Driver behaviour profiling based on trajectory analytics

Harris Georgiou*
University of Piraeus
Piraeus, Greece
hgeorgiou@unipi.gr

Nikos Pelekis*
University of Piraeus
Piraeus, Greece
npelekis@unipi.gr

Yannis Theodoridis*
University of Piraeus
Piraeus, Greece
ytheod@unipi.gr

ABSTRACT

Driver behaviour profiling, specifically in relation to identifying ‘good’ versus ‘bad’ driving patterns, is one of the most challenging problems in mobility data analytics. In this paper, the core task of driver behaviour profiling is addressed at the minimum level of pre-requisites, i.e., GPS-only trajectory data (no accelerometer or other sensors) of very low sampling rate (less than 0.1 Hz). A dynamic temporal resampling algorithm is employed for transforming GPS data into three distinct location-invariant time series, namely speed, acceleration, and turn rate, after map-matching and noise elimination pre-processing steps. A wide range of statistical, time series and spectral methods are implemented as feature functions or ‘encoders’ of various aspects of short-term mobility tracking. In our experimental study, a large real-world trajectory dataset is processed and transformed into such a feature-vector dataset, which is subsequently used in unsupervised training and adaptive category identification for the various driving behaviour ‘states’. The proposed approach is designed for online/streaming mode and lightweight yet powerful analytics. The results show that such an approach is feasible, despite its challenging context of constraints, providing a data-driven adaptive way to recognizing ‘normal’ vs. ‘abnormal’ driving patterns on-the-fly.

CCS CONCEPTS

• **Computing methodologies → Feature selection**; **Cluster analysis**; **Anomaly detection**; • **Information systems → Data analytics**.

KEYWORDS

trajectory analytics, driver behaviour, mobility patterns, anomaly detection

1 INTRODUCTION

Trajectory analytics is one of the most commonly addressed tasks in the general context of geolocation data mining, usually involving mobility patterns, mobility graphs, points of interest, hotspot detection, etc. A special topic that has been advancing steadily over the past few years is analysing the driving patterns and mobility dynamics as the driver ‘behaviour’, in the long- and in the short-term [17, 19]. In the case of long-term analytics, global trends and aggregated models can be discovered for regional and large-set statistics regarding driving habits, location- and route-specific risks of accidents, fuel consumption, delays due to traffic jams, points of interest (POI), etc. In the case of short-term, which today is the cornerstone in developing fully-autonomous driving vehicles [20], ‘spot’ analytics of the driving patterns within a limited time frame, usually no more than few minutes at most, provide hints about erratic driving, ‘unpredictable’ or risky movements, instantaneous violations of speed limits, etc. Both these cases are very useful and challenging research problems, but their context, data modalities used and inherent methodological approaches are distinct and very different.

While the long-term approach in Driving Behaviour Profiling (DBP) has been explored using location-only data, e.g. from single GPS sensors, the short-term approach is inherently more demanding in terms of spatio-temporal resolution, data quality and additional sensing modalities. In practice, tracking the movement of a single car or driver for an entire month to extract commonly used routes, visited POIs or risk of car crash within this context is inherently more straight-forward and well-studied than having to analyse movement patterns in the context of few minutes or seconds to distinguish between ‘good’ and ‘bad’ driving. The short-term case, being more challenging, is normally approached by employing multi-modal, high-resolution sensing, e.g. location tracking together with accelerometer measurements, while at the same time having pre-determined training routes and confirmed driver ‘events’ as ground truth for model training [9, 21, 31]. However, these pre-requisites cannot always be satisfied, as multiple sensing and/or ground truth may not be unavailable, sampling rates may be too low, etc.

In general, short-term DBP is based on one or more of the following assumptions about the problem setup: (a) multi-modal sensing, typically location plus accelerometer, driver- or environment-sensing apparatus, etc; (b) high-resolution data especially in the temporal dimension, typically many samples per second; (c) specific annotation of ‘good’ and ‘bad’ driving patterns, either with some pre-determined set of driving ‘events’ that are introduced during test runs or by the labelling of training samples by a human expert [31]. In this paper, the most challenging problem setup for the short-term DBP task is treated, i.e., when none of the previous assumptions is satisfied: neither (a) multi-modal sensing or (b) high-resolution data or (c) ground truth are available. This essentially translates into designing unsupervised predictive models for DBP [1, 33] in the short-term context when having location-only, sparse, variable-rate, unlabelled data. Additionally, in this work the methodological approach and the proposed solution is developed in a way that leads to lightweight and on-the-fly processing, in order to be able to implement it as online/streaming service, which is essentially the true importance and value of having short-term DBP.

In summary, the novelties of this work in the DBP topic are the following:

- Fully unsupervised, data-driven predictive models for DBP.
- Use of sparse, variable-rate, GPS-only location data as input.
- Online map-matching of the raw input to the road network & robust noise filtering.

- Dynamic temporal resampling method for high-quality fixed-rate upsampling.
- Treatment of three different data series: speed, acceleration, turn rate.
- Extensive study on feature functions as ‘encoders’ of DBP patterns.

The rest of the paper is organized as follows: In Section 2 the short-term DBP, referred to simply as DBP from here on, is clearly defined in term of the modalities available, the definition of ‘good’ and ‘bad’ driving and the limitations posed by the current approaches; in Section 3 the complete methodology of the proposed approach is described in detail, addressing each one of the individual challenges presented above; in Section 4 the datasets, experimental protocol and results are presented; in Section 5 the methodology is discussed in view of the presented results; finally, in Section 6 some conclusions are drawn for the proposed approach and its applicability to real-world DBP setups.

2 PROBLEM DESCRIPTION

Before the DBP problem is explored in detail in various aspects and limitations, a more formal definition of the context is required in relation to ‘good’ and ‘bad’ driving. Although there is no universal definition of the DBP problem, the most generic aspect that defines what is the core value at stake is *safety*, translated as not being causally involved in car accidents, i.e., not suffering from or causing them to others [7, 12, 25].

