Journal of Ambient Intelligence and Humanized Computing
https://doi.org/10.1007/s12652-020-02855-6

ORIGINAL RESEARCH

Using neural nets to predict transportation mode choice: Amsterdam network change analysis

Ruurd Buijs[1] · Thomas Koch[2] · Elenna Dugundji[1]

Received: 15 July 2020 / Accepted: 5 November 2020

Abstract
In the Amsterdam metropolitan area, the opening of a new metro line along the north–south axis of the city has introduced a significant change in the region's public transportation network. Mode choice analysis can help in assessment of changes in traveler behavior that occurred after the opening of the new metro line. As it is known that artificial neural nets excel at complex classification problems, this paper aims to investigate an approach where the traveler's transportation mode is predicted through a neural net, trained on choice sets and user specific attributes inferred from the data. The method shows promising results. It is shown that such models perform better when it is asked to predict the choice of mode for trips which take place on the same underlying transportation network as the data with which the model is trained. This difference in performance is observed to be especially high for trips from and to certain areas that were impacted by the introduction of the north–south line, indicating possible changes in behavioural patterns, entailing interesting possible directions for further research.

Keywords Transportation mode choice · Artificial neural nets · Machine learning · Public transportation network change · Travel behaviour

1 Introduction

In 2018, the region of Amsterdam witnessed the most comprehensive structural change of their public transportation network in more than a century. The opening of a new metro line serving the entire length of the north–south axis of the city has led to rigorous changes in the existing tram and bus network. Analyzing the behaviour of transport movements by individuals is an effective way to assess the impact of a rigorous network change. A standard approach carried out to model transportation behavior is discrete choice analysis, using statistical techniques for parameter estimation. Other approaches involve simulation (Li and Xu 2019).

This study explores a relatively new method that can contribute to behavioral analysis of transport movements, using a novel data set collected in Amsterdam with an app on smartphones that automatically recognizes activity signals. Thakur and Biswas (2020) present a comprehensive survey of smartphone sensor based human activity monitoring and recognition techniques using machine learning and deep learning. Considering the fact that artificial neural networks are extremely capable of performing well when assigned complex classification tasks (Long 2020), we consider possible application of this technique within the field of behavioral analysis in transportation.

The paper is structured as follows: Firstly, a brief literature review is presented, followed by a description of the Amsterdam case study. Next, the data set used and methodology to process it will be discussed. After that, suggestions are made for a neural net implementation to classify mode choice. Finally, results are presented assessing both the proposed methodology and to what extent one might say that behavioral patterns are affected by the network change. Based on these results and the discussion of the proposed methods, recommendations for future research are made.

✉ Thomas Koch
thomas.koch@cwi.nl

[1] Vrije Universiteit Amsterdam, De Boelelaan 1111, 1081 HV Amsterdam, The Netherlands

[2] Centrum Wiskunde en Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands

Published online: 04 February 2021

2 Background

For several decades, discrete choice modeling has been dominated by statistical models, such as the logit and probit models. This paradigm dates as far back as the 1970s (McFadden 1973) and 1980s (e.g. Coslett), and is an approach that is built upon in more recent publications [e.g. Guevara and Ben-Akiva (2013)].

In 2003, Vythoulkas and Kotsopoulos applied a different approach to this problem. Trying to beat the results obtained by conventional statistical methods, they introduced a neural net structure based on fuzzy set theory to model discrete choice behaviour in transportation. They then applied this algorithm to a small data set that obtained by the Dutch Railways. The data used in this study was collected from surveys and related to transportation mode alternatives in the Dutch city of Nijmegen (Bradley and Gunn 1990; Morikawa 1989). The proposed model performed slightly better in the case study than a logit model constructed for the same purpose. One of the key underlying assumptions in this study was that travelers decide based on simple underlying rules rather then complicated functions $F : X \rightarrow Y$. Those rules were then incorporated into a neural net system.

In the context of Market Share forecasting, a study has been carried out by Agrawal and Schorling in 1996, regarding a comparison between the ANN and multinomial logit method. In brand choice analysis, a hybrid model has been suggested by Bentz and Merunka (2000).

Recently, van Cranenburgh and Alwosheel (2019) have been among a growing kernel of researchers to again use neural nets in practice in a similar context. In their paper, they describe how an ANN can be trained to investigate decision rule heterogeneity. Their method trains a multinomial classification network to assign users to one of four quintessential decision rules, based on theoretical choice data, where each user was presented a series of choices in order. The results of each user are then combined and fed to the network that classifies the user into one of the four categories. A good overview of papers that have applied neural nets and other machine learning (ML) techniques to the problem of transportation mode choice can be found in the literature review by Hillel et al. (2019).

Currently, one of the particular aims of some of the works that apply specifically to neural net structures to study human choice or behaviour is to focus on the interpretation of the proposed model. One interesting paper from a different field focused mainly on extracting decision rules from data using a neural net has been presented by Hayashi et al. (2010). They make use of the *Re-RX* algorithm to extract rules from a pruned neural network. The data set used by the authors contains user characteristics and preferences on eating behaviour, which is also an application of neural nets in a behavioral context.

Although neural nets are essentially black box algorithms, it is possible to look beyond merely assessing the predictive power of ML models, using them for the same kinds of analyses that are commonly applied when applying conventional logit models. A study by Wang and Zhao (2019) focuses on the interpretability of a deep neural net (DNN), proposing a way to numerically compute economical information such as choice probabilities and probability derivatives from the DNN.

Another way of developing a better understanding of the workings of neural nets when used in the context of mode choice, trying to introduce some conventional knowledge from the field into the model by modifying the architecture, is proposed by Wang et al. (2020a). They showed that using a sparse neural net architecture, based on underlying assumptions of the random utility mixing (RUM) model, could lead to significantly better results than using a generic fully connected DNN. To get a better interpretability Wang et al. (2020a) visualize choice probability functions and compute elasticity coefficients in DNN models using numerical simulations. A special type of neural net, called multitask learning deep neural networks (MTLDNNs) is applicable to choice modelling situations where it is useful to combine data from different sources, such as bridging the gap between combining revealed and stated preference data, with the capabilities of automatic feature learning that DNNs possess (Wang et al. 2020b).

In earlier research, we introduced a novel way of extracting user-specific features from choice set data and applied this data to a neural net model for classifying mode choice, and tested this method on a relatively small subset of an Amsterdam data set (Buijs et al. 2020). In this study, we aim to extend the application of this method to the scope of the entire data set (see Sects. 3 and 5 for more context about this data set). We will assess how well our model deals with the changes in the network, and discuss what information regarding the network change can be inferred from our model.

