Temporal Misalignment Challenges in Data Collection from Multiple Devices and Resolution

CIS 695

Neel Khakhar

Abstract

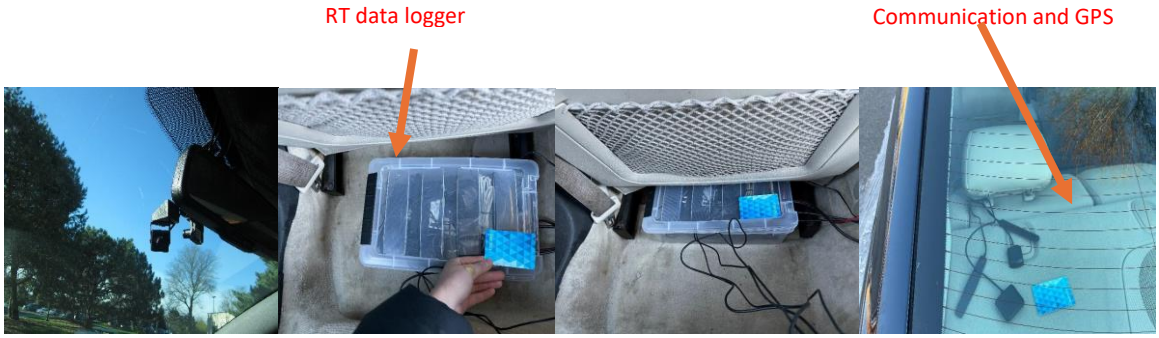
In today's increasingly interconnected world, the use of multiple devices for data collection has become commonplace. Especially in the field of medical sciences, wherein large amounts of data, be it physiological or behavioral, are collected from a large population to derive correlations and prove or disprove theories. These devices, such as smartphones, wearables, and IoT devices, gather vast amounts of data that can provide valuable insights for various industries [1]. However, one of the major challenges in data collection from multiple devices is the issue of misalignment. Misalignment refers to the discrepancies or inconsistencies that can occur in the data collected from different devices. When observations are available at different temporal scales across same spatial locations, we name it temporal misalignment [2]. In order to draw inferences for studies, it becomes necessary to align the time series of multi-modal input data. We confront this issue with perspective of temporal misalignment in data collection of physiological signals and behavioral patterns observed during Naturalistic and Fixed Course driving of amyloid positive and negative participants for National Institute of Health study on classification of Alzheimer's.

Introduction

The NIH R1 study aims at classifying amyloid positive and negative participants based on their behavioral and physiological signals during Naturalistic and Fixed course driving. Due to the advances in sensor, computer, information, and telecommunication technologies, now it is possible to obtain some automatically collected information about individuals' driving performance, habits, and some other related information [3,4]. These kinds of technologies, which can be attached to driver's vehicle, gives researchers the ability to investigate the driving behavior of individual drivers under naturalistic conditions for their everyday driving [5]. Each driving trip can involve hundreds of signals about the vehicle, roadway types, and the traffic density levels. In order to fully understand the content of these data, it is important to examine naturalistic driving data considering the context in which it occurs. An essential step in examination of these data points is the temporal alignment of data sources.

For the purpose of the study, we have included following Vehicle data acquisition system (VDAS) designed to collect multimodal sensor data during naturalistic driving for obtaining driver and trip data:

- A vehicle module (Vehicle-M) to record vehicle signals during real-time driving, which includes a position sensor. Vehicle-M included a commercially available vehicle data logging device, marked DL1 MK3 and is capable of recording many things related to the current trip (RACE, 2024). Time, latitude, longitude, speed, acceleration, GPS heading data are some examples that the logger device can provide.
- A video module (Video-M) that can record video stream data from two video cameras.



This VDAS system could be installed/ uninstalled from multiple participants for reuse. A dedicated researcher observes the consistency by monitoring device logs to the cloud. Once the study is complete, the data is extracted for further analysis.

