbf{z}$, and the integral is over all possible values of $\mathbf{z}$ (or $p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})$ in case of discrete $\mathbf{z}$).

3.2.2. Approximate inference in LVMs

The posterior distribution in (4) compactly represents our beliefs about the latent variables after having observed the data $\mathcal{D}$, and is a key component for probabilistic reasoning in LVMs. In many cases of practical interest however, the posterior is intractable. Specifically, this intractability often derives from the lack of an analytical solution for the integral appearing in the denominator of (4). To address this intractability, we focus on deterministic techniques such as *variational inference* (VI) (Jordan et al., 1999; Blei et al., 2017; Zhang et al., 2018). At a high-level, in VI we use ideas from the calculus of variations to find a parametric approximation $q(\mathbf{z})$ that minimizes a measure of dissimilarity between $q(\mathbf{z})$ and the true, intractable posterior $p(\mathbf{z} \mid \mathbf{x})$. Out of the many different ways to measure dissimilarity between two distributions, variational inference uses the *Kullback–Leibler (KL)* divergence. That is, we are interested in minimizing the following divergence between the variational (or approximate) distribution $q(\mathbf{z})$ and the posterior distribution $p(\mathbf{z} \mid \mathbf{x})$, defined as:

$$\mathbb{KL}\left[q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x})\right] = -\mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{z} \mid \mathbf{x})}{q(\mathbf{z})}\right], \tag{5}$$

where $\mathbb{E}_{q(\mathbf{z})}$ denotes an expectation over $q(\mathbf{z})$. In order to define a tractable objective for our inference problem (i.e., one where the intractable posterior $p(\mathbf{z} \mid \mathbf{x})$ does not appear in the formulation), we can rewrite (5) using (4) (as well as the properties of the logarithm) as

$$\mathbb{KL}\left[q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x})\right] = -\mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} - \log p(\mathbf{x})\right] \tag{6}$$

$$= \underbrace{-\,\mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}\right]}_{\mathcal{L}(q)} + \log p(\mathbf{x}), \tag{7}$$

where the marginal log-likelihood $\log p(\mathbf{x})$ can be taken out of the expectation because of its independence from $\mathbf{z}$. The quantity $\mathcal{L}(q)$ is know as *Evidence Lower Bound* (ELBO) and represents a lower bound on the marginal log-likelihood, or evidence, $\log p(\mathbf{x})$, i.e. $\log p(\mathbf{x}) \geq \mathcal{L}(q)$ for all $q(\mathbf{z})$. Concretely, this reformulation gives us a way to minimize the $\mathbb{KL}\left[q(\mathbf{z}) \parallel p(\mathbf{z} \mid \mathbf{x})\right]$ by maximizing the ELBO with respect to the distribution $q(\mathbf{z})$, and therefore find the variational distribution best approximating the unknown posterior. In other words, the closer the ELBO is to the marginal log-likelihood, the closer (in KL sense) the variational approximation will be to the posterior distribution. Thus, variational methods allow us to reduce an inference problem into an optimization problem.

In practice, the variational distribution $q(\mathbf{z})$ is often restricted to a known parametric family for which the ELBO is tractable or simple to approximate, such as a Gaussian distribution. Thus, the maximization of the ELBO refers to a maximization with respect to the parameters ϕ of the variational distribution q_ϕ (e.g. $q_\phi(\mathbf{z}) = \mathcal{N}(\mathbf{z} \mid \phi)$, where $\phi = \{\boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ in the case of a Gaussian approximation).

In traditional variational inference, we learn a distinct set of parameters ϕ_i for each data point $\{\mathbf{x}_i\}_{i=1}^{N}$, which can be problematic when facing large, high-dimensional datasets. To avoid the linear growth in parameters with the number of data points, *amortized inference* offers a viable alternative. Specifically, rather than defining a set of parameters ϕ_i for each data point, amortized inference shares a unique set of parameters ϕ across all data points — thus, *amortizing* the cost of variational inference. As in the case of Variational Autoencoders (VAE) (Kingma and Welling, 2014; Rezende et al., 2014), we define an inference network, also known as encoder, that allows us to compute the parameters of the posterior approximation for any given data point. Specifically, in the case of a (diagonal) Gaussian variational approximation, we define an inference network with output characterizing the mean and variance vectors as:

$$\begin{aligned} q_\phi(\mathbf{z}_i \mid \mathbf{x}) &= \mathcal{N}(\mathbf{z}_i \mid \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2 I), \\ [\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2] &= f_\phi(\mathbf{x}_i), \end{aligned} \tag{8}$$

where f_ϕ can be any parametric function such as a deep neural network, and I is the identity matrix.

3.2.3. Recurrent neural networks

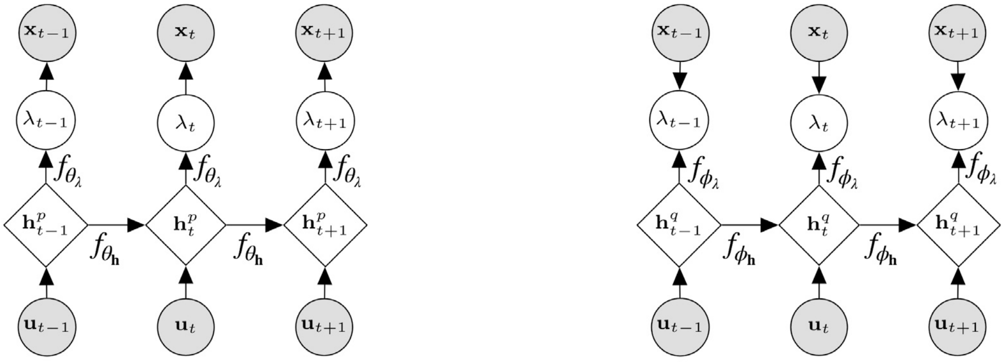
We summarize the usage of recurrent neural networks (RNNs) for sequential data modeling. RNNs are widely used to model variable-length sequences $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, possibly influenced by external covariates $\mathbf{u} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_T)$. The core assumption underlying these models is that all observations $\mathbf{x}_{1:t}$ up to time t can be summarized by a learned deterministic representation $\mathbf{h}_t$. At any timestep t, an RNN recursively updates its hidden state $\mathbf{h}_t \in \mathbb{R}^p$ by computing:

$$\mathbf{h}_t = f_{\theta_\mathbf{h}}(\mathbf{u}_t, \mathbf{h}_{t-1}), \tag{9}$$

where f is a deterministic non-linear transition function parametrized by $\theta_\mathbf{h}$, such as an Long Short-Term Memory (LSTM) cell or a Gated Recurrent Unit (GRU). The sequence is then modeled by defining a factorization of the joint probability distribution as the following product of conditional probabilities:

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T) &= \prod_{t=1}^{T} p(\mathbf{x}_t | \mathbf{x}_{<t}) \\ p(\mathbf{x}_t | \mathbf{x}_{<t}) &= g_{\theta_\mathbf{x}}(\mathbf{h}_t), \end{aligned} \tag{10}$$

where g is typically a non-linear function with parameters $\theta_\mathbf{x}$.



(a) Generative model

(b) Inference Network

Fig. 2. Graphical representation of the generative model p_θ (left) and inference network q_ϕ (right) characterizing the proposed VP-RNN. Shaded nodes represent observed variables, while non-shaded nodes represent either deterministic (diamond-shaped) or latent (circle-shaped) variables.

3.2.4. VP-RNN neural architecture

In this section, we define the generative model p_θ and inference network q_ϕ characterizing the proposed Variational Poisson RNN (VP-RNN) for the purpose of pickup and return demand modeling. A schematic illustration of the VP-RNN is shown in Fig. 2.

