ADHD Prediction via Time Series Ensemble fed Driving Simulator Data

David Grethlein

Drexel University
djg329@drexel.edu

Aleksanteri Sladek

University of Pennsylvania
asladek@seas.upenn.edu

Santiago Ontañón

Drexel University
santi@cs.drexel.edu

Abstract

In this paper, we identify the on-road scenarios within a simulated driving environment where a group of clinical trial participants ($n = 30$) with and without *Attention Deficit Hyperactivity Disorder* (ADHD) drive perceivably different from one another. We partition the simulated routes into smaller non-overlapping sections in order to determine which sections elicit behaviors that are predictive of ADHD. Then, we develop section-specific classifiers, which are used as voters in bagging ensemble classifiers. Our results show gains in classifying ADHD (increase in 5-fold average evaluation accuracy) over our previous efforts, as well as providing explainable evidence that driving behaviors indicative of ADHD tend to be exhibited in turns and curves.

Introduction

Adolescent drivers are one of the most at-risk demographics for fatal *motor vehicle collisions* (MVCs) (IIHS 2019), and those with impairments to their executive functions, such as *Attention Deficit Hyperactivity Disorder* (ADHD), are a particularly vulnerable population (Walshe et al. 2017). Driving simulators are a source of diverse data that can be scrutinized for indicators of dangerous driving behaviors in a safe environment (Mei 2017; Grethlein et al. 2020).

In this paper, we classify $n = 30$ clinical trial participants (15 with and 15 without clinical diagnoses of ADHD) who were tasked with navigating planned routes within a simulated driving environment. This classification task is difficult, as drivers (all adolescents) with and without ADHD in this dataset are known to be hard to separate when their simulated driving records are examined unabridged (Grethlein and Ontañón 2020). We hypothesize that due to the sparseness of predictive features within the time series data previous classification efforts were hindered by the inadvertent inclusion of features that conflate the classes of drivers.

Hence, we present an approach that isolates predictive features by spatially partitioning these longer simulated routes into smaller non-overlapping sections. We then train time series classifiers on the data recorded in each section individually. Next, we build ensembles that use these classifiers to vote on an overall classification for each complete simulated driving record. Finally, we analyze the performance of these section-specific classifiers and

cross-reference them with known on-road scenarios scripted within each section (referred to as *events*) to conclude which ones best separated the classes of drivers.

Results from applying our approach in ADHD classification show gains in accuracy when compared to our previous efforts. We also provide evidence that curves and turns in the route tend to elicit driving behaviors indicative of ADHD.

Background

Detecting ADHD. ADHD is a difficult condition to diagnose in adults, as most symptoms tend to manifest in childhood and may be the result of other conditions when present in adulthood. Drivers with ADHD tend to be more aggressive behind the wheel and participate in risk-taking behaviors such as driving under the influence (Pollak et al. 2019). While there are driving behaviors associated with ADHD, to the best of our knowledge, there is no work dedicated to isolating on-road risk factors for those driving with ADHD.

Time Series Similarity. In this paper, we use a derivative algorithm of *Dynamic Time Warping* (DTW) (Sakoe and Chiba 1978), called *FastDTW* (Salvador and Chan 2007) that finds the near-minimizing alignment of frames between two time series. FastDTW aims to reduce the computation time for quantifying similarity between series of length n from $\mathcal{O}(n^2)$ for DTW to $\mathcal{O}(n)$. FastDTW has been previously used in related tasks (Lohrer and Lienkamp 2015).

Time Series Classification. We use an approach to time series classification based on *k-Nearest Neighbor* (*k*-NN) algorithms. *k*-NN using DTW to generate the similarity matrix is a well-documented time series classification algorithm (Elsahar et al. 2018; Yesilli, Khasawneh, and Otto 2019). We also evaluate *convolutional neural networks* (CNN), which are also regularly employed to classify time series data (Gao, Murphey, and Zhu 2018). However, to our knowledge, CNNs have not previously been leveraged in detecting ADHD from driving simulator time series data.

Driving Simulator Data

Data used in the experiments reported on in this paper were recorded in a clinical trial performed by the Children’s Hospital of Pennsylvania and George Mason University. The trial had a population of 30 adolescent participants (age 18-24), of which 15 had clinically diagnosed and treated

cases of ADHD. These participants drove 4 planned simulated routes (Drive 1, Drive 2, Drive 3, Drive 4) twice: once while taking their ADHD medication, given the label *regulated* (Reg); and once while taking a placebo, given the label *delayed* (Del). The remaining 15 participants did not have ADHD diagnoses and were instructed to drive the simulated routes only once, and given the label *control* (Cont). One participant experienced motion sickness and opted to stop recording, so we removed the incomplete drive (Del, Drive 3) from the dataset ($N = 179$ total records).