Data Acquisition

As mentioned above, we collect data of participants from two feeds, video feed (from SafetyTrack video cameras) and RT data log (DL1 MK3 RaceTechnology). The data from both the devices is recorded on SD card. The SafetyTrack also publishes live trip data on cloud, which is used for daily monitoring of participant trips. Below are the data specifications of collected data:

- RT Data Logger: GPS reading at 5Hz and Wheel Speed at 100hz. Start of logging timestamp is recorded and data is extracted at 100Hz (without extrapolating GPS signal) wherein the time is shown in seconds (0.01, 0.02, 0.03.. seconds)
- SafetyTrack Video logger: Video start datetime is saved within the filename and video is extracted at 30 fps and 1280×720 resolution.

From collected data and manual inspection, we observe that *there is difference in expected speed of vehicle at different events*. Meaning, there is a presence of lag between the observed vehicle speed (from RT logger) and expected vehicle speed from front view video (video from this point, refers to front view video feed). This is the temporal misalignment or

data lag issue that we intend to tackle in this study. Examples of such events are mentioned below:

- Participant is stopped at a stoplight but RT logged data, at the same timestamp as video shows car moving at 30kmph.
- Participant has slowed down for the turn but RT logged speed shows no such change in the duration of the event.

However, we notice that if we account for the 'lag' in RT logging, we are able to fit the RT logged speed with expected speed from video completely. This suggests that the lag value we are looking for is consistent throughout the trip. A visual representation of the above mentioned issue is shown below:

For an event recorded on Video as well as RT logger

Ground Truth (Event Start/End on Video)
Observed Value (Event Start/End on RT Speed)

Here $t_{v1}\ and\ t_{v2}$ represent the start and end of the event on video feed. Similarly, $t_{r1}\ and\ t_{r2}$ represent the start and end of the event on RT logger. For initial observation purposes, we have chosen only vehicle stopping at signal or stop sign as events to observe lag. We also assume that vehicle RT logged speed <10kmph is considered a 'moving stop' for a stop sign (same as speed =0kmph)

Since our ground truth is video feed, we can set base true timeline as video feed timeline. Therefore, any deviation of speed logged from video feed, is considered as lag.

Hence we can define the lag as :

$$\delta = avg(t_{v1} - t_{r1}, t_{v2} - t_{r2} \dots)$$

From manual inspection the data collected shows the following temporal misalignment (lag) pattern. The data collected is from a Fixed course trip, meaning all below participants have driven through the same route:

Participant T1

Event	Actual Event Start (t_{v1})	RT event start (t_{r1})	Actual Event End (t_{v2})	RT event end (t_{r2})	Lag/ δ (s)
1	13:49:17	13:49:54	13:49:21	13:49:58	34
2	13:50:01	13:50:37	13:50:45	13:51:20	35
3	14:00:43	14:01:20	14:00:56	14:01:31	37

Participant T2

Event	Actual Event Start (t_{v1})	RT event start (t_{r1})	Actual Event End (t_{v2})	RT event end (t_{r2})	Lag/ δ (s)
1	10:48:14	10:48:50	10:50:06	10:50:42	36
2	10:59:04	10:59:42	10:59:46	11:00:22	38
3	11:02:50	11:03:27	11:03:00	11:03:37	37

Participant T3

Event	Actual Event Start (t_{v1})	RT event start (t_{r1})	Actual Event End (t_{v2})	RT event end (t_{r2})	Lag/ δ (s)
1	11:25:22	11:25:40	11:26:58	11:26:54	18
2	11:30:10	11:30:32	11:30:14	11:30:35	22

Methodology

We have at our hands RT logged data and video feed from front view camera, wherein we have observed discrepancy in expected speed readings at same point in space and different points in time. Our approach to this problem is to observe and infer information from ground truth. That is, we estimate speed from the video feed to determine our temporal misalignment with the RT logged speed. Our goal is to find δ with the assumption that lag (δ) is constant throughout the trip.