Generative model: We assume data $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ to represent a sequence of realizations of a Poisson process over the pickup (or, the return) process. That is, we recognize that our data is represented by count variables taking values $\mathbf{x}_t \in \mathbb{N} \cup \{0\}$ that we wish to model using a Poisson process specified by an appropriate rate parameter λ_t. For simplicity, we will always refer to a generic rate parameter λ_t when describing the proposed VP-RNN. However, it is important to underline how λ_t can represent any arbitrary Poisson rate in the context of bike-sharing demand prediction, such as independent pickup and return rates μ_t, λ_t, or even a 2-dimensional rate $\lambda_t = [\mu_t, \lambda_t]$, jointly modeling the pickup and return processes. We represent the rate λ_t as a latent variable whose time-dependent dynamics are modeled through an RNN. Specifically, the VP-RNN defines the following factorization of the joint probability distribution:

$$p(\mathbf{x}_{1:T}, \lambda_{1:T}, \mathbf{h}^p_{1:T} \mid \mathbf{u}_{1:T}, \mathbf{h}^p_0) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \lambda_t)\, p_{\theta_\lambda}(\lambda_t \mid \mathbf{h}^p_t)\, p_{\theta_\mathbf{h}}(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t), \tag{11}$$

$$p(\mathbf{x}_t \mid \lambda_t) = \mathrm{Pois}(\mathbf{x}_t \mid \lambda_t)$$

$$p_{\theta_\lambda}(\lambda_t \mid \mathbf{h}^p_t) = \mathcal{N}\left(\lambda_t \mid \boldsymbol{\mu}_{0,t} \mathrm{diag}(\boldsymbol{\sigma}^2_{0,t})\right), \text{with } [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = f_{\theta_\lambda}(\mathbf{h}^p_t),$$

where $\boldsymbol{\mu}_{0,t}$ and $\boldsymbol{\sigma}_{0,t}$ represent the parameters of the conditional prior distribution over the latent variable λ_t and where we assume $p_{\theta_\mathbf{h}}(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t) = \delta(\mathbf{h}^p_t - \tilde{\mathbf{h}})$, i.e. $\mathbf{h}^p_t$ follows a delta distribution centered in $\tilde{\mathbf{h}}_t = f_{\theta_\mathbf{h}}(\mathbf{h}^p_{t-1}, \mathbf{u}_t)$, such that the RNN hidden state (at time t) is itself a function of the hidden state at the previous time-step, and external data $\mathbf{u}_t$. In our implementation, f_{θ_λ} and $f_{\theta_\mathbf{h}}$ are respectively a feed-forward neural network and a GRU cell with parameters θ_λ and $\theta_\mathbf{h}$. At its core, the VP-RNN exploits the representational power of RNNs to capture potentially complex long-term dependencies in the temporal evolution of demand. It then leverages the learned representation $\mathbf{h}^p_t$ as conditioning variable for the conditional prior distribution over the Poisson rate variable λ_t.

Inference: The variational approximation defining the VP-RNN directly follows the generative model's factorization as follows:

$$q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T}) = \prod_{t=1}^{T} q_\phi(\lambda_t \mid \mathbf{x}_t, \mathbf{h}^q_t, \mathbf{u}_t), \tag{12}$$

$$q_\phi(\lambda_t \mid \mathbf{x}_t, \mathbf{h}^q_t, \mathbf{u}_t) = q_{\phi_\lambda}(\lambda_t \mid \mathbf{h}^q_t)\, q_{\phi_\mathbf{h}}(\mathbf{h}^q_t \mid \mathbf{h}^q_{t-1}, \mathbf{u}_t),$$

$$q_{\phi_\lambda}(\lambda_t \mid \mathbf{h}^q_t) = \mathcal{N}\left(\lambda_t \mid \boldsymbol{\mu}_{\lambda,t},\ \mathrm{diag}(\boldsymbol{\sigma}^2_{\lambda,t})\right), \text{with } [\boldsymbol{\mu}_{\lambda,t}, \boldsymbol{\sigma}_{\lambda,t}] = f_{\phi_\lambda}(\mathbf{h}^q_t),$$

where $q_{\phi_\mathbf{h}}(\mathbf{h}^q_t \mid \mathbf{h}^q_{t-1}, \mathbf{u}_t)$ follows a delta distribution centered in $\tilde{\mathbf{h}}^q_t = f_{\phi_\mathbf{h}}(\mathbf{h}^q_{t-1}, \mathbf{u}_t)$. Concretely, f_{ϕ_λ} and $f_{\phi_\mathbf{h}}$ together describe the encoder network defining the parameters $\boldsymbol{\mu}_{\lambda,t}$ and $\boldsymbol{\sigma}_{\lambda,t}$ of the approximate posterior distribution. In our implementation, f_{ϕ_λ} and $f_{\phi_\mathbf{h}}$ are respectively a feed-forward neural network and an LSTM cell with parameters ϕ_λ and $\phi_\mathbf{h}$. By explicitly resembling the model's factorization, the inference network defined in (12) also exhibits an implicit dependence on the entire history of $\mathbf{x}_{1:t}$ and $\mathbf{u}_{1:t}$ through $\mathbf{h}^q_t$. This implicit dependency on all information from the past can be considered as resembling a *filtering* approach from the state-space model literature (Durbin and Koopman, 2001). Denoting θ and ϕ as the set of model and variational parameters respectively, variational inference offers a scheme for jointly optimizing parameters θ, ϕ and computing an approximation to the posterior distribution by maximizing the following step-wise evidence lower bound[2] (i.e. ELBO) through gradient ascent:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t \mid \lambda_t) + \log p_\theta(\mathbf{h}_t^p \mid \mathbf{h}_{t-1}^p, \mathbf{u}_t) \right] \tag{13}$$
$$- \sum_{t=1}^{T} \mathbb{KL} \left(q_\phi(\lambda_t \mid \mathbf{h}_t^q, \mathbf{x}_t, \mathbf{u}_t) \mid\mid p_\theta(\lambda_t \mid \mathbf{h}_t^p) \right).$$

The learning (i.e., inference) problem can thus be re-written as the resulting (unconstrained) optimization problem:

$$\arg \min_{\theta,\phi} \mathcal{L}(\theta, \phi).$$

From a mathematical perspective, $\mathcal{L}(\theta, \phi)$ is a differentiable function, where the expectation is approximated through Monte Carlo sampling techniques, and is thus amenable for (stochastic) gradient-based optimization. Denoting with $\Theta = \{\theta, \phi\}$ the set of both model and variational parameters, we can compute the gradient of the loss function $\nabla_\Theta \mathcal{L}(\theta, \phi)$ through back-propagation and iteratively update the values of the parameters in the direction of the gradient, via the following update rule:

$$\Theta \leftarrow \Theta + \alpha \nabla_\Theta \mathcal{L}(\theta, \phi),$$

where α is a pre-determined step-size.

4. Empirical results

In this section, we demonstrate the performance of our proposed approach. Specifically, the goal is to answer the following questions: (1) Can we learn to reliably predict future pickup and return rates? (2) Does predictive performance align with decision-making performance? (3) In case of a misalignment, what aspects should be taken into consideration when working on frameworks combining prediction and decision-making? To answer these questions, we first analyze the performance of the proposed VP-RNN in predicting pickup and return processes of bike-sharing demand compared with other learning-based approaches (Section 4.1). We then explicitly evaluate the predictions when used for inventory management tasks both quantitatively (Section 4.2) and qualitatively (Section 4.3).

We use a real-world dataset from New York Citi Bike (Citi Bike, 2021). Citi Bike operates a station-based system, whereby the user of the service is not free to pick up or drop off a bike in any location, but is restricted to a certain number of physical stations around New York. Our objective is to model the temporal evolution of station-level pickup and return demand in the bike-sharing system and use this understanding to decide on effective starting inventory levels.

In all our experiments, we use data from the 30 most active stations in the Citi Bike's system from 1 January 2018 until 31 December 2018. The concept of *activity* is defined as the average sum of daily pickups and returns in each station. The 30 most active stations together cover approximately 25% of all rides in Citi Bike's system. As of December 2018, the system consisted of approximately 11,500 bikes, with 147,090 total annual memberships and an average demand in December 2018 of 41,172 rides per day. The stations which we consider in this work are representative of a number of different demand patterns, such as morning pickup (return) peaks and evening return (pickup) peaks in e.g., residential (business) areas, as well as more balanced situations.

The data consists of individual records of users renting and returning bikes, which we aggregate to three distinct temporal aggregation levels: 15-, 30- and 60-min intervals. Once aggregated, the data at our disposal is characterized by the time series of station-level pickups and returns, which we aim to predict one-day ahead at the start of every new day, to reflect the decision that is to be made. For all stations, we split the 12 months of data into train, validation and test sets using a ratio of 9/1/2 months, which we use respectively for training, model selection and early stopping, and the final evaluation of the implemented models.

For all models, we consider additional external explanatory variables to encode both meteorological and temporal information. Specifically, our features are characterized by the following sources of information: (i) temperature [°C], (ii) probability of rain $\in [0, 1]$, (iii) Day-of-Week (DoW), (iv) Time-of-Day (ToD), where we express both (iii) and (iv) as one-hot-encoded vectors. We use hourly weather measurements as recorded by the National Climatic Data Center (Rossow et al., 2016). In case of smaller temporal aggregation (15- and 30-min intervals), we assume the weather measurements to remain constant throughout the hour.

4.1. Predictive results

In this section, we analyze the performance of the proposed model on the task of pickup and return bike-sharing demand prediction. We compare the performance of the proposed VP-RNN with other learning-based approaches that are often used in empirical bike-sharing literature. We further place this comparison in the context of an ablation study to better analyze the contribution of each individual component of the VP-RNN. Concretely, we compare the performance of the following models:

1. Historical Average (HA): given a temporal aggregation (e.g. 60-min), the historical average for every combination of day-of-week and time-of-day (e.g. Monday 8 am) is calculated.

[2] A complete derivation is provided in Appendix.

Table 1
Test prediction performance. We report average (std. dev.) performance over all stations considered.

