Extracting Sections of Simulated Driving Routes that Elicit Driving Responses Predictive of ADHD via Recursively Constructed Ensembles

David Grethlein and Santiago Ontañón
{djg329, so367}@drexel.edu
Drexel University
Philadelphia, Pennsylvania, USA

Aleksanteri Sladek
asladek@seas.upenn.edu
University of Pennsylvania
Philadelphia, Pennsylvania, USA

Abstract

In this paper we introduce a novel algorithm called Iterative Section Reduction (ISR) to automatically identify spatial regions wherein time series were recorded that are predictive of a target classification task. Specifically, using data collected from a driving simulator study, we identify which spatial regions (dubbed sections) along the simulated routes tend to manifest driving behaviors that are predictive of the presence of Attention Deficit Hyperactivity Disorder (ADHD). Identifying these sections is important for two main reasons: (1) to improve predictive accuracy of the trained ADHD screening models by filtering out non-predictive time series data, and (2) to gain insights into which on-road scenarios (dubbed events) elicit distinctly different driving behaviors from patients undergoing treatment for ADHD versus those that are not. Our experimental results show both improved classification performance over prior efforts and good alignment between the predictive sections identified and scripted on-road events in the simulator (negotiating turns and curves).

Introduction

Motor Vehicle Collisions (MVCs) are still a leading cause of death in the United States (Kochanek et al. 2019), claiming over 36,000 lives in 2018 alone (NCSA 2019). Traffic safety experts estimate that adolescent drivers are 3 times as likely to be involved in a MVC than others (IIHS 2019). Those with cognitive impairments to their executive functions like Attention Deficit Hyperactivity Disorder (ADHD) are particularly vulnerable (Walsh et al. 2017). ADHD-associated cognitive impairment has been linked to risk-generating driving behaviors such as being easily distracted and unawareness of one’s surrounding (Chang et al. 2014). However, some effective clinical methods for detecting ADHD, rely on invasive Functional Magnetic Resonance Imaging (fMRI) technologies that require specialized equipment and trained staff to operate it (Iannaccone et al. 2015).

In this work, we aim to non-invasively identify which particular on-road hazards elicited responses from adolescent drivers (age 18-24) that most strongly indicated whether the driver had less-advanced forms of ADHD and was undergoing treatment or not (what we call ADHD status). Armed with this knowledge, driving instructors can design meaningful interventions in the simulator to mitigate dangerous behaviors adolescents with untreated ADHD might exhibit behind the wheel. The immediate goal of this work is to deliver feedback to driving simulator route designers detailing which scripted scenarios (dubbed events) they employed actually identified a driver’s ADHD status. To deliver this feedback, we build off existing work done designing common crash scenarios to assess adolescent driver attention and skill levels in a driving simulator (McDonald et al. 2012). Our work departs from these efforts in that we uncover the most predictive events without any a priori assumptions of which will work well and which won’t.

Background

Driving simulators are a rich source of time series data for capturing the indicators of risk-generating behaviors (Chandrasiri, Nawa, and Ishii 2016), as well being a widely available technology that requires minimal supervision to perform an assessment. Time series output recorded in a driving simulator (throttle, brake, and steering wheel inputs) has been fed into k-Nearest Neighbor (k-NN) classifiers and detected driver inattention under cognitive load (Chakraborty and Nakano 2016). Dynamic Time Warping (DTW) similarity-based time series clustering has also been used for modelling and differentiating different driving behaviors by analyzing vehicle velocity in various sections of road (Lohrer and Lienkamp 2015).

Furthermore, bagging ensembles composed of Long Short-Term Memory (LSTM) neural networks have been used to successfully predict lane departures and lateral driving speed (Altché and de La Fortelle 2017). Likewise, other recurrent neural network (RNN) architectures have been used successfully to assess driver performance in simulators, often tracking the driver’s on-screen gaze (Hori et al. 2016). Successes in these similar driver classification tasks motivated our decision to employ a deep neural network incorporating an LSTM layer in some of our experiments. Moreover, due to the small size of our dataset, we also independently considered several similarity based models.

