Transportation Research Part C 82 (2017) 273–289

Contents lists available at ScienceDirect

Transportation Research Part C

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

Travel mode estimation for multi-modal journey planner

Heikki Mäenpää [*], Andrei Lobov [**], Jose L. Martinez Lastra

Tampere University of Technology, FAST Laboratory, P.O. Box 600, FIN-33101 Tampere, Finland

ARTICLE INFO

Article history:
Received 27 January 2017
Received in revised form 27 June 2017
Accepted 27 June 2017
Available online 11 July 2017

Keywords:
Travel mode detection
Bayes classifier
Neural network
Feature selection

ABSTRACT

For route planning and tracking, it is sometimes necessary to know if the user is walking or using some other mode of transport. In most cases, the GPS data can be acquired from the user device. It is possible to estimate user's transportation mode based on a GPS trace at a sampling rate of once per minute. There has been little prior work on the selection of a set of features from a large number of proposed features, especially for sparse GPS data. This article considers characteristics of distribution, auto- and cross-correlations, and spectral features of speed and acceleration as possible features, and presents an approach to selecting the most significant, non-correlating features from among those. Both speed and acceleration are inferred from changes in location and time between data points. Using GPS traces of buses in the city of Tampere, and of walking, biking and driving from the OpenStreetMap and Microsoft GeoLife projects, spectral bins were found to be among the most significant non-correlating features for differentiating between walking, bicycle, bus and driving, and were used to train classifiers with a fair accuracy. Auto- and cross-correlations, kurtoses and skewnesses were found to be of no use in the classification task. Useful features were found to have a fairly large (>0.4) correlation with each other.

1. Introduction

Knowing the mode of transportation currently employed by the user has many applications. Among these are tracking the user's exercise goals (Parkka et al., 2006), more accurate household travel surveys (Gong et al., 2012) and better informed participant selection for such surveys (Reddy et al., 2009), as a precursor to trip purpose recognition (Xiao et al., 2016), and even targeted advertisements (Zhu et al., 2016).

Studies have been carried out on learning users' daily activity patterns for demand forecasting (Allahviranloo and Recker, 2013), and for building agent-based simulations (Liao et al., 2013). Knowing the users' mode of transport to the latest activity would assist in prediction of scheduling.

The effect of travel time variability on flexible scheduling has also been studied (Jenelius, 2012; Fosgerau and Karlström, 2010). Since the variability of travel time is dependent on the mode of travel, knowledge of the user's previous travel mode could be used to further optimize the scheduling.

The motivation for the system described in this paper is the intent to incentivize ecological transportation choices in the city of Tampere. For this purpose, a method to identify walking, biking, driving and taking a bus is needed. The intent is to use data gathered by logging the user's location at intervals of one minute, which means that any features must be inferred from the coordinates and timestamps of datapoints.

[*] Principal corresponding author.
[**] Corresponding author.
E-mail addresses: heikki.a.maenpaa@gmail.com (H. Mäenpää), andrei.lobov@tut.fi (A. Lobov), jose.lastra@tut.fi (J.L. Martinez Lastra).

http://dx.doi.org/10.1016/j.trc.2017.06.021
0968-090X/© 2017 Elsevier Ltd. All rights reserved.

In this paper, skewness, kurtosis, auto- and cross correlations, and energy spectra of speed and acceleration are considered as potential features for travel mode identification.

Three criteria are used for the selection of the most significant features for a machine learning algorithm. To validate these three methods, three different machine learning algorithms are trained based on the results of the three selection criteria, and compared.

The selected features are used to train a four-class feed-forward neural network, a four-class Bayes classifier, and a decision tree based on consecutive Bayes classifiers. The principle behind these classifiers is explained in Section 3.

The training data consisted of GPS traces of Tampere city buses and GPS traces acquired from the OpenStreetMap and GeoLife project. The preprocessing for this data is described in Section 4.

The selection criteria used are Welch's *t*-test, Mann-Whitney *U*-test, and *F*-test. The process of selecting features is outlined in Section 5. For comparison, a stacked autoencoder was also used in conjunction with a single-layer neural network.

A neural network was found to have the best F1-score with each selection criterion. Adding a second or third layer to a neural network yielded marginal improvements for two out of three criteria and marginally worse results in a third. Using an autoencoder rather than the manual dimension reduction produced the worst results. More in depth results are outlined in Section 6, and discussion of the results is given in Section 7.

2. Background

2.1. Related work

Travel mode recognition has been studied a lot. Most studies have used an accelerometer alone or combined with a GPS unit. A common theme in the research is that walking can be identified with over 90% accuracy.

Su et al. (2014) made a fairly broad literature review on the subject of recognizing the user's activity using smartphone sensors. They listed a large variety of features that have been used, classified to time domain and frequency domain features applied to an accelerometer's output.

A more recent literature review, Prelipcean et al. (2017), identified three research areas that use travel mode recognition: Location based services, transportation science and human geography. Each of these has slightly different demands. Location based services require close to real-time information about the user's location and current mode of transport. Transportation science concerns itself with statistics of users' modes of transport, and requires accuracy more than speed. Human geography seeks to enrich traces with domain-specific information, and its travel mode detection is focused on segmenting traces into stops and moves.

Using only timestamped coordinates to extract the features was rare. In particular, using the GPS speed data was common (Reddy et al., 2008; Sun and Ban, 2013; Bolbol et al., 2012). Using an accelerometer in conjunction with a GPS receiver has also been studied (Reddy et al., 2008).

Table 1 on page 6 shows an overview of classifiers and features considered in various studies, as well as the accuracies achieved. If more than one classifier was considered, the best-performing one is in bold.

Features that have been considered have varied. Means and variances of acceleration and speed are common. Rarer examples are other features of the distributions, and spectral features. Some studies (Gong et al., 2012; Stenneth et al., 2011) used Geographical Information Systems (GIS) to estimate a trace's similarity to a public transport route. The amount of heading change has also been considered, although it has generally been found lacking.

Selecting the best features to use presents a problem in any classification task, and particularly in travel mode recognition where the literature provides a wealth of features to select from. Bolbol et al. (2012) used Wilks' Lambda test and between-groups *F*-test to select the most significant features. They also noted that, by 2012, travel mode recognition studies rarely considered feature selection. Stenneth et al. (2011) ranked the features by Chi Square and Information gain algorithms.

Sun and Ban (2013) raised privacy concerns regarding the frequency of sampling the GPS data, and suggested that research should be done on how sparse data can be used. Bobol and Cheng (2010) demonstrated that 30–60 seconds' sample time is sufficient for this purpose.

2.2. Hypotheses

Hypothesis 1. Frequency-domain features can be used to differentiate between modes of transportation.

A number of factors could be expected to create fluctuation in the speed at which a person or vehicle is traveling at. For buses, there are stops along the way which would necessitate coming to a full stop to load and unload passengers. All traffic would be expected to pause or slow down at intersections, but cars and bicycles would reach the intersections faster than walkers.

Frequency domain features have been used in travel mode recognition via accelerometer (Kwapisz et al., 2011), and on dense GPS data (Reddy et al., 2008). However, little research was found studying the spectral features of sparse GPS data.

Table 1
Overview of classifiers and features considered by prior studies.