3 Case study

In Amsterdam, a new metro line has been opened in 2018 serving the north–south axis of the Dutch capital. In order to improve integration of the new metro line, the existing transportation network underwent significant changes. A large number of the bus and tram lines were re-routed to connect different areas. One particular aim of these changes was to create more east–west links, that connect to the new north–south line at one of the metro stations in the centre of the city. The design moved away from a network that was heavily focused on lines to and from the central train station to a network where instead the new north–south metro line forms a spine.[1] An abstract visualization of the network

[1] https://www.vervoerregio.nl/pagina/20160131-ov-lijnennetvisie.

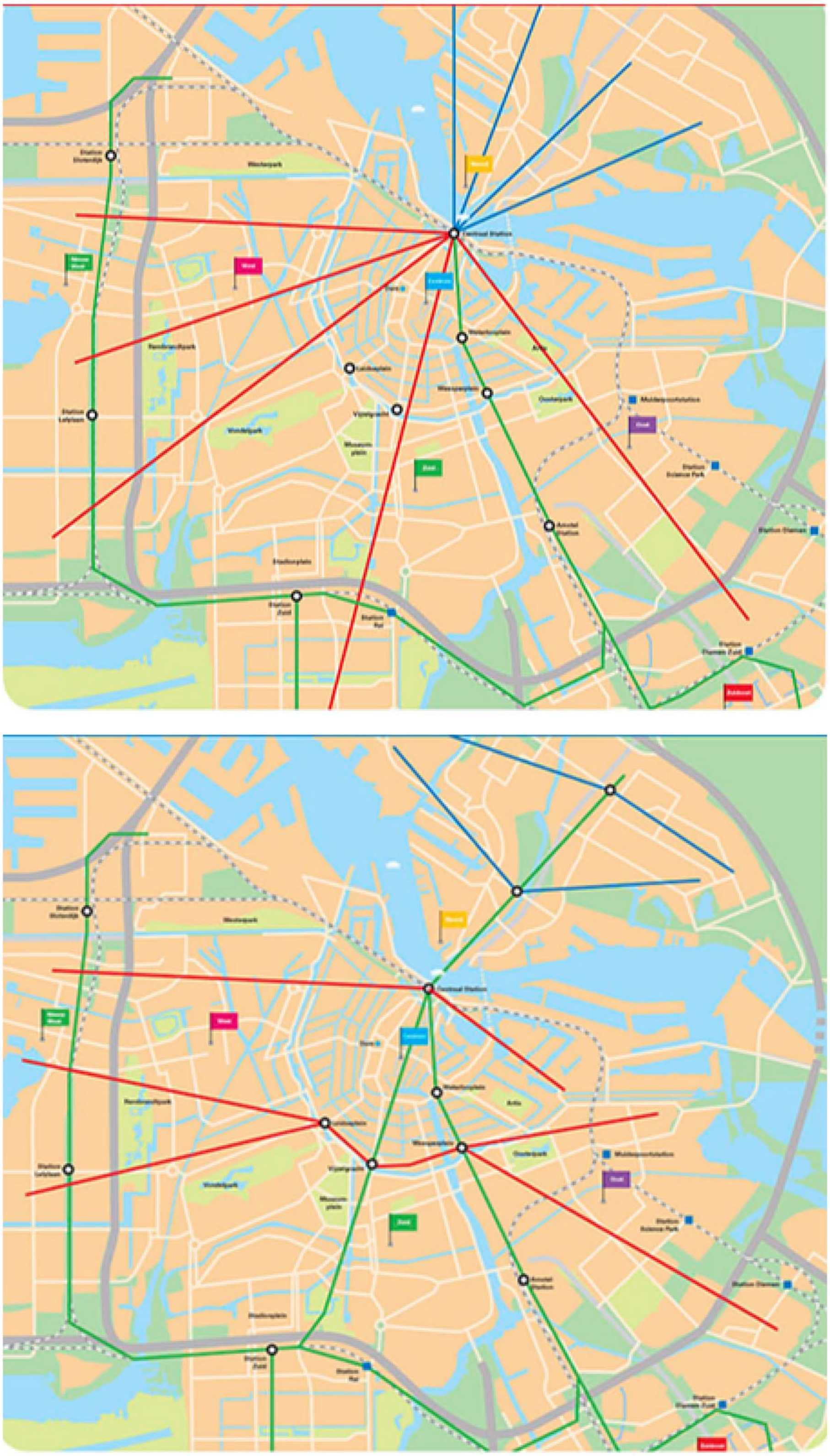

Fig. 1 The Amsterdam transportation network change visualized by GVB. Conceptually, the upper map shows the old network structure centered around the Central Station, which serves as a hub. The lower map shows the new network structure where the new north–south metro line (the central green line) forms a spine

change is shown in Fig. 1, which was published by the GVB, the main local public transportation provider in the municipality of Amsterdam.[2] For many inhabitants of the city, these changes in the network meant that their personal travel itineraries were affected. At the same time, car drivers were also confronted with the introduction of new restrictions in the inner city and around Amsterdam Central train station to avoid through-traffic in the inner city.

Policy makers from the regional transportation authority in Amsterdam and the city of Amsterdam are keen to assess the impact of the introduction of this new network. For this analysis, data was collected using a smart phone GPS application that was installed by a panel of participants recruited via several existing survey panels. Additional participants were recruited on the street. The smart phone application tracks the activities of the user in the background of the smart phone using sensors on the phone such as GPS and acceleration sensors.

4 Data

4.1 Choice set generation

In order to explore what other transportation modes were available for each user for each of their observed choices, we generated a number of alternatives using an open source library developed by Conveyal , R5—rapid realistic routing on real-world and reimagined networks[3]. This router has been used previously by other studies such as Conway et al. (2017) and de Freitas et al. (2019). R5 is able to return a large set of feasible, fast routes within a given time-range. This permits a more realistic assessment about accessibility than would be possible using estimations based on fixed frequencies.

We used two separate general transit feed specification (GTFS)[4] data files to feed the router with the correct timetable before and after the opening of the metroline. For the street network we used a temporally appropriate extract from OpenStreetMap. Additionally, we directed R5 to generate transit routes specifically including and excluding metro. For each observation and alternative we then categorized a route into one of seven different non-overlapping strata:

1. Walk trip (generated if walking stays under 60 min).
2. Car trip (generated if destination is reachable by under 60 min).
3. Bicycle trip (generated if bicycling stays under 60 min).
4. Transit trip, with use of train and metro.
5. Transit trip, with use of train (no metro).
6. Transit trip, with use of metro (no train).
7. Transit trip, not using train or metro.

To generate choice sets, we looked at the observations and categorized each observation with a stratum and subsequently took the best (fastest) from the alternatives that fit each alternatives. In some cases not each alternative was available, for example walking is not always an option if the distance between origin and destination is long. It could be possible to address the unavailability of walking in the loss function using the study by Wang et al. (2020a).

4.2 Feature engineering

From the observations and the generated alternatives we collected a number of explanatory variables as listed in Table 1. We used a walking speed of approximately 5 km/h and a bicycling speed of 14.4 km/h. We based our car speed on the speed limits in OpenStreetMap.