The classes of drivers (ADHD status) in this dataset have been previously shown to be hard to separate when globally (using the complete time series as input) comparing simulator sessions to one another (Grethlein and Ontañón 2020). We hypothesize that previous classification efforts on this dataset have failed (< 50% accuracy on a 3-way classifica-
tion task) as the classes of drivers largely drive uniformly except in response to certain on-road scenarios. In previous experiments using this dataset, each route was broken into a sequence of manually defined, non-overlapping sections of road (Grethlein, Sladek, and Ontañoén 2021). Experiments with an ensemble of classifiers (nearest neighbours and 1-D convolutional neural networks (1-D CNN)) trained on each of these sections separately revealed high sensitivity to ADHD status concentrated in data recorded within sections containing turns and curves along the route. Moreover, we suspect that manually defining sections might have cut off certain predictive behaviors.

Methods

In this section, we first describe the spatiotemporal time series dataset used in our experiments. We then define our Iterative Section Reduction (ISR) ensemble-building algorithm, which leverages a known alignment domain (such as spatial or temporal axes) of a time series dataset to: (1) automatically identify discriminatory sections along it, and (2) build an ensemble of arbitrary classifiers restricted to evaluating data from the identified sections. Next, we describe the classifiers used for composing these ensembles, and define measures which evaluate an ensemble’s ability to identify the classes of drivers by only examining parts of the alignment domain. Finally, we detail our grid-search strategy for tuning ISR to produce optimal results from our dataset.

Driving Simulator Time Series Data

Our dataset was collected under National Science Foundation Grant No. 1521943 from $n = 30$ participants using a Realtime Technologies RDS-1000 driving simulator; its raw form consists of 91 synchronized time series channels recorded at 60 Hz. In order to reduce the overall computation time for all models considered, we reduced the dimensionality of the data via Piecewise Aggregate Approximation (PAA) (Keogh et al. 2001) to 10 Hz and only use a subset of 7 channels from each recording as input. After down-sampling, simulator sessions varied in length from 3,829 to 9,746 frames, with an average of 5,728 frames; this after certain predictive behaviors.

Our experimental group consisted of 15 of the 30 total participants with confirmed clinical diagnoses of ADHD prior to recruitment into this study (Lee et al. 2018). These participants were recorded driving 4 planned routes (Drive 1, ..., Drive 4) twice; one session while receiving prescribed treatment (regulated), and once on a placebo (delayed). The other 15 participants, referred to as the control (no confirmed ADHD diagnoses) were each recorded driving the same 4 routes once. One partial Drive 3 session from the delayed group was discarded due to simulator motion sickness, leaving a total of $N = 179$ complete sessions.

Distributed along each simulated route are sequences of events (see Figure 1), regions within the route that have been scripted to elicit responses that are telling of driver attentiveness and reactivity (collision avoidance, negotiating intersections, etc.). The position of these events along the route are static, and therefore all participants driving the route encounter the same events in the same frequency and order. We view these events as potential sections to discriminate drivers with untreated forms of ADHD from those without. Comparing the annotated events to the sections identified by the ISR algorithm and determine whether the events truly elicit driving behaviors predictive of ADHD status.

To facilitate iteratively dividing the simulated routes into sections, we created lattice data structures to represent the routes. These lattices consist of ordered sequences of waypoints, points spaced at 5-meter intervals in the center of the road along each route. As a result, the smallest time series that could be extracted from our dataset will be recorded strictly within a 5-meter section of a route. For each route we define a sequence of top-level sections, 1600-meter long sections of the route displaced with a stride of 800 meters. We chose 1600-meter top-level sections as we assumed any behavior could be completely expressed within such a section. Top-level sections have 50% overlap for flexible section boundaries, since it isn’t known where along the route behaviors predictive of ADHD will manifest. With 31 top-level sections defined across all 4 routes, we expanded our dataset of 179 global time series to 1,567 clipped time series.

Iterative Section Reduction Ensembles

Iterative Section Reduction (ISR) is an ensemble-building algorithm (see Algorithms 1 and 2) designed to leverage an alignment domain for the purpose of automatically isolating time series that are highly predictive of class labels. By iteratively dividing an inherent alignment domain (in our case the route), ISR constructs ensembles composed of multiple section-specific classifiers (dubbed as modules) that strictly process the time series recorded within their respective sections. These modules then classify time series of previously unseen data and vote (with equal weighting) on a global prediction for each complete simulator session.

To isolate sections in the alignment domain that contain indicators of class sensitivity, ISR begins by dividing the alignment domain into top-level sections (see Algorithm 1, line 2). The corresponding data in each section is then partitioned (in our case by driver) into $k$ disjoint fold (ideally preserving proportions of classes in each fold). The evaluation fold is immediately reserved from further consideration
Figure 1: Scripted on-road scenarios (dubbed events) distributed along the 4 simulated routes. We partition the routes into 1600-meter long top-level sections starting every 800 meters; explicitly including the last 1600 meters of the route as a section.