Study	Classifiers	Features	Accuracy
Reddy et al. (2008)	Naïve Bayes classifier, decision tree, support vector machine, continuous hidden Markov model, **DT + discrete hidden Markov model**.	Mean, variance, energy and spectrum of speed and acceleration	98.8%
Sun and Ban (2013)	Kernel SVM	Maximum acceleration, proportions of acceleration more than 1 m/s^2, standard deviation of acceleration. (Only cars and trucks considered)	95.8%
Bolbol et al. (2012)	Support vector machine	Distance, mean speed, acceleration and heading change	88%
Gong et al. (2012)	Set of rules	GIS features, average speed, maximum speed, speed and acceleration distribution	82.6%
Stenneth et al. (2011)	Naïve Bayes classifier, Bayesian net, decision tree, multilayer perceptron, **random forest**	GIS features, average speed, average acceleration, average heading change, trace quality	92.8%
Stopher et al. (2008)	Set of rules	Bicycle ownership, GIS features, average speed, maximum speed, most common speed	93.6%
Zhu et al. (2016)	Bayesian net, decision tree, random forest, support vector machine, **stacked autoencoder + neural network**	Distance traveled, average speed, average acceleration, heading change, GIS features	93%
Bantis and Haworth (2017)	Random forest, svm, multilayer perceptron, **HMM**	Speed, GIS features, personal preferences, mobility-affecting disabilities	71 and 78% (two subjects)

Hypothesis 2. Auto- and cross correlation of speed and acceleration can be used to differentiate between various modes of transport.

A person walking would maintain a fairly steady speed from minute to minute. Similarly, a car that is moving slower than the speed limit would be expected to accelerate, as can a bus at a stop.

Features such as this seemed conspicuously absent from prior research, and are therefore considered here.

Hypothesis 3. Skewness and kurtosis of speed and acceleration can be used to differentiate between modes of transport.

A car's speed distribution would be expected to be skewed toward the speed limit, only dropping for intersections and such, whereas a walker's speed would remain fairly constant and a bicycle would accelerate and decelerate as the road's inclination and other traffic allow.

Lower statistical moments and other features of the distribution of acceleration and speed have been studied (Zhu et al., 2016; Reddy et al., 2008; Sun and Ban, 2013). However, skewness and kurtosis seem not to have been considered as features before.

To verify these three hypotheses, statistical tests will be used to identify the most suitable features, as proposed by Bolbol et al. (2012). A number of features considered by previous research, namely the mean, median and variance of speed and acceleration, will be considered alongside these for reference.

3. Design of the classifiers

Because walking has been consistently found easy to identify (Bolbol et al., 2012; Gong et al., 2012; Reddy et al., 2008), it is possible that eliminating modes individually could be a good approach, which in turn suggests that decision trees would be a good choice. Reddy et al. (2008) also received good results with decision trees. Because of this, a random forest classifier will be used as one classifier.

A Bayes classifier's ease of implementation and intuitive function make it a tempting choice as a machine-learning algorithm. Therefore, it will be included in this comparison.

To measure the above two against a third party machine learning algorithm, a feedforward neural network trained with the Encog framework is used, both with and without autoencoders.

3.1. Bayes classifier

A Bayes classifier is a classifier based on the Bayes theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

Applied to classification, the problem becomes finding a class Y for feature vector $\bar{x}$ so that

$$Y = \arg\max_{Y} P(Y|\bar{x}) = \arg\max_{Y} \frac{P(\bar{x}|Y)P(Y)}{P(\bar{x})} \tag{2}$$

In this paper, the probability density function of a multivariate normal distribution is used as a proxy for $P(\bar{x}|Y)$

$$f_Y(\bar{x}) = \frac{e^{-\frac{1}{2}(\bar{x}-\bar{\mu}_Y)^T \Sigma_Y (\bar{x}-\bar{\mu}_Y)}}{\sqrt{(2\pi)^d |\Sigma_Y|}} \tag{3}$$

where d is the dimension of the feature vector, $\bar{\mu}_Y$ is the mean of the class Y, Σ_Y is the covariance matrix of the class, and $|\Sigma_Y|$ is the determinant. A similar proxy is used for $P(\hat{x})$, substituting the mean and covariance matrix of the population for those of each class Y.

The computational complexity of this rises quadratically as the number of features rises. There are methods for estimating the probability density function using a sampling of as few as thirty vectors from the class (He et al., 2014; John and Langley, 1995). In this article a small number of significant, non-correlated features is selected, as explained in Section 5.3, leaving the exact PDF as the less complex solution.

The training of the Bayes classifier is achieved by calculating the covariances and means for each class and for the population.

3.2. Random forest

A random forest is an aggregate of several decision trees trained with slightly different data- or feature sets. The likelihoods produced by the individual trees are averaged, and the highest likelihood class is returned as the most likely class.

In this paper, the random forest is trained through bootstrap aggregating. Bootstrap aggregating, shortened to "bagging" (Breiman, 1996), creates random subsets of the dataset without removing elements from the original dataset. In other words, the datasets will overlap by a large margin. Bagging makes the aggregate classifier robust against perturbations in the training data (Breiman, 1996).

In this paper's implementation, each class is separately bagged. This is to ensure that each class is proportionally represented in the training dataset.

Individual decision trees are built by finding the feature and cutoff value that produce the largest drop in entropy, dividing the data accordingly, and recursing over those two data sets. Entropy drop is defined as the difference between the dataset's entropy and the average of the child datasets' entropies, weighted by the child datasets' sizes.

If the initial entropy or the drop in entropy is below a small threshold, a decision tree leaf is created. The leaf will contain the ratio of each class's members contained within the data used to build it. These ratios are used as the likelihoods returned by the tree.

3.3. Neural network

A feed-forward neural network consists of an input layer, output layer and zero or more hidden layers in between. Neurons of each layer take the outputs of each neuron in the layer before as input, and the neuron's output is a function of a weighted sum of these.

The Feedforward network is trained by Encog Java library of machine learning tools (Heaton, 2015), using the RPROP algorithm. The RPROP algorithm is a gradient-descent based algorithm that adapts its rate of descent in response to changes in each partial derivative's sign (Riedmiller and Braun, 1993).

The default for Encog's feedforward networks is one hidden layer with 1.5 times the amount of input and output neurons. In this paper, the maximum amount of features was used as a base for this. One to three hidden layers were tested, since the efficiency of the RPROP algorithm starts tapering off as layers are added (Hinton and Salakhutdinov, 2006).

3.4. Autoencoder

An autoencoder is a special case of a neural network where the desired output is the input, and the hidden layer has less neurons than the input layer (Xu et al., 2016). This means that the output of the smaller layer is a lower-dimensional representation of the input vector that the input can be reconstructed out of. A stacked autoencoder consists of several nested autoencoders.

Hinton and Salakhutdinov (2006) describes a method for pretraining a deep autoencoder by training each layer as a restricted Boltzmann machine. The resulting network is then trained further with a backpropagation algorithm.

For this paper, Encog's tools for feedforward networks were used to train a four layer autoencoder, and the encoding portion's output was used to train a three-layer neural network, similar to Zhu et al. (2016). Since the Encog library did not at time of writing provide tools to train RBMs, pretraining was achieved by training four single-layer autoencoders, each with the previous encoder's encoded output as the input.

4. Data collection and preprocessing

GPS traces were acquired from two sources. Bus traces were acquired from Innovative Tampere Site' journey APIs over one full week. Car, walking and bicycle traces were acquired from the OpenStreetMap and GeoLife projects.

Microsoft GeoLife is a location-based social networking service. The purpose of the service is to mine multiple users' data for typical travel sequences and to use individual location histories to measure similarity between users and provide friend- and location recommendations.

