TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
2020, Vol. 2674(9) 820–831
© National Academy of Sciences:
Transportation Research Board 2020
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0361198120932166
journals.sagepub.com/home/trr
SAGE

Predicting Commercial Vehicle Parking Duration using Generative Adversarial Multiple Imputation Networks

Raymond Low[1], Zeynep Duygu Tekler[2], and Lynette Cheah[1]

Abstract
As the world rapidly urbanizes in pace with economic growth, the rising demand for products and services in cities is putting a strain on the existing road infrastructure, leading to traffic congestion and other negative externalities. To mitigate the impacts of freight movement within commercial areas, city planners have begun focusing their attention on the parking behaviors of commercial vehicles. Unfortunately, there is a general lack of information on such activities because of the heterogeneity of practices and the complex nature of urban goods movement. Furthermore, field surveys and observations of truck parking behavior are often faced with significant challenges, resulting in the collection of sparse and incomplete data. The objective of this study is to develop a regression model to predict the parking duration of commercial vehicles at the loading bays of retail malls and identify significant factors that contribute to this dwell time. The dataset used in this study originates from a truck parking and observation survey conducted at the loading bays of nine retail malls in Singapore, containing information about the trucks' and drivers' activities. However, because of the presence of incomplete fields found in the dataset, the authors propose the use of a generative adversarial multiple imputation networks algorithm to impute the incomplete fields before developing the regression model using the imputed dataset. Through the parking duration model, the activity type, parking location, and volume of goods delivered (or picked up) were identified as significant features influencing vehicle dwell time, corroborating with findings in the literature.

As the world rapidly urbanizes in pace with economic growth, major cities are becoming increasingly dense with large urban freight traffic generators such as retail malls producing and attracting many daily truck trips to the area (*1*). The constant flow of freight activities concentrated at these localized sites leads to the issue of traffic congestion which may propagate to other parts of the road network, leading to system gridlocks and other negative externalities (*2*). With the continuing growth in demand for retail products and services, city planners and urban authorities have begun shifting their attention to the parking behaviors of commercial vehicles (*3*) and exploring various logistics initiatives aimed at reducing freight-driven congestion (*4–7*).

The general lack of information on the conduct of logistics delivery activities, caused by the heterogeneity of practices between multiple stakeholders and the complex nature of freight transportation, has led to a limited understanding of the factors that contribute to the dwell time of commercial vehicles and the resulting congestion issues caused (*8*). As a result, only a handful of studies have attempted to develop duration models to explore the factors influencing commercial vehicle parking duration (*9, 10*). In an attempt to shed light on this growing problem, past field surveys conducted at the loading bays of urban retail malls are often faced with significant challenges, while relying heavily on traditional data collection methods such as observational studies and on-site interviews (*10*). As these methods are often labor-intensive, error-prone, and subject to the consent of the interviewee, this often leads to the collection of sparse and incomplete data, thus limiting the validity of the post-data analysis conducted.

The issue with incomplete and sparse data is not just limited to freight studies but is also encountered in other

[1]Engineering Systems and Design, Singapore University of Technology and Design, Singapore
[2]Engineering Product Development, Singapore University of Technology and Design, Singapore

Corresponding Author:
Raymond Low, raymond_low@mymail.sutd.edu.sg

fields such as the medical and social studies where participants may choose to omit certain personal information in their survey response because of privacy issues (*11–13*). Therefore, various imputation methods have been proposed in the past by researchers to deal with the issue of incomplete data to facilitate a more meaningful analysis.

García-Laencina et al. (*14*) attempted to categorize the different methods of data imputation into two groups, whereby the first group involved the use of statistical analysis to perform imputation, while the second group relied on machine learning approaches. Out of all of the statistical methods of dealing with missing data, one of the most successful methods was that proposed by Rubin (*15*) with the concept of multiple imputation (MI). In MI, the procedure attempts to replace each missing component with a set of possible values to capture accurately the variability in the feature containing missing fields. By generating multiple completed datasets from the original dataset, the former can be analyzed using standard methods before combining their results for further inference. A particular imputation method that uses the concept of MI is the multivariate imputation by chain equations (MICE) algorithm (*16*). In MICE, a series of regression models are used to impute the missing data sequentially by conditioning on the other variables that are both observed initially and previously imputed. This process is repeated in a round-robin fashion over multiple iterations until the parameters governing the imputation converges.

On the other hand, imputation methods based on machine learning approaches generally consist of a data-driven approach to perform estimations of the missing components based on the observed data. Under this category, the most popular approach is the K-nearest neighbor (*K-NN*) method whereby a missing component is estimated from a set of its K-nearest neighbors (determined according to a distance metric) containing the complete feature set. Two other approaches that fall under this category are the multi-layer perceptron imputation (*17*) and the autoencoder (*18*). Both of these approaches use a similar idea of performing imputation whereby a multi-layer network is trained on a complete dataset with the observed data introduced into the network as input features and produces the missing data as output. Different loss functions are selected to ensure that the distribution of the imputed values approaches the true distribution of the missing data. The limitation of these approaches is that they need to be trained initially on a complete dataset which may not be available. With the recent successes in the application of generative adversarial networks (GANs) to many real-life problems, Yoon et al. (*19*) proposed the generative adversarial imputation network (GAIN) algorithm to overcome these limitations encountered by other imputation approaches. The GAIN algorithm works by training two neural networks, a discriminator and a generator, pitted against each other in an adversarial relationship. By passing the incomplete data vector into the generator, imputation is performed based on the observed components to output a complete dataset. This dataset is subsequently passed into the discriminator where it will attempt to differentiate between the observed components against those that are imputed. As the objective of the generator is to impute the missing components such that the discriminator is unable to differentiate between the imputed and the observed components, both neural networks possess opposing objective functions. A theoretical analysis has also been conducted to prove that the generator can replicate the joint distribution of the original data. This approach has been tested on various benchmarking datasets and was shown to outperform many state-of-the-art imputation methods.

The dataset used in this study originates from a truck parking and observation survey conducted at the loading bays of nine urban retail malls in Singapore, containing visual information about the delivery activity as well as the drivers' delivery patterns. Given the presence of incomplete fields found in the dataset, the primary objective of this study is to develop a regression model to predict the parking duration of each commercial vehicle using the incomplete data and to identify significant factors related to dwell time during delivery activities. The regression model is developed using a two-step process whereby the presence of incomplete fields found in the dataset is addressed during the imputation step through the introduction of a generative adversarial multiple imputation networks (GAMIN) algorithm. The GAMIN algorithm is an extension of the GAIN algorithm proposed by (*19*), which reduces the implementation complexity and allows for MI. By generating multiple imputed versions of the original dataset, these datasets will be passed through separate regression models in the regression step to produce multiple predictions and will be combined through averaging to output the final prediction.

Through this study, the significant factors related to the dwell time of commercial vehicles can be identified, allowing city planners and building management to review the effectiveness of existing and future logistic management schemes. Future extensions of this study can also include the proposal of more effective parking management policies at existing loading bays to lessen the impacts of congestion during delivery peak hours.