**Algorithm 1** Build Iterative Section Reduction Ensemble

**data**: Labelled time series dataset ($X$ and $y$).

**domain**: The alignment domain (simulated driving route).

**$\Lambda$**: Size (in our case, length) of top-level sections.

**$\Delta$**: The displacement stride of top-level sections.

**$k$**: Number of folds to split dataset into for cross-validation.

**$i$**: The index of the single data fold reserved for evaluation.

**$M$**: The max height ISR trees may grow to.

**$\tau$**: Threshold value for module inclusion in ensemble.

**$C$**: The type of classifiers (modules) voting in the ensemble.

**$p$**: Model selection paradigm (ALL vs. ANY).

1: function ISR(data, domain, $\Lambda$, $\Delta$, $k$, $i$, $M$, $\tau$, $C$, $p$)
2: sections ← get_top_level_sections(domain, $\Lambda$, $\Delta$)
3: $x_{data}$, $e_{data}$ ← stratified_split(data, $k$, $i$)
4: ensemble ← []
5: for sec ∈ sections do
6:   $s_{mods}$ ← mods($x_{data}$, sec, $k-1$, 0, $M$, $\tau$, $C$, $p$))
7:   ensemble.append($s_{mods}$)
8: return ensemble

while building the ensemble (Algorithm 1, line 3). The remaining $k = k - 1$ folds are the experimental set; used to construct an ensemble of modules processing data recorded strictly within each top-level section (Algorithm 1, line 6).

The experimental set is used to consider $k$ independently trained modules per section via cross-validation (Algorithm 2, lines 2-13). Note, this means each module was trained using a unique combination of $k - 1$ folds of data (Algorithm 2, line 5-6), each producing an initial estimate of accuracy called the development accuracy (Algorithm 2, line 8). This estimate, is then saved by ISR for use in model selection, determining whether a module (and hence a specific section of the alignment domain it was trained on) will be included in the final ensemble (Algorithm 2, lines 10-17). ISR imposes a threshold ($\tau$) on modules as model selection to tune the degree of sufficient sensitivity to class labels for inclusion.

If any suitable (mean development accuracy $\geq \tau$) candidate modules for inclusion are found in a section, the ISR algorithm attempts to leverage its sub-sections to further separate non-predictive parts of the alignment domain from conflating with predictive ones (Algorithm 2, lines 2-13). Sub-section modules must out-perform section modules to be included in the ensemble (Algorithm 2, lines 14-17). Sub-sections are only examined if the ISR tree hasn’t reached its max height $M$ (Algorithm 2, line 13). Each section is broken down into three halves: 50% the size of the original section and collected with stride of 25% the size; in other words, a first half, middle half, and last half (Algorithm 2, line 19). By using overlapping section boundaries, predictive behaviors are less likely to be split into multiple sections.

**Section-Specific Classifiers**

Our time series dataset was used in two sets of experiments, where the types of modules (denoted as $C$) used as voters in an ensemble were varied. In the first set of experiments, we built ISR ensembles from strictly either k-Nearest Neighbors (k-NN) or logistic regression (LogReg) classifiers. These modules were fed similarity matrices that relate all multivariate time series recorded in a section to one another as input. These models were chosen as they require minimal parameter tuning to get up and running. In all such experiments time series were down-sampled using PAA to 1 Hz to further expedite similarity matrix construction.
In the second set of experiments, we developed our own deep learning modules, dubbed DeepLSTM. Each DeepLSTM in the ensemble is composed of the following sequence of layers: two 1-D CNN layers each with 64 filters (filter size 5, stride 1) to function as feature extractors, an average pooling layer, an LSTM layer (with a hidden layer size of 64) and one Fully Connected (FC) layer of 64 neurons. The FC layer, using the Rectified Linear Unit (ReLU) activation function, feeds into an output layer of three neurons using soft-max activation, which produces the final class-wise probabilities. For all networks we used the Adam optimizer and the categorical cross entropy loss function during the training phase. To reduce overfitting, 20% dropout layers were added between the LSTM and FC layer, as well as between the FC and output layer. Each network was fed inputs in mini-batches of 32 windows at a time, trained for 50 epochs with a learning rate of $10^{-4}$. This fixed number of training epochs was chosen given the small size of datasets fed to each DeepLSTM module and the large volume of models trained with ISR, as this reduces overfitting and computation time.