The project has made the GPS traces of 182 users from a period of April 2007 to August 2012 available. The dataset was collected by Microsoft Research Asia. Parts of this dataset were labeled by mode of transport, which made them useful for this paper's purposes.

OpenStreetMap is an open-source cartography project. Its purpose is to provide and maintain an open-data map of the world. The project has amassed a large collection of GPS traces, some of which were tagged with a mode of transport. Up to one thousand newest GPS traces tagged "Car", "Walking" or "Bicycle" were downloaded. Unfortunately, some of the traces were tagged with more than one trace so an initial heuristic was used to reduce the amount of false positives.

4.1. Segmentation

The continuously logged Bus traces were tagged by route and direction, and split into segments according to those.

The OpenStreetMap and GeoLife traces covered large distances and often incorporated stops. Some of the OpenStreetMap traces had also been tagged with multiple travel modes. Similarly to Gong et al. (2012), a segment was ended if a route point did not pass outside a 50 meter radius of the current route point in three minutes. This corresponds to an average speed of 1 km/h, considerably slower than the mean comfortable walking gait of an adult (Bohannon, 1997).

The re-sampling described in Section 4.3 would result in two consecutive interpolated route points on a three minute gap. A segment was therefore ended if there was more than a three minute interval between route points.

In addition, to improve the classification accuracy on continuous classification, the segments were split to the length of ten minutes or less. The continuous classification uses a sliding window, and is expanded upon in the first author's Master's thesis (Mäenpää, 2017 p.26–27).

4.2. Initial filtering

Some of the traces received from OpenStreetMap were tagged with more than one of the three mode tags, which necessitated further filtering. After segmentation, the segments were rejected from a mode if they lasted less than 5 min, did not maintain the speed v_{min} for 30 consecutive seconds, or maintained an average speed greater than v_{max} for 30 consecutive seconds. The time limit for speeds was intended to permit short bursts of speed, such as biking downhill.

Speed was always inferred from the timestamps and coordinates, even if speed was included in the trace. This was because the speed datapoint was absent from most of the traces. Acceleration was inferred from the timestamp and speed.

The parameters are listed in Table 2.

In order to weed out cars, maximum speed for bicycles is set at 10 km/h less than Finland's default speed limit in an urban area. Minimum speed is set by rounding down to the previous ten the mean speed of recreational bicyclists measured by Thompson et al. (1997).

The bus traces were known to be logged from buses, and they were included in the filtration only in case any of the traces were corrupt.

The speed was inferred from the time and distance between track points. If the time between track points was zero, or the speed was over 200 km/h, the track point was dismissed as corrupt. If more than ten consecutive corrupt track points were detected, the segment was dismissed as corrupt.

Finally, after the feature extraction described in Section 5, outliers were filtered out by Mahalanobis distance, analogous to the three-sigma rule with a single variable. The distance is calculated by

$$d_M = \sqrt{(\overline{x} - \overline{\mu})^T \Sigma^{-1} (\overline{x} - \overline{\mu})} \tag{4}$$

where $\overline{x}$ is the feature vector, $\overline{\mu}$ is the mean vector, and Σ is the covariance matrix. The square of the Mahalanobis distance is χ^2-distributed with degrees of freedom equal to the amount of features. The three-sigma interval contains approximately

Table 2
The parameters for initial classification.

Mode	v_{max}	v_{min}
Walking	15 km/h	0
Bicycle	40 km/h	10 km/h
Bus	∞	0
Car	80 km/h	40 km/h

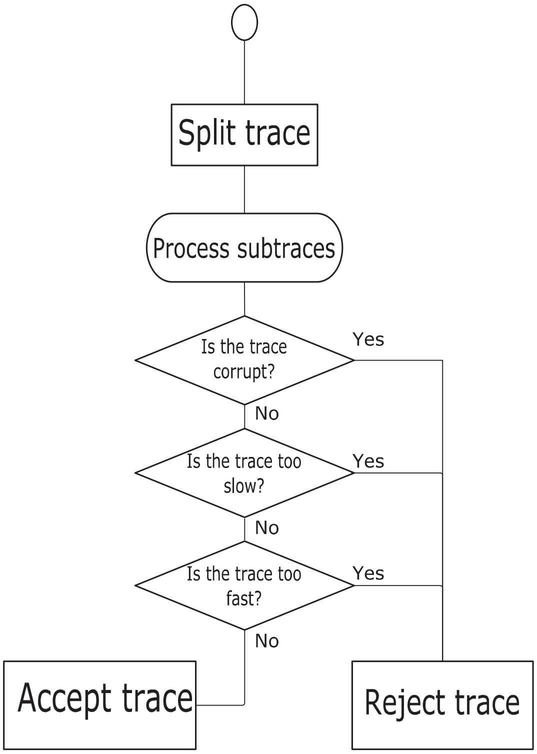

Fig. 1. Flowchart of the track filtration.

99.7% of normally distributed variables, and the 99.7% threshold of the χ^2_{41}-distribution is $d^2_M < 66.2$. Therefore, the equivalent cutoff for a three-sigma outlier is $d_m < 8.13$.

Because the bus traces constituted over three quarters of the data, the traces were truncated to twice the car class's size. This was to ensure that k-fold cross-validation wouldn't give excessive weight to the walking class. Had this not been done, a classifier that simply labeled everything as "bus" would have gotten an approximately 80% accuracy and, conversely, a single percent point's error in buses would have been worth, approximately, six for walking, thirty for bicycles and sixty for cars.

Fig. 1 shows the process for initial filtering of traces. The traces are split first, in order to separate different modes of transport in each trace. Second, the traces are filtered according to whether or not they are corrupt and whether they reach the minimum speed or exceed the maximum speed for each mode of transport. The filtration step is carried out for each class.

The post-filtration segment counts, N, and the amount of segments dismissed for each reason are in Table 3.

Twenty percent of the smallest class's (car) size, i.e. 263, of the traces in each class were randomly removed from the training data for purposes of calculating confusion matrices and plotting visualizations. The final size of each training dataset is in the column labeled "final".

The complete preprocessing flowchart is in Fig. 2. This process was run individually on each class.

4.3. Re-sampling

The tracks from OpenStreetMaps and GeoLife had various sampling rates, most being several samples per minute, depending on the submitting user's preference. The API used for the bus traces was polled once every minute, that being the upper limit of sample time previously shown to be useful for travel mode recognition by Bobol and Cheng (2010). For these reasons, a re-sampling was carried out.

The resampling was initialized by making the first point in the original trace, x_0 the first point in the resampled trace $\hat{x}_0$, and the index k was set to zero. Then, the following was iterated:

1. Find x_i, x_{i-1} so that x_i is at least one minute after $\hat{x_k}$ and x_{i-1} is less than a minute after $\hat{x}_k$.
2. Linearly interpolate a point $\hat{x}_{k+1}$ between x_i and x_{i-1} so that $\hat{x}_{k+1}$ is exactly one minute after $\hat{x}_k$.
3. Increment k.

The remainder of less than one minute at the end of the trace was discarded.

Table 3
Number of accepted and rejected segments by mode and rejection criterion.