2.1 Defining the problem

In general, there are two sets of specifications or constraints that dictate if a driving behaviour is *safe* or not: (a) ‘hard’ limits that need to be strictly satisfied and (b) ‘soft’ restrictions that indicate some strong preference. In practice, (a) are regulations defined by laws and, thus, are almost always quantifiable and in some way inferred directly from data measurements, as for example *overspeeding* is a direct violation of the speed limit in some road. On the other hand, (b) can be an informal or qualitative description of safe driving for a single car and the others around it, as for example avoiding *cornering* (harsh turns or lateral movements), *harsh accelerating* or *harsh braking*, etc [5, 8, 26, 30]. The first set of restrictions are typically well-defined and easy to encode into DBP, however the second set is not; even if these driving *events* of interest are formalized and somehow extracted automatically from the data, a behavioural norm has to also be defined in order for them to be compared to some safe driving baseline [34, 35].

In view of the relevant literature and the current state-of-the-art, in this paper the DBP is treated in the context of two specific factors for road safety:

(1) *Speed limits*: Driving patterns include checks against overspeeding conditions; these are hard limits that are available locally for each road, according to official regulations.
(2) *Path predictability*: Driving patterns are associated to road safety at a lower or higher degree according to how predictable the trajectory of the car is; in other words, the more predictable a car’s path is, the safer its driving profile is for everyone (anticipate and avoid the risk of accidents).

It is clear that, in the context of safety as the primary criterion, both these factors need to be treated in the short-term or ‘spot’ estimations (minutes or seconds), rather than long-term or aggregated ones (over weeks or months). Additionally, the creation of a well-defined and reliable set of some safe driving baseline to be used as ground truth, the normal approach of careful planning and execution of experimental measurements is usually hard to achieve in real-world driving conditions, as ‘bad’ driving in this sense would result in risk of causing real accidents. Thus, it is very hard to precisely plan and execute data generation experiments for DBP, as it is inherently hard to perform for real with the intention to record artificial driving ‘violations’.

It should be noted that other criteria for ‘good’ and ‘bad’ driving may also be applied, including economic factors, environmental impact, time schedule, etc. However, safety is typically the single most important factor and the highest priority when viewing the DBP task in the short-term, e.g. when designing systems for fully autonomous driving [3, 20].

2.2 Data availability and modalities

Regarding data availability, DBP can be categorized according to the sensing modalities that are available for use, according to the triplet of context choices [15]:

- *single- or multi-vehicle*: Sensing data by/for individual cars, e.g. location or acceleration [27], versus being able to correlate or simultaneously track multiple cars close by, e.g. cars inside a buffer zone around it [13, 23].
- *without or with driver tracking*: Vehicle data may be supplemented with sensors that are tracking actual driver attributes, e.g. attention drift (eyes), sleepiness (steering wheel), etc [2, 24].
- *without or with environment tracking*: Vehicle data may be supplemented with sensors that are tracking external factors other than neighbouring cars, e.g. road lines/edges/signs, obstacles, etc [32].

In real-world DBP applications, there are several limitations that may arise in relation to one or more of the aspects described above. The most common one is the lack of additional modalities other than location tracking and (maybe) accelerometer measurements, as these are readily available in of-the-shelf portable devices like typical smartphones [4, 6, 11, 14, 21]. In contrast, any of the other options usually require special devices installed inside the car (e.g. tracking cameras), around the car externally (e.g. proximity sensors, LiDAR), in combination with the other cars in traffic (e.g. inter-vehicle networking) or in combination with environmental guides (e.g. UV painting on road edges/signs). For obvious reasons, the cheapest and most preferable DBP solution would require location-only data, perhaps accompanied with acceleration measurements from sensors, if available.

2.3 Novelties of the proposed approach

As previously described, the context of DBP can be very restrictive in terms of data availability and quality, sensor modalities employed and the existence of a reliable baseline to be used as ground truth for the models. Moreover, the type and complexity of the required processing can be prohibitive for on-the-fly DBP models that need to work with new data as they are generated, instead of processing them offline in batches with little or no processing time restrictions.

This paper presents a new approach to DBP in the short-term context and with on-the-fly processing in mind. More specifically, the main focus and contributions in this work are the following:

- Data-driven, purely unsupervised model training, without any labelled ground truth available.
- Dynamic temporal resampling method for high-quality fixed-rate upsampling.
- Application of high-quality map-matching to the underlying road network and robust noise filtering (pre-/post-processing).
- Use of sparse (< 0.1 Hz), GPS-only location data of variable sampling rates for the single-vehicle DBP task.
- Generation of high-quality multi-modal time series from the GPS data (speed, acceleration, turn rate).
- Instead of simple thresholds, in-depth analysis of the data series with optimally selected higher-order ('texture'), curve and spectral features.
- Association with external data enrichments, e.g. weather and road/vehicle types, as additional DBP features.
- Employment of multi-stage clustering as 'blind' DBP state tracking, i.e., driving 'categories' that are discovered naturally from the data.
- Employment low-complexity, on-the-fly processing, to enable DBP applications for online/streaming modes.

This DBP description is essentially addressing the problem at the minimum level of pre-requisites, i.e., GPS-only trajectory data (no accelerometer or other sensors) of very low and varying sampling rate (less than 1 sample per 10 seconds). Before using the input, the raw GPS location data are map-matched to the underlying road network and noise-filtered for removal of artifacts, thus providing high-quality estimations about the actual route and distance travelled within each temporal step in the road network, e.g. calculate speed using the actual network distance instead of the Euclidean norm. Additionally, the map-matched location data are supplemented with context-related enrichments such as the road speed limits. This is achieved by a dynamic temporal resampling method that is employed for transforming the sparse GPS-only trajectory data into three distinct, optimally upsampled to a fixed-rate and location-invariant time series, namely speed, acceleration and turn rate. Additionally, the feature functions are optimally selected for analysing these reconstructed data series as content-rich 'encoders' of DBP patterns but yet lightweight enough to be applicable to on-the-fly processing architectures.

Given the data restrictions and the challenging setup of the DBP problem here, only few works from the current best-practices in DBP are comparable with this proposed approach [5, 17, 19, 25, 34, 35]. The most 'compatible' work in terms of unsupervised DBP categorization via clustering context-sensitive (per road segment) speed and acceleration descriptive statistics is [33]; this method is also implemented and included in the experimental work for providing comparative results in the same dataset, as described in Section 4.

3 MATERIAL AND METHODS

The overall 'pipeline' view of the proposed approach can be summarized in the following sequence of phases:

(1) Map-matching & filtering of the raw GPS data.
(2) Dynamic Temporal Resampling Buffer (DTRB).
(3) Feature extraction for DBP via trajectory analytics.
(4) DBP evaluation based on unsupervised models (clustering).

The following sections describe the methods developed and applied in each phase.