Data Description

The dataset used in this study originates from a commercial vehicle parking and observation survey that was conducted at nine urban retail malls in Singapore over 12 separate weekdays between 2015 and 2018 (*10*). For the first two urban retail malls, a combination of road-side video recordings, a loading bay observation survey, and electronic parking records was used to capture a comprehensive view of the activities occurring at both retail malls under observation.

Road-Side Video Recordings

Video cameras were placed at several strategic locations around the shopping mall to capture the flow of delivery traffic. These locations include the entrance and exit of the service road as well as the entrances and exits to various parking facilities such as the loading bay and the customer car park. The video recordings were subsequently post-processed using a license plate recognition algorithm to create a sequence of timestamps describing different phases of each delivery. These timestamps include the time of arrival and departure from the retail mall, the amount of time spent queuing at the service road, as well as the amount of time spent parked at the loading bay.

Driver Survey and Vehicle Observation

During the conduct of the driver survey, surveyors were stationed at various parking locations around the retail mall, such as the customer car park, the loading bay, and on the streets to capture illegal on-street parking. The surveyors were also trained to observe and record specific details about the activity conducted, including the activity type, commodity type, volume of goods picked up or delivered, vehicle type, and vehicle stop duration, among other visual information. Next, the surveyors were instructed to approach the delivery crew to conduct a short face-to-face interview to gain more information about their delivery patterns. This information includes the number of tours made daily, the number of stores they serve in the retail mall, as well as the number of retail malls they serve in the vicinity among other details. However, because of their busy schedules, it was not uncommon for the delivery crew to refuse to participate in the interview portion of the survey, resulting in incomplete fields found in the dataset.

Electronic Parking Records

The parking facilities in Singapore's retail malls are also equipped with an electronic gantry system that recognizes and records the entry and exit times of every vehicle via its in-vehicle unit. Vehicle owners will be charged based on the amount of time spend in these parking facilities when exiting through the gantry. By accessing these electronic records, it is possible to obtain detailed information about the vehicles' arrival and departure times, parking duration, and parking location (i.e., customer car park or loading bay).

For the remaining seven urban retail malls, data collection was mainly concentrated at the loading bay where only the driver survey and vehicle observation were conducted. Table 1 presents a short description of each feature captured in the dataset, together with its respective missing rate.

Methodology

Because of the presence of incomplete fields found in the dataset, the development of the parking duration model follows a two-step approach. The first step involves the imputation of missing values found in the dataset by implementing the GAMIN algorithm to obtain a complete dataset. The dataset is subsequently passed through a regression algorithm to develop the final parking duration model in a supervised fashion.

Notation Definition

The following set of notation will follow the same notation used by Yoon et al. (*19*) for ease of comparison between the two algorithms for the interested reader.

Consider a d-dimensional space $\chi = \chi_1 \times \ldots \times \chi_d$ where d represents the number of features in the dataset. Suppose that data vector $\mathrm{X} = (X_1, \ldots, X_d)$ is a random variable containing either continuous or binary values in χ, while mask vector $\mathrm{M} = (M_1, \ldots, M_d)$ is a random variable taking values in $\{0,1\}^d$. For each $i \in \{1, \ldots, d\}$, $\tilde{\chi}_i = \chi_i \cup \{*\}$ where * represents an unobserved value not found in χ_i. Let $\tilde{\chi} = \tilde{\chi}_1 \times \ldots \times \tilde{\chi}_d$ where we define a new variable $\tilde{\mathrm{X}} = \left(\tilde{X}_1, \ldots, \tilde{X}_d\right) \in \tilde{\chi}$ based on Equation 1:

$$\tilde{X}_i = \begin{cases} X_i, if\ M_i = 1 \\ *, \text{otherwise} \end{cases} \tag{1}$$

Therefore mask matrix M indicates the components of X that can be observed.

Throughout the remainder of the paper, lower-case letters will denote the realization of a random variable. For instance, n independent and identically distributed (i.i.d.) copies of $\tilde{\mathrm{X}}$ will be denoted by $\tilde{x}^1, \ldots, \tilde{x}^n$, while the dataset $\mathcal{D}$ is denoted by $\{\tilde{x}^i, m^i\}_{i=1}^n$, where m^i is the mask matrix corresponding to $\tilde{x}^i$.

Imputation: GAMIN Algorithm

The imputation of the missing values found in dataset $\mathcal{D}$ is achieved by implementing the GAMIN algorithm, which is an extension of the GAIN algorithm by improving the latter's implementation and introducing the concept of MI into the algorithm. The GAMIN algorithm, similar to the GAIN algorithm, seeks to impute the

Table 1. Description of Data Features and Their Respective Missing Rates

Feature	Missing rate	Description	Data source
Mall	0.000	The retail mall where the activity was conducted. Possible values range from *Mall A* to *Mall I* to represent the nine urban retail malls.	na
Entry hour	0.000	The time where the vehicle arrives at the mall (rounded to the nearest hour).	Video recordings, observations, parking records
Parking location	0.002	The location where the vehicle is parked when conducting activities. Possible values include *Car Park, Loading Bay*, and *Street.*	Video recordings, observations, parking records
Vehicle type	0.051	Possible values include *Truck* or *Van.*	Observations
Activity type	0.424	The type of activity conducted. Possible values include *Deliver, Pick up, Deliver & Pick up*, and *Service.*	Observations
Refrigerated	0.174	Is the commercial vehicle refrigerated? Possible values include *Yes* and *No.*	Observations
Commodity type	0.528	The type of goods delivered. Possible values include *Clothing and Accessories, Cosmetics and Cleaning, Electronics*, etc.	Observations
Payload (%)	0.456	How full was the commercial vehicle when first arriving at the retail mall? Possible values include *0–25%, 25–50%, 50–75%*, and *75–100%.*	Observations
Initial payload (%)	0.737	How full was the commercial vehicle at the beginning of the tour? Possible values include *0–25%, 25–50%, 50–75%*, and *75–100%.*	Driver survey
Delivery volume (m^3)	0.473	The amount of goods delivered to the retail mall.	Observations
Pickup volume (m^3)	0.550	The amount of goods picked up from the retail mall.	Observations
Single/bundle	0.053	*Single* indicates that only a single type of commodity is being delivered, picked up, or both. *Bundle* indicates that different types of commodities are being delivered, picked up, or both.	Observations
Number of workers	0.563	The number of workers helping out with the activity.	Observations
Store count	0.654	The number of stores served by the delivery crew in that retail mall.	Driver survey
Mall count	0.934	The number of malls served by the delivery crew in the vicinity.	Driver survey
Employer	0.639	The delivery crew's employer. Possible values include *Carrier, Receiver, Retailer, Shipper, Supplier*, and *Transport Provider.*	Driver survey
Number of tours	0.839	The number of tours made by the delivery crew on a daily basis.	Driver survey
Is service vehicle	0.974	Is the vehicle a service vehicle? Possible values include *Yes* and *No.*	Observation
Number of stops/tour	0.980	The average number of stops made during each tour.	Driver survey
System occupancy	0.233	The total number of commercial vehicles in the retail mall, including those that are queuing up to make a delivery.	Video recordings, observations, parking records
Parking duration (min)	0.000	The total amount of time the commercial vehicle is parked while the delivery crew is conducting their activities.	Video recordings, observations, parking records

Note: na = not applicable.

missing values in each $\tilde{x}^i$ by generating complete data vectors according to the distribution $P(X|\tilde{X} = \tilde{x}^i)$. This imputation step is achieved by pitting a generator network against a discriminator network in an adversarial relationship, following the architecture of GANs.

