A novel activity pattern generation incorporating deep learning for transport demand models

Danh T. Phan* and Hai L. Vu*

*Department of Civil Engineering, Monash University, Clayton 3800, VIC, Australia

Abstract

Activity generation plays an important role in activity-based demand modelling systems. While machine learning, especially deep learning, has been increasingly used for mode choice and traffic flow prediction, much less research exploiting the advantage of deep learning for activity generation tasks. This paper proposes a novel activity pattern generation framework by incorporating deep learning with travel domain knowledge. We model each activity schedule as one primary activity tour and several secondary activity tours. We then develop different deep neural networks with entity embedding and random forest models to classify activity type, as well as to predict activity times. The proposed framework can capture the activity patterns for individuals in both training and validation sets. Results show high accuracy for the start time and end time of work and school activities. The framework also replicates the start time patterns of stop-before and stop-after primary work activity well. This provides a promising direction to deploy advanced machine learning methods to generate more reliable activity-travel patterns for transport demand systems and their applications.

Keywords: Activity generation; Neural networks; Deep learning; Entity embedding; Random forest; Activity patterns; Travel patterns; Activity-based models; Agent-based models

1 Introduction

Activity-based travel demand models have been increasingly used to support transport planners to forecast mobility patterns (Castiglione et al., 2015). These demand models aim to describe the process of how individuals plan and schedule their daily activities that affect travel demand (Axhausen and Gärling, 1992). Specifically, it models and imitates the individual activity decision of what, when, how long, where, and with whom involved. Activity-based models usually include activity generation, activity location choice, and mode choice modules. An activity generation module is responsible for answering what, when and how long the activity will be carried out; a location choice module addresses the question of where the activity occurred; a mode choice module is used to model how individuals travel between activities. Some activity-based models may additionally have an activity scheduling module that simulates the activity scheduling process and interaction among individuals or members in households. An activity-based model is often integrated with a traffic assignment simulation to form an agent-based system (Briem et al., 2019), as shown in Fig. 1. The outcomes of activity-based models are individuals’ activity schedules or plans. These schedules are the demand input for agent-based simulations. Being the first module, the activity generation plays a crucial role in developing an accurate and realistic transport demand model.

Two popular approaches for activity generation are utility-based and rule-based models (Rasouli and Timmermans, 2014). The utility-based models are based on an econometric assumption that individuals select a travel pattern alternative that maximises their utilities. Discrete choice methods are often used to estimate and select the choice with the highest utility among different alternatives. While discrete choice methods could effectively estimate output variables having a small number of discrete values like travel modes, these methods could face difficulties in representing and estimating continuous output variables like activity time and duration. As traditional discrete choice methods can

*Corresponding author: the.phan@monash.edu, hai.vu@monash.edu
only work with categorical outputs, continuous outputs in utility-based models are often segmented into several groups to make them suitable for discrete choice. For example, activity start time or trip departure time were grouped into four or six separate periods (Bowman, 1998; Bhat et al., 2004). This aggregate time representation reduces the accuracy of activity generation modules in utility-based models. To enhance discrete choice capacity, several studies have proposed different methods such as joint discrete-continuous models and hazard-based models to estimate continuous activity duration or time expenditure.

In contrast, rule-based models represent individual activity decisions as heuristics or rules (Rasouli and Timmermans, 2014). These models usually capture the activity patterns of individuals reasonably well. Their results may stem from the use of observed distributions in activity generation modules. These distributions are usually built from travel survey data. However, rule-based models might depend on complex manually-coded expert knowledge to generate activity patterns (Drchal et al., 2019). This complexity may reduce the transferability of rule-based models, as these expert-designed components may differ among different cities or regions.

Machine learning is a practical approach to automatically derive rules from data. It could help to improve rule-based models by reducing the complexity in expert-designed components. Machine learning also has the potential to address the challenge of large datasets and new travel data sources (Miller, 2017). Another advantage of machine learning is its systematic validation process, which separates data into training and validation sets (Miller, 2017). Different machine learning methods are increasingly being applied in activity-travel behaviour research, where popular methods are neural networks and decision trees (Koushik et al., 2020). For activity generation tasks, for instance, simple neural networks were applied by Kato et al. (2002) in a comprehensive activity-based modelling system. In addition, decision trees were used for generating activity attributes in activity-based models like ALBATROSS and DDAS (Arentze et al., 2000a; Drchal et al., 2019). When summarising the performance of different machine learning methods for classification tasks in various travel behaviour studies, random forest usually delivers better results (Koushik et al., 2020).

The recent advancement of deep learning has led to breakthroughs in machine learning research (Goodfellow et al., 2016). Deep learning is widely used to address various tasks in Natural language processing and Computer vision fields. In transportation research, many studies have focused on the application of deep learning for mode choice and traffic flow prediction (Do et al., 2019; Veres and Moussa, 2019). Due to the capacity in modelling nonlinearity, deep neural networks can predict traffic flow with high accuracy. Deep neural networks have recently been complemented with entity embedding to encode categorical features. Entity embedding technique not only helps to effectively encode categorical features, but also improves the accuracy of deep learning models (Guo and Berkhahn, 2016; Zheng et al., 2019).

Till date, most research in activity-based modelling has focused on applying machine learning for mode choice and activity type detection (Koushik et al., 2020). However, there have not been many studies on incorporating machine learning into activity generation modules (Koushik et al., 2020; Ozonder and Miller, 2021). Moreover, the emergence of deep learning and new entity embedding
techniques are yet to be explored for activity generation tasks. To address this gap, we develop in this paper a framework to apply deep learning and entity embedding for the improvement of the activity generation module. Our goal is to automatically learn the distributions from the observed survey data, and then use the learned distributions to generate activity and travel patterns.

To this end, we introduce a novel activity pattern generator framework by leveraging the advantages of state-of-the-art machine learning methods including deep neural networks, random forest, and entity embedding techniques. Our contributions include:

- Develop a novel activity pattern generator framework by complementing deep learning techniques with domain knowledge. The framework can effectively work with high-cardinality categorical features by incorporating entity embedding in deep neural networks.
- Replicate tour-based activity patterns by leveraging skeleton schedule knowledge. These tour-based travel patterns could play an important role in modelling travel mode choice constraints among activities.
- Propose a new approach that represents activity type as discrete, and both activity start time and end time as continuous variables. This approach can capture the activity start time and end time of the primary activity, as well as the start time pattern of stop-before and stop-after of the primary activity.

The rest of the paper is outlined as follows: Section 2 reviews existing approaches for activity generation tasks. Section 3 presents the proposed activity generator framework, from high-level architecture to detailed components. Section 4 describes the data used to train and evaluate the model. Finally, Section 5 discusses the results and important implications of our approach for developing the next generation of activity-based demand models.