3.1 Road matching and filtering

As described earlier, instead of using the raw GPS data, the location points are map-matched against the underlying road network for removing GPS uncertainty and some of the noise. In practice, the distances of each point from the nearest road segments are estimated geometrically using the Haversine function (spherical approximation) and they are used as input to an online map-matching module that is based on Hidden Markov Model [10]. This enables the correction of GPS errors not for single points but for entire sequences along the 'most probable' path in the maximum-likelihood sense. Furthermore, in this work the core HMM-based map-matching process has been augmented with an additional step of localized pre-fetching and thresholding of the underlying OSM network, in order to speed up the process and avoid singular road matches at excessive distances (drop the point instead as noise).

Given the map-matched GPS trajectory, with some outlier points already removed as noise, the 'most probable' path is examined for any 'spot' violations against a set of predefined thresholds relevant to the expected values for distance versus time step. Validity checks can be asserted as additional post-processing for realistic maximum speed, acceleration, braking and turn rates, hence any location points resulting in such violations are also removed as GPS noise. The end result from this entire process is the maximum-likelihood 'corrected' trajectory of the low-resolution GPS track, which is subsequently used as input for the next steps of the proposed method. Figure 1 illustrates a close-view comparison of the raw GPS location data (in red) and the map-matched & noise-filtered trajectory (in blue).

3.2 Dynamic Temporal Resampling Buffer (DTRB)

The constraint of having sparse GPS-only location data and no other modality available is one of the most demanding challenges addressed in this work. The reason is that DBP in the short-term context requires high-resolution movement analytics, i.e., detection of over-speed at one moment not on averaged values, 'spikes' in the acceleration or turn rate, etc. Sparse GPS location data do not provide such information, thus it must be inferred the best way possible by other means.

The core idea of the DTRB is that the map-matched & filtered low-resolution GPS track is analysed for detecting sequences where the sampling rate is adequately high, even for short periods of time or data 'slices'. A much higher and fixed sampling rate is applied and the data series is upsampled with a high-accuracy algorithm, namely shape-preserving cubic spline interpolation. For each such

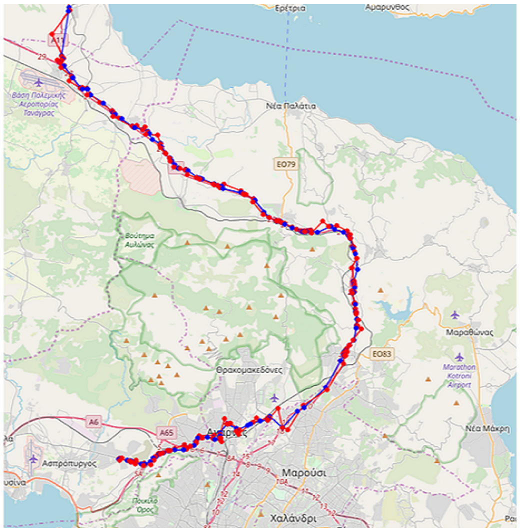

Figure 1: Example of raw GPS data map-matching & filtering from the dataset used.

slice of sparse GPS location data, the most recent part (temporally) of its upsampled transformation is used as the basis for producing the three main data series used here for DBP, i.e., speed, acceleration and turn rate. These are used as input to the feature extractors in the next step, which essentially detect, encode and quantify the properties that are relevant to the DBP task.

The 'pipeline' outline of the DTRB algorithm can be described as follows:

- Continuously scan the incoming location data for 'dense' slices.
- When a valid slice is detected, upsample to a fixed rate.
- From the upsampled location data, generate speed, acceleration, turn rate series.
- Perform another set of validity checks for the generated data series (filtering).
- Forward the processed (3x) data series for feature vector generation.

Taking all the design constraints into account, as well as the need to continuously process the input on-the-fly as new GPS location data arrive, the DTRB algorithm satisfies the following requirements, re-checked upon every new input:

(1) *Input*: Wait for new GPS location points (or read next from offline file).
(2) *Spatial span* (check): Current slice contains at least N_s^{min} GPS location points.
(3) *Temporal span* (check): Current slice spans at least L_s^{lim} sec.
(4) *Temporal inter-distances* (check): Not larger than L_s^{max} and not smaller than L_s^{min}.
(5) *Detect gaps*: Whenever L_s^{max} is violated, gap is detected and the sequence buffer is flushed, keeping only the current location point.
(6) *Short slices*: If L_s^{min} is satisfied **and** $N_s = N_s^{max} > N_s^{min}$ location points are available, consider the slice as valid even when its total time span $L_s < L_s^{lim}$.
(7) *Density criterion*: As soon as new input results in $N_s \geq N_s^{min}$ **and** $L_s \geq L_s^{lim}$ **and** L_s^{min}, L_s^{max} are satisfied, the slice is marked as valid and is forwarded for further processing; otherwise return and continue from (1).
(8) *Upsampling*: For every valid slice detected, produce speed U_t, acceleration A_t and turn rate R_t data series from the GPS location points, upsampled at fixed rate T_s.
(9) *Validation*: For each of the three new data series, perform a set of additional range checks[1]; discard the slice if any check is invalidated and return to (1).
(10) *Output*: If all checks validate ok, use the most recent $n \cdot T_s$ part of the upsampled data series U_t, A_t, R_t for DBP feature vector generation.

The DTRB configuration parameters can be tuned according to the specific dataset at hand. Considering all these constraints and after extensive experimentation with the DTRB configuration, the nominal process is defined as having a slice of least $N_s^{min} = 4$ location data points within a span of $L_s^{lim} = 32$ sec, being at least $L_s^{min} = 1$ and no more than $L_s^{max} = 32$ sec apart. If $N_s = N_s^{max} = 6$ location points and L_s^{min} is satisfied, then the slice is considered as valid regardless of L_s. This means that the total temporal extent of the slice may be from $N_s^{max} \cdot L_s^{min} = 6 \cdot 1 = 6$ up to $N_s^{min} \cdot L_s^{max} = 4 \cdot 32 = 128$ sec. The upsampling is implemented with using shape-preserving cubic spline interpolation in the entire slice, the fixed rate is set to $T_s = 1$ sec and the most recent $n \cdot T_s = 32 \cdot 1 = 32$ sec is the span of the most recent part of the slice that is produced as output. In general, any upsampling configuration with $n \cdot T_s \geq N_s^{max} \cdot L_s^{min}$ is valid. Figure 2 presents a simplified example of DTRB functionality in various data input conditions.