Generator. The generator network G takes in realizations of $\tilde{X}$, M, and Z as input to generate a complete data vector $\bar{X}$. Let $G : \tilde{\chi} \times \{0,1\}^d \times [0,1]^d \rightarrow \chi$ be the generator function, and Z = $(Z_1, \ldots, Z_d)$ be a d-dimensional noise matrix. The random variables $\bar{X}, \hat{X} \in \chi$ are therefore defined in Equations 2 and 3 as:

$$\bar{X} = G\big(M \odot \tilde{X} + (1 - M) \odot Z\big) \tag{2}$$

$$\hat{X} = M \odot \tilde{X} + (1 - M) \odot \bar{X} \tag{3}$$

where the symbol $\odot$ denotes element-wise multiplication. $\hat{X}$ corresponds to the imputed data vector which is obtained by taking $\tilde{X}$ and replacing each missing value *. with its corresponding value in $\bar{X}$.

To introduce the concept of MI into the algorithm, multiple noise matrices $Z_j, j = 1, \ldots, J$ will be generated from the same noise distribution where J is the total number of imputation that the user is interested in generating from the same dataset $\mathcal{D}$. Each set of $\left(\tilde{X}, M, Z_j\right)$ will be passed into the generator as input to produce the following output as defined in Equations 2a and 3a:

$$\bar{X}_j = G\left(M \odot \tilde{X} + (1 - M) \odot Z_j\right) \tag{2a}$$

$$\hat{X}_j = M \odot \tilde{X} + (1 - M) \odot \bar{X}_j \tag{3a}$$

Note that for each noise matrix Z_j, the same M and $\tilde{X}$ are used as input to produce $\bar{X}_j$ and $\hat{X}_j$.

Discriminator. By passing the imputed data vector $\hat{X}_j$ into a discriminator network D, the objective of discriminator D is to accurately distinguish between the components that are initially observed against those components that are imputed by generator. This process is equivalent to predicting the mask matrix M, which can be obtained from the original data vector X. The formal representation of discriminator D is a function $D : \chi \rightarrow [0, 1]^d$ with the i^{th} component of $D\left(\hat{X}_j\right)$ corresponding to the probability that the i^{th} component of $\hat{X}_j$ was observed and not imputed by generator G.

Hint Mechanism. Yoon et al. (*19*) proposed in the original paper that it was necessary to pass a random variable H as an additional input into discriminator D to ensure that generator G reproduces a single distribution that will be optimal with respect to D. This matrix H, which takes on values from the space $\mathcal{H}$, serves as a hint mechanism as it contains information about M through the distribution $H|M = m$. With the introduction of H into discriminator D, the function becomes $D : \chi \times \mathcal{H} \rightarrow [0, 1]^d$, where the i^{th} component of $D\left(\hat{X}, H\right)$ corresponds to the probability that the i^{th} component of $\hat{X}$ was observed, conditional on $\hat{X}$ and H. By defining H differently, the authors were able to control the amount of information passed to D.

However, it will be shown in the subsequent sections that the introduction of H into discriminator D causes its performance to converge too rapidly and therefore affects the training process of generator G. Instead, the introduction of a noise matrix N is propose to limit the performance of discriminator D which will, in turn, improve the training process of generator G and its ability to impute missing values.

Objective Function. As the objective of discriminator D is to identify accurately the components that are imputed against those that are observed, D is trained to maximize the probability of correctly predicting M through Equations 4 and 5:

$$\min -\mathcal{L}_D\left(M, \hat{M}\right) \tag{4}$$

$$\text{where } \mathcal{L}_D\left(M, \hat{M}\right) = M\log\left(\hat{M}\right) + (1 - M)\log\left(1 - \hat{M}\right) \tag{5}$$

On the other hand, generator G is trained to fool discriminator D into thinking that an imputed component is observed, while at the same time output values that closely replicate the components that were originally observed. Therefore, the objective function of generator G is broken into two parts.

The first loss function $\mathcal{L}_G : \{0, 1\}^d \times [0, 1]^d \rightarrow \mathbb{R}$ is used to quantify the ability of generator G to fool the discriminator into misclassifying the imputed components as observed and generate imputed values that replicate the joint distribution of the dataset. This is given by Equation 6:

$$\mathcal{L}_G\left(M, \hat{M}\right) = -(1 - M)\log\left(\hat{M}\right) \tag{6}$$

While the second loss function, $\mathcal{L}_M : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is used to quantify the ability of generator G to reconstruct the values initially observed. This loss function is similar to artificially removing values from the dataset and evaluating how well these values can be recovered. This is given by Equation 7:

$$\mathcal{L}_M\left(\tilde{X}, \bar{X}\right) = ML_M\left(\tilde{X}, \bar{X}\right) \tag{7}$$

In the original GAIN algorithm, $L_M\left(\tilde{X}, \bar{X}\right)$ is defined differently depending on whether X contains continuous values or binary values.

$$L_M\left(\tilde{X}, \bar{X}\right) = \begin{cases} \left(\bar{X} - \tilde{X}\right)^2, if\ X \text{ is continuous} \\ -\tilde{X}\log\left(\bar{X}\right), if\ X \text{ is binary} \end{cases} \tag{8}$$

In the case where X is continuous, the scale of the values between the first loss function $\mathcal{L}_G\left(M, \hat{M}\right)$ and the second loss function $\mathcal{L}_M\left(\tilde{X}, \bar{X}\right)$ would be different, making it necessary to introduce a hyper-parameter α when combining both loss functions. This combination produces the following weighted sum loss function as shown in Equation 9.

$$\min \mathcal{L}_G\left(M, \hat{M}\right) + \alpha\mathcal{L}_M\left(\tilde{X}, \bar{X}\right) \tag{9}$$

However, it is challenging to define an appropriate α value in this case, as it changes depending on the number of missing components found in the dataset (or batch if training is conducted in batches). Failing to select the appropriate α value will cause the weight of one of the loss functions to greatly exceed the other, causing the minimization of the one of the loss functions to be prioritized over the other. Therefore, we will assume that X only contains binary values, allowing us to redefine