	Models	Pickup			Return		
		RMSE	MAE	R^2	RMSE	MAE	R^2
60 min	Historical Average	6.76 (2.40)	4.19 (1.07)	0.23 (0.33)	6.65 (2.08)	4.17 (0.99)	0.29 (0.31)
	Moving Average	5.77 (2.00)	3.45 (0.80)	0.47 (0.07)	5.80 (1.86)	3.47 (0.75)	0.50 (0.07)
	Linear Regression	6.67 (2.39)	4.46 (1.35)	0.25 (0.33)	6.67 (2.20)	4.50 (1.25)	0.29 (0.31)
	Gaussian-RNN	5.03 (1.91)	3.11 (0.81)	0.61 (0.08)	5.08 (1.88)	3.09 (0.71)	0.62 (0.08)
	Poisson-RNN	4.25 (1.15)	2.65 (0.55)	0.70 (0.05)	4.28 (1.15)	2.67 (0.51)	0.71 (0.07)
	Variational Poisson-RNN	3.91 (0.99)	2.47 (0.46)	0.74 (0.06)	3.92 (0.93)	2.49 (0.49)	0.75 (0.07)
	Multi-Output VP-RNN	**3.77 (0.96)**	**2.39 (0.46)**	**0.76 (0.06)**	**3.71 (0.82)**	**2.36 (0.41)**	**0.78 (0.06)**
30 min	Historical Average	3.61 (1.14)	2.28 (0.49)	0.25 (0.24)	3.57 (1)	2.29 (0.46)	0.3 (0.24)
	Moving Average	3.28 (1.01)	2.04 (0.42)	0.4 (0.07)	3.3 (0.94)	2.05 (0.4)	0.43 (0.08)
	Linear Regression	3.7 (1.19)	2.5 (0.67)	0.21 (0.28)	3.72 (1.13)	2.52 (0.63)	0.25 (0.26)
	Gaussian-RNN	2.73 (0.72)	1.71 (0.29)	0.59 (0.07)	2.78 (0.81)	1.74 (0.30)	0.60 (0.07)
	Poisson-RNN	2.53 (0.53)	1.61 (0.25)	0.63 (0.08)	2.53 (0.47)	1.61 (0.24)	0.64 (0.09)
	Variational Poisson-RNN	2.39 (0.43)	1.55 (0.22)	0.66 (0.09)	2.41 (0.42)	1.56 (0.22)	0.67 (0.09)
	Multi-Output VP-RNN	**2.32 (0.42)**	**1.5 (0.21)**	**0.68 (0.09)**	**2.33 (0.4)**	**1.51 (0.21)**	**0.7 (0.08)**
15 min	Historical Average	2.07 (0.56)	1.35 (0.25)	0.21 (0.20)	2.05 (0.50)	1.35 (0.24)	0.25 (0.20)
	Moving Average	1.94 (0.50)	1.24 (0.23)	0.32 (0.07)	1.95 (0.47)	1.25 (0.22)	0.34 (0.09)
	Linear Regression	2.11 (0.59)	1.44 (0.33)	0.18 (0.23)	2.12 (0.56)	1.45 (0.32)	0.21 (0.22)
	Gaussian-RNN	1.65 (0.28)	1.06 (0.15)	0.50 (0.09)	1.70 (0.36)	1.09 (0.16)	0.50 (0.08)
	Poisson-RNN	1.59 (0.28)	1.03 (0.15)	0.52 (0.10)	1.59 (0.26)	1.03 (0.15)	0.54 (0.11)
	Variational Poisson-RNN	1.55 (0.25)	1.02 (0.13)	0.54 (0.11)	1.56 (0.23)	1.03 (0.13)	0.56 (0.11)
	Multi-Output VP-RNN	**1.53 (0.25)**	**1.00 (0.13)**	**0.55 (0.10)**	**1.53 (0.23)**	**1.01 (0.14)**	**0.57 (0.10)**

2. Moving Average (MA): functionally equivalent to HA, with the only difference that the average is computed using only the last month of data in a rolling window.
3. Linear Regression (LR): parametrizes the dependency of the number of pickups/returns $\mathbf{x}_t$ on explanatory features $\mathbf{u}_t$ through a linear relationship, estimated by ordinary least squares.
4. Gaussian RNN (G-RNN): inspired by Chen et al. (2020), the model is characterized by a recurrent neural network based on gated recurrent units (GRU) and trained using a Gaussian likelihood, or equivalently, a Mean Squared Error (MSE) loss.
5. Poisson RNN (P-RNN): variation on the proposed VP-RNN not including an explicit latent variable over the Poisson rate λ_t. In line with Section 3, the P-RNN defines the following factorization of the joint distribution:

$$p(\mathbf{x}_{1:T}, \lambda_{1:T}, \mathbf{h}^p_{1:T} \mid \mathbf{u}_{1:T}, \mathbf{h}^p_0) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \lambda_t)\, p_{\theta_\lambda}(\lambda_t \mid \mathbf{h}^p_t)\, p_{\theta_\mathbf{h}}(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t), \tag{14}$$
$$p(\mathbf{x}_t \mid \lambda_t) = \mathrm{Pois}(\mathbf{x}_t \mid \lambda_t), \text{with } \lambda_t = f_{\theta_\lambda}(\mathbf{h}^p_t).$$

   Given the absence of latent variables, the P-RNN allows for exact maximum likelihood estimation of the parameters. For all stations considered, in our implementation p_{θ_h} is a GRU with 128 hidden units, and p_{θ_λ} is a 2-layer MLP with 128 hidden units per hidden layer.
6. Variational Poisson RNN (VP-RNN): the model as described in Section 3.2.4. Similarly to P-RNN, in our implementation p_{θ_h} is a GRU with 128 hidden units, and p_{θ_λ} is a 2-layer MLP with 128 hidden units per hidden layer. However, the VP-RNN defines a Gaussian distribution over λ, opposed to a single point-estimate as in P-RNN. The inference network mirrors the implementation of the generative model where q_{θ_h} is a GRU with 128 hidden units, and q_{θ_λ} is a 2-layer MLP with 128 hidden units per hidden layer.
7. Multi-Output Variational Poisson RNN (MOVP-RNN): multi-output extension to the proposed VP-RNN. Specifically, this formulation allows to jointly model pickup and return processes by defining a multivariate regression variable $\mathbf{x}_t = [x_{\mu,t}, x_{\lambda,t}]$, where $x_{\mu,t}$ and $x_{\lambda,t}$ represent pickup and return counts, respectively. By doing so, MOVP-RNN can potentially leverage correlations between the pickup and return temporal patterns. Our MOVP-RNN implementation uses the same number of model parameters as our VP-RNN.

Table 1 shows the predictive performance of the implemented models based on three commonly-used measures: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and the coefficient of determination (R^2). We now concentrate on the results for the 60-min aggregation, as presented in Table 1, because they are representative also of the results for the other two temporal aggregations. Unsurprisingly, results show how RNN-based approaches have a clear advantage when compared to the classical benchmarks that are typically used in empirical bike-sharing literature.

Table 1 further highlights the contributions of each individual component of our proposed model. First, results show how the MOVP-RNN is able to exploit its additional flexibility in modeling correlations between the pickup and return processes, obtaining better performance compared to its single-output variant (VP-RNN) across all metrics. Secondly, by comparing between Gaussian and Poisson-based approaches, results in Table 1 empirically highlight the importance of encoding within predictive models the right assumptions about the data distribution, i.e., in our case, pickup and returns as samples from a time-dependent Poisson process.

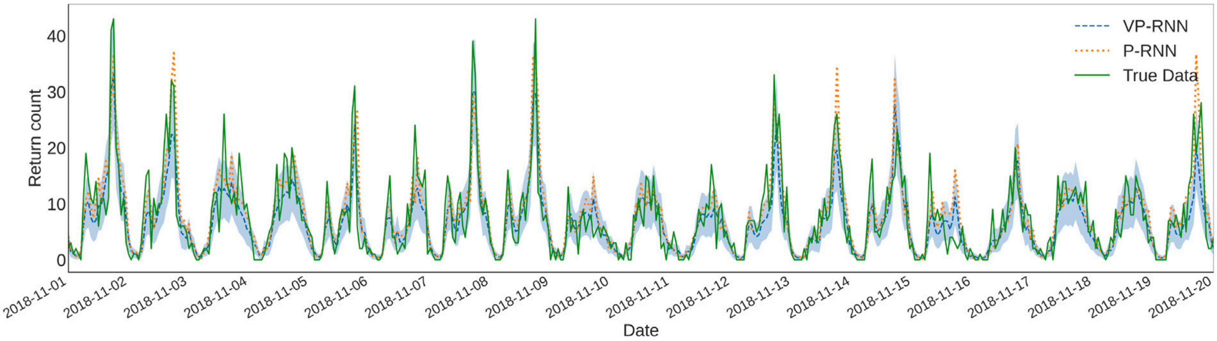


Fig. 3. A graphical representation of model prediction for 20 consecutive test days in station 504 of the Citi Bike system. The plot compares VP-RNN (blue, dashed curve), P-RNN (orange, dotted curve) against true realizations of bike return counts (green, continuous curve). The blue shaded area represents the 95% interval under the posterior predictive distribution over the return Poisson rate parameter. The plot shows how VP-RNN is able to obtain higher log-likelihood values by averaging over multiple possible rates that could have generated the data. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
Average test log-likelihood across stations. For the non-deterministic model (VP-RNN) the approximation of the marginal log-likelihood is indicated with the ≈ sign.