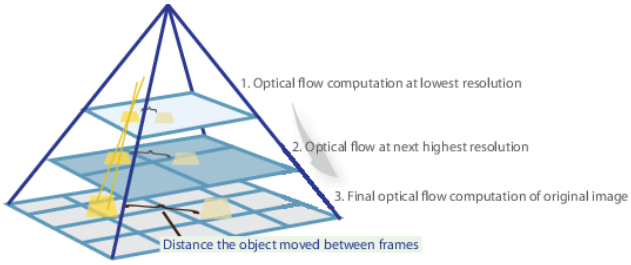
The predicted speed from video is shifted by x seconds (range 0 to 60) to minimize the covariance between the two speeds. Hence,

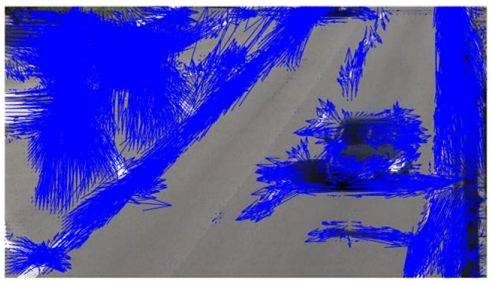
$$\delta = min \int_0^{60} var(pred_{speed}, rt_{speed})$$

This follows the heuristics of Dynamic Time Warping technique. The DTW algorithm allows two time-dependent sequences that are similar, but locally out of phase, to align in time. Its main objective consist of identifying an optimal alignment between sequences by warping the time axis iteratively [6].

To get speed predictions from video, we utilize car speed detection module in python which uses Optical Farneback Flow for feature extraction from subsequent frames, coupled with artificial neural network to estimate speed of vehicle from frames. It requires supervised learning data to generate a machine learning model which can estimate speed, given video. The idea of using Optical flow *is to approximate some neighborhood of each pixel with a polynomial expansion[7].* The effectiveness behind Farneback flow lies in extraction of pixel vector changes in subsequent images, at different levels of resolution. We set the Farneback hyperparameters as follows, i) Levels = 3 means total of 3 resolution level of calculation of optical flow polynomial; ii) pyr_scale=0.5 means a classical pyramid, where each next layer is twice smaller than the previous one; iii) Winsize = 3 ; averaging window size (functions like

'filter' in CNN); iv) Poly_n = 7 max poly deg to fit (typically 5/7); v) Poly_sigma = 1.2; std of bias in poly (typically 1-1.5)