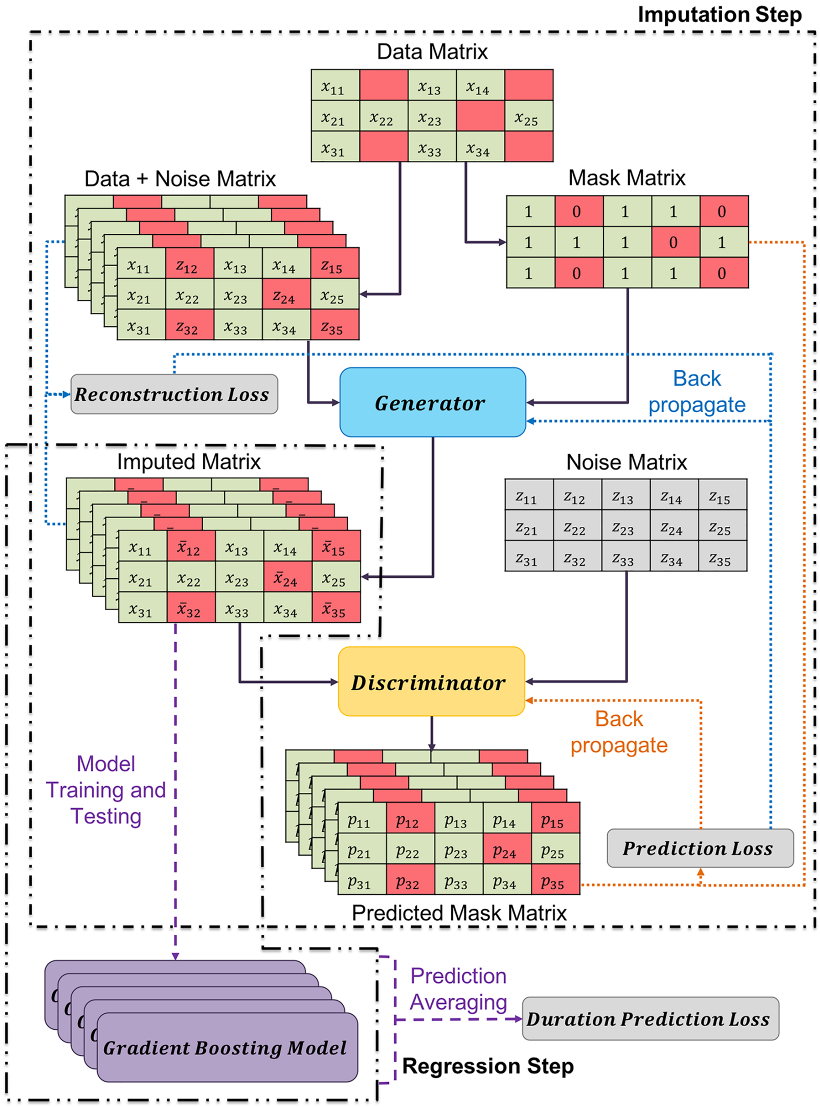
Imputation Step
Data Matrix
x_{11} x_{13} x_{14}
x_{21} x_{22} x_{23} x_{25}
x_{31} x_{33} x_{34}
Data + Noise Matrix
x_{11} z_{12} x_{13} x_{14} z_{15}
x_{21} x_{22} x_{23} z_{24} x_{25}
x_{31} z_{32} x_{33} x_{34} z_{35}
Mask Matrix
1 0 1 1 0
1 1 1 0 1
1 0 1 1 0
Reconstruction Loss
Generator
Back propagate
Imputed Matrix
x_{11} $\bar{x}_{12}$ x_{13} x_{14} $\bar{x}_{15}$
x_{21} x_{22} x_{23} $\bar{x}_{24}$ x_{25}
x_{31} $\bar{x}_{32}$ x_{33} x_{34} $\bar{x}_{35}$
Noise Matrix
z_{11} z_{12} z_{13} z_{14} z_{15}
z_{21} z_{22} z_{23} z_{24} z_{25}
z_{31} z_{32} z_{33} z_{34} z_{35}
Discriminator
Back propagate
Model Training and Testing
p_{11} p_{12} p_{13} p_{14} p_{15}
p_{21} p_{22} p_{23} p_{24} p_{25}
p_{31} p_{32} p_{33} p_{34} p_{35}
Prediction Loss
Predicted Mask Matrix
Gradient Boosting Model
Prediction Averaging
Duration Prediction Loss
Regression Step

Figure 1. Graphical representation of methodology of this study.

the second loss function, $\mathcal{L}_M : [0,1]^d \times \{0,1\}^d \rightarrow \mathbb{R}$, and remove the α value from the final loss function (since α = 1 when X is binary) to give Equation 9a.

$$\min \mathcal{L}_G\left(M, \hat{M}\right) + \mathcal{L}_M\left(\tilde{X}, \bar{X}\right) \tag{9a}$$

Regression

The imputed data vector $\hat{X}$ can be subsequently passed into a machine learning model together with its corresponding parking duration Y to perform model training in a supervised fashion. With the inclusion of MI in the proposed imputation algorithm, multiple imputed data vectors $\hat{X}_i$ can be obtained from the same data vector X and passed into separate regression models to generate different predictions γ_i. These predictions are ultimately combined via averaging to produce the final prediction γ. A graphical representation of this study's methodology can be found in Figure 1.

Implementation

Data Preprocessing

Before the dataset was passed through the GAMIN algorithm to perform the imputation step, several data preprocessing steps were taken in the following order. By plotting the distribution of the parking duration, it is observed from Figure 2 that the dataset suffers from a

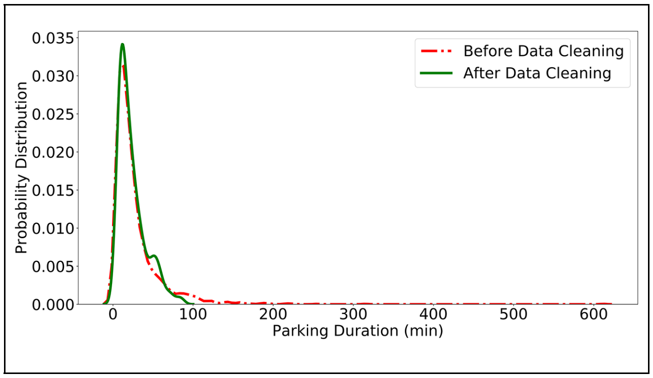

Figure 2. Distribution of parking duration before and after data cleaning.

"long tail" problem whereby a tiny percentage of the vehicles reported unusually large values for their parking durations. These extreme cases, if left in the dataset, could potentially prevent generator G from learning the true distribution of the missing features, thereby producing a prediction model that learns from incorrect data and therefore fails to generalize well to future unseen data. Therefore, to address this potential problem, a cutoff boundary was set such that any delivery with a parking duration that falls beyond two standard deviations from the mean will be considered as an outlier and removed from the dataset. In the end, 89 instances (4.30%) were removed from the original dataset, resulting in a final dataset of 1,983 delivery entries.