3.3 Trajectory analytics - feature generation

According to the definition of the DBP task as described in Section 2, it is clear that at least for the short-term context using simple thresholding in relation to fixed limits, e.g. checking for over-speeding or against the mean value of speed when traversing a specific road, is very inefficient. Instantaneous violations can be missed when the sampling rate of data is too low or when averaged over a temporal frame that is too large. Most importantly, these threshold-based methods using simple 1st-order descriptive statistics, e.g. mean value or standard deviation of speed, max value of acceleration, etc., are not adequate in capturing the actual fine-scale properties of the trajectory, as required for truly effective and robust DBP.

In this work, a very large set of candidate feature functions were employed as the initial pool of DBP trajectory analytics, ranging from 1st- and 2nd-order descriptive statistics to curve, spectral and synthetic features. In summary, this initial feature set included:

- *1st-order statistics*: min, max, (arithmetic) mean, median, mode, stdev, range, skewness, kurtosis, entropy, geometric mean.

[1] Range checks: $0 \leq U_t \leq 55.50$ m/sec (200 km/h); $A_t \leq 10.29$ m/sec^2 (0-100 km/h in 2.7 sec); $R_t \leq 90$ deg/sec (1.5708 rad/sec).

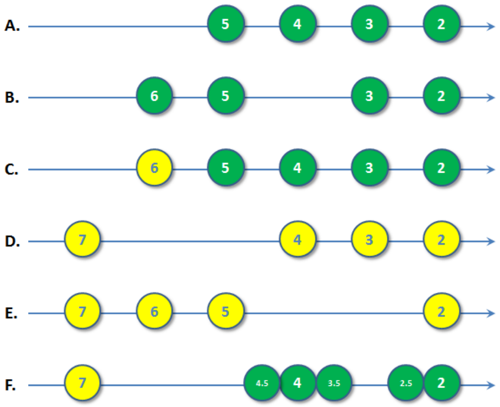

Figure 2: Simplified example of DTRB functionality. Each node represents a data point (input) and the number inside is the $|dt|$ from the current time $t = 0$ sec. DTRB configuration is: $N_s^{min} = 4$, $N_s^{max} = 5$, $L_s^{min} = 0.5$, $L_s^{lim} = 3$, $L_s^{max} = 2$. Upsampling (not shown here) configuration with any $n \cdot T_s \geq N_s^{max} \cdot L_s^{min} = 5 \cdot 0.5 = 2.5$ is valid here. Green (dark) nodes are valid slices for further processing, while yellow (light) are not.

- *Curve statistics*: zero-crossings, roughness index, correlation vs. time, linear regression coefficients, curve vs. geometric length.
- *Synthetic*: ratios between selected 1st-order statistics, e.g. range vs. stdev.
- *Spectral*: auto-regressive AR(2) coefficients, signal 'energy'.
- *2nd-order statistics*: Haralick features [18], run-length features [28].
- *Enrichments*: vehicle type, road type, road speed limit.

In the current state-of-the-art in DBP, most works exploit features from the 1st-order statistics category, mostly because they are easy and fast to calculate and straight-forward to interpret [17, 19, 33]. Some of the curve statistics are also easy to calculate, but usually less effective or significantly correlated to other 1st-order statistics, e.g. zero-crossings with standard deviation. To the best of our knowledge, most of these feature functions have not been used in this context of DBP, i.e., having only sparse, variable-rate, GPS-only location data as input. The rationale for the features described above is that they must capture an information-rich and 'compressed' form of the trajectory properties that are directly or indirectly related to the DBP task at hand.

The entire set of the initial pool of 45 feature functions is applied separately for each of the three data series, i.e., speed, acceleration and turn rate. In addition, the feature set is supplemented with several enrichments (e.g. GPS quality), from which three are DBP-related: vehicle type, road type and road speed limit. The final feature vector, generated for each valid slice produced by DTRB, contains 138 features or 'encoders' of potential DBP mobility patterns in the short-term context. Also, since all the data restrictions of the DBP task have already been addressed by the DTRB (sparsity, noise, road map-matching, upsampling), this feature generation stage is independent and in general it can be applied to any other DBP setup. Figure 3 illustrates some of the processing implemented for translating a very small set of GPS reference points into upsampled data series and feature values extracted from it.

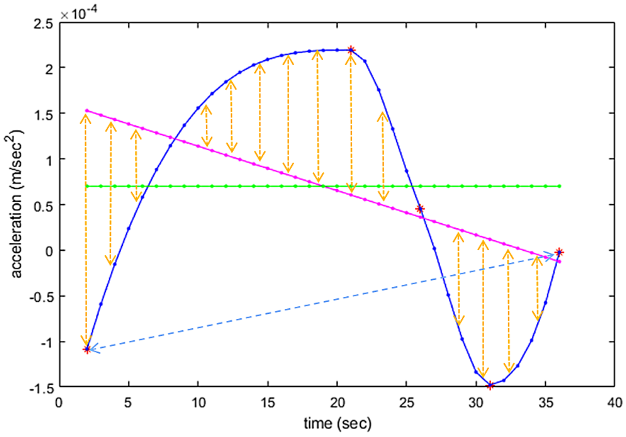

Figure 3: Example of DTRB processing for transforming a low-resolution variable-rate data 'slice' (acceleration) into an upsampled fixed-rate version and modelling for feature extraction; blue is the resampled curve length, magenta is the linear regression trend, green is the mean value, yellow is the signal energy.

Since in this study the DBP task is addressed in its fully unsupervised mode, the goal is to identify 'interesting' features that exhibit explicit statistical characteristics, for example multi-nomial distributions and/or heavy tails, in order to produce clear data groupings and/or outlier zones, respectively. The more explicit these characteristics are, the easier it is for unsupervised models to be trained for detecting 'normal' versus 'abnormal' categories, as described later on in Section 3.5. Figure 4 illustrates an example of a feature with low information content for this task, i.e., very narrow Gaussian distribution with very low skewness (no heavy left/right tails).