4.3 Data filtering

The entire GPS dataset consists of 106,647 trip entries from 712 users. The GPS data is collected during three time periods spanning about one month each: the first period of data collection took place in June and July and part of August 2018, largely before the introduction of the new north–south metro line, and the second and third period of data collection took place after the north–south line was opened: in September and October 2018 and June and July 2019, respectively.

Not every trip can eventually be used in a final data set to perform a mode choice analysis on. It was found that several trips in the data turn out to be tours or round trips. As these trips do not consider movement from A to B, these are not suitable to include for a mode choice analysis. Another example of entries that had to be filtered out in some cases, are non-public transport trips, for which different parts of the trip were traversed by a different mode, as these type of trips have not been generated as choice alternatives. These were dealt with as follows:

- Trips with the mode combination of car and cycling are all discarded.
- Trips with mode combination of car and walking are discarded if the duration of the walking part exceeds the duration of the part traversed by car. Otherwise, the trip is considered to be similar enough to a ‘car-only’ trip, that it is concerned as such, and the duration of the walk is set to 0.

[2] https://ucarecdn.com/7bad75fd-27ad-4ee6-96c4-b51b2f400619/GVB_Vervoerplan_Amsterdam_2018.pdf.

[3] https://github.com/conveyal/r5.

[4] http://gtfs.ovapi.nl/nl.

Table 1 Variables collected for choice set

Variable	Description
Group id	A unique identifier referring to a single trip from origin to destination; generated trips corresponding to the observed trip have the same groupid and also refer to a specific person and date
Strata	Categorical variable that indicates the transportation mode of a (generated or actual) trip:
	1 for walking
	2 for traveling by car
	3 for traveling by bicycle
	4 for traveling by public transportation with use of metro and train
	5 for traveling by public transportation with use of train (no metro)
	6 for traveling by public transportation with use of metro (no train)
	7 for traveling by public transportation without use of metro and train
Access mode	Categorical variable that indicates the mode of access to public transportation (i.e. mode of transportation used to reach the bus stop/train station such as walk, bicycle or car)
Egress mode	Categorical variable that indicates the mode of egress from public transportation (i.e. mode of transportation used to reach the destination after leaving the bus stop/train station)
Start time	Time that trip started at origin
End time	Time that trip ends at destination
Transfers	Number of transfers on the public transportation part of this trip
Distance	Total distance of trip
Bicycle distance	Total distance of trip traversed on bicycle
Car distance	Total distance of trip traversed by car
Walk distance	Total distance of trip traversed on foot
Bicycle duration	Total duration of trip that is traversed on bicycle
Car duration	Total duration of trip that is traversed by car
Walk duration	Total duration of trip that is traversed on foot
Waiting time	Total time spent waiting on public transportation if applicable

– Trips with mode combination of walking and cycling are discarded if the duration of the walking part is longer than 30% of the duration of the part traversed by bike. Otherwise, the trip is considered to be similar enough to a ‘bike-only’ trip, that it is concerned as such, and the duration of the walk is set to 0.

For some trips, the difference between the observed trip duration and the theoretical trip duration, determined by the duration of the generated trip with corresponding mode, is rather large, ranging from a factor 2 to a factor 10 difference. These differences may have various reasons, some of which could be a ground for excluding the trips from the data. Trips that manifested such a difference and originally consisted of multiple segments (except for public transport trips), are assumed to be an indirect trip from A to B, thus would not be of interest, and are filtered out likewise.

In addition to this, it is assumed that an artificial neural net will be able to distinguish between ‘real’ observed data and generated data if the characteristics of the observed data are too far apart from the range of values that occur in the generated data characteristics. For this reason, all data entries concerning trips spanning more than 100 min, are discarded as well. Table 2 shows the reasons why some data was discarded and how many entries were involved for each reason. The final dataset consists of 76,712 trip entries, concerning 709 users. For each of these entries, at least one alternative trip has been computed in which a different mode was used.

Table 2 Amount and fraction of data that were filtered out due to various reasons

Original data set	106,647 entries (100.0%)
No alternatives could be generated	9894 entries (8.5%)
Trip is a round trip/tour	9784 entries (9.2%)
Significant parts are traversed by different modes	5848 entries (5.5%)
Trips that were likely indirect	3524 entries (3.3%)
Trips that took longer than 100 min	885 entries (0.8%)
Final data set	76,712 entries (71.9%)

5 Methodology

This section gives an account of what operations and techniques have been used in order to make mode choice predictions based on the data set. Since we opt for a machine learning approach, most methodological decisions are made such that input is created which is suitable to train a machine learning model on.

5.1 Data preparation

Within the process of data preparation, three main steps can be distinguished: combining data, selecting features, and splitting data into a train, validation and test partition. We will briefly discuss all three of these steps.

Combining data Having obtained a filtered data set, the observed data was combined with the data concerning generated alternatives. Initially, duplicates exist in this merged data set, i.e. for a single trip, there can be two routes with the same transportation mode: one that corresponds to the trip that was made originally by the user, the other one is the generated trip having the same transportation mode. In order for the data to be used in a machine learning model, one of the two entries must be deleted, so that for each trip only one option per transportation mode remains. While there are certainly relative advantages to restricting the feature data to one source, most notably that an ML-model will not pick up any bias from the fact that different data sources are combined, it needs to be noted that redeeming the features from the observed trip data means losing valuable information, possibly causing the model to be a worse reflection of reality. Therefore, it is opted for to preserve this data and discard the generated duplicates. Reduction of bias present in the data will be taken into account specifically when selecting features that will serve as input for the ANN.

Feature selection In order to fully benefit from the power of ANNs and to get meaningful results, it is necessary to carefully select the features that will eventually be fed to the ANN. The most important reason for this is to reduce the risk of having 'false predictors' as much as possible. These arise when the neural net would be able to distinguish 'real' observed data from generated data within a choice set. From all features initially present in the data, the most reliable predictors will likely be *transfers*, *distance*, *bicycle_distance*, *car_distance* , *walk_distance* , *bicycle_duration* , *car_duration*, *walk_duration* and *waiting_time*. These features together form an initial selection of input features for the ANN. Other features have not been considered as direct input, either due to the nature of the feature or due to the feature having little explanatory value. However, when comparing the generated and observed data, it was found that a substantial number of entries in the generated data had a record of an abnormally high waiting time, resulting in an abnormally high trip duration as well. It is assumed that this is caused by users doing activities not related to transportation at a station. Because of this, it was opted to exclude the feature *waiting_time* from the data and to adjust the feature *duration* accordingly. This operation was performed before the data filtering took place as described in Sect. 4.3. It was also observed that the features related to distance and the features related to duration display different correlation patterns in the observed and generated data. This is illustrated in Fig. 2. This observation indicates that feeding a choice set including both distance and duration related attributes, might also introduce an unwanted form of bias in the data. Therefore, we choose to discard all attributes related to distance, as it is known that duration plays a more important role in mode choice considerations of individuals.