Models	Pickup	Return
Poisson-RNN (P-RNN)	3666	4012
Variational Poisson-RNN (VP-RNN) ≈	**3896**	**4210**

Moreover, Table 1 also highlights the gains in defining explicit latent variables over the rate parameter, thus allowing for a structured treatment of uncertainty and ultimately leading to more accurate predictions.

To further illustrate the potential advantages of explicitly modeling the rate as a latent variable, in Table 2 we compare the VP-RNN with its deterministic counterpart P-RNN. Specifically, results show the test log-likelihood averaged over all stations of interest for the 60-min aggregation. We report exact log-likelihoods for P-RNN, while in the case of VP-RNN, we report the importance sampling approximation to the marginal log-likelihood using 30 samples, as in Rezende et al. (2014). For both pickup and return processes, we see how the combination of RNNs with latent variable models allows the VP-RNN to better estimate the demand process, obtaining higher log-likelihood values on held-out data.

Crucially, by explicitly allowing for the presence of latent variables, the VP-RNN is able to express its uncertainty over the rate parameter of the demand Poisson process by computing a full posterior distribution. As qualitatively highlighted in Fig. 3, the VP-RNN predicts a full distribution over future demand rates possibly generating the observed data, whereas, by construction, the P-RNN only defines a point estimate for future rates.

4.2. Prescriptive results

In this section, we focus on evaluating the inventory performance of different predictive models. Specifically, we count the number of shortages (of pickups and returns) during the next day based on actual demand data (i.e., the sequence consists of actual pickup and return events), assuming the initial bike inventory level prescribed by the UDF (see Section 3.1) obtained using the computed pickup and return forecasts.

We compare our solutions to a model which receives perfect information about future demand patterns. Specifically, we use the UDF with perfect information about future rates $\hat{\mu}_t, \hat{\lambda}_t$. Therefore, this approach serves as an *oracle* that provides prescriptive performance in the limit of perfect forecasting accuracy for any algorithm within the same inventory decision model.

We present two performance indicators: (i) *Cost*: the average total penalty per day due to the unsatisfied customers for pickups and returns. Formally:

$$\text{Cost} = l_p \times \text{number of failed pickups} + l_r \times \text{number of failed returns}$$

Initially, we set the unit penalty for each lost pickup and lost return to one ($l_p = l_r = 1$). Later we test the inventory performance with different parameter settings in Section 4.4. (ii) *Relative percentage difference* (RPD) from oracle performance. Formally:

$$\text{RPD} = \frac{c_i - c_{oracle}}{c_{oracle}},$$

where c_{oracle}, c_i represent the costs obtained respectively by the oracle model and model $i \in$ {HA, MA, LR, G-RNN, P-RNN, VP-RNN, MOVP-RNN}. Results in Table 3 show that the forecasts generated by VP-RNN lead to the best decisions. Specifically, inventory decisions based on VP-RNN predictions are able to decrease costs at least 40% closer to oracle performance when compared to

Table 3
Test prescriptive performance for inventory management tasks averaged over all stations of interest.

Aggregation	Measure	HA	MA	LR	G-RNN	P-RNN	VP-RNN	MOVP-RNN	Oracle
60 min	Cost	10.14	10.01	10.88	11.93	9.95	**9.37**	10.21	8.18
	RPD	24.1%	22.5%	33.1%	45.9%	21.7%	**14.6%**	24.9%	–
30 min	Cost	10.13	10.01	10.88	9.52	9.56	**8.91**	10.25	8.18
	RPD	23.9%	22.4%	33.1%	16.4%	16.8%	**8.9%**	25.3%	–
15 min	Cost	10.13	10.00	10.88	9.38	9.17	**8.76**	10.20	8.19
	RPD	23.7%	22.1%	32.8%	14.6%	11.9%	**7.0%**	24.5%	–

Table 4
Performance statistics for the predictions shown in Fig. 4. Results compare VP-RNN and MOVP-RNN on both prediction and decision performance.

	Prediction						Decision	
	Pickup			Return			Inventory	Cost
	RMSE	MAE	R^2	RMSE	MAE	R^2		
VP-RNN	5.35	3.53	0.76	6.02	3.92	0.69	11	**13.72**
MOVP-RNN	**3.04**	**2.09**	**0.92**	**4.94**	**3.43**	**0.79**	32	29.49

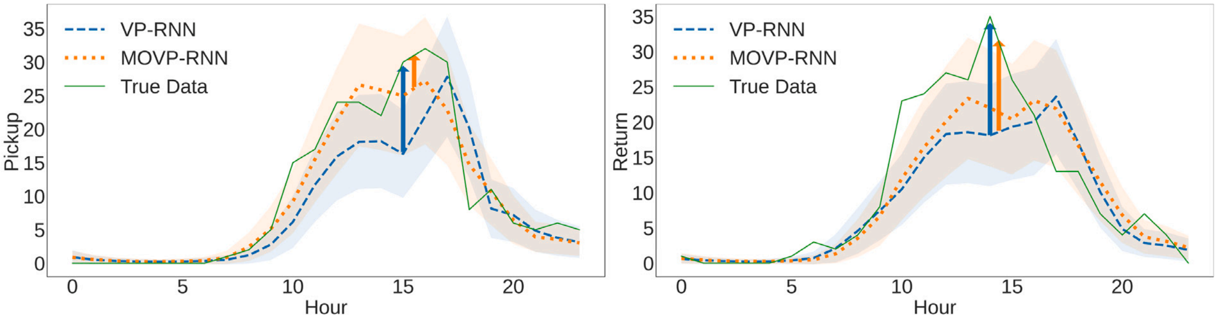


Fig. 4. A graphical representation of pickup (left) and return (right) predictions for one test day in station 3641 of the Citi Bike system. The plot shows how the two models reach different types of over/under-estimation patterns, where the arrows highlight the different biases during key moments of the day (e.g. the afternoon demand peak).

traditional HA, MA and LR performance and at least 10% when compared to VP-RNN's deterministic counterpart P-RNN. Results also highlight how predictive models which assume Poisson distributed demand (especially P-RNN and VP-RNN) benefit significantly from using smaller temporal discretizations. This is in line with the finding of Raviv and Kolka (2013) that finer time discretizations yield a better fit of the non-homogeneous Poisson process to the pickup and return time series. Contrarily, classical models that do not use the Poisson property do not show a significant improvement when time intervals are chosen smaller.

Notably, our evaluation highlights a fundamental misalignment between prediction and decision performance.

Table 1 shows that MOVP-RNN obtains the best prediction performance across all stations and test days, and using either RMSE, MAE, or R^2. However, when evaluated in the context of decision performance, the predictions of MOVP-RNN lead to costs close to those achieved by HA, MA and LR.

Fig. 4 and Table 4 show a representative example where this misalignment is particularly evident. Fig. 4 compares pickup and return predictions belonging to VP-RNN and MOVP-RNN against real pickup and return observations on a single held-out test day, and Table 4 presents both prediction and decision performance for the same day. In Fig. 4, we can observe how the two models have different error patterns, with the VP-RNN underestimating pickups and returns in a similar way, opposed to the MOVP-RNN which is approximately unbiased for the pickups but underestimates the return process. From a purely predictive point of view, it is clear that MOVP-RNN represents a better model (Table 4). However, once the generated predictions are used in the UDF, the relative performance between the two models is reversed.

The reason behind this misalignment lies in the nature of the decision-making problem at hand. Specifically, when considering the task of selecting the best starting inventory, the optimal decision is fundamentally influenced by the cumulative difference between pickups and returns, rather than only their separate evolution over the day. For example, in Fig. 4, by underestimating only the return rate, MOVP-RNN wrongly predicts a higher cumulative net demand (i.e., it predicts the correct number of pickups, but a lower number of returns), ultimately selecting higher starting inventories, and in practice leading to higher overall costs. On the other hand, by having similar biases between pickup and return predictions, the VP-RNN will have a better estimate of the optimal starting inventory level, thus describing a situation where prediction errors in the same direction might (partially) cancel out when evaluated on decision performance.

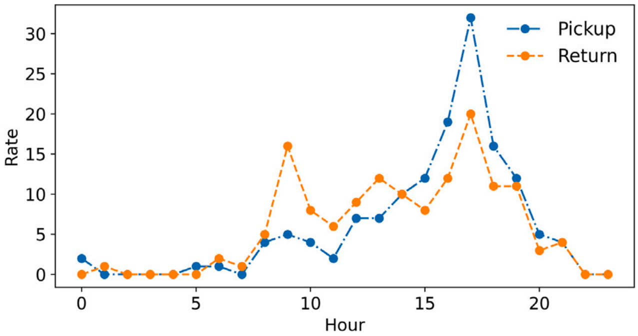


Fig. 5. Graphical representation of hourly pickup rates μ_t and return rates λ_t for one test day of station 168 of the Citi Bike system, on which we base our synthetic experiment.