2 Literature Review

In the following sections, we review related studies and approaches on activity generation tasks. We then discuss the application of machine learning to travel behaviour analysis. Furthermore, the combination of deep learning and entity embedding techniques in transportation research will also be presented.

Focusing on activity generation tasks, we categorise activity-based models into three groups: utility-based, rule-based and machine learning (ML-based) models. For each group, popular models will be discussed. Table 1 summarises and compares the temporal resolutions and methods used for activity generation in these models. A more comprehensive review of different activity-based demand models can be referred to Rasouli and Timmermans (2014).

| Model name   | Model category | Start time  | Generation method       | Duration      | Generation method          |
|--------------|----------------|-------------|-------------------------|---------------|-----------------------------|
| DaySim       | Utility-based  | Segments    | Discrete choice         | None          |                             |
| CEMDAP       | Utility-based  | Segments    | Discrete choice         | Continuous    | Hazard-based model          |
| TASHA        | Rule-based     | Continuous  | Observed distributions  | Continuous    | Observed distributions      |
| ADAPTS       | Rule-based     | Continuous  | Observed distributions  | Continuous    | Hazard-based model          |
| ALBATROSS    | ML-based       | Segments    | Decision trees          | Segments      | Decision trees              |
| DDAS         | ML-based       | Segments    | None                    | Continuous    | Decision trees              |

2.1 Existing activity generation methods

2.1.1 Utility-based models

One popular utility-based model is DaySim or Day Activity Schedule proposed by Bowman and Ben-Akiva (2001). They assumed that a prior basic activity pattern was formed before the decision of more detailed activity and travel plans. DaySim represented each individual schedule as a daily activity pattern with tours. The primary tour’s structure included a primary activity, the sub-tour type (only
for *subsistence* primary activity), and intermediate stop-type before and after the primary activity location. Bowman and Ben-Akiva (2001) used nested logit models to represent the choice of activity patterns. The highest tier of the nested hierarchy was an activity pattern, followed by a primary tour, and secondary tours at the lowest layer. The time of home-based tours was represented as the combination of the departure time from home and from the primary activity in the daily activity chain. These departure times were grouped into four time periods including AM peak, Midday, PM peak, and others (Bowman and Ben-Akiva, 2001). Two multinomial logit models were applied to estimate the choice of tour time for both primary and secondary tours. DaySim was then implemented as an activity generation component in an operational travel forecasting system named SacSim (Bowman et al., 2006).

Another utility-based model is the Comprehensive Econometric Microsimulator for Daily Activity-Travel Patterns (CEMDAP) which formed an individual activity-travel structure into three levels including pattern, tour and stop (Bhat et al., 2004). A pattern was a list of consecutive tours, and each tour included a chain of travel stops. Each travel stop was an out-of-home activity episode along with activity type, duration, and location, as well as the travel time to this stop location. CEMDAP differentiated travel patterns among non-workers, and workers (including students) (Bhat et al., 2004). Non-worker patterns were represented with a list of home-based tours with no fixed stops. Worker patterns were divided into five segments: before-work, home-to-work, work-based, work-to-home, and after-work. For work activity, CEMDAP first generated work duration, and then start time. Specifically, a hazard-based model was used for generating continuous activity duration, while discrete choice methods were applied for selecting activity start time (Bhat et al., 2004).

In utility-based models, discrete choice methods were normally used to model categorical outputs such as activity types and travel mode choices. To address the limitation of traditional discrete choice models when dealing with continuous activity time, different joint discrete-continuous choice models were proposed. For instance, Bhat (2005) developed the multiple discrete-continuous extreme value (MDCEV) model for activity type and duration, while Konduri et al. (2010) introduced a probit-based discrete-continuous model of activity type and duration choices. Furthermore, Nurul Habib (2018) proposed a comprehensive utility-based system by integrating discrete activity type and location choice with continuous-time expenditure for workers.

The advantages of utility-based models are their econometric-based travel behaviour theory and their capacity to represent activity schedules as tour-based travel patterns. The theory-driven modelling paradigm explains the behaviour of human choices, while tour-based patterns help to effectively forecast mode choices with constraints. However, the number of independent variables in these models is often small. In addition, many personal and household features are often ignored in utility functions. Despite some limitations in dealing with continuous outputs like activity start time, discrete choice methods have been extended for the development of different activity-based models due to their interpretability. For example, DaySim was integrated as the daily activity generation module for SimMobility-Midterm, as well as was extended in a weekly activity generation model named ActiTopp (Lu et al., 2015; Hilgert et al., 2017).

### 2.1.2 Rule-based models

Two popular rule-based operational models are TASHA and ADAPTS. The Travel/Activity Scheduler for Household Agents (TASHA) was developed to replicate the activity scheduling and interacting process of household members (Miller and Roorda, 2003; Miller et al., 2015). In TASHA, the sequence of activity generation was activity frequency, start time and duration. These activity attributes were generated from empirical probability distributions in a travel survey (Eberhard, 2003; Miller et al., 2015). The formulation of observed distributions for different activity types was based on the cross-classification of individual, household and schedule characteristics such as gender, age, occupation, employment status, student status, presence of children, and work project status (Yasmin et al., 2015). The validation of TASHA’s activity generation and scheduling modules showed that it can replicate observed activity with good accuracy (Roorda et al., 2008). However, the formation of distributions was dependent on the demographic characteristics of studied areas (Yasmin et al., 2015). Therefore, it remains challenging to transfer these observed rules into other cities.

Auld and Mohammadian (2009) developed the Agent-based Dynamic Activity Planning and Travel Scheduling (ADAPTS) model. ADAPTS aimed to dynamically model the activity planning and...
It treated the decision of selecting activity attributes as a separate event in each simulation time step. In any time step, if an agent planned an activity, the agent would firstly select an activity type. After that, an activity planning order module decided in what order other activity attributes were planned (Auld and Mohammadian, 2009). For example, one planning order may be activity time first, followed by location, with whom, and finally by which travel mode. To model the dynamic planning order, ADAPTS required an additional travel survey called the Urban Travel Route and Activity Choice Survey, which may not be available in other cities (Auld and Mohammadian, 2012). Furthermore, similar to TASHA, activity start time and duration in ADAPTS were drawn from observed start time duration distributions derived from a travel survey (Auld, 2011).

Both TASHA and ADAPTS have been implemented in operational agent-based systems (Miller et al., 2015; Auld et al., 2016). These systems may also incorporate discrete choice into other components such as mode choice modules. For activity generation modules, the use of observed distributions results in high accuracy of activity start time. However, the forming of these distributions might require expert domain knowledge. Machine learning methods can alleviate this limitation by automatically deriving rules from observed datasets, which will be explored in this paper.