Splitting data A common practice within the field of machine learning is to split the data before it is being used. In supervised learning (training the model to predict a known target), the data is usually split into a train set and a test set. The former is used to train the model, whereas the latter is used to evaluate model performance on a batch of unseen data. From the training set, some data is usually set apart for validation. This part of the training set is not used to train the model, but to check whether the model does not overfit. This would be the case if the model performed significantly worse on the validation set than on the training set. The data for each user is set to follow roughly this distribution over the three sets:

– Training set: 50%.
– Validation set: 20%.
– Test set: 30%.

In order for the model to be able to take into account individual user preference characteristics, it is important that data from all users is contained in the train set. Some users who have only one entry, will as a consequence only appear in the train set. The final sizes of the three partitions are as follows:

– Training set: 38,539 entries (including all single-entry users); 50.2%.
– Validation set: 15,156 entries; 19.8%.
– Test set: 23,017 entries; 30.0%.

The training set is the only set for which the target variable (in this case strata) is not hidden. Hence, all operations used for setting up the model that are described in the following sections, apply to the training set only.

5.2 Classifying choices

It is clear that individual users have different preferences. Those individual preferences should be taken into account in any predictive model, as becomes clear in the literature. Because of the nature of the data (panel data), where multiple trip entries will correspond to the same user, it makes sense to include features to the input of the ANN model that

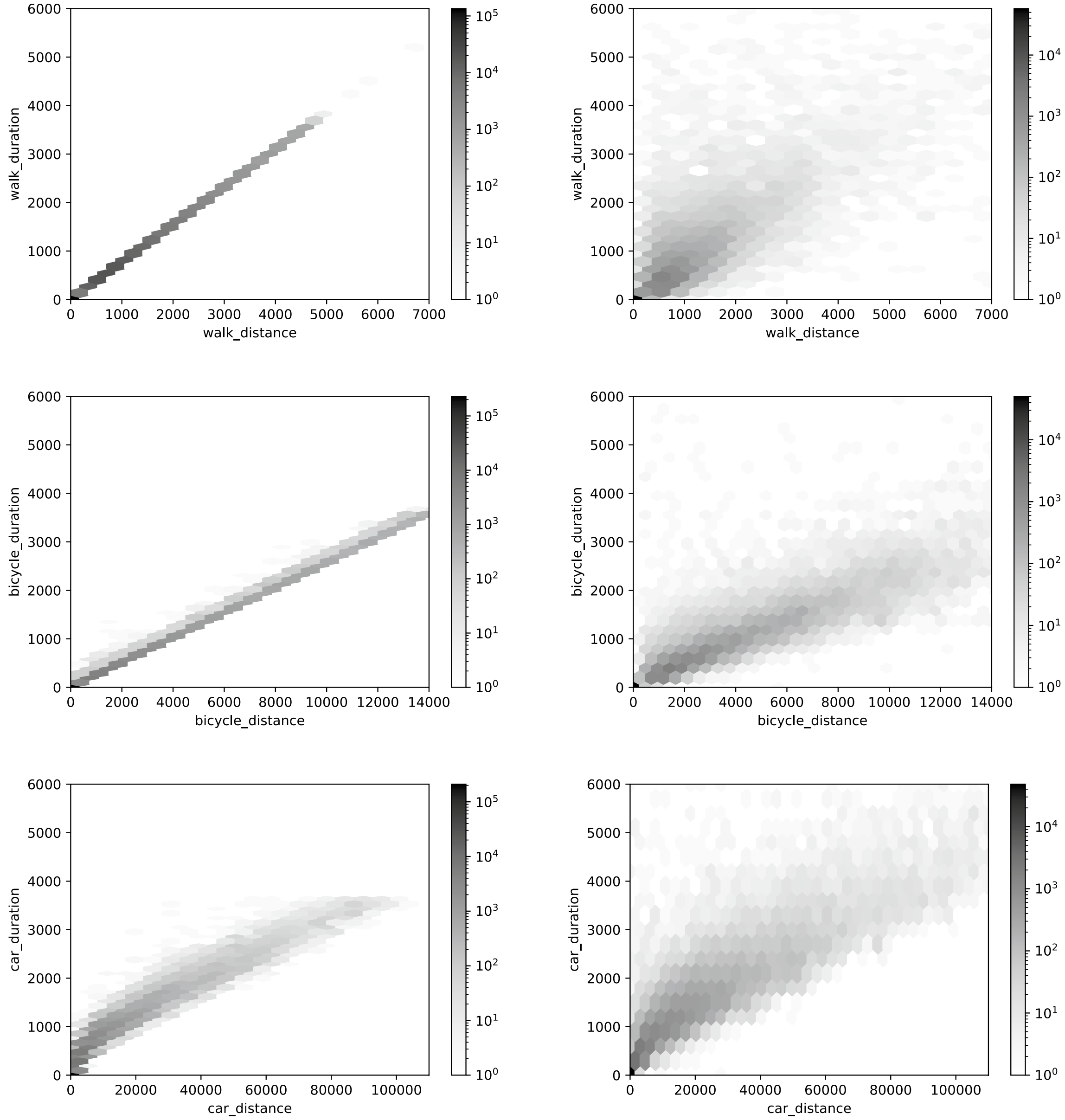

Fig. 2 Correlation between distance and duration related attributes in the generated data (left) and observed data (right)

specifically concerns individual user preference. The literature suggests different methods in order to classify users as decision-makers, yet all are based on assumptions. The most important notion that comes clear from this is that different individuals decide differently, and can be divided into classes or groups that share similar decision characteristics. Regardless of what the underlying decision functions might be (it will be nearly impossible to approximate them all due to many users having relatively few training entries), it is possible to divide the observed mode choices into different classes based on comparative measures regarding the alternative modes. The comparative measures have been computed by normalizing the attributes *transfers, duration, bicycle_duration, walk_duration* and *car_duration* within each choice set individually and extracting solely the normalized values corresponding to the chosen mode. In this way, for each trip, a singular value between 0 and 1 is obtained for each attribute, where 0 is obtained if the alternative with the lowest value of an attribute is chosen and 1 is obtained if the alternative with the highest value for this attribute is chosen.

k-Means clustering In order to subdivide the trips into different groups without having a clear target to aim for, a method called k-means clustering is used (Steinley 2006). K-means clustering is a relatively simple and intuitive clustering method. Although multiple clustering methods exist that can deal with specific types of problems, like Hierarchical clustering Murtagh and Contreras (2012) and DBSCAN Schubert et al. (2017), k-means clustering is fairly suitable for a relatively simple clustering task like the one at hand. With this particular data set, there are two main challenges in terms of clustering:

1. The most obvious underlying structure is the strata classification itself, which tells something about the choice, yet is not the particular information structure we are looking for.
2. Some of the normalized variables may be correlated, for example *walk_duration* and *duration*.