Crucially, current research fails to acknowledge these interactions between prediction and decision-making tasks, and rather focuses on either prediction or decision performance in isolation. Motivated by this, we argue that having a deep understanding of how properties of the predictions can effect downstream decision-making processes is of fundamental importance. Therefore, we further explore this phenomenon in the remainder of this section.

4.3. Qualitative analysis on the prediction–decision misalignment

To analyze the reason for the misalignment between predictive and prescriptive performance, we design a synthetic experiment. The goal of the experiment is to show how the bias of the prediction affects inventory decision making. We consider two types of bias: the *same-side* bias, i.e. when pickups and returns are both either over-estimated or under-estimated, and the *opposite-side* bias, i.e. when pickup and return estimates are biased in opposite directions. In what follows, we examine the impact of these different biasing patterns on the inventory decisions computed according to the model presented in Section 3.1.

For exposition purposes, we select the demand pattern observed in station 168 on November 13, 2018. We choose a 60-min aggregation time interval, and assume that we have perfect information about pickup and return patterns for that day. The observed counts of pickup and return arrivals within each hour t are regarded as pickup rate μ_t and return rate λ_t, respectively. Fig. 5 shows the demand pattern of the selected instance, represented as pickup rate μ_t and return rate $\lambda_t, t \in [0, 23]$. Given the information about the true count demand rates, the oracle starting inventory is then calculated according to (1)–(2).

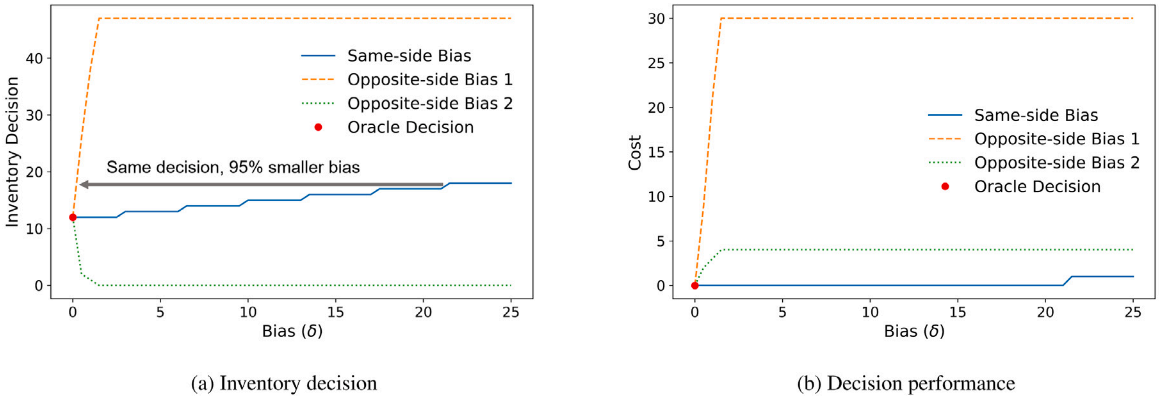
To assess the impact of different bias patterns, we select bias levels δ in the interval $[0, 25]$ in increments of 0.5. For each δ and each t, we generate the same-side biased pickup rate $\hat{\mu}_t = \mu_t + \delta$ and return rate $\hat{\lambda}_t = \lambda_t + \delta$. On the other hand, for the opposite-side bias, the pickup rate is over-estimated while the return rate is under-estimated by δ, or vise versa. Specifically, the first opposite-side biased pickup rate and return rate are calculated as $\hat{\mu}_t = \mu_t + \delta$ and $\hat{\lambda}_t = \max(\lambda_t - \delta, 0)$. Note that in the case of downward biases, we truncate the resulting prediction at 0 to retain feasible estimates for the rates. Positive estimates are typically guaranteed by any prediction method. The second opposite-side biased pickup rate and return rate are calculated as $\hat{\mu}_t = \max(\mu_t - \delta, 0)$ and $\hat{\lambda}_t = \lambda_t + \delta$, respectively. Finally, we calculate inventory decisions for all biased demand rates according to (1)–(2). Fig. 6 illustrates the relationship between inventory decisions and bias level δ, and the performance of decisions under bias level δ.

In Fig. 6, the inventory decision gradually deviates from the oracle decision as δ increases. Under the opposite-side bias 2, where pickup rates are under-estimated and return rates are over-estimated, the inventory decision quickly converges to the lower bound 0. Under the opposite-side bias 1, where pickups are over-estimated and returns are under-estimated, the inventory decision deviates rapidly towards the upper bound of station capacity. On the other hand, the inventory decision computed under the same-side bias is always very close to the oracle decision, even if the bias level δ is very large. Plot (b) shows that the resulting cost for any same-side bias smaller than 22 is 0, indicating that there are still no dissatisfied customers, despite the slightly different decision. Also for larger biases the cost only increases to 1. Contrarily, a small over-estimation of 2 units of pickups together with under-estimated returns already leads to 30 dissatisfied customers, a cost that is not caused by any same-side bias level in our experiment. Concluding, pickup and return rates with a very large bias on the same side still yield much better inventory decisions than those with a small bias on opposite sides.

4.4. Cumulative Error metric

The qualitative analysis highlights that improved forecasting accuracy on pickup and return rates individually, does not necessarily lead to better inventory decisions. Specifically, the inventory decision model is based on a complex function of all pickup and return rates during the day and the difference between them, which creates a misalignment between prediction and decision objectives. To deal with this misalignment, we introduce a new metric to measure forecast quality, the *Cumulative Error* (CE) of net demand during a day. We formally define CE as follows:

$$\mathrm{CE} = \left| \sum_{t=0}^{T} [(\mu_t - \lambda_t) - (\hat{\mu}_t - \hat{\lambda}_t)] \right| \tag{15}$$



(a) Inventory decision

(b) Decision performance

Fig. 6. Impact of different bias patterns on inventory decisions as a function of bias level δ. Plot (a) compares inventory decisions under bias predictions against the oracle inventory decision (red dot) under predictions with perfect information ($\delta = 0$). To get the same deviation from the oracle inventory decision, the same-side bias can sustain a 95% increase in forecasting error compared with the opposite-side bias. Plot (b) evaluates the performance of the decisions using the cost of lost sales.

Table 5
Average CE over all test days and all 30 stations.

Aggregation	HA	MA	LR	G-RNN	P-RNN	VP-RNN	MOVP-RNN
60 min	12.01	11.54	12.23	24.25	15.10	**11.50**	12.82
30 min	11.93	11.54	12.23	14.59	11.89	**7.29**	11.72
15 min	11.93	11.54	12.23	11.63	11.16	**6.19**	10.95

In (15), $\mu_t - \lambda_t$ represents the difference between the true pickup and return rates (i.e., the actual mean of net demand) within time interval t, while $\hat{\mu}_t - \hat{\lambda}_t$ represents the difference between the predicted pickup and return rates (i.e., predicted net demand) within time interval t.

Table 5 shows the average CE over all test days and stations. When evaluated on CE, VP-RNN clearly outperforms all the other models, which is consistent with the prescriptive performance shown in Table 3. Even though its predictions of pickup and return rates are separately not the most accurate, VP-RNN still makes the best prescriptive decision. This finding indicates that, compared with the traditional measures like MAE, RMSE and R^2, CE yields a measure of prediction quality that is better aligned with the eventual decision performance. In the remainder of this section, we analyze the effectiveness of CE as a measure of predictive bias, its correlation to prescriptive performance, and its sensitivity to problem parameters.

4.4.1. CE to quantify the effect of prediction bias

Given the newly introduced CE metric, we further quantify the intuitions built in Section 4.2 (Fig. 4) on an aggregated station and system level, opposed to a single representative example. Specifically, we use the definition of CE to quantify prediction bias. In order to have a valid comparison of bias values across different days and stations (possibly characterized by varying magnitudes),

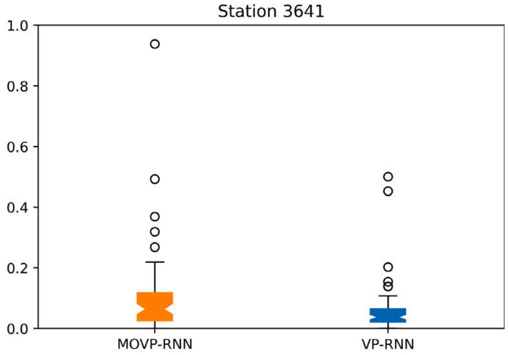


Fig. 7. Test percentage daily bias of MOVP-RNN and VP-RNN for Station 3641.

Table 6
Pearson correlation coefficients between the Cost and different measures of predictions. The results are averaged over all stations of interest.