The DeepLSTM classifiers, in contrast to LogReg and k-NN, were fed the section-specific 10 Hz clippings directly. Training instances were extracted from the clippings with the sliding window method, with a window size of 30 frames (roughly 3 seconds) and a window stride of 15 frames. A small window size was chosen as many of the smallest sections considered had as little as 10 seconds worth of data for a given participant. Presented with the need to evaluate a small number of already very short time series clippings, DeepLSTM was fed the 10 Hz data to preserve both the number of overall data points, and any scarce time series features resulting from flattening via PAA.

### Evaluation Measures

For each route (Drive 1, ..., Drive 4) in our dataset, we constructed $k = 5$ ensembles by choosing each of the $k$ folds created by ISR once (via $i$) to be reserved as an evaluation fold for a given ensemble (Algorithm 1, line 3). After being built with the remaining $k-1$ folds, the ISR modules classify previously unseen time series in the evaluation fold clipped to the identified sections. Finally, ISR ensembles tally the (evenly weighted) votes of all member modules and produce a final classification for each complete driving simulator session examined. The reliability of these predictions is then measured by computing the evaluation accuracy. We report herein the 5-fold average evaluation accuracy (denoted Acc) of ISR ensembles constructed on a single route.

Every ISR ensemble built also produced a subset $S_i$ of the route lattice way-points $L$, defining sections of the planned route found to have elicited driving behaviors predictive of the driver’s ADHD status. As each route contained pre-determined events depicted in Figure 1, the subset $E$ of the lattice way-points covering these events was known to us. This allowed us to analyze the overlap of ISR’s unanimously-selected route sections with on-road events using Jaccard similarity of Intersection with Events (JIwE) (equation 1). JIwE is defined as the ratio of the number of lattice way-points both shared by all $k$ ensembles and overlapping known events $E$ to the number of way-points composing both the events and the union of all ensembles.

$$JIwE = \frac{|\bigcup_{i=1}^{k} S_i \cap E|}{|\bigcup_{i=1}^{k} S_i \cup E|}$$

JIwE was motivated to quantify the degree to which the events designed by the route developers to elicit ADHD-predictive behaviors were effective in doing so. By evaluating which sections of the route all 5 ensembles ISR found to contain predictive behaviors, we can provide route designers feedback on how commonly their scripted events made a difference in predicting the ADHD status of drivers.

### Hyper-parameter Grid Search

We experimented with two different paradigms, strategies for determining which development fold modules are included in the ensemble: all folds per section (ALL) and any fold above (ANY). In the ALL paradigm, if the average development accuracy from the $k-2$ modules is above the threshold, then all are considered as candidates for inclusion in the ensemble. In the ANY paradigm, any individual module with development accuracy above the threshold is considered as a candidate. In both paradigms, if no potential
candidates are selected, the section is deemed non-predictive and discarded from further consideration. Additionally, for both paradigms the average development accuracy of all ensemble candidates in a section is made the new threshold value for candidates in any subsequent sub-section to beat.

In order to gauge the maximum sensitivity to ADHD status achievable we tested hundreds of ISR parameterizations for each route. In the first set of experiments, we constructed similarity matrices to compare the driving performance of all participants to one another within a defined section. We elected to use brute force DTW and its approximations Sakoe-Chiba Dynamic Time Warping (SC DTW) and FastDTW to gauge how precisely time series similarity must be defined between samples to render accurate classifications (Salvador and Chan 2007). The radius (R) parameter in both extensions to the DTW algorithm characterize the size of a search area for finding the alignment between time series frames that nearly minimizes the sum of residual differences, supposedly at a fraction of the computational cost. For all similarity-based (Sim) experiments, we permuted the 15 similarity functions (DTW, SCDTW and FastDTW both with \( \tau \in \{1, 5, 10, 15, 20, 25, 30\} \)) and the types of base classifier (1-NN and LogReg).

For both sets of experiments, we also permuted \( M \), i.e. the max ISR tree depth, of top-level section trees allowed using values 1, 2, 3, and 4. Additionally, we observed the effects of setting a higher initial threshold for the ISR algorithm with both ANY and ALL paradigms by testing the values 0%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, and 50%. This resulted in 2,640 unique ISR parameterizations being evaluated per route in the similarity-based experiments and 88 unique DeepLSTM parameterizations per route. We performed these grid searches and report on the initializing ISR hyper-parameters that yielded the highest \( \text{Acc} \) across all \( k = 5 \) folds.