In order to overcome these obstacles, the following solutions have been suggested:

- Choose the number of clusters *k* such that *k* exceeds the number of significantly different modes (In this case, 4: walking, cycling, car, and public transportation) by a comfortable margin (but not higher than necessary) to create substantial 'classes' that are composed of entries from different strata. In this case, *k* was set to 10.
- Perform principal components analysis prior to performing k-means clustering (Jolliffe and Cadima 2016). This method creates linearly independent vectors (i.e. vectors that have covariance 0). The resulting vectors are then used as input for the k-means classification algorithm.

Principal component analysis Our next step is to gather information about users using the obtained choice classification. Based on the outcomes of the labeling phase described earlier, each user now has a characteristic 'label distribution'. The relative frequencies of the different choice types are stored in a DataFrame for each user. Again, PCA is conducted to reduce these values to a set of five vectors that aim to capture the users' behavior and taste.

In order to assess the usefulness of applying PCA here, we trained 75 models with hyperparameters randomly selected from the hyperparameter space as described in Table 4, with PCA applied to the user specific features and ranked the models based on validation loss. After that, the same procedure was applied to assess the performance of models that were trained based on input where PCA was not applied to the user specific features. This was done once applying an early stopping condition of 2 epochs, and then repeated once more applying an early stopping condition of 5 epochs.

The results of this procedure are shown in figure is assessed in Fig. 3. The figure indicates that for the top-ranked models, models where PCA was applied to the user specific input features have a slightly lower loss than models where this was not the case. The reported accuracy values of the highest ranked models are roughly the same for both the models where PCA was applied to the input and the models where PCA was not applied.

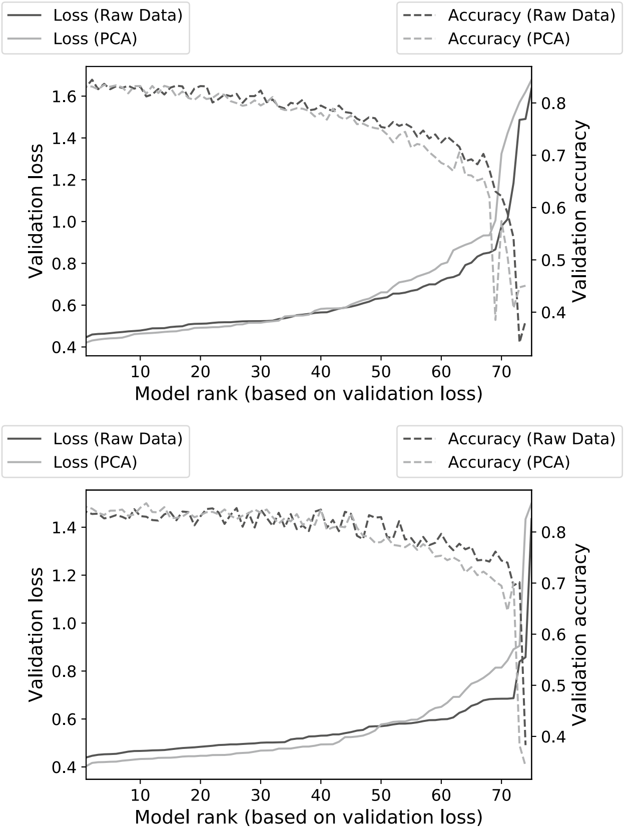

Fig. 3 Ranked performance of models (in terms of sparse categorical crossentropy loss and prediction accuracy) of ANNs with full user-specific input feature data (i.e. ten distinct features) and input where user-specific features are reduced in dimension (i.e. five distinct features). In the upper plot, early stopping is applied when validation loss has not decreased for 2 consecutive epochs, whereas the lower plot shows models that were trained with the application of early stopping condition when validation loss has not decreased for 5 consecutive epochs

5.3 Prediction

In order to predict which strata will be chosen in different situations for different users, we feed the acquired data concerning trips made, alternatives, and user preference to an artificial neural network (ANN).

Neural networks for multiclass classification Neural nets have extended the scope of machine learning beyond linear models. A feedforward neural network consists of one or more hidden layers, that each consist of a number of nodes. In a basic, fully connected neural net, each node gets input from all nodes in the previous layer, and outputs to all nodes in the next layer. Each layer is assigned a type of activation function, i.e. a function that generates the output of a node based on the input from a previous node. Commonly used non-linear activation functions include *sigmoid, tanh* and *reLU* functions, with respective domains (0,1), (–1,1) and [0,∞). All nodes except for those in the output layer have *reLU* as activation function, which is the most commonly used activation function nowadays. For a multiclass classification, the method that will be used in this study, another activation function is usually used in the final (output) layer. The so-called *softmax* activation takes exponents of the output of the previous layer and scales them such that they sum to 1. The network is trained by a back-propagation algorithm that works upon a chosen loss function. Commonly used loss functions include least-squares and cross-entropy loss. The network is trained with *rate* η. After each iteration the weights are adjusted in the direction of the gradient of the chosen loss function, based local derivatives and the chain rule. The *training rate* η is the parameter determining the magnitude of the change of weights after each iteration. Choosing a higher value for η increases the training speed but may result in not being able to find optimal values. It is possible to train a neural net with a constant value for η for all parameters, or with adaptive η, meaning that η can be different for each parameter update. Different optimizers have been introduced that make use of adaptive η, like Adagrad (Duchi et al. 2011), RMSProp (Tieleman and Hinton 2012) and Adam. In this study, we use Adam as optimizer, since it is "robust and well-suited to a wide range of non-convex optimization problems in the field machine learning" and the method is computationally efficient, which is convenient for large data sets (Kingma and Ba 2014).

Data shape and processing In this case, the shape of the individual data entries fed to the network is a table of 7 by 10; ten attribute values for each of the seven different alternative strata. Each *groupid* in the training set corresponds to one mode choice scenario and therefore to one of these tables. The values of the ten attributes are not always available for every stratum as not every mode of transportation is possible on every trajectory. If no route was generated for a certain stratum in a certain scenario, all attributes corresponding to this stratum (including user-specific attributes that are essentially known even for alternatives that do not have a route generated) are set to 0 in this scenario, and will be passed to the network as such. This is done because the network requires the data entries passed to it to have a consistent shape (*Stratum 1= row 1, Stratum 2 = row 2 etc*), while in the meantime strata without a generated route option must not affect the working of the model. Before creating the data entries, all data has been normalized using the minimum and maximum values of the entire combined dataset.

6 Results

6.1 Classifying choices

Table 3 shows the composition[5] of the clusters that result from the k-means clustering algorithm with $k = 10$. After inspection of the clusters, a description has been added based on the values of the used comparative measures observed among the choices in each cluster.