Measures		60 min	30 min	15 min
CE		0.51	0.62	0.70
Pickup	MAE	0.21	0.20	0.18
	RMSE	0.37	0.38	0.38
	R2	0.08	0.12	0.21
Return	MAE	0.22	0.22	0.19
	RMSE	0.42	0.44	0.43
	R2	0.15	0.21	0.31

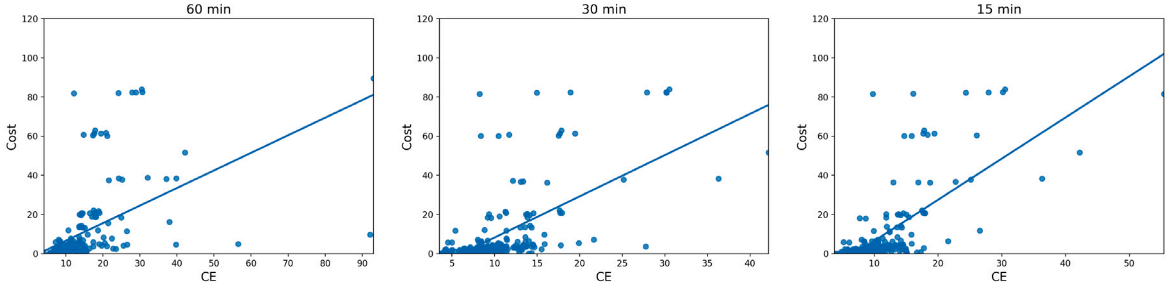


Fig. 8. Scatter plots showing the correlation between CE and cost under the 60- (left), 30- (middle), and 15-min (right) aggregations. The Pearson correlation coefficients are 0.51, 0.62 and 0.70 for the 60-, 30- and 15-min aggregations, respectively.

we normalize CE by the total number of daily pickups, as follows:

$$\frac{\left|\sum_{t=0}^{T}[(\mu_t - \lambda_t) - (\hat{\mu}_t - \hat{\lambda}_t)]\right|}{\sum_{t=0}^{T}\mu_t}.$$

In Fig. 7, we report the boxplot of the resulting percentage bias computed for the entire test set on Station 3641 (the same used for the representative example).

Results highlight a clear tendency of the VP-RNN to be more concentrated towards smaller bias numbers, confirming the intuition of Fig. 4 over the entire test set for this station. Appendix B extends these results to all stations under study.

4.4.2. *Correlation between CE and prescriptive performance*

In this section, we test the relationship between CE-performance and the costs of the decisions for all prediction models and stations. In Fig. 8, the three scatter plots correspond to the 60-, 30- and 15-min aggregations, respectively. The Pearson correlation coefficients are all above 0.5, which indicates a notable positive correlation between CE and cost. As the aggregation time shortens, results highlight how the correlation between CE and cost becomes stronger.

For comparison, we calculate the Pearson correlation coefficients between Cost and different measures of predictions, including MAE, RMSE, and R^2. The results in Table 6 show that under all problem settings, CE has stronger correlations with Cost than the other measures. This indicates its superior prediction quality regarding the decision performance.

4.4.3. *Sensitivity of CE to hyper-parameters*

The unit penalties for each lost pickup and lost return – in particular their ratio l_p/l_r – are important inputs for the UDF inventory model. On the other hand, the CE measure does not use these parameters. Therefore, we test its robustness against different settings of the ratio. So far, we assumed that $l_p = l_r = 1$. Here we consider two different settings of l_p/l_r: $1.5/0.5$ and $0.5/1.5$. These different parameter settings are fed as input to the UDF model in order to obtain different inventory decisions under the 15-min aggregation. Table 7 shows the average decision performance over all the 30 stations under different parameter settings.

Table 7 shows that the change in the values of these parameters does not change the optimal solution, but only scales its value. A higher ratio of l_p/l_r leads to overall smaller costs compared with the lower ratio of l_p/l_r. However, the comparison results between different models remain in the same order. Particularly, VP-RNN, which attains the lowest value of the CE measure (see Table 5), remains the best-performing method. Thus, although the CE measure does not depend on the hyper-parameters, its selected method performs best also under variations of these hyper-parameters. This indicates a certain robustness of CE to variations of the hyper-parameters in selecting the best-performing method.

Table 7
Test decision performance under different hyper-parameters. l_p and l_r denote the unit cost of unsatisfied pickup and return, respectively. The results are averaged over all stations of interest.

Parameters	Measure	HA	MA	LR	G-RNN	P-RNN	VP-RNN	MOVP-RNN	Oracle
$l_p = 1.5, l_r = 0.5$	Cost	9.17	9.06	9.86	8.46	8.35	**8.04**	9.12	7.56
	RPD	21.3%	19.9%	30.4%	11.9%	10.4%	**6.4%**	20.7%	–
$l_p = 0.5, l_r = 1.5$	Cost	10.43	10.31	10.92	9.86	9.62	**9.41**	10.43	8.95
	RPD	16.4%	15.1%	21.9%	10.2%	7.5%	**5.0%**	16.4%	–

5. Conclusion

Taking effective operational-level decisions in real-world bike-sharing systems unavoidably entails a combination of demand forecasting and inventory decision-making. Whereas both have separately received considerable attention, their interface has been left largely unaddressed. Bike pickups and returns jointly determine the inventory dynamics of a bike-sharing station. Therefore, a classical forecast accuracy evaluation of both streams separately is not perfectly indicative for the quality of the resulting inventory decision. This paper illuminates this mismatch by considering a UDF to determine daily starting inventory levels in combination with various forecasting methods for the pickup and return rates. Among these are variations of a novel, deep generative model that represents pickup and return rates as latent variables, as well as several classical and learning-based benchmarks.

We show that the proposed method outperforms the benchmarks in terms of both forecast accuracy and the service quality of the resulting inventory decisions. Explicitly using a Poisson likelihood at prediction time and modeling the pickup and return rates as latent variables yields a better distributional fit and higher forecast accuracy on the 2018 Citi Bike dataset than classical methods, as evaluated using RMSE, MAE, and R^2. However, whereas these three measures agree on the model variant with the highest prediction accuracy (the MOVP-RNN), another variant of our approach (the VP-RNN) gives the best inventory decisions in terms of customer service quality. Classical measures of forecast accuracy, when used separately on pickup and return predictions, are not fully indicative of prescriptive (decision) accuracy. Our experiments show that conforming signs of the errors in pickups and returns during the day can to a large extent cancel out the consequences of their magnitudes. Therefore, we propose a different accuracy measure, the Cumulative Error of net demand, and show that the ranking of forecasting methods based on this measure is in line with their ranking based on inventory performance. We furthermore show – in line with theoretical literature on non-homogeneous Poisson models for user dissatisfaction – that our approach can additionally benefit from narrowing down the prediction interval to 30 or 15 min, whereas classical benchmarks cannot.

In the context of bike-sharing, this paper builds an intuition for what predictive properties are relevant for taking effective inventory decisions. We highlight that (1) using simple averages, or even a linear regression, to estimate pickup and return rates leads to poor empirical decision performance, (2) learning-based approaches that exploit the Poisson likelihood in combination with a latent variable model for the rates lead to better decisions, but (3) it is crucial that predictive accuracy is measured in a way that aligns with the eventual inventory decision. That is, the joint effect of pickups and returns, their forecasts and biases should be taken into account. This paper proposes an accuracy measure that accomplishes this in the often-encountered case of static rebalancing.

Future research should proceed on the interface of demand prediction and decision making for shared mobility. In the context of station-based bike-sharing, directly embedding decision-performance incentives within deep learning architectures would enable end-to-end learning of “decision-aware predictors”. Integrating forecasting and decision-making further by means of imitation learning or data-driven optimization provides a different approach. Applying the proposed accuracy measure to other objectives, such as dynamic rebalancing, may lead to new insights in the forecasting methods’ performances. In this paper, we took the daily starting inventory levels of individual stations as decisions to directly link station-level forecasting accuracy to decision performance. A natural extension is to consider joint, system-wide forecasting and rebalancing. This yields additional opportunities, such as that of exploiting spatial dependencies in demand patterns. Efficiently exploiting decision performance in system-level forecasting of demand patterns, auto- and cross-sectional correlations, remains an open challenge. Possible extensions to free-floating or hybrid systems would open a plethora of new application areas, such as cars, scooters, and urban air mobility.

CRediT authorship contribution statement

Daniele Gammelli: Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Yihua Wang:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Visualization. **Dennis Prak:** Conceptualization, Methodology, Writing – review & editing, Visualization, Supervision. **Filipe Rodrigues:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Stefan Minner:** Conceptualization, Writing – review & editing, Project administration, Supervision, Funding acquisition. **Francisco Camara Pereira:** Conceptualization, Writing – review & editing, Project administration, Supervision, Funding acquisition.

Acknowledgments

This research was supported by TUM International Graduate School of Science and Engineering (IGSSE), Germany through the project ILOMYTS. Dennis Prak was supported by the Dutch Research Council (NWO), Netherlands [grant nr. 019.191SG.003].