2.1.3 Machine learning (ML-based) models

First generation neural networks

The idea of applying neural networks for travel demand analysis could be traced back to the 1990s. The Activity-Mobility Simulator (AMOS) was probably the first attempt to use neural networks in a comprehensive activity-based travel demand system (Kitamura et al., 1993). AMOS was based on the behavioural principle of adaptation, which aimed to replicate a learning process where individuals try to find the best activity-travel alternative using a trial-and-error process. AMOS included a response option generator component which produced and ranked a list of choices when an individual encountered changes in their travel environment (Kitamura et al., 1996). Hence, the role of neural networks in AMOS was not for activity pattern generation, but for forecasting the change of activity pattern, given the change in the travel environment.

In another study, Kato et al. (2002) used neural networks to build an activity-based travel demand model for work-tour mode and related discretionary activities. They assumed the work travel pattern, particularly work travel mode, will affect the characteristics of discretionary tours before and after work. Firstly, a model was built to predict a travel mode to work. Given the work travel information, they then developed two models to estimate trip generation, destination choice, mode choice and activity duration for discretionary activities before and after work. This model showed its practical capability to forecast the impacts of travel demand measures on daily travel patterns (Kato et al., 2002).

While the above studies showed the practical capability of neural networks for travel demand modelling, the architecture of these first-generation neural networks was simple, using only one or two layers with a small number of neurons. The data sets used for model estimation were also small-size samples. Thus, these models did not fully exploit the advantages of neural networks.

Decision trees

Arentze et al. (2000a) proposed A Learning-Based, Transportation-Oriented Simulation System (ALBATROSS), which was the first tree-based comprehensive activity-based model. ALBATROSS used different decision tree algorithms for generating activity attributes, including activity type, with whom, duration, start time and location. Activity start time was divided into six periods, while duration was grouped into three categories: short, average or long (Arentze et al., 2000b). The alternatives of activity attributes relied on the choice situation which was based on different conditional variables. A limitation of ALBATROSS was its scheduling process which is based on a pre-assumption that is not validated on empirical data (Rasouli and Timmermans, 2014).

Drchal et al. (2019) have recently developed a Data-Driven Activity Scheduler (DDAS) to generate sequential activity schedules. For activity generation, they assumed that previous activities affected the current activity. Hence, the current activity attributes such as activity type and end time were used to forecast the next activity attributes. Drchal et al. (2019) employed a decision tree classifier to select the next activity type, and used a continuous probability distribution to generate the next
activity duration. However, DDAS did not capture the start time of activities well (Drchal et al., 2019). In addition, DDAS framework cannot produce tour-based activity schedules, which could reduce its accuracy in the mode choice module.

Nevertheless, the advantage of decision trees is their interpretability, which allows tree-based models such as ALBATROSS and DDAS to infer causal relationships among input factors and outputs. However, the accuracy of activity generation in these models seems weak, leading to more inaccurate outcomes in other modules such as location choice and mode choice modules.

Several studies have recently applied more advanced decision tree techniques such as random forest for trip generation (Ghasri et al., 2017) or activity generation (Ozonder and Miller, 2021). They showed the usefulness of random forest in producing higher accuracy for trip generation, as well as in identifying the most important explanatory variables for activity generation. Our proposed framework also exploits the advance of random forest for activity generation tasks.

2.2 Deep neural networks for travel behaviour analysis

Deep learning, or deep neural network, is a powerful tool for analysing not only images and text data but also tabular or relational data (Arik and Pfister, 2019). With the improvement of high-performance computing systems and hardware like graphics processing units, machine learning researchers and practitioners are now able to train deep learning models with large datasets. Another advantage of deep learning is its ability in solving regression tasks, which can forecast continuous outputs. In travel behaviour analysis, there is an increasing application of machine learning, especially neural networks for predicting travel mode and travel destination (Koushik et al., 2020). However, there are not many studies on applying machine learning for activity generation tasks (Koushik et al., 2020; Ozonder and Miller, 2021). Especially, the integration of deep learning for developing a complete activity generation module is yet to be developed.

Deep learning has recently been complemented with entity embedding techniques. The use of entity embedding for encoding categorical features has improved the accuracy of deep neural networks and their capacity to work with categorical variables (Guo and Berkhahn, 2016). Entity embedding has a similar idea as word embedding like Word2Vec encoding which represents each word as a continuous vector (Mikolov et al., 2013). Using entity embedding, which creates embedded vector representation for categorical variables, deep learning models can work well with categorical features with a large number of discrete values. Entity embedding helps to reduce the dimensions of categorical features, thus reducing the computation costs and speeding up neural networks compared with one-hot encoding1 (Guo and Berkhahn, 2016). Moreover, the embedding vectors obtained from training deep neural networks can improve the performance of other machine learning methods like random forest. However, the combination of deep learning with entity embedding has only recently been applied in transportation research.

In the next section, we introduce a novel activity pattern generator that incorporates random forest, deep learning and entity embedding techniques to generate reliable mobility patterns.

3 Proposed activity pattern generator

In this section, we first introduce several concepts and the high-level architecture of the proposed framework, followed by more detailed components. Furthermore, the implementation of advanced machine learning techniques is described along with performance evaluation metrics.

3.1 Tour-based activity pattern concept

We assume that each person’s daily out-of-home activity schedule includes one primary activity and several secondary activities, as similar to (Bowman, 1998). The primary activity then forms a home-based primary tour, while secondary activities create several home-based secondary tours. The criteria to decide which activity is the primary activity will depend on activity type and activity duration.

1One-hot encoding represents each categorical value as a vector of a single one (1) and several zeros (0)
Activity type and activity group

Activity types are categorised into three groups: subsistence, maintenance, and discretionary (Table 2). As activity type and activity group have a similar role, they can be used interchangeably. Subsistence activities include work for workers and study for students. Maintenance activities are those for the household to maintain their daily life. Shopping and personal business, for instance, are maintenance activities. Discretionary activities are those with lower priority and constraints. For example, discretionary activities include recreational, social, and other activities. However, different from (Bowman, 1998), we explicitly consider pickup-dropoff activities as a separate activity group. While pickup-dropoff activities normally take a short duration, they hold important information regarding individuals travel patterns.

Table 2: Activity group along with its activity types

| Activity group   | Code | Activity type                  |
|------------------|------|--------------------------------|
| Subsistence      | W    | Work; Study                    |
| Maintenance      | M    | Shopping; Personal business    |
| Discretionary    | D    | Recreation; Social; Other      |
| Pickup–dropoff   | P    | Pick up or Drop off            |

Primary and secondary activity

We define a primary activity and secondary activities for three different personal groups: workers, students, and nonworkers (Table 3). For workers and students, the primary activity could be subsistence, maintenance, or discretionary; while secondary activities could be maintenance, discretionary, or pickup-dropoff. For nonworkers, since they do not have a subsistence activity, their primary activity could be maintenance or discretionary. Nonworkers also have similar secondary activity types like workers and students.