6.2 Prediction

For this paper, we trained 75 ANNs with different hyperparameter configurations. All models are trained for a maximum of 100 epochs, where early stopping is applied if the loss on the validation set does not decrease for 2 consecutive epochs, to prevent the model from overfitting on the training data. The hyperparameters concerning architecture were randomly chosen for each run from a pre-determined set of possible values, as given in Table 4. The models are trained using the Adam optimizer which was mentioned in Sect. 5.3, with fixed α, β_1, β_2 and ϵ (Table 4). Based on the validation loss, the best 5 models were selected, as can be seen in Table 5. For these models, we looked into the confusion matrices for the classifications. Table 6 shows the confusion matrix for the highest ranked model, based on the predictions made on the test set. As we see, the model performs very well on choice sets where the actual mode was car. Decent scores are also reported for all other Strata, except for Stratum 4. This is likely due to the fact that this is the smallest class containing only 96 entries in the training set.

6.3 Analysis in the light of the network change

In addition to the assessment of our model in general, we have explored the effect of training our model on the different partitions of the collected data. For this, we compared two different settings: One where the entire model was trained and validated based only on data relating to trips made before the introduction of the north–south line (this is data stemming from the first collection period), one where the model was trained and validated based only on data relating to trips made after the north–south line was introduced (data from the second collection period). These models were then tested on data from the final collection period. As a reference, we also trained a model with roughly the same amount of data (about 20,000 entries) sampled randomly

[5] When running the model multiple times, the exact composition of the clusters will slightly deviate due to the nature of the clustering algorithm. The nature and sizes of the identified clusters however have shown to be consistent over multiple runs.

Table 3 Composition and description of clusters obtained by k-means clustering after applying PCA

	Stratum							
Cluster	1	2	3	4	5	6	7	Total
0	1270	0	0	0	5	6	3	1284
1	0	13,412	0	0	0	0	0	13,412
2	0	0	9994	0	0	0	0	9994
3	4273	0	0	0	26	5	34	4338
4	0	0	0	34	558	353	372	1317
5	0	0	3472	0	0	0	0	3472
6	0	707	0	0	0	0	0	707
7	0	0	0	2	269	88	578	937
8	0	0	0	60	324	404	548	1336
9	273	0	0	0	302	264	903	1742
Total	5816	14,119	13,466	96	1484	1120	2438	38,559

Cluster	Size	Description
0	1284	Trips involving walking which are not the most time-consuming alternative
1	13,412	Trips by car which are generally (among) the quickest alternative(s)
2	9994	Trips by bicycle which are generally among the quickest alternatives
3	4338	Trips where relatively more walking is involved, generally the slowest
4	1317	Public transportation trips with relatively many transfers and (relatively) longer walk involved
5	3472	Trips by bicycle which are generally among the slowest alternatives
6	707	Trips by car which are generally among the slowest alternatives
7	937	Relatively slow public transportation trips with relatively few transfers
8	1336	Public transportation trips with relatively many transfers with relatively fewer walking
9	1742	Relatively fast trips without using car or bicycle or making a lot of transfers

Table 4 Hyperparameter configuration for the ANN

Hyperparameter	Values to sample from
Hyperparameters: randomly sampled for each run	
Number of hidden layers (excluding output layer)	{1, 2, 4, 6, 8, 10}
Number of nodes in each hidden layer	{5, 10, 25, 100, 200, 500, 1000, 2000}
Batch size	{8, 32, 64, 128, 512, 2048, 8192, 38539}
Hyperparameters: fixed for every run	
Activation function for every layer except output layer	*reLU*
Activation function output layer	*Softmax*
Number of nodes output layer	7
Loss function	Sparse categorical Cross-Entropy Loss
Learning rate α	0.001
β_1	0.9
β_2	0.999
ϵ	10^{-7}

Table 5 Results and characteristics of the five best models selected on validation loss

Rank	Number of hidden layers	Number of hidden nodes	Batch size	Number of epochs trained	Validation loss	Validation accuracy
1	4	200	128	17	0.422	0.843
2	4	200	64	13	0.433	0.834
3	4	1000	512	11	0.438	0.835
4	4	100	128	16	0.440	0.834
5	6	100	32	16	0.440	0.835

Table 6 The confusion matrix of the highest ranked model

		Predicted stratum							
		1	2	3	4	5	6	7	Predicted correctly (%)
Actual stratum	1	2883	148	490	0	2	6	7	81.5
	2	164	7866	635	2	65	97	40	88.7
	3	744	714	6323	0	6	69	33	80.1
	4	0	66	2	25	8	5	2	23.1
	5	2	190	31	0	697	3	0	75.5
	6	1	44	39	0	5	492	5	84.0
	7	9	145	97	0	1	23	831	75.1

from all three collection periods and tested this against a test set containing a random sample of the remaining data. For all of these settings, ten models were run with four hidden layers, 200 hidden nodes per layer and batch size 128. This approach is mainly used to obtain a relative insight into how much the classification task will become 'different' when the underlying network is different. If the assumption that the data on which the model is trained and the data on which the model is tested are i.i.d. is not valid, the model will not be able to perform as well on the test data in comparison to situations where this assumption does hold. This principle is related to the theory underlying transfer learning. An interesting further research direction, yet outside the scope of this paper, would be to try to further generalize our model using the transfer learning techniques discussed by Yosinski et al. (2014), by subjecting our model to two similar waves of data with a different underlying network.

Table 7 shows the differences in (relative) cluster sizes for the clusters that were obtained based on the two different collection periods. We can see that most clusters formed

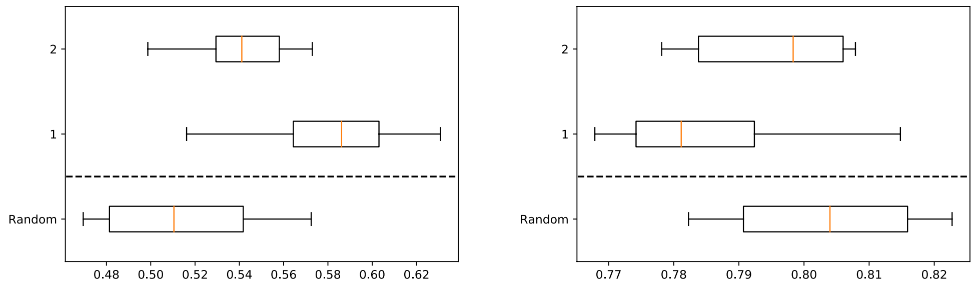

Fig. 4 The loss (left) and accuracy (right) of models trained on data collected in the first and second period, when tested against data collected in the final period, compared to the loss and accuracy of a model with randomly sampled train and test set of roughly equal size