All simulator data were recorded at a frequency of 60 Hz, and were then down-sampled to 10 Hz for our experiments using the *Piecewise Aggregate Approximation* (PAA) technique (Keogh and Pazzani 2000). The PAAs varied in length 6-17 minutes (3,829-9,746 frames) per drive, with an average of 9.5 minutes (5,728 frames) in terms of duration.

Time Series Channels. The Realtime Technologies RDS-1000 driving simulator used to collect our data recorded the driver inputs using the physical controls (pedals, wheel, signals) and each participant’s interactions with other vehicles and traffic controllers. To avoid complex feature engineering that depends heavily on domain knowledge, we use a subset of commonly available channels to represent driver performance: throttle, brake, steering, forward velocity, forward jerk (derivative of acceleration), lateral lane position, and heading error (route versus vehicle heading).

Clipping by Section. We hypothesized that it is beneficial to decompose driving simulator time series data into sequences of sub-intervals. This is due to two main reasons: first, driving simulator time series can be very long, where only a few sub-intervals within a series are indicative of the class of driver. Second, by splitting time series into sub-intervals, we obtain a larger number of training instances to train a model with. Thus, we use a system of *sections* (see Figure 1), manually delineated segments of the road, to isolate driving behaviors aligned by exposure to the same scripted conditions. Doing so allows us to test our hypothesis that certain on-road conditions, specifically turns in the route, are more useful for separating the classes of drivers. We partitioned 179 simulator records into sub-intervals using the sections along the driven route in each case. The 23 defined sections yield a total of 981 individual sub-intervals.

Methods

For all experiments conducted, we performed a 3-way classification (Cont vs Del vs Reg) to determine where along each route the classes of drivers are most frequently conflated. This allows us to infer what effect the ADHD treatment, or its absence (placebo), has on impaired drivers.

Section-Specific Classifiers. In each experiment, we trained section-specific classifiers, referred to as *modules*, that assess driver performance strictly within their assigned section. Modules from all sections defined along a route are used as voters in an *ensemble* to classify unabridged driving simulator records. We constructed ensembles using 3 different types of modules: *Logistic Regression* (LogReg), *k Nearest Neighbors* (k-NN), and *1-Dimensional Convolutional Neural Networks* (1-D CNN). Each ensemble was

Figure 1: All 4 routes contain common on-road scenarios and are broken down into sequences of non-overlapping sections isolating distinct road features (curves, straight-aways). (a) *Drive 1* consists of 8 sections. (b) *Drive 2* consists of 4 sections. (c) *Drive 3* consists of 9 sections. (d) *Drive 4* consists of 2 sections.

built using strictly one type of module, to determine if one approach was better suited to the task at hand.

For the LogReg and k-NN ensembles, we constructed a FastDTW similarity matrix relating all sub-intervals recorded within a section to one another. These similarity matrices were constructed using further down-sampled time series (PAAs at 1 HZ) and a FastDTW radius of 25 frames to expedite computation. k-NN and LogReg modules use a similarity matrix directly, taking a vector of similarities between a given sub-interval to all other sub-intervals in the training data as an input feature vector. For k-NN we tested this approach using $k = 1$, $k = 3$, and $k = 5$.

1-D CNN modules learn directly from the time series sub-intervals extracted in each section. A 1-D CNN module is composed of two 1-D convolutional layers (32 filters, size 5), followed by an average pooling layer (pool size 2). The pooled output is flattened and passed through two *fully connected* (FC) layers of 64 then 32 units (with *ReLU* activations). The first FC layer used dropout (Srivastava et al. 2014) to reduce overfitting, and outputs were produced by a final 3 unit FC layer (with *softmax* activation). Hence, a module had 33,123 parameters. Each 1-D CNN module was trained for 100 epochs and fed mini-batches of 16 non-overlapping *windows*, roughly 3 second sub-intervals of 32 frames sampled at 10 Hz, at a time. Optimization was done with the ADAM optimizer (Kingma and Ba 2015), a learning rate of 10^{-5} , and categorical cross-entropy was used as the loss function. Classifications were formulated as a probability vector indicating which of the three classes an input window might belong to. Since a sub-interval is further divided into windows when training the 1-D CNN module, its class probability vector is the average of the probability vectors for all of its component windows. We tested 1-D CNN modules using 0%, 5%, 10%, 15%, and 20% dropout.