Appendix A. Elbo derivation of (13)

We hereby report the derivation of the evidence lower bound used for inference in the VP-RNN:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}_{1:T}) &= \log \int p_\theta(\mathbf{x}_{1:T}, \lambda_{1:T}, \mathbf{h}_{1:T}) \mathrm{d}\lambda\, \mathrm{d}\mathbf{h} \\
&= \log \int \frac{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})}{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})} p_\theta(\mathbf{x}_{1:T}, \lambda_{1:T}, \mathbf{h}^p_{1:T}) \mathrm{d}\lambda\, \mathrm{d}\mathbf{h} \\
&= \log \mathbb{E}_{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})} \left[\prod_{t=1}^{T} \frac{p_\theta(\mathbf{x}_t \mid \lambda_t) p_\theta(\lambda_t \mid \mathbf{h}^p_t) p_\theta(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t)}{q_\phi(\lambda_t \mid \mathbf{h}^q_t, \mathbf{x}_t, \mathbf{u}_t)} \right] \\
&\geq \mathbb{E}_{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t \mid \lambda_t) + \log p_\theta(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t) + \log \left(\frac{p_\theta(\lambda_t \mid \mathbf{h}^p_t)}{q_\phi(\lambda_t \mid \mathbf{h}^q_t, \mathbf{x}_t, \mathbf{u}_t)} \right) \right] \\
&= \mathbb{E}_{q_\phi(\lambda_{1:T} \mid \mathbf{x}_{1:T})} \left[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t \mid \lambda_t) + \log p_\theta(\mathbf{h}^p_t \mid \mathbf{h}^p_{t-1}, \mathbf{u}_t) \right] \\
&- \sum_{t=1}^{T} \mathbb{KL} \left(q_\phi(\lambda_t \mid \mathbf{h}^q_t, \mathbf{x}_t, \mathbf{u}_t) \ || \ p_\theta(\lambda_t \mid \mathbf{h}^p_t) \right)
\end{aligned}
$$

Appendix B. System-level bias performance

This appendix extends the analysis of Fig. 4 to all stations in our dataset. Specifically, each plot represents the boxplot of the CE obtained over all days in the test set, for every station in the system. As for the individual case of Station 3641, results show that

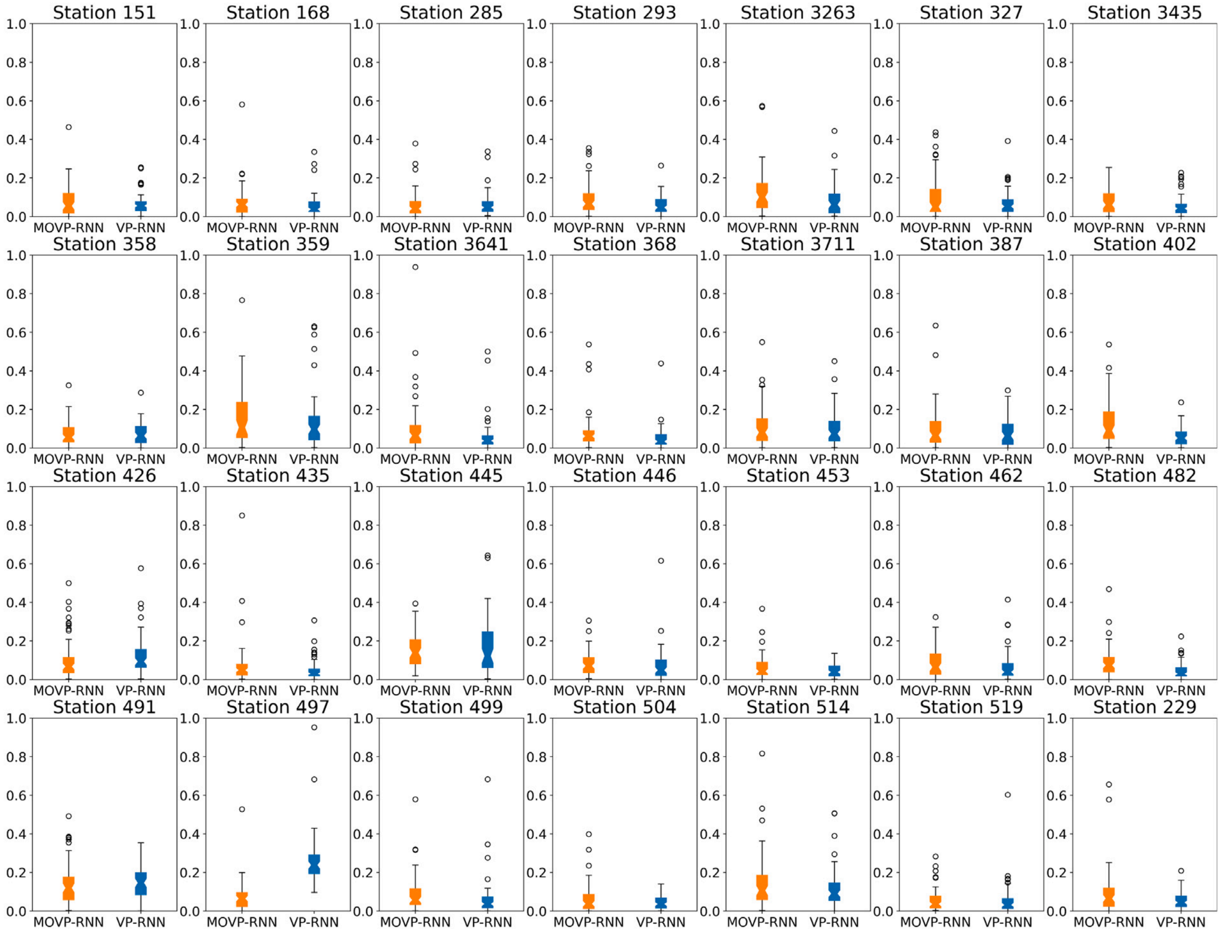


Fig. B.9. Test percentage daily bias of MOVP-RNN and VP-RNN for all stations.

VP-RNN is clearly characterized by a smaller bias in the estimate of the pickup and return process when compared to MOVP-RNN in more than 75% of the stations considered. Because of this, results in Fig. B.9 strongly support our claim that bias on the cumulative difference of pickup and return demand is highly correlated with downstream decision performance.

References

Ahmadi, A., Tani, J., 2019. A novel predictive-coding-inspired variational RNN model for online prediction and recognition. Neural Comput. 31 (11), 2025–2074.

Albiński, S., Fontaine, P., Minner, S., 2018. Performance analysis of a hybrid bike sharing system: A service-level-based approach under censored demand observations. Transp. Res. E Logist. Transp. Rev. 116, 59–69.

Alvarez-Valdes, R., Belenguer, J.M., Benavent, E., Bermudez, J.D., Muñoz, F., Vercher, E., Verdejo, F., 2016. Optimizing the level of service quality of a bike-sharing system. Omega 62, 163–175.

Angelopoulos, A., Gavalas, D., Konstantopoulos, C., Kypriadis, D., Pantziou, G., 2018. Incentivized vehicle relocation in vehicle sharing systems. Transp. Res. C 97, 175–193.

Babai, M.Z., Syntetos, A., Teunter, R., 2014. Intermittent demand forecasting: An empirical study on accuracy and the risk of obsolescence. Int. J. Prod. Econ. 157, 212–219.

Blei, D.M., Kucukelbir, A., McAuliffe, J.D., 2017. Variational inference: A review for statisticians. J. Amer. Statist. Assoc. 112 (518), 859–877.

Boufidis, N., Nikiforiadis, A., Chrysostomou, K., Aifadopoulou, G., 2020. Development of a station-level demand prediction and visualization tool to support bike-sharing systems' operators. Transp. Res. Procedia 47, 51–58.

Caggiani, L., Camporeale, R., Ottomanelli, M., Szeto, W.Y., 2018. A modeling framework for the dynamic management of free-floating bike-sharing systems. Transp. Res. C 87, 159–182.

Çelebi, D., Yörüsün, A., Işık, H., 2018. Bicycle sharing system design with capacity allocations. Transp. Res. B 114, 86–98.

Chemla, D., Meunier, F., Calvo, R.W., 2013. Bike sharing systems: Solving the static rebalancing problem. Discrete Optim. 10 (2), 120–146.

Chen, P.-C., Hsieh, H.-Y., Su, K.-W., Sigalingging, X.K., Chen, Y.-R., Leu, J.-S., 2020. Predicting station level demand in a bike-sharing system using recurrent neural networks. IET Intell. Transp. Syst. 14 (6), 554–561.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., Bengio, Y., 2016. A recurrent latent variable model for sequential data. In: Conf. on Neural Information Processing Systems.

Citi Bike, 2021. System data. https://www.citibikenyc.com/system-data.

Datner, S., Raviv, T., Tzur, M., Chemla, D., 2019. Setting inventory levels in a bike sharing network. Transp. Sci. 53 (1), 62–76.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S., 2014. The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. Omega 45, 7–19.

DeMaio, P., 2009. Bike-sharing: History, impacts, models of provision, and future. J. Public Transp. 12 (4), 3.