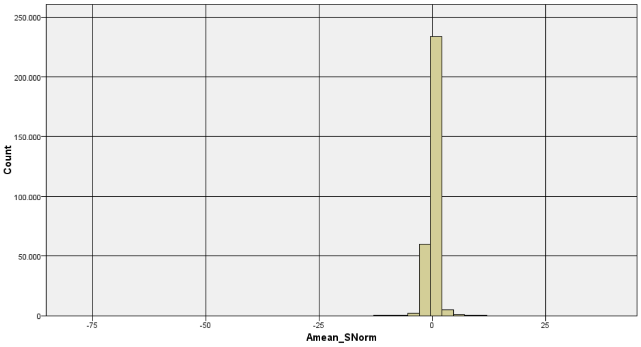

Figure 4: Example of 'bad' feature function for DBP (acceleration: A_t^{mean}).

On the other hand, Figures 5, 6and 7 are examples of such information-rich 'encoders' of clear and distinct groupings of DBP patterns - these are actually the four best-ranked features selected at the end of the dimensionality reduction process, as described next in Section 3.4.

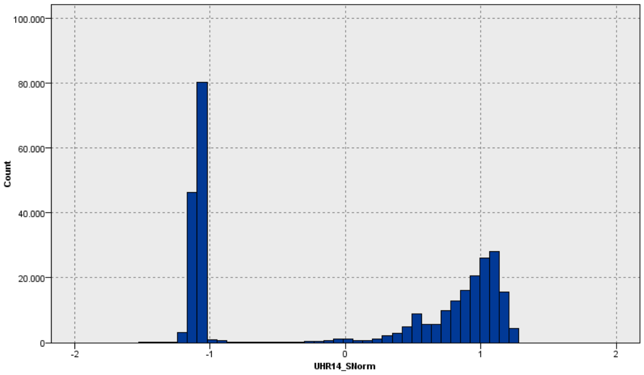

Figure 5: Example of 'good' feature function for DBP (speed: U_t^{HR14}).

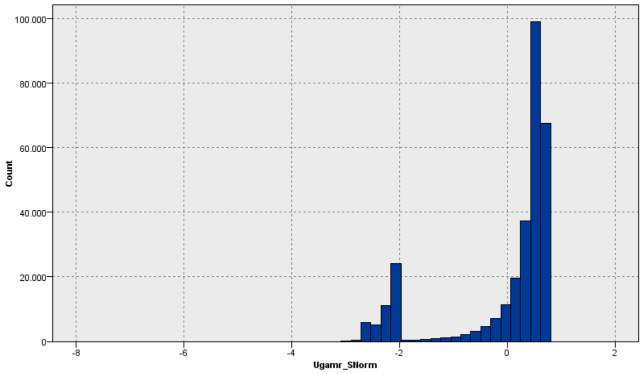

Figure 6: Example of 'good' feature function for DBP (speed: U_t^{gamr}).

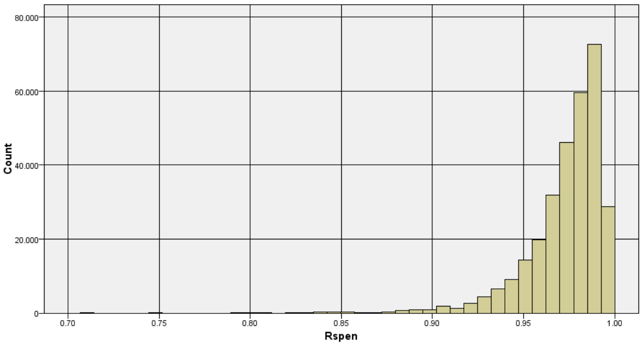

Figure 7: Example of 'good' feature function for DBP (speed: U_t^{vpen}).

3.4 Dimensionality reduction - feature selection

The initial set of feature functions employed is 45 for each data series, i.e., speed, acceleration and turn rate, plus three more included from data enrichments (vehicle type, road type, road speed limit), thus resulting in a total of 138, as described in Section 3.3. This collection of candidate 'encoders' of DBP patterns includes essentially various categories of time series analytics, statistics, signal processing and image analysis algorithms, adapted here for 1-D data series. Before any model training, the features set has to be refined and significantly reduced in size, in order to significantly decrease the dimensionality of the DBP feature vectors dataset and, thus, the complexity of the models.

Since in this work the DBP problem is addressed in its fully unsupervised mode (no ground truth available), most of the standard statistical or heuristic approaches for feature selection are not applicable, since there is no 'target' upon which to investigate the differentiation between features subsets. Thus, the quality and the usefulness of each one of the implemented features is a matter of unsupervised feature selection process, which is a very challenging task by itself.

The feature selection and dimensionality reduction process consists of a multi-step approach, incorporating statistical ranking, factor analysis, predictive model evaluation, etc. More specifically, the first stage was comprised of the following:

(1) *Single-variate analysis* (SVA): Entropy, kurtosis, quartiles, standard deviation.
(2) *Limits-based analysis* (LVA): Adaptive labelling & hypothesis testing against outlier/extreme zones.
(3) *Goodness-of-Fit analysis* (GoF): Kolmogorov-Smirnov test, Jarque-Bera test, Lilliefors test.
(4) *Multi-variate analysis* (MVA): Pairwise correlation, mutual information, cross-entropy.
(5) *Factor analysis* (PCA): Principal Component Analysis for ranking based on eigenvectors.
(6) *Fractal dataset analysis* (FDA): Intrinsic dataset dimensionality analysis.

Next, feature selection via model testing was employed using the refined subset of 31 features, in order to investigate and identify even smaller features subsets, still capturing most of the DBP information. Finally, a third stage of feature refinement produces further shrinkage of the dimensionality is achieved (for the dataset of our study), from 31 down to 4, plus a supplementary subset of potentially useful features that are combined into PCA components, as described later in Section 3.5. More details regarding the internals of the feature selection process above are described in the Appendix.

3.5 Unsupervised learning - Clustering

Using the refined features subset from the two-stage selection process described in Section 3.4, the design of the unsupervised models includes clustering. More specifically, K-Means with Euclidean distance and speed & acceleration statistics as input was implemented as a reference baseline of the most comparable state-of-the-art approach in the relevant DBP literature [33]. Additionally, Two-step clustering with log-likelihood as distance function was employed in this study [36].

There are two main reasons why Two-step clustering is selected as the main algorithm here instead of K-Means. First, it incorporates a pre-clustering step that enables the automatic selection of k for the number of clusters. Second, it incorporates a log-likelihood function as distance metric instead of the Euclidean distance in standard K-Means, hence it is more distribution-agnostic. In practice, this means that the underlying probability distribution for each dimension is not assumed as strictly Gaussian and, hence, the cluster boundaries are more well-fitted to the actual training data. This was verified in the experimental part of this work, where in very similar clustering setups with K-Means and Two-step algorithms, the second one produced cluster boundaries that were more orthogonal against each axis (input dimension), i.e., a model more easily implementable via optimal thresholding per-feature instead of minimum-distance calculations against the centroids in the entire feature space.