## Results

ISR ensembles out-performed previous efforts to classify the ADHD status of drivers (Grethlein, Sladek, and Ontaños 2021) in routes Drive 1, Drive 3, and Drive 4. Each route yielded some ISR models with partial sensitivity (\( \text{Acc} \geq 33\% \)) to ADHD status. Table 1 shows the most accurate ensemble results and hyper-parameters per route from all similarity-based models. Table 2 shows the same for the DeepLSTM models.

| Route | Similarity Measure | C | \( \tau \) | \( M \) | \( \text{Acc} \) | Prior \( \text{Acc} \) | JwaE |
|-------|--------------------|---|---------|------|--------|-------------|-----|
| Drive 1 | FastDTW \( \tau = 1 \) | LogReg | 0.00 | 2 | 0.58 | 0.49 | 0.80 |
| Drive 2 | SC DTW \( \tau = 15 \) | LogReg | 0.00 | 2 | 0.44 | 0.44 | 0.63 |
| Drive 3 | FastDTW \( \tau = 1 \) | 1-NN | 0.30 | 4 | 0.62 | 0.59 | 0.72 |
| Drive 4 | DTW | 1-NN | 0.35 | 4 | 0.44 | 0.38 | 0.28 |

Table 1: Ensembles achieving highest \( \text{Acc} \) per route from all similarity-based grid searches. Prior \( \text{Acc} \) is the highest 5-fold evaluation accuracy obtained for each route so far (Grethlein, Sladek, and Ontaños 2021). Best classification results for each route highlighted in bold.

All ensembles benefited from at least one level \( (M > 1) \) of sub-sections to extract predictive scenarios; choosing higher \( \tau \) values yielded less accurate ISR ensembles overall, with results typically worsening significantly when \( \tau > 33\% \) (the class \( \text{a priori probability} \)). Setting too high a value for \( \tau \) at shallow levels of the ISR tree tended to preclude the discovery of more predictive candidates at deeper levels. Similarity-based ensembles tended to be more accurate for Drive 1 and Drive 3; in the DeepLSTM experiments JwaE tended to be lower as DeepLSTM ensembles agreed less consistently on sections deemed predictive (Figure 2).

## Discussion

Events involving stop sign intersections, curves, and avoiding collisions with other vehicles were often selected as predictive of ADHD status in most ensembles with high accuracy (\( \text{Acc} > 50\% \)). Overall, the \textit{control} group was easily identified by most ensembles, however the \textit{delayed} were confused with \textit{regulated} and the \textit{regulated} with the \textit{control} group. This could suggest that the medication being taken by some participants was partially effective in mitigating ADHD-associated risk-generating driving behaviors.

While our results suggest that driver ADHD status may be inferred from certain on-road scenarios, as part of our future work, we would like to test ISR against techniques like shapelet decision trees (Cetin, Mueen, and Calhoun 2015). Modest model performance reiterates the difficulty of our task, matching global ADHD status labels to long time series. We also acknowledge improved ADHD classification accuracy could likely be achieved if we had collected eye-tracking or other physiological time series channels from participants along with their driving performance data. A purpose of our study, and perhaps a limitation of it was to predict driver ADHD status without informative biometrics.

We posit that similarity-based ISR ensembles performed marginally better in most cases for this small dataset as deep neural nets are notoriously greedy for large sample sizes to perform well. On the other hand, we anticipate that given sufficient processing resources DeepLSTM ensembles would scale better to accommodate a larger time series dataset than the similarity-based methods, due to quadratically increasing computation costs for similarity matrices.

## Conclusions

Since we only studied one small dataset of \( n = 30 \) drivers, our findings about which on-road scenarios elicit responses predicative of ADHD status while statistically robust, must be treated as anecdotal. That being said, the ISR algorithm demonstrated a capacity for isolating predictive sub-sections
Figure 2: (Top) The best performing Drive 3 ISR ensembles were built using FastDTW with $R = 1$ as the underlying similarity function comparing time series, 1-NN modules, the ANY inclusion paradigm, a max depth of 3, and an initial threshold of 30%. (Bottom) The best performing Drive 3 DeepLSTM ensembles were built using the ALL paradigm, a max depth of 4, and an initial threshold accuracy of 30%. All 5 folds produce ensembles with high sensitivity and their intersection aligns with scripted on-road events. This supports our hypothesis that ADHD would measurably affect driving behaviors in curves and during collision avoidances.

of the spatial alignment domain to improve classification accuracy. Simultaneously, ISR isolated on-road events that adolescents driving with untreated forms of ADHD are likely to react to in a noticeably different manner (e.g. jerky braking and acceleration, over-steering on turns and curves) than their peers (Grethlein, Sladek, and Ontañón 2021). This is valuable information for researchers designing driving simulator routes to effectively screen for ADHD. By only including events that have been found to be useful for driver classification tasks, both time and resources can be saved in the driving simulator design and usage processes.