Mode	N	$v > v_{max}$	$v < v_{min}$	Corrupt	d_M	Final
Walking	10,575	153	0	180	1135	2630
Bicycle	3107	388	277	162	207	2630
Bus	60,836	0	0	1	4781	2630
Car	1411	1590	25,175	3417	108	1315

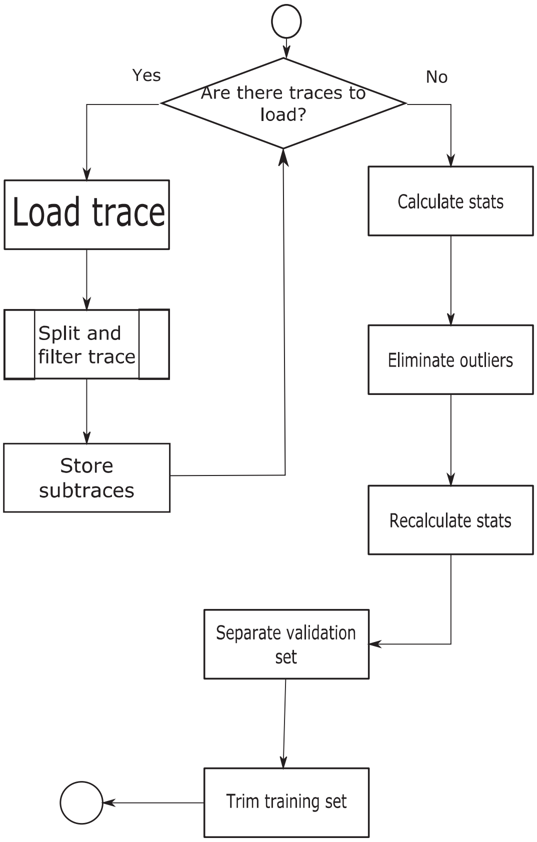

Fig. 2. Flowchart for trace preprocessing.

5. Feature selection

5.1. Features considered

For each segment, the following statistics were calculated for speed and magnitude of acceleration: Minimum, maximum, median, average, variance, skewness and kurtosis.

The minimum and maximum speed were ignored, because they had previously been used to filter out the data. Magnitude of acceleration was used because the sum of accelerations would be zero for a segment between two stops.

In the frequency domain, a spectrum was calculated up to the Nyquist frequency, $8mHz$, in seven one-octave bins with 50% overlap. Cross- and autocorrelations for speed and acceleration were calculated for up to three samples to the past and future.

The segments were classified by mode, and the means and standard deviations were calculated for the features inside each mode.

Since the sampling frequency was one sample per minute, sampling momentary speed and acceleration would not have produced an accurate representation of the GPS receiver's motion. Therefore, speed and acceleration were inferred from the distance and time between two timestamps. First, speed was inferred by dividing the distance traveled between points with the time between them. This means that for a trace of n track points, there were $n - 1$ speeds. Next, the change in speed was divided by the difference in time, giving a total of $n - 2$ accelerations.

5.2. Spectrum calculation

The frequency components of speed and acceleration were calculated with a discrete-time Fourier transform on the entire dataset. The dataset was windowed with a Hamming window.

The spectrum was calculated out of frequency components by integrating the square of the magnitude of the frequency component over the frequency bin's width in 16 steps with the trapezoidal rule. Finally, a base-10 logarithm was taken of the result of integration.

The bins were spaced so they had a 50% overlap in logarithmic frequency, with the width of one octave. In other words, the end frequency of a bin was twice the starting frequency, and the next bin would start at the end frequency divided by $\sqrt{2}$. The number of bins was selected so that the first bin was 0–1 mHz.

5.3. Feature selection process

Zhu et al. (2016) used autoencoders to reduce the dimensionality of a feature vector. In this paper, however, the aim is to reduce the amount of features that need to be calculated. To discover which features were the most suitable for classification, the data was first split to three pairs of classes. Because walking has been found to be easy to classify, the first sub-classification was Walking/Wheeled. Since bicycles were the only muscle-powered mode left Bicycle/Motorized was second, leaving the Car/Bus distinction last. Two statistical tests were then run for all features on each pair of classes.

The first test was Welch's *t*-test. Because of the large degrees of freedom produced, a comparison of p-values proved impractical. Therefore, the features were ranked on based on their *t*-values, a larger *t*-value being more significant.

Because some of the features, such as variances, were not normally distributed, Mann-Whitney U test for large samples was used as the second test.

In the Mann-Whitney U test, each feature was ranked by value, and the output was the smaller of $R - \frac{n(n+1)}{2}$, n being the class size and R being the sum of ranks. A lower output was considered more significant.

A third test, the *F*-test, could be used to test multiple groups of features at once. The *F*-test ranks features by the ratio of between-groups variability to within-group variability. A larger *F*-statistic was therefore better.

A number of features for each pair of classes were selected with the following procedure:

1. Select the next most significant feature not already considered. Call this feature f_0.
2. Select all other features for which correlation with $f_0, |C_{corr}| > C_t$.
3. Choose f_0 as an accepted feature.
4. Mark all the selected features as already considered.
5. Iterate until all features are considered or desired number of features are selected.

Since experimentation proved that auto- and cross-correlations were weak distinguishing features (as shown in Fig. 4 on page 21), the amount of good features returned by each statistical test were estimated by the number of features more significant than the first correlation feature. These amounts are in Table 4. For the *t*- and *U*-tests, the smallest value for any split (typically the car/bus split) was used.

Correlation threshold of 0.5 was selected, which corresponded to three features per split. The selections from the U- and *t*-tests were combined. Because this increased the amount of features used, for the *F*-test, six indices were selected by that with the correlation threshold 0.9. The autoencoder's output size was also six features.

Ranking by the *t*-test produced the features, eight in total, in Table 5. Ranking by the *U*-test produced the features, six in total, listed in Table 6. Ranking by the *F*-test produced the features listed in Table 7. The correlations between all features, calculated from the entire dataset, are in Appendix A. The *F*-statistics for each feature are in Appendix B.

Table 4
Number of good features for each correlation threshold.

C_t	*t*-test	*U*-test	*F*-test
0.1	1	1	1
0.2	1	1	1
0.3	1	1	2
0.4	1	1	2
0.5	3	3	3
0.6	3	3	4
0.7	4	3	4
0.8	6	5	5
0.9	8	7	9
1.0	10	9	26

Table 5
Features selected with Welch's *t*-test, *t*-values in parentheses.

#	Walk/wheeled	Bicycle/motorized	Car/bus
1	Speed spectrum 1.4–2.8 mHz (143.3)	Average acceleration (102.6)	Average speed (47.27)
2	Minimum acceleration (60.46)	Average speed (53.88)	Speed skewness (15.80)
3	Speed skewness (59.68)	Speed skewness (18.62)	Speed spectrum 4.0–8.0 mHz (12.06)

Table 6
Features selected with *U*-test, *u*-values/1000 in parentheses.

#	Walk/wheeled	Bicycle/motorized	Car/bus
1	Speed spectrum 1.4–2.8 mHz (307)	Average acceleration (232)	Average speed (248)
2	Minimum acceleration (1176)	Average speed (1598)	Speed skewness (921)
3	Speed skewness (2627)	Speed skewness (3510)	Speed spectrum 4.0–8.0 mHz (1043)

Table 7
Features selected with *F*-test and *f*-statistics.

#	Feature	f
1	Speed spectrum 4.0–8.0 mHz	10,920
2	Speed spectrum 1.4–2.8 mHz	9588
3	Acceleration spectrum 0.71–1.4 mHz	8686
4	Average acceleration (m/s^2)	7771
5	Average speed (m/s)	7173
6	Speed spectrum 2.0–4.0 mHz	6446

Table 8
Average F1 scores, for each classifier and feature selection method.