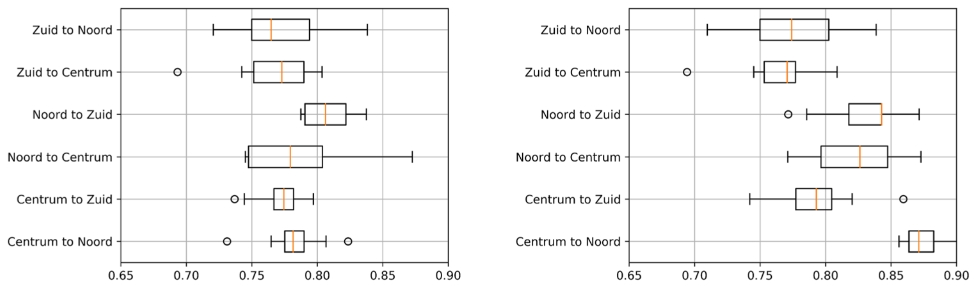

Fig. 5 The accuracy of predictions of trips grouped by the neighbourhoods of origin and destination, for models trained on the first (left) and second (right) data collection period

Table 7 Comparison of cluster size and composition between the first and second data collection periods

Relative size 1st period (%)	Description	Relative size 2nd period (%)
34.2	Trips by car which are generally (among) the quickest alternative(s)	36.9
13.1	Walking trips or public transportation trips where relatively more walking is involved, generally the slowest	10.4
3.1	Relatively slow public transportation trips with relatively many transfers and (relatively) longer walk involved	3.1
1.8	Trips by car which are generally among the slowest alternatives	1.8
2.4	Relatively slow public transportation trips with relatively few transfers	2.2
5.9	Relatively fast walking trips and relatively fast public transportation trips with relatively less transfers	4.4
3.2	Public transportation trips with relatively many transfers and relatively less walking	4.1
6.7	Trips by bicycle that are generally among the slowest alternatives	8.2
19.9	Trips by bicycle that are generally among the fastest alternatives	25.4
9.8	Trips by bicycle that are generally among the slowest nor the fastest alternatives	
	Walking trips in general, that went faster than a generated public transportation trip with a great walking component	3.5

are very comparable in composition and size for the two periods. However, it is worth noting that the share of fast public transportation trips with relatively less transfers has dropped, and the share of public transportation trips with relatively many transfers has risen. This is in line with what one would expect given the new network structure with the north–south line as a spine.

Figure 4 shows that the models that were trained on data from the second period of data collection, generally had a higher performance on the test set (data from the final collection period) than models that were trained on data from the first period of data collection. This implies that the neural net is better able to capture mode choice relations if the underlying transportation network is the same. Both groups of models however generally performed worse than those trained and tested on data that was randomly sampled throughout the entire data set. It was also found that the difference in performance between the models trained data from the first and second collection period depends on the origin and destination of the trips in the test set. If we compare, for example, the subsets of trips that had an origin in one neighbourhood containing a north–south line station and destination (*North*—Noord, *City Centre*—Centrum and *South*—Zuid) in another neighbourhood containing such a station, we mainly observe an increase in prediction accuracy for the second group of models compared to the first when looking at trips going from Centrum to Noord (see Fig. 5). The differences in accuracy between the two model

Table 8 10%-trimmed mean of loss and accuracy of model predictions on the test set when certain attributes are excluded during training

Excluded attributes	Test set loss (10% trimmed mean)	Test set accuracy (10% trimmed mean)
None	0.513	0.807
User-specific attributes	0.540	0.782
Walk_duration, *car_duration*, *bicycle_duration*	0.602	0.776
Transfers	0.504	0.810
Duration	0.522	0.801
User-specific attributes, *transfers*, *duration*	0.721	0.727

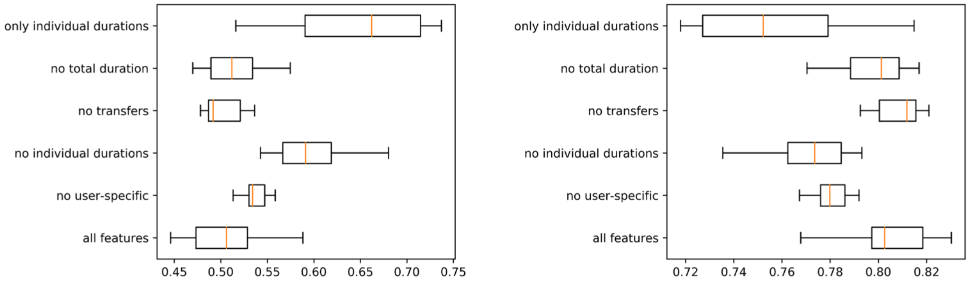

Fig. 6 The test loss (left) and accuracy (right) of models trained with different attributes excluded during training

categories in Fig. 5 may be an indication of the similarity between the classification tasks with differing underlying networks for trips between these areas. Especially, the remarkable difference observed for the Centrum-Noord trips might indicate that the underlying behavioral patterns relating to travel between these areas are to some degree different prior to and after the network change.

7 Discussion

As Tables 5 and 6 suggest, using a multi-layer ANN with sufficiently many nodes in each hidden layer can be a useful and promising technique in predicting mode choice for a large GPS dataset. As mentioned earlier, one of the disadvantages of using a fully connected neural net in order to predict mode choice based on multiple sources of data, is that the neural net will likely pick up on any pattern that is related to the source of the data, which can cause the model to learn non-meaningful relations. In order to investigate which are predictors the model most heavily relies on, we also trained models where several attributes were excluded from the neural net input (feature selection by elimination). We tested six different settings that are relevant for the assessment of our model, for which ten models were trained each. In 5 of the 6 settings, a specific attribute or set of attributes was removed from the input data, and in one setting, no attributes were removed. All models were trained using four hidden layers with *reLU* activation and 200 hidden nodes per layer, with batch size 128. The performance of these models in terms of the losses on the test set and accuracy are displayed in Table 8 and Fig. 6. The results suggest that the ANN is generally less prone to overfitting on the train set if the attribute *transfers* is excluded. The variation in performance can also be reduced by removing the user-specific attributes, however the high losses that are obtained can partially by the fact that the training process is stopped when the validation loss does not decrease for a period of 2 consecutive epochs, while in fact the validation loss could likely decrease more if the model training would not have stopped early. Excluding *car_duration*, *bicycle_duration* and *walk_duration* leads to a greater reduction in model performance. This fact may suggest that these are important predictors, hinting at a possible relation between these variables that would lead the neural net to detect which trip was a real record (and not a generated one). However, the models based on only these predictors perform significantly worse than all other tested settings, which weakens this assumption somewhat.