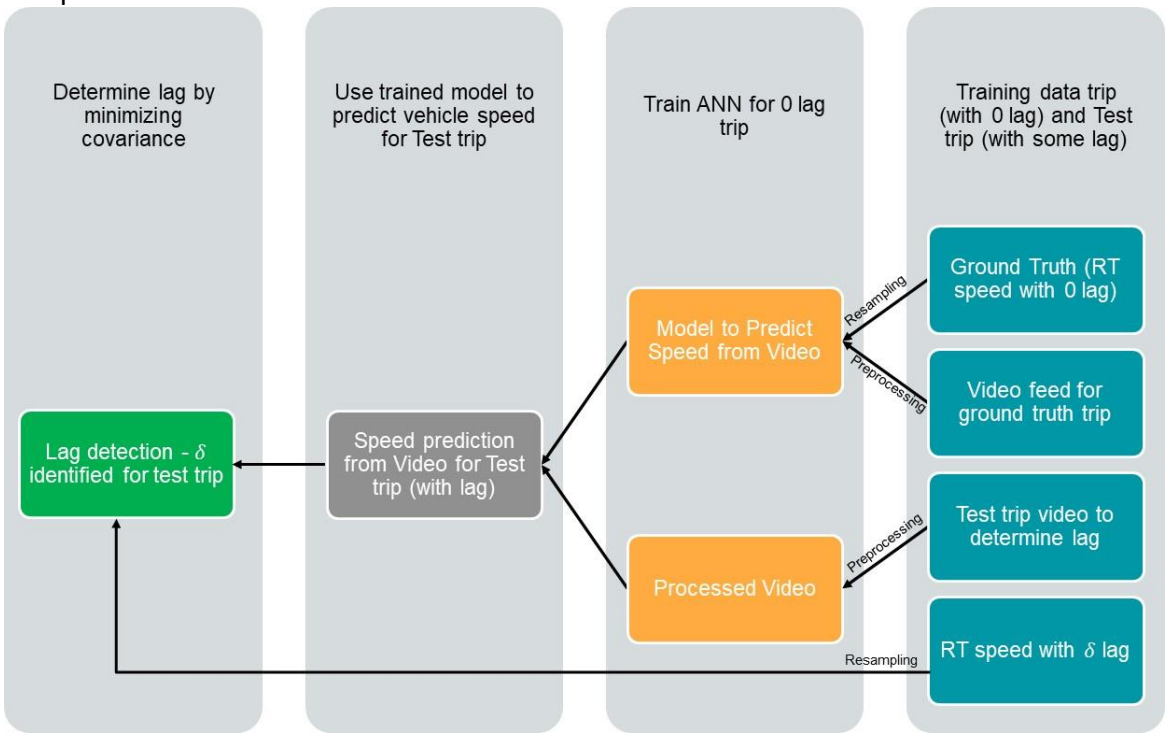




Farneback flow pyramid and sample of vector extraction [7]

Using optical flow we intend to extract features from video frames which in turn will be utilized by our ANN model to estimate the speed of vehicle. This estimate of speed of vehicle will help us 'align' our RT logged speed to correct timestamp.

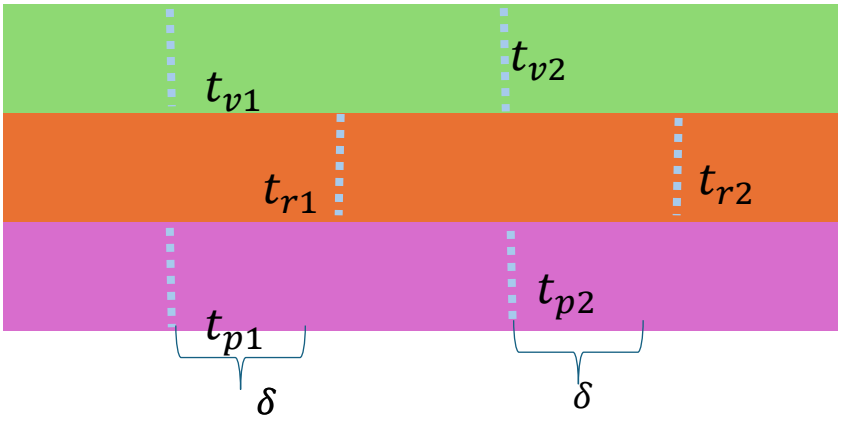
The process flow is as below:



Ground Truth (Event Start/End on Video)

Observed Value (Event Start/End on RT Speed)

Predicted speed from front cam



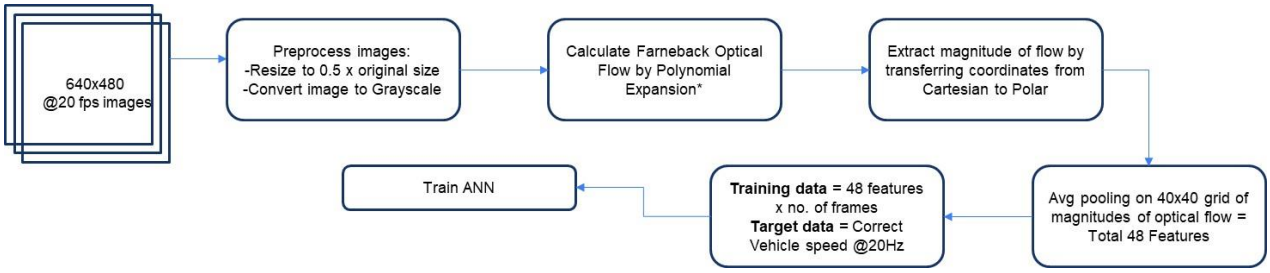
Data Preprocessing

To calculate optical farneback flow from images, we convert the video to 640x480 resolution, 20 fps video. Similarly, for our supervised ANN ground truth, we consider a participant with the same Fixed course route trip RT logged data with zero lag. We utilize this temporally aligned RT speed data and front view video data feed to train our model. We also downsample our RT logged speed at 100Hz to 20Hz to match fps of video.