Ranking criterion	BC	NN1	NN2	NN3	RF	AE
Welch's *t*	0.830	0.895	0.899	0.897	0.903	0.830
U-test	0.837	0.896	0.898	0.899	0.904	
F-test	0.849	0.894	0.898	0.898	0.901	

The feature selection and subsequent classifier training was run two more times with various features included. The second run excluded the "traditional" features, i.e. minima, maxima, means, medians and variances, to gauge the stand-alone usefulness of the hypothesized features.

To further gauge the stand-alone usefulness of features, the feature selection was run individually for each hypothesis' features.

6. Results

As explained in Section 4, 263 traces from each class were set aside as a validation set. The rest of the data was used to train each classifier with 5-fold cross-validation.

The average F1 scores, weighed with the inverse of class size, over the four classes are in Table 8. Only one value is shown for the autoencoder because the feature selection step was bypassed for that method.

The *t*- and *U*-tests' outputs were identical, so the Bayes classifier and Random forest performed equally. However, the neural networks' initial weights were randomized separately, so there was some fluctuation in the end result.

The neural network was slightly improved by a second hidden layer, but the results from adding a third hidden layer were mixed. The difference was of the same order as the fluctuation caused by the different starting weights.

Table 9 presents the confusion matrix and recall rates for a Bayes classifier, Tables 10–13 present the same for the neural networks, and Table 14 presents the same for a random forest. Each represents the highest F1-score.

The percentages in the confusion matrices are the ratio of the known class classified as each class. In other words, the recall rates can be found in the diagonal of the matrix. The precision, or percentage of traces classified to a mode belonging to that mode, is given below each confusion matrix, as is the F1 score.

Buses were the most problematic class for the Bayes classifier, with 10.1% classified as bicycles and 4.22% classified as cars, and a full 31.6% of cars and 8.64% of bicycles classified as buses.

Table 9
Confusion matrix of the Bayes classifier.

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	95.9%	4.07%	0.00%	0.00%
Bike	3.69%	87.6%	8.64%	0.114%
Bus	0.00%	10.1%	85.7%	4.22%
Car	0.00%	1.62%	31.6%	66.8%
Recall	96.3%	85.5%	80.1%	86.0%
F1-score	0.961	0.865	0.828	0.752

Table 10
Confusion matrix of the neural network (1 hidden layer).

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	99.2%	0.763%	0.00%	0.00%
Bike	2.29%	91.6%	6.11%	0.00%
Bus	0.00%	6.11%	90.8%	3.05%
Car	0.00%	1.53%	30.2%	68.3%
Recall	97.7%	91.6%	71.5%	95.7%
F1-score	0.985	0.916	0.800	0.797

Table 11
Confusion matrix of the neural network (2 hidden layers).

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	99.6%	0.382%	0.00%	0.00%
Bike	1.91%	89.7%	8.40%	0.00%
Bus	0.00%	6.49%	89.7%	3.82%
Car	0.00%	1.53%	30.9%	67.6%
Recall	98.1%	91.4%	69.5%	94.7%
F1-score	0.989	0.906	0.783	0.788

Table 12
Confusion matrix of the neural network (3 hidden layers).

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	99.6%	0.00%	0.382%	0.00%
Bike	1.53%	91.6%	6.87%	0.00%
Bus	0.00%	6.49%	89.3%	4.20%
Car	0.00%	1.91%	29.0%	69.1%
Recall	98.5%	91.6%	71.1%	94.3%
F1-score	0.991	0.916	0.792	0.797

Table 13
Confusion matrix of the autoencoder neural network.

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	96.2%	3.82%	0.00%	0.00%
Bike	6.49%	84.4%	8.40%	0.763%
Bus	0.00%	7.63%	86.3%	6.11%
Car	0.00%	1.53%	33.2%	65.3%
Recall	93.7%	86.7%	67.5%	90.5%
F1-score	0.949	0.855	0.757	0.758

Table 14
Confusion matrix of the random forest.

Correct label	Classified as			
	Walk	Bike	Bus	Car
Walk	99.1%	0.875%	0.00%	0.00%
Bike	1.86%	90.8%	7.27%	0.0381%
Bus	0.00%	6.70%	88.1%	5.21%
Car	0.00%	2.00%	23.8%	74.2%
Recall	98.2%	91.6%	84.0%	85.0%
F1-score	0.986	0.912	0.860	0.792

The neural networks had similar troubles with the bus class, with 29–30.9% of cars misclassified as buses.

Even with autoencoding, the pattern persists, with 33.2% of cars and 8.4% of bicycles misclassified as buses.

The random forest continues the trend of misclassifying cars as buses, at a rate of 23.8%, with 7.27% of bicycles also classified as buses. The inverse misclassification rates are 5.21 and 6.7%, respectively.

The F1 scores from running the experiment with only the hypothesized features are in Table 15, and the result with only the "traditional" features is in Table 16. In the latter case, each criterion of feature selection selected only three or four features.

F1-scores achieved without spectral features are in Table 17, and those achieved with only spectral features are in Table 18. Using only spectral features resulted in only two features from the *t*-test, four from the *U*-test and five from the *f*-test.

Table 19 shows the F1 scores achieved with only auto- and cross correlations. The number of selected features was four features for *t*- and *U*-tests and six for the *F*-test. The result of excluding auto- and cross-correlations is the same as in Table 8.

Table 21 shows the average F1 scores achieved when using only skewnesses and kurtoses. Unsurprisingly this resulted in three or four features per selection criterion. Table 20 shows the F1 scores achieved when skewnesses and kurtoses were excluded.

7. Discussion

7.1. Hypothesis testing

For all three hypotheses outlined in 2.2, the *F*-test produced p-values less than 0.01 for at least one feature.

Spectral components for acceleration and speed were selected for the walk/wheeled and car/bus sub-classifications. They were also among the most significant features selected by the *F*-test.

Using only spectral features produced worse classification results than using only traditional features. Excluding spectral components impacted the Bayes classifier, although the other classifiers performed better by a similar margin. Oddly, the effect was least visible with the *F*-test, which also selected the most spectral features.

These discrepancies may be explained by the high correlation between spectral features, which reduced the information content of a feature vector containing several of them. Therefore, Hypothesis 1 can be tentatively accepted.

As an illustration of the spectral components' suitability for the classification task, a scatterplot of two speed spectral components is shown in Fig. 3.

Auto- and cross-correlations were used as a yardstick for poor features, since Hypothesis 2 was rejected during initial experimentation. Indeed, Bayes classification using only auto- and cross correlations performed barely better than chance.

Looking at the scatterplot of cross-correlations in Fig. 4 reveals that the classes are heavily overlapped.[1] Therefore the hypothesis can be discarded.

Skewness of speed was selected as one of the best features to differentiate bicycles and motorized by the *t*- and *U*-tests. Therefore, Hypothesis 3 might be accepted. However, as with the correlations, the scatterplot in Fig. 5 tells a different story, as does the classification accuracy using only skewnesses and kurtoses. The hypothesis can also be discarded.

It appears that for both discarded hypotheses, the walking class had a cluster that was slightly offset from the other modes' combined cluster, as did the bus class. These two factors may have lead to the high test scores for the features.

7.2. Classification

Selecting the features by the *F*-test produced a noticeably better-performing classifier only one case, although even then the difference in F1 score was in the order of 0.05. Where the *F*-test stood out, was ranking the generally poorly-performing features low. In general, ease of implementation makes the *F*-test a good choice for the algorithm laid out in this paper.