Several conditions with thresholds are used to decide which is a primary activity. The thresholds depend on activity groups and the dataset’s characteristics. The highest priority is for subsistence activity, followed by maintenance (with a minimum duration above a threshold), and then discretionary (again with a minimum duration above a threshold). When no subsistence primary activity is found, we consider a primary activity as the longest duration one among maintenance and discretionary activities in that person’s activity schedule.

Table 3: Primary activity and secondary activity in different personal groups

| Personal group | Primary activity   | Secondary activity |
|----------------|--------------------|--------------------|
| Workers        | W or M or D        | M or D or P        |
| Students       | W or M or D        | M or D or P        |
| Nonworkers     | M or D             | M or D or P        |

Formulation of primary tour and secondary tour

Each activity schedule is represented as a home-based primary tour along with several home-based secondary tours. The primary tour may have stop-before or stop-after or both. The primary tour with subsistence primary activity might contain sub-tours. These concepts are illustrated in Fig. 2 where an activity schedule is transformed into one home-based primary tour (with a work-based sub-tour) and one home-based secondary tour.

For example, one exemplary daily activity schedule of a worker is described as follows: In the morning, the worker dropped off (P) their child at 8:30 am on the way to a workplace. The worker then started to work (W) at 9:10 am, and at noon, went out for a personal business (M) at 12:30 pm for around one hour, and after that returned to work. The worker finished working at 5:15 pm and went shopping (M) on the way home. The person arrived at a shopping centre at 5:40 pm and spent around 30 minutes shopping. The worker then came home, rested, and had dinner with the family.
At night, the person went out to a nearby cafe to socialise (D) with their friends for one hour, and then came back home. From this information, which can be derived from a travel diary survey, we can construct an activity schedule for that worker (as in the left of Fig. 2).

The activity schedule is translated into one primary tour and one secondary tour (as in the right of Fig. 2). The primary tour has work (W) as a primary activity. There is one stop-before (P), and one stop-after (M) the primary activity. In addition, the person also has a work-based sub-tour (M) at noon during work activity. The transformation from activity schedule into a primary tour and secondary tours helps to form the architecture of the proposed activity generator framework.

Figure 2: Translating an activity schedule to tour-based activity patterns

3.2 High-level architecture

The activity pattern generator framework aims to generate individual tour-based activity schedule (Fig. 3). The inputs include household demographics, personal characteristics, and zone information. The framework first generates primary activity attributes which are considered as the skeleton schedule of each person. Based on the skeleton information, the framework then produces a full primary activity pattern of stop-before, stop-after, and sub-tour attributes. Finally, the framework creates secondary tour-based activities based on the primary activity pattern information.

In the probability terms, each activity schedule is represented as a set of secondary home-based tours tied together by a primary activity pattern (Eq. (1)). The primary activity pattern is a home-based primary activity tour which includes primary activity attributes along with stop-before and stop-after it (Eq. (2)).

$$\text{prob}(\text{schedule}) = \text{prob}(\text{primaryPattern}) \cdot \text{prob}(\text{secondaryPattern}|\text{primaryPattern})$$  
(1)

$$\text{prob}(\text{primaryPattern}) = \text{prob}(\text{primaryActivity}) \cdot \text{prob}(\text{stops}|\text{primaryActivity})$$  
(2)

The assumption here is that the primary activity affects other activities and stops in a daily schedule of individuals. The primary activity forms a skeleton pattern that constraints stop-before, stop-after, and secondary activities. The benefit of this formulation is that it could generate tour-based travel patterns, which could yield more reliable outcomes in the subsequent modules such as location choice and mode choice of activity-based models. In contrast, models like DDAS can only generate the current activity based on the previous activity information, without constraints on tours (Drchal et al., 2019). Specifically, the mode choice module of DDAS is a trip-based model. Compared to trip-based models, tour-based models deliver a major advance as they explicitly account for the logical interconnections between individual trips (Miller, 2019).

To this end, our proposed activity pattern generator framework includes two modules: a primary activity pattern generator and a secondary activity generator (Fig. 3). The framework first uses the
primary pattern generator to generate individuals primary tour-based activity patterns. Then, based on the primary activity information, the framework will generate secondary tour-based activities using the secondary pattern generator. In the current setup, the secondary activity generator component is simpler than the primary activity generator. Similar to Miller and Roorda (2003), the sequence in predicting secondary activity attributes is activity type, and then start time and duration. Detailed components and implemented methods are presented in the next sections.

3.3 Primary activity pattern generator

A primary activity pattern generator module is used to generate a home-based primary tour. This process includes two main stages: first generating primary activity attributes, and then generating stop-before, stop-after, and sub-tour for this primary activity (Fig. 4).

Primary activity attributes

In the first stage, we classify the primary activity type as one of subsistence, maintenance or discretionary categories. Given the primary activity type, the model then predicts its continuous start time. After that, the model will estimate the end time of the primary activity based on its type, start time, and other inputs. The primary duration will be calculated from the gap between the start time and end time. The output of stage one is the primary activity type, its start time and end time.

It is worth noting that in contrast to other existing models, we predict the end time of the primary activity instead of its duration. This is because there may be a sub-tour during the primary activity. In this case, using the end time of the primary activity represents more accurate activity time patterns. When a sub-tour occurs, the duration of primary activity can be derived from the differences in primary activity’s end time and start time, less the duration of the sub-tour.

Stop-before and stop-after of a primary activity

The second stage produces the stop-before and stop-after of the primary activity. Stop information is generated based on its primary attributes, which can be considered as a skeleton schedule. Given the primary activity type, and start time, the model predicts the type, start time and duration of the stop-before of a primary activity. The activity type of stop-before can be maintenance, discretionary, pickup-dropoff, as well as zero-stop type in case there is no stop. Similarly, the model also forecasts the stop-after given the primary activity type and end time.

This second stage could also generate sub-tour information if the primary activity type is work or education. The sub-tour activity type can be maintenance, pickup-dropoff, discretionary or None. In this situation, the sub-tour attributes are generated using the primary activity type, and both the start time and end time of the primary activity.

3.4 Machine learning techniques

We develop different deep neural networks and random forest models for each component in the primary activity generator and secondary activity generator. Classification models are implemented for classifying stop or activity types, while regression models are built for activity time prediction.
Figure 4: The detailed architecture of primary activity pattern generator. The bottom shows an example of final primary activity patterns for the daily scheduler mentioned in Fig. 2.
Stage 1: Generating primary activity attributes, with the (red) output of each step on the left.
Stage 2: Generating stops and sub-tour information, with the (red) output of each step on the right.

Random forest

Random forest (RF) is a combination of several decision trees \(h(x, \Theta_k), k = 1, \ldots\), where the \(\Theta_k\) represents independent identically distributed random vectors, and each tree decides a weight for the most common features of the input \(x\) (Breiman, 2001). Random forest can be applied to sort the feature importance of input variables in both classification tasks (like activity type classification) and regression tasks (like activity start time prediction) (Breiman, 2001).