Du, Y., Deng, F., Liao, F., 2019. A model framework for discovering the spatio-temporal usage patterns of public free-floating bike-sharing system. Transp. Res. C 103, 39–55.

Durbin, J., Koopman, S.J., 2001. Time Series Analysis By State Space Methods. Oxford University Press.

Elmachtoub, A.N., Grigas, P., 2021. Smart "predict, then optimize". Manage. Sci. Forthcoming.

Erdoğan, G., Laporte, G., Calvo, R.W., 2014. The static bicycle relocation problem with demand intervals. European J. Oper. Res. 238 (2), 451–457.

Eren, E., Uz, V.E., 2020. A review on bike-sharing: The factors affecting bike-sharing demand. Sustainable Cities Soc. 54, 101882.

Fabius, O., van Amersfoort, J.R., 2015. Variational recurrent auto-encoders. In: Workshop in International Conference on Learning Representations.

Fournier, N., Christofa, E., Knodler Jr., M.A., 2017. A sinusoidal model for seasonal bicycle demand estimation. Transp. Res. D Transp. Environ. 50, 154–169.

Frade, I., Ribeiro, A., 2015. Bike-sharing stations: A maximal covering location approach. Transp. Res. A Policy Pract. 82, 216–227.

Freund, D., Henderson, S.G., O'Mahony, E., Shmoys, D.B., 2019. Analytics and bikes: Riding tandem with motivate to improve mobility. INFORMS J. Appl. Anal. 49 (5), 310–323.

Gammelli, D., Peled, I., Rodrigues, F., Pacino, D., Kurtaran, H.A., Pereira, F.C., 2020a. Estimating latent demand of shared mobility through censored Gaussian processes. Transp. Res. C 120, 102775.

Gammelli, D., Rolsted, P.K., Pacino, D., Rodrigues, F., 2020b. Generalized multi-output Gaussian process censored regression. https://arxiv.org/abs/2009.04822.

Guo, R., Jiang, Z., Huang, J., Tao, J., Wang, C., Li, J., Chen, L., 2019. BikeNet: Accurate bike demand prediction using graph neural networks for station rebalancing. In: 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI). pp. 686–693.

Ho, S.C., Szeto, W., 2017. A hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem. Transp. Res. B 95, 340–363.

Jian, N., Freund, D., Wiberg, H.M., Henderson, S.G., 2016. Simulation optimization for a large-scale bike-sharing system. In: 2016 Winter Simulation Conference. WSC, pp. 602–613.

Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K., 1999. An introduction to variational methods for graphical models. Mach. Lang. 37 (2), 183–233.

Kadri, A.A., Kacem, I., Labadi, K., 2016. A branch-and-bound algorithm for solving the static rebalancing problem in bicycle-sharing systems. Comput. Ind. Eng. 95, 41–52.

Kingma, D.P., Welling, M., 2014. Auto-encoding variational Bayes. In: 2nd International Conference on Learning Representations. ICLR2014.

Kourentzes, N., Trapero, J.R., Barrow, D.K., 2020. Optimising forecasting models for inventory planning. Int. J. Prod. Econ. 225, 107597.

Krishnan, R.G., Shalit, U., Sontag, D., 2016. Deep Kalman filters. In: Conf. on Neural Information Processing Systems.

Laporte, G., Meunier, F., Calvo, R.W., 2015. Shared mobility systems. 4OR 13 (4), 341–360.

Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transp. Res. C 97, 258–276.

Maggioni, F., Cagnolari, M., Bertazzi, L., Wallace, S.W., 2019. Stochastic optimization models for a bike-sharing problem with transshipment. European J. Oper. Res. 276 (1), 272–283.

Meddin, R., DeMaio, P., O'Brien, O., Rabello, R., Yu, C., Seamon, J., Benicchio, T., Han, D., Mason, J., 2021. The meddin bike-sharing world map. https://bikesharingworldmap.com.

Nair, R., Miller-Hooks, E., 2011. Fleet management for vehicle sharing operations. Transp. Sci. 45 (4), 524–540.

Nair, R., Miller-Hooks, E., Hampshire, R.C., Bušić, A., 2013. Large-scale vehicle sharing systems: Analysis of vélib'. Int. J. Sustain. Transp. 7 (1), 85–106.

Negahban, A., 2019. Simulation-based estimation of the real demand in bike-sharing systems in the presence of censoring. European J. Oper. Res. 277 (1), 317–332.

O'Mahony, E., Shmoys, D., 2015. Data analysis and optimization for (Citi) bike sharing. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence.

Prak, D., Teunter, R., Syntetos, A., 2017. On the calculation of safety stocks when demand is forecasted. European J. Oper. Res. 256 (2), 454–461.

Rangapuram, S.S., Seeger, M.W., Gasthaus, J., Stella, L., Wang, Y., Januschowski, T., 2018. Deep state space models for time series forecasting. In: Conf. on Neural Information Processing Systems.

Raviv, T., Kolka, O., 2013. Optimal inventory management of a bike-sharing station. IIE Trans. 45 (10), 1077–1093.

Raviv, T., Tzur, M., Forma, I.A., 2013. Static repositioning in a bike-sharing system: Models and solution approaches. EURO J. Transp. Logist. 2 (3), 187–229.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic backpropagation and approximate inference in deep generative models. In: Proceedings of the 31st International Conference on Machine Learning. pp. 1278–1286.

Richter, F., 2018. The global rise of bike-sharing. https://www.statista.com/chart/13483/bike-sharing-programs.

Rixey, R.A., 2013. Station-level forecasting of bikesharing ridership: Station network effects in three US systems. Transp. Res. Rec. 2387 (1), 46–55.

Ross, S.M., 2014. Introduction to Probability Models. Academic Press.

Rossow, W., Walker, A., Golea, V., Knapp, K.R., Young, A., Inamdar, A., Hankins, B., 2016. International satellite cloud climatology project climate data record, H-series 2018 NOAA national centers for environmental information. In: National Centers for Environmental Information. NESDIS, NOAA, U.S. Department of Commerce.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36 (3), 1181–1191.

Sathishkumar, V.E., Cho, Y., 2020. Season wise bike sharing demand analysis using random forest algorithm. Comput. Intell. 2020, 1–26.

Schuijbroek, J., Hampshire, R., van Hoeve, W.-J., 2017. Inventory rebalancing and vehicle routing in bike sharing systems. European J. Oper. Res. 257 (3), 992–1004.

Shaheen, S.A., Guzman, S., Zhang, H., 2010. Bikesharing in europe, the americas, and Asia: Past, present, and future. Transp. Res. Rec. 2143 (1), 159–167.

Shui, C., Szeto, W., 2020. A review of bicycle-sharing service planning problems. Transp. Res. C 117, 102648.

Sohrabi, S., Paleti, R., Balan, L., Cetin, M., 2020. Real-time prediction of public bike sharing system demand using generalized extreme value count model. Transp. Res. A Policy Pract. 133, 325–336.

Syntetos, A., Babai, M., Davies, J., Stephenson, D., 2010. Forecasting and stock control: A study in a wholesaling context. Int. J. Prod. Econ. 127 (1), 103–111.

Szeto, W., Liu, Y., Ho, S.C., 2016. Chemical reaction optimization for solving a static bike repositioning problem. Transp. Res. D Transp. Environ. 47, 104–135.

Tian, Z., Zhou, J., Szeto, W., Tian, L., Zhang, W., 2020. The rebalancing of bike-sharing system under flow-type task window. Transp. Res. C 112, 1–27.

Tratar, L.F., 2010. Joint optimisation of demand forecasting and stock control parameters. Int. J. Prod. Econ. 127 (1), 173–179.

Vogel, P., Saavedra, B.A.N., Mattfeld, D.C., 2014. A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In: Hybrid Metaheuristics: 9th International Workshop. pp. 16–29.

Wang, B., Kim, I., 2018. Short-term prediction for bike-sharing service using machine learning. Transp. Res. Procedia 34, 171–178.

Wang, Y., Szeto, W., 2021. An enhanced artificial bee colony algorithm for the green bike repositioning problem with broken bikes. Transp. Res. C 125, 102895.

Warrington, J., Ruchti, D., 2019. Two-stage stochastic approximation for dynamic rebalancing of shared mobility systems. Transp. Res. C 104, 110–134.

Xu, C., Ji, J., Liu, P., 2018. The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets. Transp. Res. C 95, 47–60.

Yang, Z., Hu, J., Shu, Y., Cheng, P., Chen, J., Moscibroda, T., 2016. Mobility modeling and prediction in bike-sharing systems. In: MobiSys '16: Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. pp. 165–178.

Zhang, C., Bütepage, J., Kjellström, H., Mandt, S., 2018. Advances in variational inference. IEEE Trans. Pattern Anal. Mach. Intell. 41 (8), 2008–2026.

Zhang, J., Meng, M., Wong, Y.D., Ieromonachou, P., Wang, D.Z., 2021. A data-driven dynamic repositioning model in bicycle-sharing systems. Int. J. Prod. Econ. 231, 107909.