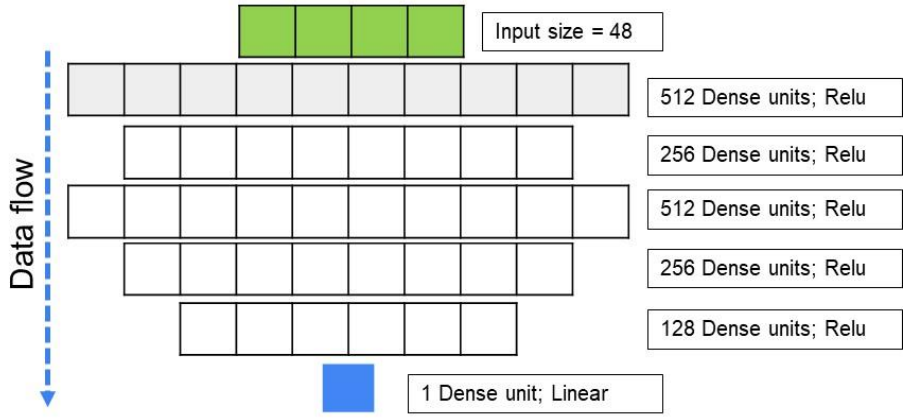
For cold starting issue, i.e difference in start times of RT logger and SafetyTrack video recorder device, we observe that the video recorder usually starts before the RT and that initial few minutes are spent within the car (speed = 0kmph). Hence, we fill our RT speed data with zeros to match the video length. This aligns our start times of both devices to the same timestamp.

For our training data, we create two models to study the impact of different roads and size of training dataset, on accuracy of lag determined. We use Fixed course trip data of length ~16 mins and for larger model we combine this trip with ~150 mins of trip. The speed varies in Fixed course trip from 0-100kmph, whereas speed range for longer trip is 0-150kmph.

Feature extraction from video follows the below routine:



Finally, the extracted features are passed through below model, along with RT logged speed, resampled with mean at 20Hz as the ground truth (with 0 temporal misalignment). The training is ran for 100 epochs or until the loss <1e-6, with a learning rate of 0.001, Adam optimizer and MSE loss function.



Results and Discussion

Model trained on shorter Fixed course trip front view video feed and resampled RT logged data gives a training RMSE of 4.47 kmph. The baseline set by comma.ai gives an RMSE on training at 2.64 kmph, with the same model architecture and similar hyperparameters, with a trip length of ~34 mins.

Model trained on longer trip, with merged Fixed course and naturalistic drive, with combined length of ~186 mins, gives a training RMSE of 2.8 kmph.

Using the predicted speed time series and resampled RT logged data timeseries of speeds, we calculate the minimum 'shift' we need to introduce in RT logged data series to minimize the covariance between the two series. We utilize the similarity in variability of predicted speed and RT speed to find a lag value, which minimizes the covariance. An illustration is shown below:

RT speed	1	2	3	4	5	6	7		Covariance
Predicted speed	0	0	1.5	2.5	3.5	4.5	5.5	6	7	3.929
RT speed shifted 2 s	→		1	2	3	4	5	6	7	Covariance
Predicted speed	0	0	1.5	2.5	3.5	4.5	5.5	6	7	3.643

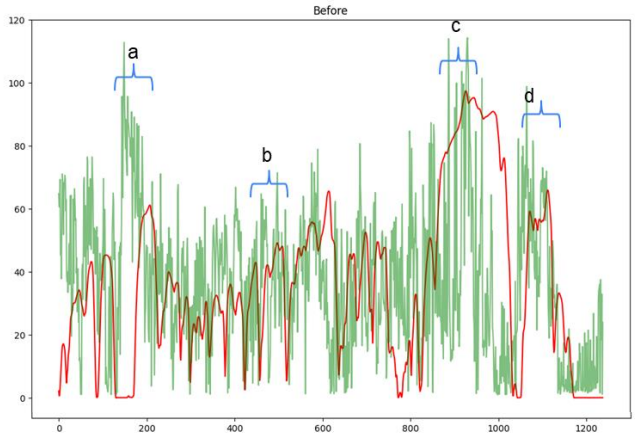
Applying this methodology to our test cases. We observe that inspite of speed prediction from front view video cam being highly deviated from actual noted, but lagging RT logged speed, it is able to preserve the variability of stops, lows and high speeds.