We intend to employ ISR in other spatially-motivated driver classification tasks, to further study the insights that ISR may be capable of generating.

References

Altche, F., and de La Fortelle, A. 2017. An lstm network for highway trajectory prediction. In 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 353–359. IEEE.

Cetin, M. S.; Mueen, A.; and Calhoun, V. D. 2015. Shapelet ensemble for multi-dimensional time series. In Proceedings of the 2015 SIAM International Conference on Data Mining, 307–315. SIAM.

Chakraborty, B., and Nakao, K. 2016. Automatic detection of driver’s awareness with cognitive task from driving behavior. In 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 003630–003633. IEEE.

Chandrasiri, N. P.; Nawa, K.; and Ishii, A. 2016. Driving skill classification in curve driving scenes using machine learning. Journal of Modern Transportation 24(3):196–206.

Chang, Z.; Lichtenstein, P; D’Onofrio, B. M.; Sjolander, A.; and Larsson, H. 2014. Serious transport accidents in adults with attention-deficit/hyperactivity disorder and the effect of medication: a population-based study. JAMA psychiatry 71(3):319–325.

Grethlein, D., and Ontañón, S. 2020. Spatially aligned clustering of driving simulator data. In The Thirty-Third International Flairs Conference.

Grethlein, D.; Sladek, A.; and Ontañón, S. 2021. Adhd prediction via time series ensemble fed driving simulator data. The International FLAIRS Conference Proceedings 34.

Hori, C.; Watanabe, S.; Hori, T.; Harsham, B. A.; Hershey, J.; Koji, Y.; Fujii, Y.; and Furumoto, Y. 2016. Driver confusion status detection using recurrent neural networks. In 2016 IEEE International Conference on Multimedia and Expo (ICME), 1–6. IEEE.

Iannaccone, R.; Hauser, T. U.; Ball, J.; Brandeis, D.; Walitza, S.; and Brem, S. 2015. Classifying adolescent attention-deficit/hyperactivity disorder (adhd) based on functional and structural imaging. European child & adolescent psychiatry 24(10):1279–1289.

IIHS. 2019. Fatality facts 2018: Teenagers. Insurance Institute for Highway Safety.

Keogh, E.; Chakrabarti, K.; Pazzani, M.; and Mehrotra, S. 2001. Dimensionality reduction for fast similarity search in large time series databases.

Kochanek, K.; Murphy, S.; Xu, J.; and Arias, E. 2019. Deaths: Final data for 2017 volume 68 number 9.

Lee, Y.-C.; McIntosh, C. W.; Winston, F.; Huang, P.; Ontañón, S.; and Gonzalez, A. 2018. Design of an experimental protocol to examine medication non-adherence among young drivers diagnosed with adhd: A driving simulator study. Contemporary clinical trials communications 11:149–155.

Lohrer, J., and Lienkamp, M. 2015. Building representative velocity profiles using fastdtw and spectral clustering. In 2015 14th International Conference on ITS Telecommunications (ITST), 45–49. IEEE.

McDonald, C. C.; Tanenbaum, J. B.; Lee, Y.-C.; Fisher, D. L.; Mayhew, D. R.; and Winston, F. K. 2012. Using crash data to develop simulator scenarios for assessing novice driver performance. Transportation research record 2321(1):73–78.

NCSA. 2019. 2018 fatal motor vehicle crashes: Overview, traffic safety facts research note. report no. dot hs 812 826.

Salvador, S., and Chan, P. 2007. Toward accurate dynamic time warping in linear time and space. The Internationals conference on Data Mining 2015 SIAM International Conference on Data Mining, 307–315. SIAM.

Verster, J. C., and Roth, T. 2011. Standard operation procedures for conducting the on-the-road driving test, and measurement of the standard deviation of lateral position (sdlp). International journal of general medicine 4:359.

Walshe, E. A.; Ward McIntosh, C.; Romer, D.; and Winston, F. K. 2017. Executive function capacities, negative driving behavior and crashes in young drivers. International journal of environmental research and public health 14(11):1314.