8 Conclusion and further research

This paper examined the usage of ANNs in order to predict transportation mode choice using a combination of a large GPS-based data set and additionally generated data. After combining and filtering the data, extra user-defining features were extracted using k-means clustering and PCA. Several ANN models were trained based on the choice sets and these extracted features, with different (randomly sampled) hyper-parameter settings. The best model initially reported a validation accuracy of over 84% and performed well in predicting trips from every category, except for trips from the smallest one (public transportation with use of train and metro). It was found that the model performed better on unseen data if the data on which it was trained and tested, were collected on a very similar underlying transportation network, than if the underlying transportation network would be somewhat different between the train and test sets. This difference in performance was observed to differ based on the origin and destination area of the trip for which the mode had to be predicted. After further analysis, it was found that excluding durations of each individual mode from the training data has the highest negative impact on model performance, whereas excluding the number of transfers has little to no negative impact, and might even reduce overfitting.

Building on this study, interesting for future research might be to investigate a more problem-tailored neural net architecture, as well as to infer all information that could be inferred from classical statistical mode choice models from the ANN.

As the source data for this neural network is panel data, it's possible that better results could be achieved by a model that is not blind to panel effects. The work by Yang et al. (2020) proposes a new class of interpretable neural network models achieving both high prediction accuracy and interpretability in regression problems with time series cross-sectional data, might improve the accuracy achieve by this model.

Due to the methodological focus of this paper, the selection of input features for the choice set has been limited to some extent. Another interesting suggestion for future research would be to investigate how well the presented models are suited when the input feature space is extended beyond core attributes like duration. Additional features that could be considered for inclusion would be economic features like parking tariffs, fuel cost or public transportation cost. Also, it would be interesting to see how the model performs when external features are added to the input, such as weather (which is easily extracted from the original trip record database) or trip purpose [which, although not readily available, could be extracted using an activity detection algorithm, see e.g. Reumers et al. (2013)].

Since the results also seem to hint at a clear change in behavioral patterns following the opening of the new metro line and the restructuring of the network, it would certainly be interesting to investigate these in more detail. Given this change in behavioral patterns, studying and tuning a tailored transfer learning model to this data is another, final possible direction for further research.

Acknowledgements This research has been conducted in the framework of the Impact Study North/Southline, funded in part by the Municipality of Amsterdam and the regional transportation authority of Amsterdam. We would gratefully like to acknowledge the Vrije Universiteit Amsterdam legal affairs office, IT department, and university library research data management team as well as the IT and Facilities department of CWI for support in the design and implementation of the computing infrastructure.

Data availability GTFS and OSM are publicly available. The mobile phone app data is privacy-sensitive and strictly protected under the European General Data Protection Regulation (GDPR), EU 2016/679, which went into effect May 25, 2018.

Code availibility We made use of Conveyal R5 and several publicly available Python libraries, among which tensorflow and scikit-learn.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Agrawal D, Schorling C (1996) Market share forecasting: an empirical comparison of artificial neural networks and multinomial logit model. J Retail 72(4):383–408

Bentz Y, Merunka D (2000) Neural networks and the multinomial logit for brand choice modelling: a hybrid approach. J Forecast 19(3):177–200

Bradley MA, Gunn HF (1990) Stated preference analysis of values of travel time in the Netherlands. Transp Res Rec 1285:78–88

Buijs R, Koch T, Dugundji E (2020) Using neural nets to predict transportation mode choice: an Amsterdam case study. Proc Comput Sci 170:115–122

Conway MW, Byrd A, van der Linden M (2017) Evidence-based transit and land use sketch planning using interactive accessibility methods on combined schedule and headway-based networks. Transp Res Rec 2653(1):45–53

Cosslett SR (1981) Efficient estimation of discrete-choice models. Struct Anal Discrete Data Econ Appl 3:51–111

Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12(7):2121–2159

de Freitas LM, Becker H, Zimmermann M, Axhausen KW (2019) Modelling intermodal travel in Switzerland: a recursive logit approach. Transp Res Part A Policy Pract 119:200–213

Guevara CA, Ben-Akiva ME (2013) Sampling of alternatives in logit mixture models. Transp Res Part B Methodol 58:185–198

Hayashi Y, Hsieh MH, Setiono R (2010) Understanding consumer heterogeneity: a business intelligence application of neural networks. Knowl Based Syst 23(8):856–863

Hillel T, Bierlaire M, Jin Y (2019) A systematic review of machine learning methodologies for modelling passenger mode choice. Tech. rep., Technical Report TRANSP-OR 191025. EPFL

Jolliffe IT, Cadima J (2016) Principal component analysis: a review and recent developments. Philos Trans R Soc A Math Phys Eng Sci 374(2065):20150202

Kingma D, Ba J (2014) Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980

Li Z, Xu WA (2019) Path decision modelling for passengers in the urban rail transit hub under the guidance of traffic signs. J Ambient Intell Humaniz Comput 10(1):365–372

Long T (2020) Research on application of athlete gesture tracking algorithms based on deep learning. J Ambient Intell Human Comput 11(9):3649–3657

McFadden D (1973) Conditional logit analysis of qualitative choice behavior. In: Zarembka P (ed) Frontiers in Econometrics. Academic Press, New York, pp 105–142

Morikawa T (1989) Incorporating stated preference data in travel demand analysis. PhD thesis, Massachusetts Institute of Technology

Murtagh F, Contreras P (2012) Algorithms for hierarchical clustering: an overview. Wiley Interdiscip Rev Data Min Knowl Discov 2(1):86–97

Reumers S, Liu F, Janssens D, Cools M, Wets G (2013) Semantic annotation of global positioning system traces: activity type inference. Transp Res Rec 2383(1):35–43

Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Trans Database Syst 42(3):1–21

Steinley D (2006) K-means clustering: a half-century synthesis. Br J Math Stat Psychol 59(1):1–34

Thakur D, Biswas S (2020) Smartphone based human activity monitoring and recognition using ML and DL: a comprehensive survey. J Ambient Intell Human Comput 11(11):5433–5444

Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw Mach Learn 4(2):26–31

van Cranenburgh S, Alwosheel A (2019) An artificial neural network based approach to investigate travellers' decision rules. Transp Res Part C Emerg Technol 98:152–166

Vythoulkas PC, Kotsopoulos HN (2003) Modeling discrete choice behavior using concepts from fuzzy set theory, approximate reasoning and neural networks. Transp Res Part C Emerg Technol 11(1):51–73

Wang S, Zhao J (2019) An empirical study of using deep neural network to analyze travel mode choice with interpretable economic information. Tech. rep., Massachusetts Institute of Technology

Wang S, Mo B, Zhao J (2020a) Deep neural networks for choice analysis: architecture design with alternative-specific utility functions. Transp Res Part C Emerg Technol 112:234–251

Wang S, Wang Q, Zhao J (2020b) Multitask learning deep neural networks to combine revealed and stated preference data. J Choice Model 37:100236

Yang Y, Zheng Z (2020) Interpretable neural networks for panel data analysis in economics. arXiv preprint arXiv:2010.05311

Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? Advances in Neural Information Processing Systems 27:3320–3328