Next, given that the GAMIN algorithm assumes that X contains only binary values, the numerical features found in the dataset (i.e., delivery volume, pickup volume, etc.) will be discretized into binary classes. This is achieved by calculating the first ($Q1$), second ($Q2$), and third ($Q3$) quantiles of each numerical feature, and grouping the values of a particular feature into the same class if it falls into any one of the following cases: (i) below $Q1$, (ii) between $Q1$ and $Q2$, (iii) between $Q2$ and $Q3$, or (iv) higher than $Q3$. The result is four discrete and similarly sized classes for each numerical feature. By ensuring that the distribution of each class is fairly even any class imbalance issue that biases generator G toward imputing the most frequently occurring classes during the imputation step is avoided. Finally, by performing one-hot-encoding on the resulting dataset, we end up with a data matrix X, which contains only binary values.

Although the implementation of the discretization step was deemed necessary to obey the assumption of the GAMIN algorithm, the second reason for discretizing the dataset stems from the data collection approach adopted in this study. Given that most of the features are obtained via traditional data collection approaches, the observation and response errors that come with the visual observation of the activity and face-to-face interviews can be reduced if these features take on a range of values (through discretization) instead of an absolute numerical value. Therefore, by reducing the precision of the numerical features, we are also attempting to improve the accuracy of the information captured in the dataset, which will improve the performance of the parking duration model.

Algorithm

The GAMIN algorithm follows the usual GAN approach of iteratively training the discriminator and generator by sampling mini-batches of size K from the dataset during each training iteration. By modeling the discriminator D and generator G as fully connected neural networks, the algorithm begins by optimizing discriminator D with a fixed generator G by computing a corresponding noise matrix $z(j)$ for each sample in the mini-batch $(\tilde{x}(j), m(j))$. Next, using $\tilde{x}(j)$, $m(j)$, and $z(j)$ as input, generator G produces a complete data matrix $\bar{x}(j)$ which will be combined with $\tilde{x}(j)$, for the observed components, to result in the imputed data matrix $\hat{x}(j)$. The parameters of discriminator D are adjusted during each training cycle to optimize its ability to differentiate between the imputed components against those originally observed.

Next, generator G is optimized against the updated discriminator D by using $\tilde{x}(j)$, $m(j)$, and $z(j)$ to calculate its loss function. This loss function is dependent on generator G's ability to fool discriminator D into thinking that the imputed components are observed, while at the same time generating values that replicate the originally observed components. This process is repeated until the performance of both discriminator D and generator G converges.

MI is achieved by generating multiple $z_i(j)$ values for the same mini-batch $(\tilde{x}(j), m(j))$ and passing it through generator G to produce I imputed data matrices $\hat{x}_i(j)$, where $i = 1, \ldots, I$ and I equal to the number of imputations carried out. By passing each imputed data matrix $\hat{x}_i(j)$ through a separate regression model, the predictions made by each regression model are combined through averaging to produce the final prediction. In this study, the gradient boosting algorithm (*20*) is selected, because of its superior predictive performance over the other regression algorithms tested, and allows us to calculate the importance score of each feature to identify the significant factors related to parking duration. The algorithm follows an iterative functional gradient descent approach that minimizes its loss function $\mathcal{L}_R(y_i, \gamma)$ by iteratively introducing base learners $b_m(\tilde{x}(j))$, defined based on the errors made by the current model $F_{m-1}(\tilde{x}(j))$ to boost model performance. Because of its robust performance, it has also been used in many other application areas such as activity recognition and indoor localization (*21–24*).

A preliminary implementation of the GAMIN algorithm described up to this point showed a rapid convergence in the performance of discriminator D within the first few iterations and prevents generator G from further improving its performance. This phenomenon is also commonly known as the vanishing gradient problem (*25*). Therefore, to address this particular issue, the following measures were introduced in the GAMIN algorithm to slow down the convergence rate of discriminator D. The first measure was to select an optimizer using a slower algorithm for discriminator D, such as the stochastic gradient descent optimizer, while the Adam optimizer was used to train generator G. The second measure involves allowing generator G to update its parameters T times for every time the parameters of discriminator D is updated. By setting different T values, this allows us to adjust the amount of advantage that is given to generator G during the training process. Finally, the last measure involves introducing a noise matrix N as input into discriminator D instead of the hint matrix H proposed by Yoon et al. (*19*) to further disrupt the training process of discriminator D. The pseudo-code for the final algorithm is presented in Algorithm 1.

Algorithm 1 Pseudo-code for GAMIN algorithm and regression step

```
(1) Training discriminator D and generator G using GAMIN algorithm
for number of training iterations do
    (1a) Optimize discriminator D
    Draw $K$ i.i.d. samples from the dataset $\{(\tilde{x}(j), m(j))\}_{j=1}^{K}$
    Draw $K$ i.i.d. samples $\{z(j)\}_{j=1}^{K}$ from $Z$
    Draw $K$ i.i.d. samples $\{n(j)\}_{j=1}^{K}$ from $N$
    for $j = 1, \ldots, K$ do
        $\bar{x}(j) \leftarrow G(m(j) \odot \tilde{x}(j) + (1 - m(j)) \odot z(j))$
        $\hat{x}(j) \leftarrow m(j) \odot \tilde{x}(j) + (1 - m(j)) \odot \bar{x}(j)$
    end for
    Update D using Stochastic Gradient Descent optimizer
        $\nabla_D - \sum_{j=1}^{K} \mathcal{L}_D(m(j), D(\hat{x}(j), n(j)))$

    (1b) Optimize generator G
    for $t = 1, \ldots, T$ do
        Draw $K$ i.i.d. samples from the dataset $\{(\tilde{x}_t(j), m_t(j))\}_{j=1}^{K}$
        Draw $K$ i.i.d. samples $\{z_t(j)\}_{j=1}^{K}$ from $Z$
        Draw $K$ i.i.d. samples $\{n_t(j)\}_{j=1}^{K}$ from $N$
        for $j = 1, \ldots, K$ do
            $\bar{x}_t(j) \leftarrow G(m_t(j) \odot \tilde{x}_t(j) + (1 - m_t(j)) \odot z_t(j))$
            $\hat{x}_t(j) \leftarrow m_t(j) \odot \tilde{x}_t(j) + (1 - m_t(j)) \odot \bar{x}_t(j)$
        end for
        Update $G$ using Adam optimizer
            $\nabla_G - \sum_{j=1}^{K} [\mathcal{L}_G(m_t(j), D(\hat{x}_t(j), n_t(j)))$
             $+ \mathcal{L}_M(\tilde{x}_t(j), \bar{x}_t(j), m_t(j))]$
end for

(2) Training the gradient boosting model
for $i = 1, \ldots, I$ do
    (2a) Imputation step
    Draw $Z_i$ from $Z$
    $\bar{X}_i \leftarrow G(M \odot \tilde{X} + (1 - M) \odot Z_i)$
    $\hat{X}_i \leftarrow M \odot \tilde{X} + (1 - M) \odot \bar{X}_i.$

    (2b) Regression step
    $F_0(\hat{X}_i) = argmin_{\gamma_i} \mathcal{L}_R(Y, \gamma_i)$
    for number of estimators do
        $F_m(\hat{X}_i) = F_{m-1}(\hat{X}_i) + argmin_{b_m \in B} \mathcal{L}_R(Y, F_{m-1}(\hat{X}_i) + b_m(\hat{X}_i))$
    end for
end for
$\gamma = \frac{\sum_{i=1}^{I} F_m(\hat{X}_i)}{I}$
```