[1] The correlations were calculated using the means and standard deviations calculated from the entire trace, which caused some of the autocorrelations to lie outside the ±1-range.

Table 15
Average F1 scores with only hypothesized features.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.790	0.839	0.841	0.853	0.851
U-test	0.817	0.874	0.881	0.884	0.882
F-test	0.815	0.891	0.897	0.897	0.892

Table 16
Average F1 scores with only traditional features.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.814	0.891	0.894	0.893	0.902
U-test	0.806	0.890	0.891	0.890	0.900
F-test	0.829	0.888	0.890	0.891	0.899

Table 17
Average F1 scores without spectral features.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.829	0.903	0.902	0.904	0.905
U-test	0.829	0.902	0.899	0.905	0.905
F-test	0.833	0.895	0.898	0.902	0.902

Table 18
Average F1 scores with only spectral features.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.767	0.817	0.817	0.819	0.829
U-test	0.784	0.861	0.869	0.870	0.873
F-test	0.799	0.882	0.886	0.888	0.886

Table 19
Average F1 scores with only auto- and cross correlations.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.356	0.478	0.482	0.494	0.554
U-test	0.356	0.479	0.480	0.476	0.554
F-test	0.364	0.488	0.500	0.508	0.562

Table 20
Average F1 scores without skewnesses and kurtoses.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.835	0.900	0.900	0.902	0.907
U-test	0.841	0.894	0.900	0.904	0.903
F-test	0.849	0.889	0.898	0.897	0.901

Table 21
Average F1 scores with only skewnesses and kurtoses.

Ranking criterion	BC	NN1	NN2	NN3	RF
Welch's *t*	0.333	0.472	0.475	0.482	0.553
U-test	0.333	0.478	0.488	0.485	0.553
F-test	0.344	0.479	0.486	0.493	0.555

The neural network performed the best with all selection criteria. One hidden layer slightly outperformed two hidden layers with two tests, and performed worse with Welch's *t*-test.

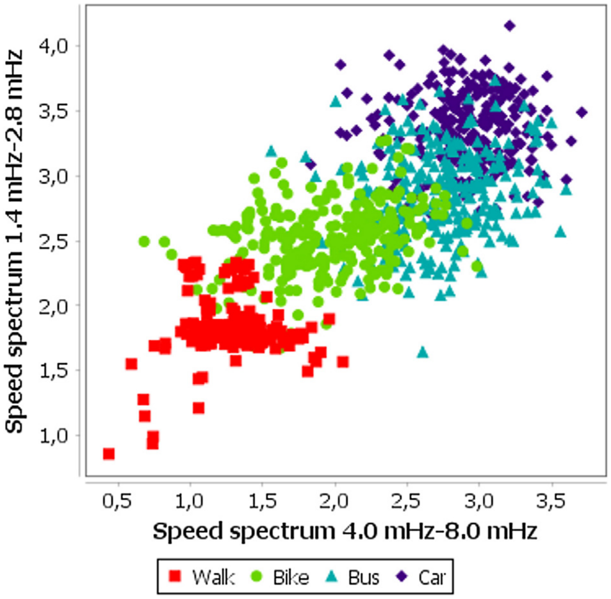

Fig. 3. Scatter plot two spectral components of speed.

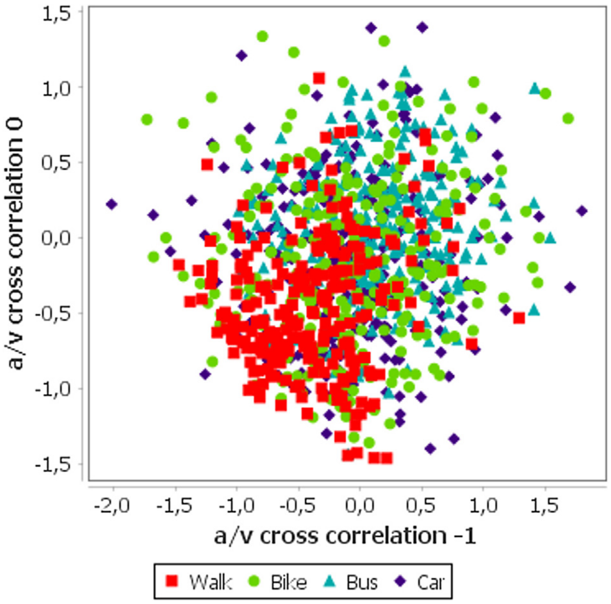

Fig. 4. Scatter plot of the highest *f*-statistic correlations.

The critical value for a Mann-Whitney *U*-test at $p < 0.05$ is zero when both sample sizes are four or less. In other words, it can be said the random forest outperforming the other methods is statistically significant according to the *U*-test, although there is little real-world significance to the difference.

Reducing the feature vector's dimensionality with an autoencoder performed poorly compared to feature selection, most likely due to the large amount of low-information features it needed to account for.

Buses stood out as a confounding class, as can be deduced from Fig. 3. In particular, the class had a precision approximately ten percent points worse than the others on the neural networks and the Bayes classifier, and just under seven percent on the random forest.

Visualizations were made for the highest F1 score classifier of each class, by assigning points at $(0, \pm 1)$ and (± 1.0) to the four classes counter-clockwise from (1.0), and assigning each datapoint a location as a sum of these points weighted by the likelihood, normalized to between 0 and 1, given by each classifier.

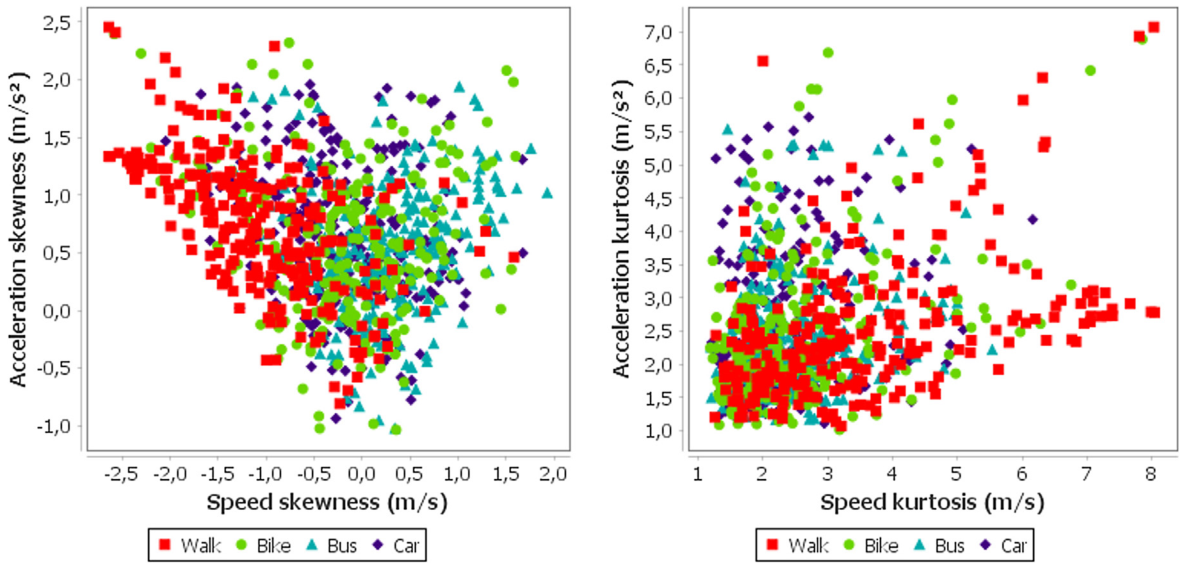

Fig. 5. Scatter plots of speed and acceleration skewnesseses and kurtoses.