Best Choice Per Section Ensemble of Classifiers									
Route	Linear	k-NN			1-D CNN (% dropout)				
	LogReg	k=1	k=3	k=5	0%	5%	10%	15%	20%
Drive 1	46.7	44.4	37.8	44.4	26.7	35.6	28.9	37.8	37.8
Drive 2	31.1	40.0	37.8	37.8	44.4	42.2	44.4	35.6	46.7
Drive 3	52.8	57.2	45.6	36.1	38.9	36.4	36.7	58.9	32.2
Drive 4	35.6	26.7	22.2	33.3	35.6	35.6	28.9	33.3	28.9

Table 1: 5-fold average evaluation accuracy from the BCPS ensemble experiments. The results from the best performing ensembles for each route are highlighted in bold.

All Modules Per Section Ensemble of Classifiers									
Route	Linear	k-NN			1-D CNN (% dropout)				
	LogReg	k=1	k=3	k=5	0%	5%	10%	15%	20%
Drive 1	40.0	48.9	42.2	42.2	31.1	31.1	31.1	31.1	26.7
Drive 2	24.4	31.1	35.6	44.4	46.7	44.4	44.4	51.1	46.7
Drive 3	52.2	59.4	43.3	41.1	38.3	34.2	31.7	38.3	29.4
Drive 4	31.1	31.1	37.8	26.7	31.1	31.1	28.9	31.1	26.7

Table 2: 5-fold average evaluation accuracy from the AMPS ensemble experiments. The results from the best performing ensembles for each route are highlighted in bold.

Experimental Procedure. For each route and type of module used, we performed stratified 5-fold leave-one-out cross-validation, preserving the distribution of classes in each split (Cano, Herrera, and Lozano 2005). The dataset was also split by participant, so that no module would become biased towards learning the driving behaviors unique to a particular driver. Each fold also always had the same participants across experiments. For each ensemble, an *evaluation fold* of data is separated and reserved for later. Every module is trained on 3 folds of data, and an initial estimate of its performance is measured by computing the classification accuracy for a *development fold* of unseen data. We cycle the development fold 4 times, and as a result generate 4 modules that can assess driver performance in each section; the unseen 5th fold having been reserved for evaluation.

We implement 2 strategies in building an ensemble out of modules trained from all sections along a given route. The first, *best choice per section* (BCPS), uses the module with highest development accuracy from each section as a voter in the ensemble. The second, *all modules per section* (AMPS), uses all 4 trained modules from each section as voters in the ensemble (4 times as many votes as in BCPS). The voting system used was standard majority voting. We repeat the experiment 5 times, each time reserving a different evaluation fold for the ensemble to classify. This helps statistically verify results in case the folds of our small dataset are significantly different from one another.

Results

Herein we report the performance of all classifiers evaluated. Tables 1 and 2 show the 5-fold *average evaluation accuracy* (AEA) from the BCPS and AMPS ensemble experiments respectively. Rows of the tables are organized by the route driven and columns by the type of the modules used to construct ensembles. Our ensemble training process produced

Figure 2: Plots showing all sub-intervals of the 3 classes of drivers superimposed over an S-shaped curve found in *Drive 3 Section V*. Drivers enter the curve from the right. Each plot point shows vehicle position in a frame, with color indicating driver input (greener for throttle, redder for brake, bluer for steering, black for coasting). Each point’s size indicates the driver’s velocity in a frame (larger is faster).

intermediary results for each module. We analyzed these results to see if certain sections, and subsequently the scripted events in these sections, were more predictive than others.

Ensemble Results. This dataset has class prior probabilities of 33%; and the unabridged time series recorded by the different classes of drivers were shown to be non-separable with roughly 42% AEA achieved in a prior study (Grethlein and Ontañón 2020). Therefore any improvements over such baselines are significant, particularly if findings deepen our understanding of ADHD’s risk factors for young drivers.

Drive 1 produced few ensembles with sensitivity to the ADHD class labels. *Drive 2* produced several 1-D CNN ensembles with modest AEA, with *Section II* and *Section III* modules capable of separating *control* (Cont) sub-intervals from the rest. *Drive 3* ensembles yielded the highest AEA, reliably identifying Cont and *regulated* (Reg) drivers while struggling with *delayed* (Del) ones. None of the *Drive 4* ensembles were sensitive to class labels.

LogReg ensembles using the BCPS strategy outperformed their AMPS counter-parts. For k-NN ensembles, lower values of k yielded the most accurate ensembles in both AMPS and BCPS. 1-D CNN ensembles using 15% dropout tended to perform well for both strategies.

Predictive Sections. *Drive 1 Section V* produced modules with higher AEA across experiments; with LogReg modules exceeding 55% AEA and nearly all k -NN *Drive 1 Section V* modules surpassing 45% AEA. *Drive 3 Section V* (see Figure 2) consistently produced the most accurate modules of all sections considered. 5-NN modules in BCPS ensembles had an AEA of 61.7%, though the corresponding ensembles performed poorly with an AEA of 36.1%. *Drive 3* ensembles using AMPS and 1-NN modules had the highest AEA (59.4%); with *Section V* yielding 50.4%. *Section V* modules with AEA > 45% contributed to all but one of the most successful *Drive 3* ensembles (AEA > 50%).