Each decision tree has nodes and leaves. The top nodes represent independent features, while the bottom contains tree leaves, which represent final targets. For example, one simple decision tree for the workers’ primary activity type classification is shown in Fig. 5. This tree has five nodes with variable names (MainRole, WorkType, Age, and PriActLocation) along with conditions to divide it into branches. At the bottom, there are six leaf nodes, which show the predicted label of primary activity types (W: Work, M: Maintenance, or D: Discretionary).

Figure 5: A decision tree for workers’ primary activity type classifier
A grid search algorithm\(^2\) with five-fold cross-validation is used to find the best random forest configuration of hyperparameters (Table 4). For simplicity, we investigate the performance of different random forest models by varying the number of decision trees, and the minimum of sample leaves in each decision tree. The Gini index is used to measure the node impurity for classifiers, while the residual sum of squares metric is applied for regression models (Liaw and Wiener, 2012). The max samples depend on the size of the training sets. For example, there is a maximum of 10000 samples in case of workers’ data set, while there is a maximum of 5000 samples in each decision tree for students and nonworkers’ data set since the number of observations is smaller. The investigated configurations of different random forest models are detailed in Table 4. In addition, the importance of variables is also calculated to examine the explanatory variables in random forest models.

### Table 4: Random forests’ hyperparameters

| Hyperparameter     | Values            |
|--------------------|-------------------|
| max_samples        | 5000, 10000       |
| min_samples_leaf   | 3, 5, 6, 10, 20   |
| n_estimators       | 40, 60, 100       |

#### Deep neural networks with entity embedding

We use Deep feedforward network architecture, also called Multilayer perceptrons (MLPs), to build deep neural networks. Feedforward neural networks are a conceptual stepping stone for the development of other architectures like recurrent networks (RNNs) and convolutional networks (CNNs) (Goodfellow et al., 2016). We choose MLP architecture due to its simplicity and suitability for tabular data which is a travel survey as in our case.

MLPs are the composition of many different hidden layers (Goodfellow et al., 2016). Each hidden layer is represented as a function \( h \), which is the combination of an activate function \( g \) on top of an affine transformation (Eq. (3)). The affine transformation is used to transform the input \( x \) from the previous layer by using parameters \( (W) \) and bias \( (b) \). The activate function is a nonlinear transformation. Popular activate functions include Sigmoid, Tanh, Rectified linear unit (ReLU), and LeakyReLU. For \( n \) hidden network layers, the output function \( f \) is represented in Eq. (4) with the input \( h^{(n)} \). The \( h^{(n)} \) function is calculated from its previous hidden layers \( h^{(n-1)} \) as presented in Eq. (5).

\[
h = g(W^T x + b) \tag{3}
\]

\[
f = g^{(n)}(W^{(n)}T h^{(n)} + b^{(n)}) \tag{4}
\]

\[
h^{(n)} = g^{(n-1)}(W^{(n-1)}T h^{(n-1)} + b^{(n-1)}) \tag{5}
\]

When implementing different deep neural networks for different components in the activity pattern generator, we also use a simple grid search approach to seek the best architecture of feedforward neural networks. We vary the number of hidden layers from one to six layers, and find that two or three hidden layers produce the best results. The configuration of other hyperparameters is presented in Table 5.

The above MLPs architecture is complemented with the entity embedding of high-cardinality categorical features to form our activity pattern generator framework. We apply entity embedding to encode categorical variables of the household, individual and zone information. These encoded features include variables such as person type, main occupation, weekday and home postcode. The encoded vectors are then concatenated with continuous variables like person age, household size, and the number of cars. For example, a deep neural network architecture for primary activity start time prediction includes three hidden layers with ReLU activation, as well as an entity embedding layer for categorical variables (as shown in (a) of Fig. 6).

\(^2\)https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
Table 5: Deep neural networks’ hyperparameters

| Hyperparameter       | Values                                      |
|----------------------|---------------------------------------------|
| number of hidden layers | 1, 2, 3, 4, 5, 6                           |
| learning rate        | $1e^{-5}$, $1e^{-4}$, $1e^{-3}$, $1e^{-2}$, $1e^{-1}$ |
| activation           | $\text{Sigmoid}$, $\text{ReLU}$            |
| loss function        | Cross entropy, RMSE                        |
| optimiser            | SGD, Adam                                   |

Figure 6: Transferring categorical embedding vectors from a deep neural network to a random forest.
(a) A deep neural network with 3 hidden layers and 1 entity embedding layer for categorical features.
(b) A random forest model which uses the categorical encoded vectors from the deep neural network.

One-hot encoding is a popular technique to encode categorical variables in Machine Learning. Compared to one-hot encoding, entity embedding requires less memory (see Table 6 for an example). With entity embedding, for instance, a categorical variable with $n$ unique values is encoded by a vector size of $\frac{n+1}{2}$ in our setup, while one-hot encoding requires a vector size of $n$. For instance, Table 6 shows the difference between one-hot encoded vector and entity embedded vector for a weekday, which is derived from the deep neural network for classifying the workers’ primary activity type.

To investigate the efficacy of the entity embedding technique, we compare the performance of deep neural networks with one-hot encoding and with entity embedding. The learned entity embedded vectors are also integrated into RF models to check its transferability by copying the learned embedded vectors from a deep neural network to a random forest model, as shown in Fig. 6. We use Scikit-learn\(^3\) library for building the random forest models, as well as Pytorch\(^4\) and Fastai\(^5\) libraries for implementing the deep neural networks with different embedding techniques.

Model evaluation metrics

The accuracy rate (Eq. (6)) and $F_1$ score (Eq. (7)) is used to measure the fitness of classification models like activity type classification, while Root Mean Square Error (RMSE) is used to measure the error of regression models (Eq. (8)) for activity time prediction. These metrics will be compared for the results of different deep learning (DL) and random forest (RF) models, with and without entity embedding.

---

\(^3\)https://scikit-learn.org/

\(^4\)https://pytorch.org/

\(^5\)https://www.fast.ai/
Table 6: An example of one-hot encoding and entity embedding representation for weekday.

| Weekday   | One-hot encoding | Entity embedding |
|-----------|------------------|-----------------|
| Monday    | [1, 0, 0, 0, 0, 0, 0] | [0.56, -0.87, 0.57, -0.76] |
| Tuesday   | [0, 1, 0, 0, 0, 0, 0] | [-0.73, -1.33, 0.29, -1.68] |
| Wednesday | [0, 0, 1, 0, 0, 0, 0] | [0.09, -1.42, -0.48, 0.81] |
| Thursday  | [0, 0, 0, 1, 0, 0, 0] | [0.14, 0.22, 0.21, -1.82] |
| Friday    | [0, 0, 0, 0, 1, 0, 0] | [-0.74, -0.80, 1.22, 0.03] |
| Saturday  | [0, 0, 0, 0, 0, 1, 0] | [-1.04, 2.45, 0.14, -0.56] |
| Sunday    | [0, 0, 0, 0, 0, 0, 1] | [0.27, 0.89, -0.56, 1.58] |

\[
Accuracy^6 = \frac{TP + TN}{TP + FP + TN + FN} \quad (6)
\]

\[
F_{1-score}^7 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{TP}{TP + 0.5 \cdot (FP + FN)} \quad (7)
\]

\[
RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \quad (8)
\]

4 Data

The Victorian Integrated Survey of Travel and Activity (VISTA)\(^7\) is used for the framework’s implementation. This travel diary survey includes around 174,000 trips of 64,500 persons in 25,000 households from Victoria, Australia from 2012 to 2018. The survey’s variables include personal variables, household features and zone information. Detailed description of these features are presented in Table 7.