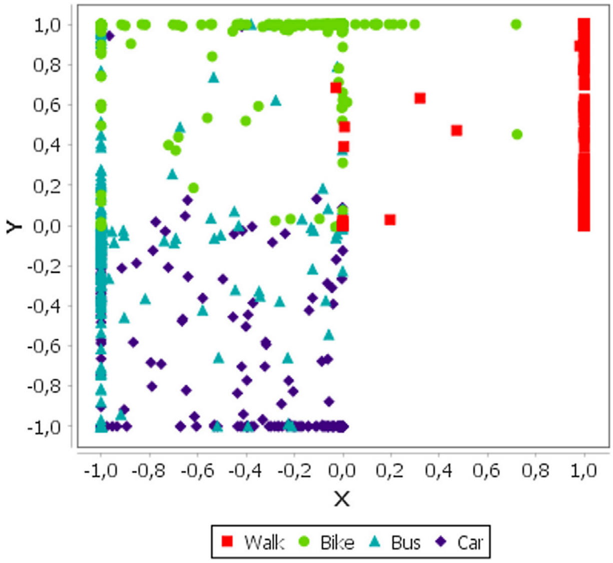

Fig. 6. Visualization of the Bayes classifier.

As can be seen from Fig. 6, the Bayes classifier often assigned high likelihoods to two or more classes. The confusion between buses and bicycles can be seen from the way the data of the two classes are commingled, and a large chunk of the car class can be seen clustered near the origin and skewed toward buses.

Fig. 7 shows the visualization for the neural network. The classifier did not, as a rule, assign a high likelihood to more than one class, but often assigned middling likelihoods to two classes. The intermingling of buses with cars and bicycles is clearly visible.

Fig. 8 shows the visualization of the random forest. The forest's likelihoods tended to land on a line between two classes, and appear to form a discrete set. The discreteness makes the magnitude of confusions harder to spot from the visualization, but the car and bicycle classes can be seen tending towards the bus class.

Fig. 9 shows the autoencoder's visualization. It would appear that using an autoencoder forced the results closer to a line between two classes compared to the neural network. However, walking was considerably more problematic for this classifier than a regular neural network.

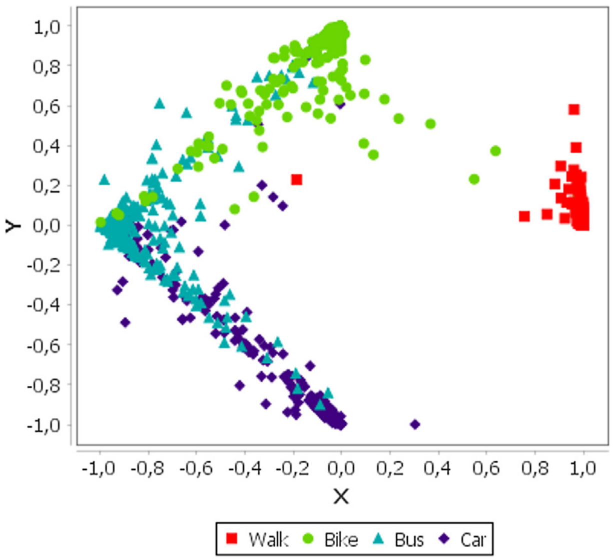

Fig. 7. Visualization of the neural network.

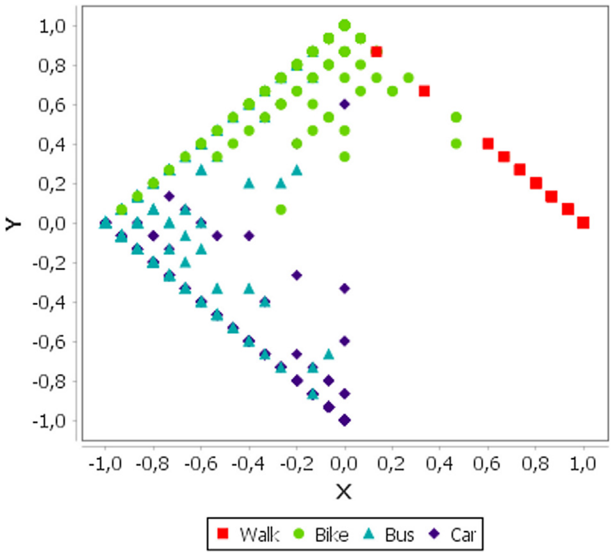

Fig. 8. Visualization of the random forest classifier.

From the first three visualizations, it would appear that the classifiers would have a near-perfect accuracy if it were not for the bus class.

8. Conclusion

This paper had three main contributions. The first contribution was furthering the research into the use of spectral components in travel mode recognition. In particular, frequency components were found to be useful in separating walking from wheeled modes of transport, and buses from cars.

The second contribution was establishing that auto- and cross-correlations, skewnesses and kurtoses of speed or accelerations at granularity of 1 sample/min are not useful.

The third contribution was a comparison of combined two sample statistical tests with an *F*-test. Somewhat counterintuitively, selecting features by an *F*-test produced slightly worse results than using a two-class statistical test like the *t*-test for three manually selected classifications and combining the results.

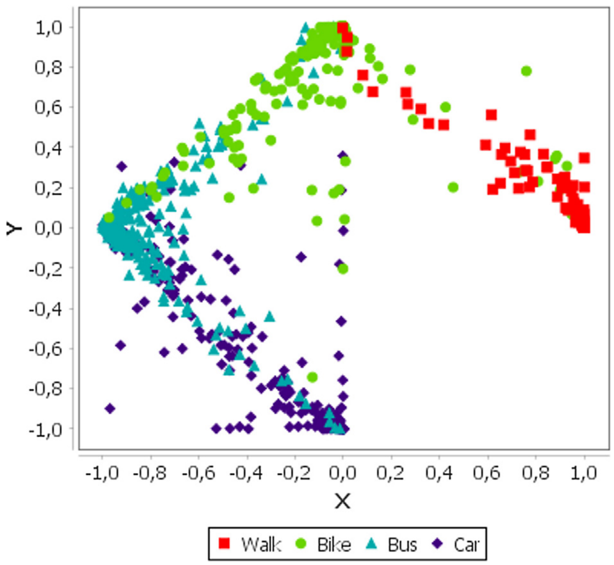

Fig. 9. Visualization of the autoencoder classification.

A random forest was found to offer a better combination of precision and recall than a Bayes classifier, and slightly better than a neural network. Adding a second hidden layer to a neural network increased the accuracy by a small amount, but the increase was considerably smaller when adding a third.

Applying an autoencoder to the entire feature space produced poor classification results. This is worse than Zhu et al. (2016), which was partly due to the other study's inclusion of public transport similarity among the features, and partly the low quality of some of the features the autoencoder was trained to encode.

The most comparable study to this paper, Stenneth et al. (2011), achieved worse results when not using public transit similarity for classification. However, they were using one more class, train, which explains the discrepancy. Compared to the other studies, the classification results were fairly average, although the differences in data sources, features considered and sampling times reduce the comparability. In particular, this paper fell short of the state-of-the-art set by Zhu et al. (2016).