Comparison of Results

The performance of the imputation algorithm is evaluated by randomly removing 10% of the initially observed information from the commercial vehicle parking and observation survey dataset and calculating the imputation error. Since X is assumed only to contain binary values, the imputation error will be calculated using a cross-entropy loss function, as shown in Equation 10. Furthermore, to ensure that an accurate regression model can be trained based on the imputed dataset, the mean absolute error (MAE) between the predicted parking duration γ and the true parking duration Y will be used as a secondary performance metric when evaluating the imputation algorithm (refer to Equation 11).

$$\mathcal{L}_{\text{imputation}} = -M_{\text{removed}} \tilde{X}_{\text{removed}} \log(\bar{X}_{\text{removed}}) \tag{10}$$

$$\text{MAE} = \frac{1}{P} \sum_{p=1}^{P} |\gamma_p - Y_P| \tag{11}$$

where M_{removed} is a mask matrix indicating the artificially dropped values, $\bar{X}_{\text{removed}}$ represents the imputed matrix for the dropped values, and P indicates the size of the test dataset.

The performance of the GAMIN algorithm is evaluated against other baseline imputation algorithms by passing the same training and test datasets through the K-NN algorithm, the MICE algorithm, as well as the original GAIN algorithm before using the imputed datasets to perform model development. Each gradient boosting model uses an identical set of hyperparameters to ensure a fair comparison between the different imputation methods. On top of the proposed GAMIN algorithm, GAMIN (Noise), we have also included the evaluation results for different variants of the GAMIN algorithm. The second variant, GAMIN (Hint), involves replacing the proposed noise matrix N, which is part of the input into discriminator D, with a hint mechanism H as recommended in the original GAIN algorithm (*19*). A third variation of the GAMIN algorithm, GAMIN (No Hint), involves removing the noise matrix N entirely such that the sole input into discriminator D is $\hat{X}_i$. The performance of each imputation method is captured by repeating the imputation step using

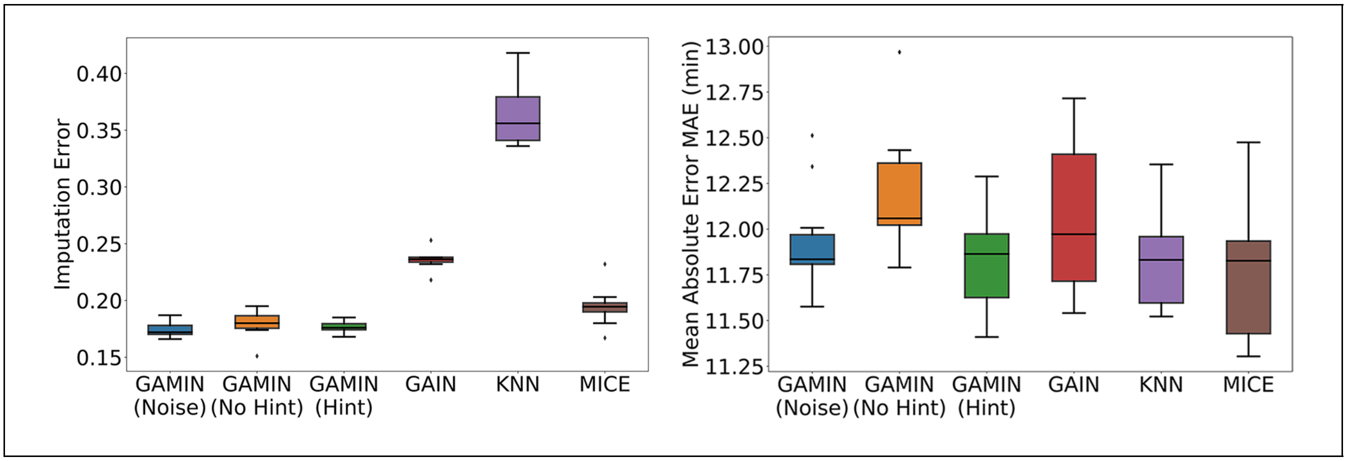

Figure 3. Performance comparison between different imputation methods. The figure on the left shows the imputation error of each method, where the lower the imputation error, the better the imputation performance. The figure on the right shows the performance of the resulting parking duration models that were developed based on the same dataset imputed using different imputation methods.
Note: GAMIN = generative adversarial multiple imputation networks; GAIN = generative adversarial imputation network; KNN = K-nearest neighbor; MICE = multivariate imputation by chain equations.

different pairs of training and test datasets and plotting the results, as in Figure 3.

As shown in Figure 3, the GAMIN (Noise) algorithm was able to outperform the baseline imputation algorithms, such as K-NN, MICE, and the original GAIN algorithm, by producing a lower imputation error but also producing a parking duration model that is more robust to changes in the training dataset. These results can be attributed to the GAMIN algorithm's better ability to represent the variability in the missing features through the generation of multiple completed datasets from the original dataset.

Furthermore, while a comparison of the resulting imputation error and predictive performance among different variants of the GAMIN algorithm did not produce a clear winner, especially between GAMIN (Noise) and GAMIN (Hint), GAMIN (Noise) was more robust to changes in the training dataset as demonstrated by its smaller variance in predictive performance. This result may be attributed to the introduction of noise matrix N into discriminator D, which disrupted its training process and increased its convergence time toward optimality. This statement is supported in Figure 4, which shows a slower convergence rate for discriminator D when using the GAMIN (Noise) algorithm, as compared with other variants of the algorithm. Because of discriminator D's slower convergence rate, this provides generator G with more time to update its parameters, thereby possibly producing a more stable generator G.

Practical Findings and Applications

After implementing the GAMIN (Noise) algorithm to address the issues of incomplete data and using the gradient boosting algorithm to develop the parking duration model based on the completed datasets, this section will explore the significant factors that are related to vehicle dwell time. This objective is achieved by assigning a numerical value, or feature importance score, to each input feature describing the value of a particular feature in constructing the regression model. In the case of a single decision tree, this feature importance score is defined by the amount of performance improvement achieved by each attribute split point, weighted by the number of samples that are affected by the split. Given that a gradient boosting model, which is made up of an ensemble of decision trees, is adopted in this study, the feature importance score is calculated by averaging over the scores from each decision tree used within the model (*26*).