A data cleaning process is performed and after removing inconsistent rows, there are around 158,000 trips of 41,700 persons in 19,600 households, which are used for model estimation and validation as described below. These trips are then converted into 41,700 personal activity schedules including 23,900 workers, 9,900 students (from primary schools to universities), and 7,900 nonworkers.

Individual trips are translated into activity attributes which then form each person’s activity schedule as a list of consecutive activities (as described in Section 3.1). For each person, the primary activity is derived from the activity schedule based on activity priority. Similar to (Bowman, 1998), we define thresholds with several conditions to select primary activities. From the primary activity and activity schedule, a primary activity pattern is formed by combining primary activity attributes, stop-before, and stop-after of the primary activity (as explained in Fig. 2). Similarly, secondary activities are those in the activity schedule, but not in the primary activity home-based tour.

Data analysis shows that more than 96% of people have zero-stop or one stop-before or one stop-after each primary activity. Hence, to simplify the model implementation, we only consider the maximum of one stop-before and one stop-after each primary activity. In case a person has two or more stop-before or stop-after, we select the longest duration stop, and remove other stops. The number of secondary tours is also insignificant; thus it is omitted in the current model’s implementation.

Finally, the data is split into training and validation sets. The data from 2012 to 2017 is used for training models, while the 2018 data is kept as validation sets.

The results of the activity generator framework, which incorporates machine learning techniques proposed in Section 3.4 for both training and validation sets, will be presented in the next section.

\(^6\)TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives
\(^7\)https://transport.vic.gov.au/about/data-and-research/vista
Table 7: The description of the dataset’s variables

| Variable                  | Data type    | Range/#Values | Description                                                                 |
|---------------------------|--------------|---------------|-----------------------------------------------------------------------------|
| **Personal attributes**   |              |               |                                                                             |
| PersonType                | Categorical  | 3             | Worker; Student; Nonworker                                                  |
| Age                       | Numerical (int) | [0-116]       | Age of the person                                                           |
| Gender                    | Categorical  | 2             | Male; Female                                                                |
| CarLicence                | Categorical  | 2             | Yes; No                                                                    |
| MainOccupation            | Categorical  | 10            | Main occupation                                                             |
| ANZSCO2                   | Categorical  | 53            | Detailed occupation                                                         |
| MainIndustry              | Categorical  | 21            | Main industry                                                               |
| ANZSIC2                   | Categorical  | 106           | Detailed industry                                                           |
| PersonIncomeLevel         | Categorical  | 11            | Income level                                                                |
| MainRole                  | Categorical  | 7             | Full-time worker; Part-time worker; Student; Pupil; Child; Retired; Nonworker|
| WorkType                  | Categorical  | 5             | Fixed hours; Flexible hours; Roster shifts; Work from home; Not in workforce |
| EmpType                   | Categorical  | 5             | Employment type                                                             |
| **Household features**    |              |               |                                                                             |
| OwnDwell                  | Categorical  | 5             | Fully owned; Being purchased; Being rented; Occupied rent-free; and Something else |
| TravelYear                | Numerical (int) | [2012-2018]  | Year travel, from 2012 to 2018                                             |
| TravelMonth               | Numerical (int) | [1-12]        | Travel month                                                                |
| TravelDay                 | Categorical  | 7             | Day in week                                                                |
| NumPersons                | Numerical (int) | [1-11]        | Number of persons in a household                                           |
| NumKids                   | Numerical (int) | [0-7]         | Number of children in a household                                          |
| NumFulltimeWorkers        | Numerical (int) | [0-6]         | Number of full-time workers in a household                                  |
| NumParttimeWorkers        | Numerical (int) | [0-5]         | Number of part-time workers in a household                                  |
| NumCasualWorkers          | Numerical (int) | [0-4]         | Number of casual workers in a household                                     |
| NumCars                   | Numerical (int) | [0-7]         | Number of cars                                                              |
| NumBikes                  | Numerical (int) | [0-14]        | Number of bikes                                                             |
| HhIncome                  | Numerical (cont) | [0-12500]    | Household income                                                            |
| YearsLived                | Numerical (int) | [0-88]        | Number of years lived at the house                                         |
| **Zone information**      |              |               |                                                                             |
| HomeRegion                | Categorical  | 2             | Home region                                                                |
| HomeLGA                   | Categorical  | 32            | Home local government area                                                  |
| HomePostcode              | Categorical  | 245           | Home postcode                                                               |

5 Results and discussion

**Activity type classification**

In most cases, random forest (RF) performs better than deep learning (DL) in the primary activity type classification task (Table 8). RF also mostly produces better accuracy rates and F1-score than DL in validation sets. This outcome is consistent with the results in previous studies (Koushik et al., 2020). Both RF and DL deliver reasonable accuracy for workers and students; however, the accuracy outcomes of classifying activity type in validation sets are weak. This is understandable, since the decision of selecting a primary activity type may depend on other factors outside the information in the dataset. For example, the decision not to travel to work should be based on the workers’ weekly work schedule. Similarly, the decision to go to work on a specific day of a part-time worker should depend on their roster.

For nonworkers, however, the accuracy is weak in both training and validation sets for RF and DL (Table 8). This may be due to the diversity and uncertainty of activity type selection of nonworkers. As they do not have to participate in subsistence activities like work or school, there are fewer constraints in their decision making for primary activity type selection. Thus, there is a need for more details such as a weekly travel survey to get a comprehensive picture of activity schedules.
The integration of entity embedding helps to improve machine learning models. Combining entity embedding into deep neural networks mostly increases the accuracy and $F_1$ score in both training and validation sets (on DL&Emb columns of Table 8). The embedded vectors, which are trained from DL, can also enhance the performance of RF models. Entity embedding not only helps RF to produce better outcomes in training sets, but it may also help RF deliver better results in validation sets (on RF&Emb columns of Table 8). This is due to the entity embedding’s capacity to capture the intrinsic properties of categorical variables by mapping similar values close to each other in the embedding space (Guo and Berkhahn, 2016). Entity embedding helps deep neural networks and random forest models to generalise better. Hence, entity embedding provides a useful technique when dealing with datasets with many high cardinality features such as postcodes and detailed occupations.