Further research is needed into correctly classifying buses, by excluding regular cars and differentiating buses from bicycles. Stenneth et al. (2011) reported a fairly large jump in accuracy when public transit similarity was included, as did the proof of concept in Mäenpää (2017).

The dataset used for this paper was very tentatively labeled, and underwent considerable pre-processing to make the sampling rate uniform. Repeating this study with a dataset with more definitively labeled traces, and with consistent sampling rates, should produce more reliable results. Another improvement to the dataset would be to restrict the traces to the area of a specific city or cities, in order to add map-matching related features to the selection.

Appendix. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.trc.2017.06.021.

References

Allahviranloo, M., Recker, W., 2013. Daily activity pattern recognition by using support vector machines with multiple classes. Transport. Res. Part B: Methodol. 58, 16–43. URL <http://www.sciencedirect.com/science/article/pii/S0191261513001689>.

Bantis, T., Haworth, J., 2017. Who you are is how you travel: a framework for transportation mode detection using individual and environmental characteristics. Transport. Res. Part C: Emerg. Technol. 80, 286–309. URL <http://www.sciencedirect.com/science/article/pii/S0968090X1730133X>.

Bobol, A., Cheng, T., 2010. GPS data collection setting for pedestrian activity modelling. In: GISRUK 2010: Proceedings of Geographical Information Science Research UK Conference 2010.

Bohannon, R.W., 1997. Comfortable and maximum walking speed of adults aged 20–79 years: reference values and determinants. Age Age. 26 (1), 15–19. URL <http://ageing.oxfordjournals.org/content/26/1/15.abstract>.

Bolbol, A., Cheng, T., Tsapakis, I., Haworth, J., 2012. Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. Comput. Environ. Urban Syst. 36 (6), 526–537 (Special Issue: Advances in Geocomputation). URL <http://www.sciencedirect.com/science/article/pii/S0198971512000543>.

Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140. http://dx.doi.org/10.1007/BF00058655.

Fosgerau, M., Karlström, A., 2010. The value of reliability. Transport. Res. Part B: Methodol. 44 (1), 38–49. URL <http://www.sciencedirect.com/science/article/pii/S019126150900068X>.

Gong, H., Chen, C., Bialostozky, E., Lawson, C.T., 2012. A GPS/GIS method for travel mode detection in New York city. Comput. Environ. Urban Syst. 36 (2), 131–139 (Special Issue: Geoinformatics 2010) URL <http://www.sciencedirect.com/science/article/pii/S0198971511000536>.

He, Y.-L., Wang, R., Kwong, S., Wang, X.-Z., 2014. Bayesian classifiers based on probability density estimation and their applications to simultaneous fault diagnosis. Inform. Sci. 259, 252–268. URL <http://www.sciencedirect.com/science/article/pii/S0020025513006361>.

Heaton, J., 2015. Encog: library of interchangeable machine learning models for Java and C#. J. Mach. Learn. Res. 16, 1243–1247. URL <http://jmlr.org/papers/v16/heaton15a.html>.

Hinton, G.E., Salakhutdinov, R.R., 2006. Reducing the dimensionality of data with neural networks. Science 313 (5786), 504–507. URL <http://science.sciencemag.org/content/313/5786/504>.

Jenelius, E., 2012. The value of travel time variability with trip chains, flexible scheduling and correlated travel times. Transport. Res. Part B: Methodol. 46 (6), 762–780. URL <http://www.sciencedirect.com/science/article/pii/S0191261512000185>.

John, G.H., Langley, P., 1995. Estimating continuous distributions in bayesian classifiers. In: Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. UAI'95. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 338–345. URL <http://dl.acm.org/citation.cfm?id=2074158.2074196>.

Kwapisz, J.R., Weiss, G.M., Moore, S.A., 2011. Activity recognition using cell phone accelerometers. SIGKDD Explor. Newsl. 12 (2), 74–82. http://dx.doi.org/10.1145/1964897.1964918.

Liao, F., Arentze, T., Timmermans, H., 2013. Incorporating space-time constraints and activity-travel time profiles in a multi-state supernetwork approach to individual activity-travel scheduling. Transport. Res. Part B: Methodol. 55, 41–58. URL <http://www.sciencedirect.com/science/article/pii/S0191261513000751>.

Mäenpää, H., 2017. User Tracking and Incentive Management in Smart Mobility Systems Master's thesis. Tampere University of Technology. URL <http://urn.fi/URN:NBN:fi:tty-201703221202>.

Parkka, J., Ermes, M., Korpipaa, P., Mantyjarvi, J., Peltola, J., Korhonen, I., 2006. Activity classification using realistic data from wearable sensors. IEEE Trans. Inform. Technol. Biomed. 10 (1), 119–128.

Prelipcean, A.C., Gidófalvi, G., Susilo, Y.O., 2017. Transportation mode detection – an in-depth review of applicability and reliability. Transp. Rev. 37 (4), 442–464. http://dx.doi.org/10.1080/01441647.2016.1246489.

Reddy, S., Burke, J., Estrin, D., Hansen, M., Srivastava, M., 2008. Determining transportation mode on mobile phones. In: 2008 12th IEEE International Symposium on Wearable Computers, pp. 25–28.

Reddy, S., Shilton, K., Burke, J., Estrin, D., Hansen, M., Srivastava, M., 2009. Using Context Annotated Mobility Profiles to Recruit Data Collectors in Participatory Sensing. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp. 52–69. http://dx.doi.org/10.1007/978-3-642-01721-6_4.

Riedmiller, M., Braun, H., 1993. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. IEEE International Conference on Neural Networks, 1993, vol. 1, pp. 586–591.

Stenneth, L., Wolfson, O., Yu, P.S., Xu, B., 2011. Transportation mode detection using mobile phones and GIS information. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. GIS '11. ACM, New York, NY, USA, pp. 54–63. http://dx.doi.org/10.1145/2093973.2093982.

Stopher, P., FitzGerald, C., Zhang, J., 2008. Search for a global positioning system device to measure person travel. Transport. Res. Part C: Emerg. Technol. 16 (3), 350–369 (Emerging Commercial Technologies). URL <www.sciencedirect.com/science/article/pii/S0968090X07000836>.

Su, X., Tong, H., Ji, P., 2014. Activity recognition with smartphone sensors. Tsinghua Sci. Technol. 19 (3), 235–249.

Sun, Z., Ban, X.J., 2013. Vehicle classification using {GPS} data. Transport. Res. Part C: Emerg. Technol. 37, 102–117. URL <http://www.sciencedirect.com/science/article/pii/S0968090X13002040>.

Thompson, D.C., Rebolledo, V., Thompson, R.S., Kaufman, A., Rivara, F.P., 1997. Bike speed measurements in a recreational population: validity of self reported speed. Injury Prevent. 3 (1), 43–45. URL <http://injuryprevention.bmj.com/content/3/1/43.abstract>.

Xiao, G., Juan, Z., Zhang, C., 2016. Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization. Transport. Res. Part C: Emerg. Technol. 71, 447–463. URL <http://www.sciencedirect.com/science/article/pii/S0968090X16301425>.

Xu, Q., Zhang, C., Zhang, L., Song, Y., 2016. The learning effect of different hidden layers stacked autoencoder. 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), vol. 02, pp. 148–151.

Zhu, X., Li, J., Liu, Z., Wang, S., Yang, F., 2016. Learning transportation annotated mobility profiles from GPS data for context-aware mobile services. In: 2016 IEEE International Conference on Services Computing (SCC), pp. 475–482.