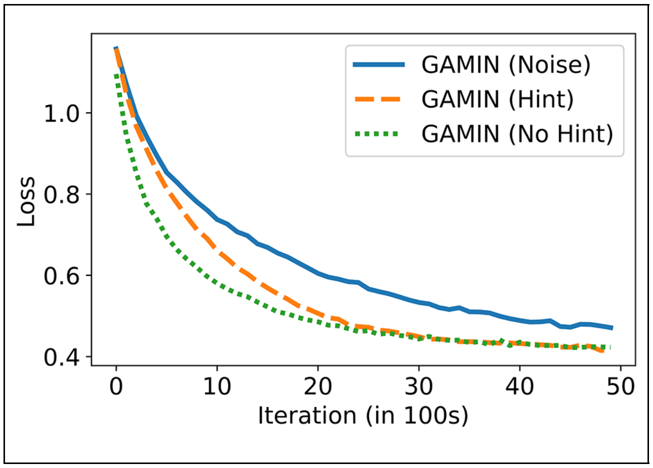

Figure 4. Convergence rate of discriminator *D* using different variants of the generative adversarial multiple imputation networks (GAMIN) algorithm.

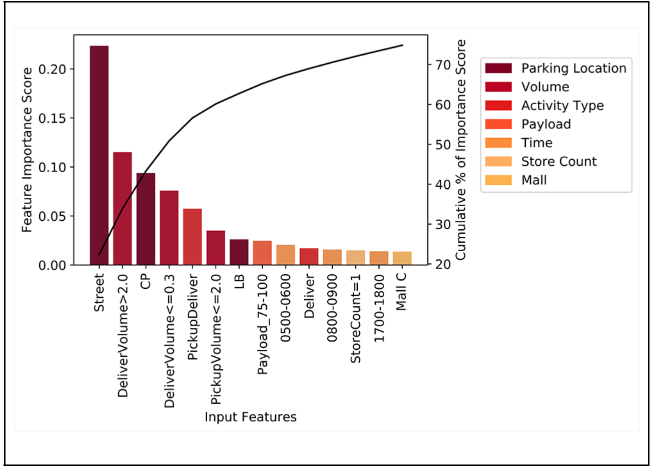

Figure 5. Pareto chart of the feature importance scores for the most significant input features.
Note: CP = customer car park; LB = loading bay.

By plotting the feature importance score of each input feature on a Pareto chart arranged in descending order, Figure 5 shows a Pareto chart of the most significant features in the parking duration model responsible for 75% of the total feature importance score (equals to 1.0). While the feature importance score of several of the features changes slightly when the algorithm is repeated on different training and test dataset pairs, features such as the parking location, volume of goods delivered (or picked up), and activity conducted consistently possess the highest feature importance scores among the other features considered in this study.

The results of this study corroborate the findings of Dalla Chiara and Cheah (*10*), which highlighted that the parking duration of commercial vehicles differs significantly depending on their parking location. More specifically, it was reported that the dwell times of commercial vehicles parked illegally along the streets are significantly lower than those parked in the customer car park and loading bay, with the vehicles in the loading bay reporting the most prolonged parking duration. This phenomenon was explained by mentioning that the driver was most likely aware of the expected amount of time that is necessary to complete the delivery. Therefore, in the case where a delivery can be completed within a short period, the delivery crew might choose to park illegally on the streets when performing their activities to avoid the queue at the loading bay and save time.

Parking duration is also found to have a positive correlation with the volume of goods delivered (0.382), with larger volumes of goods requiring a more extended amount of time to be unloaded from the vehicle, and subsequently delivered, and accounted for when the retail staff receive the delivery. A similar argument applies (0.105) when a large volume of goods is picked up by the driver, as an extended amount of time is taken to transport and load the goods onto the vehicle.

Thirdly, the activity type that the driver is conducting also provides valuable information about the dwell time of the commercial vehicle. The parking duration when the driver is conducting a *Deliver* activity or a *Pick up* activity takes an average of 18 min, with the former having a higher variance than the latter. When a *Deliver & Pick up* activity is conducted, the parking duration increases to an average of 32 min with the volume of goods that are being delivered and picked up further increasing the variance in vehicle dwell time. Finally, service vehicles parked at the loading bays have an average dwell time of approximately 17 min and a similarly high variance when compared to the commercial vehicles performing a *Deliver & Pick up* activity.

Based on the insights gained from this study, a practical application of this work for building managers is to implement a similar model based on the data collected from existing malls to determine the parking duration of each commercial vehicle arriving at the loading bay. By ascertaining the nature of the delivery activity that will be conducted (i.e., activity type, delivery volume) before the vehicle's arrival, vehicles with shorter delivery times can be assigned to express lots, while vehicles with longer delivery times are assigned to regular lots. An extension of this implementation will involve asking the carriers to provide information about their activities through a mobile application where the estimated dwell time will be passed through a scheduling system to assign a designated parking lot to each commercial vehicle. The objective of the scheduling system could be tuned to maximize the number of commercial vehicles passing through the system (i.e., throughput), thereby ensuring the efficient use of scarce parking resources and reducing congestion during delivery peak hours. Based on the type of parking lot assigned to each commercial vehicle, different parking fees can also be imposed to ensure system integrity. For instance, vehicles that are assigned to the express parking lots might not be charged for the first 30 min but parking fees would then increase exponentially with time beyond the grace period. This measure is aimed at preventing the delivery crew from underreporting their delivery information to enter the express lots. On the other hand, vehicles entering the regular parking lots will be charged with a parking fee that increases linearly with time to provide the delivery crew with more time to conduct their activities. Finally, the installation of dock levellers and belt conveyors at the regular parking lots can assist the delivery crew in loading and unloading their goods, thereby reducing the delivery time for each vehicle.

Conclusion

This study developed a regression model to predict the parking duration of commercial vehicles operating at the loading bays of urban retail malls. The dataset used in this study originates from a truck parking and observation survey in Singapore that contains information about the trucks' and drivers' activities. Because of the presence of incomplete fields found in the dataset, an imputation algorithm known as GAMIN was used to fill in the missing fields before developing a gradient boosting model using the imputed dataset. A comparison of the GAMIN algorithm with other baseline imputation methods such as K-NN, MICE, and GAIN showed that the GAMIN algorithm was able to generate datasets that developed better prediction models. Furthermore, by comparing the performance between different variants of the GAMIN algorithm, it was concluded that the introduction of a noise matrix N into discriminator D would improve the performance of the imputation algorithm. Finally, based on the parking duration model developed, activity type, parking location, and volume of goods delivered (or picked up) were identified as significant factors related to the parking duration of the commercial vehicle, which corroborates with findings in the literature. An extension of the current work would involve an in-depth study of the causal effects of each feature following the analysis conducted by McCaffrey et al. (*27*) whereby the propensity scores of different features were analyzed according to a gradient boosting regression algorithm. This information will be useful for city planners in the design of future loading bays and urban freight initiatives that are aimed at reducing congestion because of freight activities.