Table 8: The result of the primary activity type classification (Emb: Entity embedding)

| Group     | Metric | DL     | DL&Emb | RF     | RF&Emb | DL     | DL&Emb | RF     | RF&Emb |
|-----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Workers   | Acc    | 0.724  | 0.754  | 0.804  | **0.820** | 0.687  | 0.704  | **0.712** | 0.705  |
| Students  | Acc    | 0.749  | 0.837  | 0.845  | **0.848** | 0.780  | 0.798  | 0.835  | **0.841** |
| Nonworkers| Acc    | **0.703** | 0.627  | 0.625  | 0.668  | 0.584  | 0.594  | 0.576  | **0.594** |
| Workers   | F1     | 0.784  | 0.789  | 0.822  | **0.836** | 0.759  | 0.753  | 0.752  | 0.747  |
| Students  | F1     | 0.779  | 0.853  | 0.872  | **0.873** | 0.819  | 0.813  | 0.873  | **0.875** |
| Nonworkers| F1     | 0.711  | 0.668  | 0.693  | **0.714** | 0.616  | 0.624  | **0.665** | 0.662  |

Compared to DL, an advantage of RF is that it can show how it makes forecasting decision via feature importance. This helps to improve the interpretability of our framework. Fig. 7 shows the top 20 features that most affect the primary activity classifier. The most influenced factors include activity location, home postcode, age, income, occupation, industry, and travel month.

Activity time prediction and primary activity patterns

For primary activity start time and end time prediction, DL outperforms RF in training sets but is less generalised than RF in validation sets. DL can effectively replicate observed activity time of work activity for workers (Fig. 8) and school activity for students (Fig. 9). However, the prediction errors in validation sets for all three groups (workers, students, nonworkers) are significantly higher than training sets. Especially, the patterns of activity start time for maintenance and discretionary activities are not captured well in validation sets (Fig. 10, Fig. 11). RF seems less overfitting than DL in both training and validation sets.
In addition, the framework accurately predicts the start time pattern of stop-before and stop-after the primary activity. Given the primary skeleton information, both DL and RF in the framework can capture the travel pattern before and after the primary activity well. For example, our model can effectively replicate the pattern of observed data for the *pickup-dropoff*, stop-before, and stop-after of work activity (Fig. 12). Compared to observations, both DL and RF produce similar patterns of activity start time throughout the day, especially the trend of peak hours in the morning and afternoon. Our approach produces more accurate patterns for nonwork/school activities in comparison with discrete choice methods which tend to over-predict the frequency of nonwork/school activities (Dianat et al., 2020).

![Work start time [training]](image)

![Work end time [training]](image)

![Work start time [validation]](image)

![Work end time [validation]](image)

**Figure 8:** The start time and end time of work activities

![School start time [training]](image)

![School end time [training]](image)

![School start time [validation]](image)

![School end time [validation]](image)

**Figure 9:** The start time and end time of school activities
Important implications

The results show the potential use of deep learning and entity embedding in activity generation tasks. Our proposed activity generator framework accounts for a large number of explanatory features. This can help to address the challenge of big data and new travel behaviour data sources such as GPS, smartcard transactions, and smartphone applications. Machine learning methods, especially deep learning, can automatically estimate large numbers of parameters. Furthermore, combining with entity embedding, our framework can also effectively represent categorical features with a large number of discrete values. We also show that entity embedding has the potential to encode context-dependent variables to produce better outcomes for both deep neural network and random forest.

We demonstrate the need for incorporating travel behaviour domain knowledge with deep learning. While experimenting with different architecture to leverage deep learning, the use of the tour-based
skeleton schedule in our framework helps to capture accurate primary tour patterns. Besides, the tour-based activity patterns could benefit mode choice modules by leveraging tour-based mode choice models. Deep learning is advantageous in forecasting continuous activity attributes such as start time, end time, and duration, while random forest could be suitable for activity type classification. Moreover, the discrete choice could be used to model variables with limited discrete values like travel modes. Hence, using domain knowledge to design appropriate model architecture that can exploit the advantages of both machine learning methods and discrete choice models.

In future work, the accuracy of the primary activity type classification can be further improved. For example, the use of a weekly travel survey could help to derive which days individuals go to work. Given the available weekly travel data, our framework can incorporate this information for its improvement.

6 Conclusion

We show that deep learning has the potential for activity generation tasks in travel demand systems, which perform well for primary activity time prediction in the proposed framework. This is underpinned by the deep learning’s capacity in dealing with high-cardinality categorical inputs as well as continuous outputs prediction in large datasets. Combining with skeleton schedule knowledge, our approach can generate reliable activity patterns. More importantly, we also show that deep learning with entity embedding can accurately capture and reproduce the activity pattern for stop-before and stop-after of work primary activity. The framework could be expanded for activity location prediction by combining with additional data sets. It also provides a viable approach to exploit advanced machine learning techniques for generating more reliable activity and travel patterns, resulting in more reliable personal activity schedules and better accuracy performance for activity-based and agent-based transport systems.
Acknowledgement

The authors thank the Victorian Department of Transport, Australia for VISTA data access. We are also grateful for the feedback from Prof. Eric J. Miller and his research group at the Department of Civil & Mineral Engineering, University of Toronto, Canada.

References

Arentze, TA; Hofman, Frank; Van Mourik, H, and Timmermans, H. ALBATROSS: multiagent, rule-based model of activity pattern decisions. Transportation Research Record, 1706(1):136–144, 2000a.

Arentze, TA; Hofman, Frank; Van Mourik, H; Timmermans, H, and Wets, G. Using decision tree induction systems for modeling space-time behavior. Geographical Analysis, 32(4):330–350, 2000b.

Arik, Sercan O and Pfister, Tomas. Tabnet: Attentive interpretable tabular learning. arXiv preprint arXiv:1908.07442, 2019.

Auld, Joshua. Agent-based dynamic activity planning and travel scheduling model: Data collection and model development. University of Illinois at Chicago, 2011.

Auld, Joshua and Mohammadian, Abolfazl. Framework for the development of the agent-based dynamic activity planning and travel scheduling (ADAPTS) model. Transportation Letters, 1(3):245–255, 2009.

Auld, Joshua and Mohammadian, Abolfazl. Activity planning processes in the Agent-based Dynamic Activity Planning and Travel Scheduling (ADAPTS) model. Transportation Research Part A: Policy and Practice, 46(8):1386–1403, 2012.

Auld, Joshua; Hope, Michael; Ley, Hubert; Sokolov, Vadim; Xu, Bo, and Zhang, Kuilin. POLARIS: Agent-based modeling framework development and implementation for integrated travel demand and network and operations simulations. Transportation Research Part C: Emerging Technologies, 64:101–116, 2016.