Discussion

The section-specific intermediary results revealed which events best separated (modules with higher AEA) the classes of drivers. Within *Drive 1 Section V* we found that Del drivers braked prematurely in response to a collision avoidance, and then exhibited aggressive steering and over-correction through a left-hand turn. Similarly in *Drive 3 Section V*, Del drivers exhibited higher velocities entering the S-shaped curve, and over-corrective steering when exiting. These results indicate that curves in a simulated route are useful for detecting untreated forms of ADHD.

By using sequences of sub-intervals instead of unabridged time series as input, we increased the granularity of comparison when computing similarity. This provided LogReg and k-NN modules more precise time series similarities to leverage specific on-road events. We also chose to use 1-D CNNs to determine if we could directly learn time series features indicative of ADHD. While LogReg and k-NN ensembles were marginally more accurate in general, we posit that 1-D CNN may perform better given a larger dataset that could be prohibitive for constructing similarity matrices (quadratic time complexity). Additionally, having many modules vote on a final classification added robustness, as ensembles had more chances to evaluate driving behaviors.

Since our dataset is small, $n = 30$ participants (resulting in $N = 179$ records), our results must be considered anecdotal and would benefit from further validation on a larger dataset. Previous experiments using this data have found considerable variation in the driving behaviors within the classes of drivers, underscoring the difficulty in isolating the traits of each class. At the time of writing we were unaware of the existence of any larger driving simulator dataset recorded from participants with ADHD on which to validate.

Conclusions

In this paper we presented several methods for building ensembles to classify driving simulator time series data clipped by sections of the routes driven. The results presented herein show gains in accuracy over previous efforts to identify adolescent drivers with ADHD using driving simulator data (Grethlein and Ontañón 2020). Our approach accounts for the difficulties in classifying unabridged time series by using a spatial partitioning approach that aligns sub-intervals by exposure to common conditions. This allows for intervention designers to gauge the effectiveness of specific scripted events within a driving simulator environment at discriminating classes of drivers.

As part of our future work, we plan to consider dynamically partitioning routes into smaller sections iteratively. With more precisely defined sections we may further isolate predictive features and deepen our knowledge of on-road events that affect drivers with ADHD.

Acknowledgements: This project was partially supported by National Science Foundation Grant No. 1521943.

References

Cano, J. R.; Herrera, F.; and Lozano, M. 2005. Stratification for scaling up evolutionary prototype selection. *Pattern*

Recognition Letters 26(7):953–963.

Elsahar, Y.; Bouazza-Marouf, K.; Kerr, D.; Gaur, A.; Kaushik, V.; and Hu, S. 2018. Breathing pattern interpretation as an alternative and effective voice communication solution. *Biosensors* 8(2):48.

Gao, J.; Murphey, Y. L.; and Zhu, H. 2018. Multivariate time series prediction of lane changing behavior using deep neural network. *Applied Intelligence* 48(10):3523–3537.

Grethlein, D., and Ontañón, S. 2020. Spatially aligned clustering of driving simulator data. In *FLAIRS-33 Proceedings*. FLAIRS, AAAI.

Grethlein, D.; Winston, F. K.; Walhse, E.; Tanner, S.; Kendadaï, V.; and Ontañón, S. 2020. Prediction of licensing on-road exam outcomes from virtual driving test data: Novel method using time series clustering. *Journal of Medical Internet Research*.

IIHS. 2019. Fatality facts 2018: Teenagers. *IIHS*.

Keogh, E. J., and Pazzani, M. J. 2000. Scaling up dynamic time warping for datamining applications. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, 285–289. ACM.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In Bengio, Y., and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

Mei, T. T. 2017. Simulator training for learner drivers from 2019. *The Straits Times*.

Pollak, Y.; Dekkers, T. J.; Shoham, R.; and Huizenga, H. M. 2019. Risk-taking behavior in attention deficit/hyperactivity disorder (adhd): A review of potential underlying mechanisms and of interventions. *Current psychiatry reports* 21(5):33.

Sakoe, H., and Chiba, S. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26(1):43–49.

Salvador, S., and Chan, P. 2007. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis* 11(5):561–580.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(56):1929–1958.

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.

Yesilli, M. C.; Khasawneh, F. A.; and Otto, A. 2019. Chatter detection in turning using machine learning and similarity measures of time series via dynamic time warping. *arXiv preprint arXiv:1908.01678*.