Acknowledgments

The authors thank Cheung Ngai-Man, Sun Xin, Giacomo Dalla Chiara, Rakhi Manohar Mepparambath, Gabriella Marie Ricart Surribas, IMDA, and the mall operators for supporting and facilitating with the data collection efforts.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: Raymond Low, Zeynep Duygu Tekler, Lynette Cheah; data collection: Lynette Cheah; analysis and interpretation of results: Raymond Low, Zeynep Duygu Tekler; draft manuscript preparation: Raymond Low, Lynette Cheah. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received the following financial support for the research, authorship, and/or publication of this article. This research is supported in part by the Infocomm Media Development Authority of Singapore (IMDA) and the Singapore-MIT Alliance for Research and Technology. Raymond Low is supported by the SUTD President's Graduate Fellowship from the Singapore Ministry of Education.

References

1. Jaller, M., X. C. Wang, and J. Holguin-Veras. Large Urban Freight Traffic Generators: Opportunities for City Logistics Initiatives. *Journal of Transport and Land Use*, Vol. 8, No. 1, 2015, pp. 51–67.
2. Browne, M., J. Allen, T. Nemoto, D. Patier, and J. Visser. Reducing Social and Environmental Impacts of Urban Freight Transport: A Review of Some Major Cities. *Procedia-Social and Behavioral Sciences*, Vol. 39, 2012, pp. 19–33.
3. Marcucci, E., V. Gatta, and L. Scaccia. Urban Freight, Parking and Pricing Policies: An Evaluation from a Transport Providers' Perspective. *Transportation Research Part A: Policy and Practice*, Vol. 74, 2015, pp. 239–249.
4. Browne, M., A. G. Woodburn, and J. Allen. Evaluating the Potential for Urban Consolidation Centres. *European Transport/Trasporti Europei*, Vol. 35, 2007, 46–63.
5. McLeod, F., and T. Cherrett. Loading Bay Booking and Control for Urban Freight. *International Journal of Logistics Research and Applications*, Vol. 14, No. 6, 2011, pp. 385–397.
6. Holguín-Veras, J., and I. Sánchez-Díaz. Freight Demand Management and the Potential of Receiver-Led Consolidation Programs. *Transportation Research Part A: Policy and Practice*, Vol. 84, 2016, pp. 109–130.
7. Sánchez-Díaz, I., P. Georén, and M. Brolinson. Shifting Urban Freight Deliveries to the Off-Peak Hours: A Review of Theory and Practice. *Transport Reviews*, Vol. 37, No. 4, 2017, pp. 521–543.
8. Campagna, A., A. Stathacopoulos, L. Persia, and E. Xenou. Data Collection Framework for Understanding UFT within City Logistics Solutions. *Transportation Research Procedia*, Vol. 24, 2017, pp. 354–361.
9. Zou, W., X. Wang, A. Conway, and Q. Chen. Empirical Analyzis of Delivery Vehicle On-Street Parking Pattern in Manhattan Area. *Journal of Urban Planning and Development*, Vol. 142, No. 2, 2015, 04015017.
10. Dalla Chiara, G., and L. Cheah. Data Stories from Urban Loading Bays. *European Transport Research Review*, Vol. 9, No. 4, 2017, p. 50.
11. Sterne, J. A., I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter. Multiple Imputation for Missing Data in Epidemiological and Clinical Research: Potential and Pitfalls. *BMJ*, Vol. 338, 2009, p. b2393.
12. Fairclough, D., and D. Cella. Functional Assessment of Cancer Therapy (FACT-G): Non-Response to Individual Questions. *Quality of Life Research*, Vol. 5, No. 3, 1996, pp. 321–329.

13. Durrant, G. B. *Imputation Methods for Handling Item-Nonresponse in the Social Sciences: A Methodological Review*. ESRC National Centre for Research Methods and Southampton Statistical Sciences Research Institute. NCRM Methods Review Papers NCRM/002, Southampton, UK, 2005.
14. García-Laencina, P. J., J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern Classification with Missing Data: A Review. *Neural Computing and Applications*, Vol. 19, No. 2, 2010, pp. 263–282.
15. Rubin, D. B. *Multiple Imputation for Nonresponse in Surveys*. John Wiley & Sons, Hoboken, NJ, 2004.
16. Azur, M. J., E. A. Stuart, C. Frangakis, and P. J. Leaf. Multiple Imputation by Chained Equations: What Is It and How Does It Work? *International Journal of Methods in Psychiatric Research*, Vol. 20, No. 1, 2011, pp. 40–49.
17. Sancho-Gómez, J.-L., P. J. García-Laencina, and A. R. Figueiras-Vidal. Combining Missing Data Imputation and Pattern Classification in a Multi-Layer Perceptron. *Intelligent Automation & Soft Computing*, Vol. 15, No. 4, 2009, pp. 539–553.
18. Thompson, B. B., R. Marks, and M. A. El-Sharkawi, eds. On the Contractive Nature of Autoencoders: Application to Missing Sensor Restoration. *Proc., International Joint Conference on Neural Networks*, IEEE, Portland, OR, 2003.
19. Yoon, J., J. Jordon, and M. Van Der Schaar. Gain: Missing Data Imputation using Generative Adversarial Nets. *arXiv Preprint arXiv:180602920*, 2018.
20. Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, Vol. 29, No. 5, 2001, pp. 1189–1232.
21. Tekler, Z., R. Low, and L. Blessing, eds. Using Smart Technologies to Identify Occupancy and Plug-In Appliance Interaction Patterns in an Office Environment. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, Bristol, England, 2019.
22. Tekler, Z. D., R. Low, and L. Blessing, eds. An Alternative Approach to Monitor Occupancy using Bluetooth Low Energy Technology in an Office Environment. *Journal of Physics: Conference Series*, IOP Publishing, 2019.
23. Tekler, Z. D., R. Low, B. Gunay, R. L. Andersen, and L. Blessing. A Scalable Bluetooth Low Energy Approach to Identify Occupancy Patterns and Profiles in Office Spaces. *Building and Environment*, Vol. 171, 2020, p. 106681.
24. Low, R., L. Cheah, and L. You. Commercial Vehicle Activity Prediction with Imbalanced Class Distribution using a Hybrid Sampling and Gradient Boosting Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 99, 2020, pp. 1–10.
25. Goodfellow, I. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv Preprint arXiv*:170100160, 2016.
26. Hastie, T., R. Tibshirani, J. Friedman, and J. Franklin. The Elements of Statistical Learning: Data Mining, Inference and Prediction. *The Mathematical Intelligencer*, Vol. 27, No. 2, 2005, pp. 83–85.
27. McCaffrey, D. F., G. Ridgeway, and A. R. Morral. Propensity Score Estimation with Boosted Regression for Evaluating Causal Effects in Observational Studies. *Psychological Methods*, Vol. 9, No. 4, 2004, p. 403.

The opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors only.