Furthermore, a multi-stage approach is employed as a composite clustering model, with each level incorporating a separate Two-step clustering model using only specific features from the input. The optimal selection of features in each case is part of the model design in each clustering level. Again, Silhouette (mostly) and Fisher criterion are employed as quality metrics for the resulting clusterings and a quantitative ranking method for each setup, as well as some qualitative assessment by visual analytics and inspection.

In summary, the following clustering levels are trained in a cascaded form:

- Level-1 (TSL1): Two-step clustering using U_t^{HR14} and U_t^{gamr} as input, resulting in 4 clusters as output.
- Level-2 (TLS2): Two-step clustering using TSL1 cluster id and U_t^{spen} as input, resulting in 8 clusters.
- Level-3 (TSL3): Two-step clustering using TSL2 cluster id and 2 PCA factors as input, resulting in 3 clusters.

In practice, TSL1 uses only two features from the speed data series, namely U_t^{HR14} and U_t^{gamr} described below, to produce the first level of clustering; as their corresponding PDFs illustrate in Figures 5 and 6, these two features effectively produce a very clear four-cluster setup. Similarly, using the output from TSL1 and U_t^{pen} as additional input, another two-dimensions input effectively produces a very clear eight-cluster setup, as described in detail later on in Section 4. Finally, the additional clustering TSL3 can be incorporated for even finer and complementary analysis of the DBP features, if the processing complexity of PCA is acceptable for the application at hand. In this case, the input space is PCA-transformed and, thus, neither the input or the output dimensionality is directly comparable to the ones employed in TSL1 and TSL2 models.

Some of the feature functions used here are based on 2nd-order statistics or 'texture' of a data series, more specifically the well-studied set of 14 features that use the Co-Occurrence Matrix (COM) [18] and 6 features that use the Run-Length Matrix (RLM) [28]. COM is defined as a Np-by-Np matrix $p(i,j)$ that counts pairs of subsequent discrete values $\{i,j\} = \{1,\ldots,Np\}$, where Np is the number of bins used to discretize the continuous range of the target variable, i.e., the same per-series data ranges used for the validity checks in DTRB (see Section 3.2). Similarly, RLM is defined as a Np-by-Nr matrix $r(i,j)$ that counts same subsequent discrete values $i = \{1,\ldots,Np\}$ of sequence lengths or 'runs' $j = \{1,\ldots,Nr\}$, where Np is the number of bins used to discretize the continuous range of the target variable and Nr is the maximum expected 'run'. Whenever the defined Np and Nr discretization lowest or highest limits are exceeded, the corresponding marginal bins are used for the counter updates, i.e., value inside or higher/lower than the discretization limits. The optimal values used in this study were determined experimentally at $Np = 5$ and $Nr = 10$; additionally, the value ranges for U_t, A_t and R_t were scaled down by a factor of 0.5, in order to make COM and RLM more compact and decrease the counts of such updates in their marginal bins.

The features used in the TSL1 and TSL2 models are the following:

- *Maximum Correlation Coefficient*: (speed)

$$U_t^{HR14} = \sqrt{\lambda_2} \tag{1}$$

  where λ_2 is the second-largest eigenvalue of:

$$Q(i,j) = \sum_k \frac{p(i,k)p(j,k)}{p_j(i)p_i(k)} \tag{2}$$

  and: $k = \{1,\ldots,Np\}$, $p_j(i) = \sum_{j=1}^{Np} p(i,j)$, $p_i(j) = \sum_{i=1}^{Np} p(i,j)$.

- *Geometric-to-arithmetic means ratio*: (speed)

$$U_t^{gamr} = \frac{\sqrt[N]{\prod_{i=1}^{N} u_i}}{1/N \sum_{i=1}^{N} u_i} \tag{3}$$

- *Road speed penalty factor*: (speed)

$$U_t^{spen} = sign(U_t^{vpen} - 0.98) \tag{4}$$

  where:

$$U_t^{vpen} = \frac{U_{rlim}}{\max(0, U_t - U_{rlim}) + U_{rlim}} \tag{5}$$

Note that, since all calculations are applied to discrete- rather than continuous-valued series for U_t, the term u_i in Eq.3 is essentially identical to U_t for $i = t$.

Regarding the penalty factor related to the (local) road speed limit, it is $0 < U_t^{vpen} \leq 1$; in reality, in most cases $0.7 \leq U_t^{vpen} \leq 1$. When $U_t \leq U_{rlim}$ then $U_t^{vpen} = 1$, i.e., speed strictly within the permitted limit, and when $U_t > U_{rlim}$ then $U_t^{vpen} < 1$, i.e., $U_t^{vpen} = \frac{U_{rlim}}{U_t}$. Thus, the threshold $U_t^{vpen} \leqslant 0.98$ is translated to actual speed $U_t = U_{rlim}/0.98 \approx 1.02 \cdot U_{rlim}$ or speed at least 2% over the permitted limit. Here, a hard-thresholded value U_t^{spen} at the level 0.98 is used instead of U_t^{vpen}. This is valid in the sense that, as described in section 2.1, 'hard' regulations e.g. for speed limits are one of the factors that define the DBP problem.

The features used in the TSL3 model are the first two PCA factors calculated for the subset of the following eight supplementary features:

- *Value non-uniformity*: (speed, acceleration, turn rate)

$$X_t^{RL03} = \frac{\sum_{i=1}^{Nx}(\sum_{j=1}^{Nr} r(i,j))^2}{\sum_{i=1}^{Nx}\sum_{j=1}^{Nr} r(i,j)} \tag{6}$$

  where:

$$X_t = \{U_t, A_t, R_t\} \tag{7}$$

- *Sum entropy*: (speed)

$$U_t^{HR08} = \sum_{i=2}^{2N} p_{x+y}(i) \log p_{x+y}(i) \tag{8}$$

- *Mode-to-mean ratio*: (speed)

$$U_t^{gamr} = \frac{mode(U_t)}{1/N \sum_{i=1}^{N} x_i} \tag{9}$$

  where $mode(U_t)$ is the value where the peak of the PDF occurs.
- *Range-to-stdev ratio*: (turn rate)

$$R_t^{rsr} = \frac{\max R_t - \min R_t}{\sqrt{\sum_{i=1}^{N} (x_i - \mu_x)^2 / N}} \tag{10}$$

  where $\mu_x = 1/N \sum_{i=1}^{N} x_i$ is the mean value of R_t.
- *AR(2) 1st-order coefficient* : (speed, acceleration)

$$X_t^{ar1} = \alpha_1 , \quad A(z)X_t = e_t \tag{11}$$

  where $A(z) = 1 - \alpha_1 z^{-1} - \alpha_2 z^{-2}$ an AR(2) auto-regressive model of order 2 for the best-approximation (minimum error e_t) model identification of series X_t via the Yule-Walker algorithm [22] and $X_t = \{U_t, A_t\}$.