Axhausen, Kay W and Gärling, Tommy. Activity-based approaches to travel analysis: conceptual frameworks, models, and research problems. Transport Reviews, 12(4):323–341, 1992.

Bhat, Chandra R. A multiple discrete–continuous extreme value model: formulation and application to discretionary time-use decisions. Transportation Research Part B: Methodological, 39(8):679–707, 2005.

Bhat, Chandra R; Guo, Jessica Y; Srinivasan, Sivaramakrishnan, and Sivakumar, Aruna. Comprehensive econometric microsimulator for daily activity-travel patterns. Transportation Research Record, 1894(1):57–66, 2004.

Bowman, John L. The day activity schedule approach to travel demand analysis. PhD thesis, Massachusetts Institute of Technology, 1998.

Bowman, John L and Ben-Akiva, Moshe E. Activity-based disaggregate travel demand model system with activity schedules. Transportation Research Part A: Policy and Practice, 35(1):1–28, 2001.

Bowman, John L; Bradley, Mark A, and Gibb, J. The Sacramento activity-based travel demand model: estimation and validation results. In European Transport Conference, 2006.

Breiman, Leo. Random forests. Machine learning, 45(1):5–32, 2001.

Briem, Lars; Mallig, Nicolai, and Vortisch, Peter. Creating an integrated agent-based travel demand model by combining mobiTopp and MATSim. Procedia Computer Science, 151:776–781, 2019.

Castiglione, Joe; Bradley, Mark, and Gliebe, John. Activity-based travel demand models: a primer. Number SHRP 2 Report S2-C46-RR-1. 2015.
Dianat, Leila; Habib, Khandker Nurul, and Miller, Eric J. Modeling and forecasting daily non-work/school activity patterns in an activity-based model using skeleton schedule constraints. Transportation Research Part A: Policy and Practice, 133:337–352, 2020.

Do, Loan NN; Taherifar, Neda, and Vu, Hai L. Survey of neural network-based models for short-term traffic state prediction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 9 (1):e1285, 2019.

Drchal, Jan; Čertický, Michal, and Jakob, Michal. Data-driven activity scheduler for agent-based mobility models. Transportation Research Part C: Emerging Technologies, 98:370–390, 2019.

Eberhard, LK. A 24-hour household-level activity-based travel demand model for the GTA (Ontario). 2003.

Ghasri, Milad; Rashidi, Taha Hossein, and Waller, S Travis. Developing a disaggregate travel demand system of models using data mining techniques. Transportation Research Part A: Policy and Practice, 105:138–153, 2017.

Goodfellow, Ian; Bengio, Yoshua; Courville, Aaron, and Bengio, Yoshua. Deep learning, volume 1. MIT press Cambridge, 2016.

Guo, Cheng and Berkhahn, Felix. Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737, 2016.

Hilgert, Tim; Heilig, Michael; Kagerbauer, Martin, and Vortisch, Peter. Modeling week activity schedules for travel demand models. Transportation Research Record, 2666(1):69–77, 2017.

Kato, Kenji; Matsumoto, Shoji, and Sano, Kazushi. Microsimulation for commuters’ mode and discretionary activities by using neural networks. In Traffic And Transportation Studies (2002), pages 1290–1297. 2002.

Kitamura, R; Lula, CV, and Pas, EI. Amos: An activity-based, flexible and behavioral tool for evaluation of TDM measures. In Proceedings, 1993.

Kitamura, Ryuichi; Pas, Eric I; Lula, Clarisse V; Lawton, T Keith, and Benson, Paul E. The sequenced activity mobility simulator (SAMS): an integrated approach to modeling transportation, land use and air quality. Transportation, 23(3):267–291, 1996.

Konduri, Karthik C; Ye, Xin, and Pendyala, Ram M. Probit-based discrete–continuous model of activity choice and duration with history dependency. Transportation Research Record, 2156(1):17–27, 2010.

Koushik, Anil NP; Manoj, M, and Nezamuddin, N. Machine learning applications in activity-travel behaviour research: a review. Transport Reviews, 40(3):288–311, 2020.

Liaw, Andy and Wiener, Matthew. Random forest package, 2012.

Lu, Yang; Adnan, Muhammad; Basak, Kakali; Pereira, Francisco Câmara; Carrion, Carlos; Saber, Vahid Hamishagi; Loganathan, Harish, and Ben-Akiva, Moshe E. SimMobility mid-term simulator: A state of the art integrated agent based demand and supply model. In 94th Annual Meeting of the Transportation Research Board, Washington, DC, 2015.

Mikolov, Tomas; Chen, Kai; Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

Miller, Eric J. Modeling the demand for new transportation services and technologies. Transportation Research Record, 2658(1):1–7, 2017.

Miller, Eric J. Agent-based activity/travel microsimulation: what’s next? In The Practice of Spatial Analysis, pages 119–150. Springer, 2019.

Miller, Eric J and Roorda, Matthew J. Prototype model of household activity-travel scheduling. Transportation Research Record, 1831(1):114–121, 2003.
Miller, Eric J; Vaughan, James; King, David, and Austin, Matthew. Implementation of a next generation activity-based travel demand model: the Toronto case. In Presentation at the Travel Demand Modelling and Traffic Simulation Session of the 2015 Conference of the Transportation Association of Canada, 2015.

Nurul Habib, Khandker. A comprehensive utility-based system of activity-travel scheduling options modelling (custom) for worker’s daily activity scheduling processes. Transportmetrica A: Transport Science, 14(4):292–315, 2018.

Ozonder, Gozde and Miller, Eric J. Longitudinal microeconomic and machine learning models of skeletal activity generation. Travel Behaviour and Society, 23:40–64, 2021.

Rasouli, Soora and Timmermans, Harry. Activity-based models of travel demand: promises, progress and prospects. International Journal of Urban Sciences, 18(1):31–60, 2014.

Roorda, Matthew J; Miller, Eric J, and Habib, Khandker MN. Validation of tasha: A 24-h activity scheduling microsimulation model. Transportation Research Part A: Policy and Practice, 42(2): 360–375, 2008.

Veres, Matthew and Moussa, Medhat. Deep learning for intelligent transportation systems: A survey of emerging trends. IEEE Transactions on Intelligent Transportation Systems, 2019.

Yasmin, Farhana; Morency, Catherine, and Roorda, Matthew J. Assessment of spatial transferability of an activity-based model, TASHA. Transportation Research Part A: Policy and Practice, 78: 200–213, 2015.

Zheng, Zibin; Yang, Yatao; Liu, Jiahao; Dai, Hong-Ning, and Zhang, Yan. Deep and embedded learning approach for traffic flow prediction in urban informatics. IEEE Transactions on Intelligent Transportation Systems, 20(10):3927–3939, 2019.