Below plots show predicted speed from front view video (green) over RT speed (red). a,b,c,d points show the temporal misalignment in predicted vehicle speed from video and RT speed. Ideally, the highs, lows and stops of both speeds should align along the same timestamps.

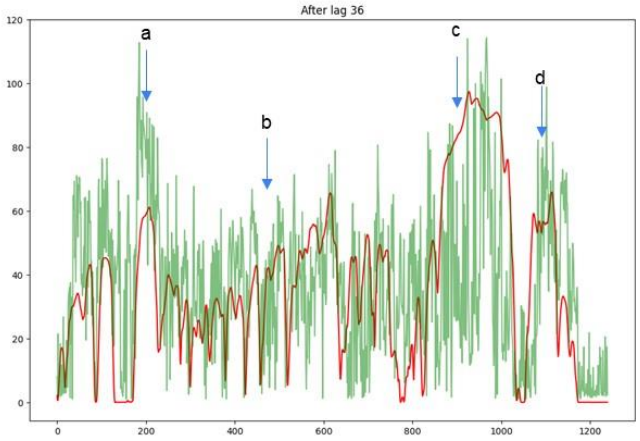
For below plots, we utilize the model with Fixed course trip (~16 min) as training with zero temporal lag resampled RT speed for the same trip.

Predicted Speed	
RT logged Speed	

Participant T1 before introducing lag, shows covariance between speeds is 1377:

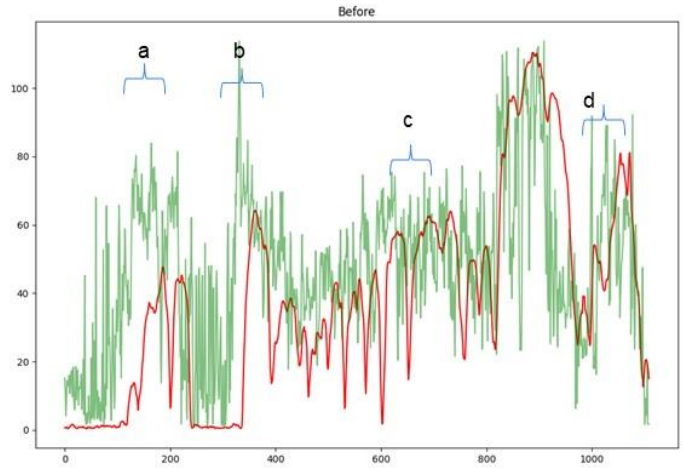


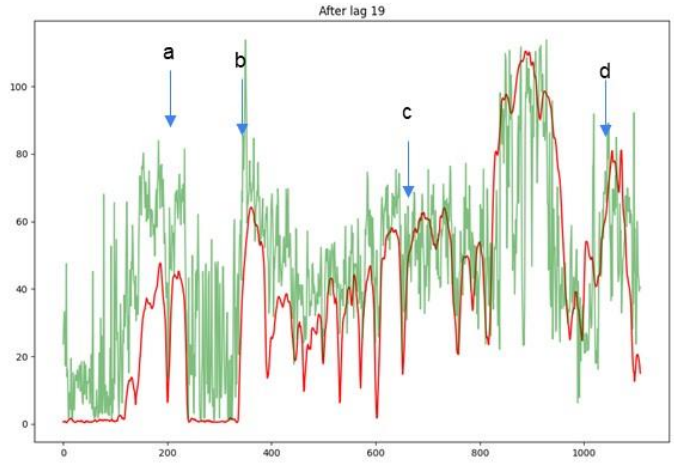
Participant T1 after introducing lag (of 36 seconds determined by minimization of covariance), shows covariance between speeds is 966:



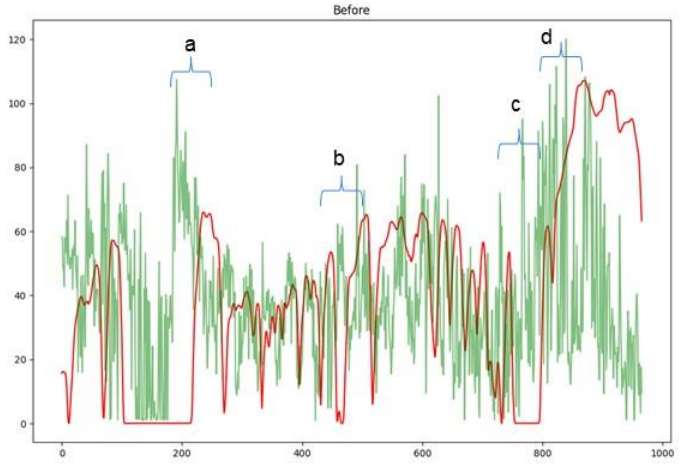
Similarly before and after plots for Participant T2 and T3 test cases.

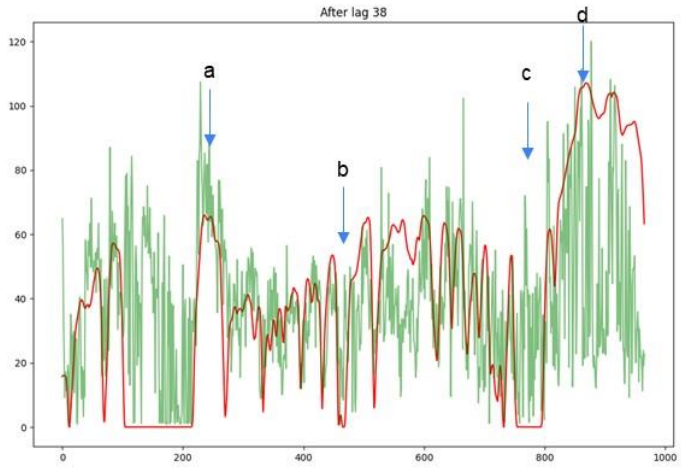
Participant T2, before and after introducing lag of 19:



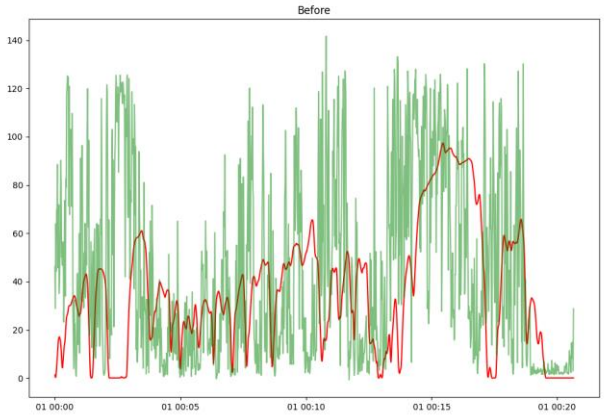


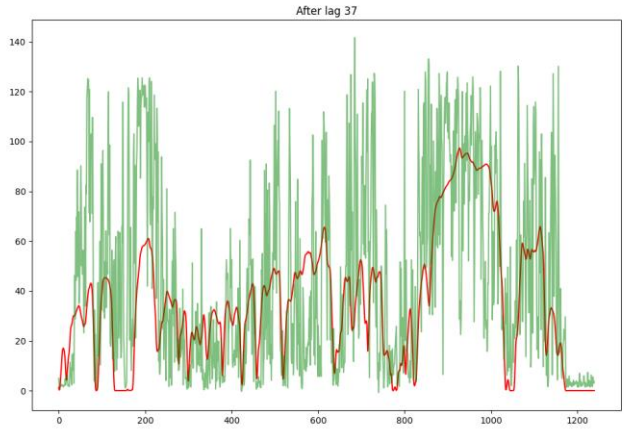
Participant T3, before and after introducing lag of 38:





We also create similar plot for Participant T1 using larger model, with training trip length ~186 mins. Participant T1, before and after applying lag by minimizing covariance:





We see high amounts of variability in predicted speed with larger model. This implies following conclusions:

- Training data trips were on very high speeds (~150 kmph) while test trip was fixed course and did not have much speed
- Separate normalization of test and training data can help. Currently no normalization is applied on ground truth speed data

We justify the variability of predicted speed for larger training data model with above pointers. It is also worth noting that no manual cropping or region selection was applied on initial set of images. This makes it difficult for ANN model to identify the exact Farneback optical flow features which indicate towards correct speed prediction.

Future work can include other methods of identifying lag and reusing accurate predicted segments to augment the pretrained model. Other methods include the Longest Common Subsequence (LCSS) distance for time series. It is derived from a solution to the problem of finding the longest common subsequence between two discrete series through edit operations. For example, if two discrete series are abaacb and bcacab, the LCSS is baab. Unlike DTW, the LCSS does not provide a path from (1, 1) to (m, m). Instead, it describes edit operations to form a final sequence, and these operations are given a certain cost. So, for example, to edit abaacb into the LCSS baab requires two deletion operations [8]. We can also employ new distance minimization which is called Dynamic Time Warping Delta (DTW-D) and is the ratio between DTW and Euclidean distances. While there is also a surfeit of possible distance measures for time series, Dynamic Time Warping (DTW), a technique from the dawn of computing, is exceptionally difficult to beat [9]

References

[1] Sarker, I H., Hoque, M M., Uddin, K., & Alsanoosy, T. (2020, September 14). Mobile Data Science and Intelligent Apps: Concepts, AI-Based Modeling and Research Directions. Springer Science+Business Media. https://doi.org/10.1007/s11036-020-01650-z

[2] Zapata-Marin, S., Schmidt, A. M., Weichenthal, S., & Lavigne, E. (2023). Modeling temporally misaligned data across space: The case of total pollen concentration in Toronto. Environmetrics, 34(8), e2820. https://doi.org/10.1002/env.2820

[3] Kerautret, L., Dabic, S., & Navarro, J. (2023). Exploration of driver stress when resuming control from highly automated driving in an emergency situation. Transportation research part F: traffic psychology and behaviour, 93, 222-234.

[4] Gong, Z., Yang, X., Song, R., Han, X., Ren, C., Shi, H., ... & Li, W. (2024). Heart Rate Estimation in Driver Monitoring System Using Quality-guided Spectrum Peak Screening. IEEE Transactions on Instrumentation and Measurement.

[5] Collet, C., & Musicant, O. (2019). Associating vehicles automation with drivers functional state assessment systems: A challenge for road safety in the future. Frontiers in human neuroscience, 13, 131.

[6] Folgado, Duarte & Barandas, Marilia & Matias, Ricardo & Martins, Rodrigo & Carvalho, Miguel & Gamboa, Hugo. (2018). Time Alignment Measurement for Time Series. Pattern Recognition. 81. 268-279. 10.1016/j.patcog.2018.04.003.

[7] Farneback, G. "Two-Frame Motion Estimation Based on Polynomial Expansion." In Proceedings of the 13th Scandinavian Conference on Image Analysis, 363 - 370. Halmstad, Sweden: SCIA, 2003

[8] Holder, C., Middlehurst, M. & Bagnall, A. A review and evaluation of elastic distance functions for time series clustering. Knowl Inf Syst 66, 765–809 (2024). https://doi.org/10.1007/s10115-023-01952-0

[9] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo E.A.P.A Batista. 2013. DTW-D: time series semi-supervised learning from a single example. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '13). Association for Computing Machinery, New York, NY, USA, 383–391. https://doi.org/10.1145/2487575.2487633