Based in this multi-stage clustering approach and the specific TSLx models designed for each stage, Section 4 describes the experimental protocol and the results for their assessment, using the real-world dataset described in Section 4.1.

4 EXPERIMENTS AND RESULTS

4.1 Datasets used

In this work, an extensive real-world trajectory dataset of GPS location data is used as the basis, consisting of 977,646 records generated by special-purpose devices installed in 2,638 large vehicles (transport trucks) travelling in the main urban area of the city of Athens (Attica region, Greece) for a period of 24 hours in a typical weekday. More specifically, the dataset was defined within the spatial bounding box: Lat = [37.8860, 38.1057] North / Lon = [23.5591, 23.9128] East and temporal frame: 2-Nov-2018 (00:00':00"-23:59':59"). The data enrichments supplemented are various parameters regarding the local weather (precipitation, temperature, wind speed & direction from NOAA) [16], the underlying road network (OpenStreetMaps – OSM) and the GPS signal quality (number of satellites tracked, map-matching distance & probability as described next).

4.2 Experimental work & results

The experimental protocol in this study was based on the real-world dataset described previously.Most of the experimental work followed the four-phase sequence summarized at the beginning of Section 3, while some parts required iterations between feature subset refinement and model design for clustering (see Sections 3.4 and 3.5, respectively). Various hardware/OS[2] and software[3] platforms were used for the experimental work, some of which is currently ported to R, Java and Python for open cross-platform use.

For DTRB, Figure 8 illustrates the 3-D histogram of number of extracted valid slices from the data per data points included and per temporal span used, which was the main guideline for the optimal configuration of the DTRB parameters for the dataset at hand.

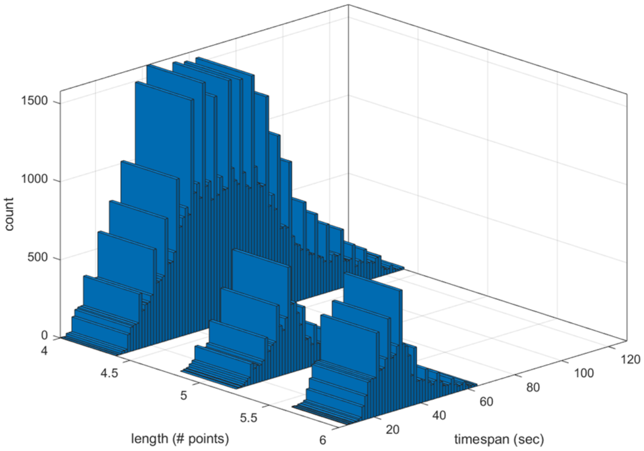

Figure 8: DTRB: Histograms of extracted slices versus data points and temporal span used.

The results from the reference baseline of the most comparable state-of-the-art DBP approach [33], using K-Means with Euclidean distance and speed & acceleration statistics as input, is presented in Figure 9. This is directly comparable to the TSL1 model, proposed in this study, with its results presented in Figure 10. It is clear that in the second case the clusters are significantly enhanced in terms of shape and separation, while retaining almost the same discrimination ratios in the dataset, i.e., in the smallest cluster ('outliers').

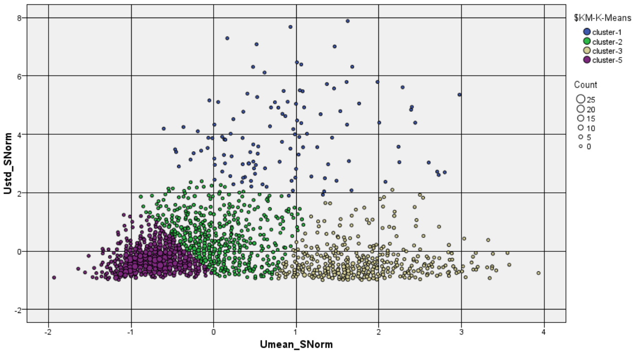

Figure 9: K-Means reference model: 4 clusters, smallest 4.2%, silhouette=0.6.

In the second stage of clustering, results from TSL2 are presented presented in Figure 11. Again, it is evident that with the addition of one more optimally-selected feature to the TSL1 output, the clusters become even more well-shaped and separated, almost orthogonally with centroid very close to the 8 corners of the hypercube.

[2] Intel core i7-3537U @2.00GHz & 8GB memory; Intel core i7-8550U @1.80GHz & 32GB memory; Microsoft Windows 8.1 & 10; Ubuntu Linux 19.04 & 18.4 LTS.

[3] Mathworks MATLAB v9.4/R2018a (x64); Octave v5.1.0; R v3.6.2; WEKA v3.9.4; IBM SPSS Modeler v14.1 & Statistics v26; custom Java & C/C++ tools for data import/export.

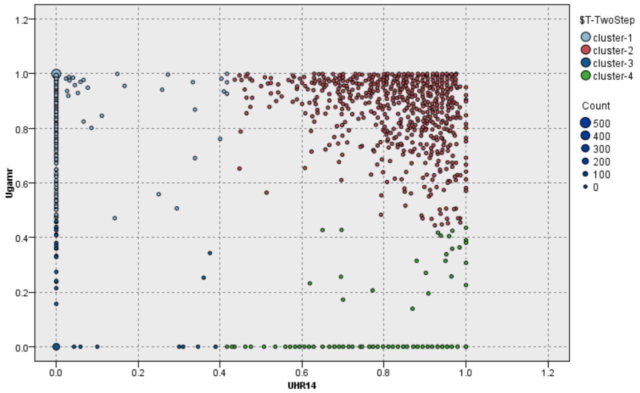

Figure 10: TSL1 model: 4 clusters, smallest 8.1%, silhouette=0.9.

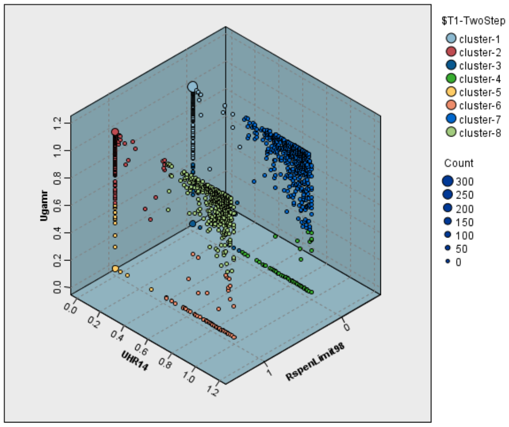

Figure 11: TSL2 model: 8 clusters, smallest 3.4%, silhouette=1.0.

Lastly, in the third stage of clustering, results from TSL3 in Figure 12 illustrate the usefulness of adding the PCA-transformed (top 2 factors used) supplementary subset of eight more optimally-selected features. Although the quality metric (silhouette) seems worse than in TSL2 and TSL1, in fact this clustering space embodies the intrinsic information content of the best 3+8 features, ranging from simple statistics to spectral model coefficients and for all three data series (speed, acceleration, turn rate), while at the same time producing well-defined clusters in a low-dimensionality space (3-D).

5 DISCUSSION

Based on the experimental results, the proposed four-phase methodology manages to successfully address all the challenges and data limitations of this DBP problem specification. DTRB together with efficient map-matching & filtering enables the necessary quality enhancement of the low-quality raw input, which otherwise would be unusable for developing the subsequent phases.

The extensive initial pool of features that are relevant to DBP patterns were gradually refined by employing both fast statistical methods (initially), as well as model testing and heuristics (later on), in order to end up with only few, very efficient and robust features subset of DBP pattern 'encoders'. Additionally, this final selection includes features of low to moderate computational complexity at least for TSL1 and TSL2 (no PCA required), with the calculation of COM and eigenvalue λ_2 in Eq.1 being the most demanding. Nevertheless, even in this case the selected COM size (5x5) is adequate for capturing the core information content from the fairly limited data involved (32 upsampled data points) and, thus, simple enough to enable on-the-fly calculations.

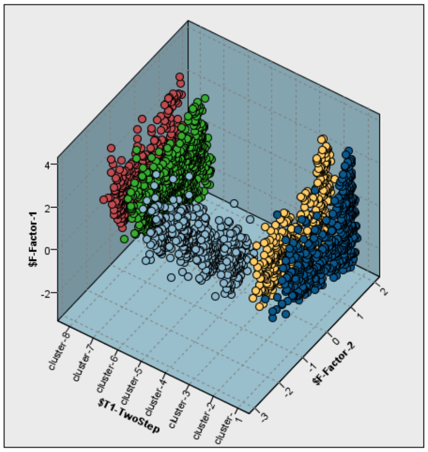

Figure 12: TSL3 model: 5 clusters (balanced), silhouette=0.3.

Finally, the multi-stage clustering approach provides a solution of scalable complexity: a very simple model in TSL1 using only two features; an additional clustering level in TSL2 using context-relevant data (road speed limit), is such enrichment data are available; and another, more demanding clustering level TSL3 using 8 additional features with PCA transformation, for low-volume or offline DBP applications.

6 CONCLUSIONS

In this work, the DBP problem is addressed in the short-term context and with the most data-restrictive setup, using as input only low-resolution GPS-only location data of variable sampling rate. The proposed approach introduces online HMM-based map-matching to the underlying road network and robust noise filtering, as well as an algorithm for dynamic temporal resampling, to generate upsampled fixed-rate data series for speed, acceleration and turn rate.

Starting from an extensive set of feature functions, ranging from simple statistics to spectral and 'texture' analytics, the most content-rich in terms of DBP are selected. For fully unsupervised predictive modelling, a multi-stage clustering is designed and tested with a real-world dataset. The results prove the feasibility and effectiveness of the proposed approach.

Further enhancements of the proposed approach are planned in relation to the optional integration of data modalities, to exploit sensor-based acceleration instead of GPS-induced, improved clustering models, designed specifically for on-the-fly processing, as well as state-sequencing of the DBP predictive process, to enable stateful instead of stateless DBP characterization.

Feature selection (details) As described in Section 3.4, the large initial size of candidate feature functions required a multi-step feature selection and dimensionality reduction process, incorporating methods of gradual complexity and descriptive power.

In SVA, entropy and kurtosis of each estimated probability distribution function (PDF) of each feature via histogram over the entire dataset was used as hint of non-Gaussianity, similarly to other established methods, e.g. in blind source separation (BSS) via Independent Component Analysis (ICA), were kurtosis is a common choice for testing non-Gaussianity of mixed data sources. In GoF, three well-established Gaussianity tests, namely Kolmogorov-Smirnov, Jarque-Bera and Lilliefors, were employed to the PDF of each feature over the entire dataset. MVA addresses the issues of correlations and complementarity between pairs of features. Besides the standard Pearson pairwise correlation, mutual information and cross-entropy were also employed as indices aggregated to per-feature vectors for ranking against non-redundancy of dimensions. Additionally, PCA components over the entire dataset and various subsets of features were analysed in terms of variance explained. At the end of the analysis above, after aggregating all the individual feature rankings and identifying consistently 'good' features across multiple analysis methods, the initial set of 138 features was reduced to 31, i.e, a 4.45:1 shrinkage of the dimensionality.

The refined subset of 31 features was used as input to model-based evaluation. More specifically, a Expectation–Maximization (EM) algorithm for fast clustering was employed with several options for heuristic features subset evaluation. Silhouette and Fisher criterion [29], were employed as quality metrics for the resulting clusterings. The significantly reduced final subset of optimal features included 3 main plus 8 supplementary candidates, with were employed as the input in the core TSLx models, as described in Section 3.5.

ACKNOWLEDGEMENTS

This work was partially supported by the project *Track & Know* (No:780754), which has received funding from the European Union's Horizon 2020 (H2020) programme. `http://trackandknowproject.eu/`

REFERENCES

[1] A. Bender, G. Agamennoni, J. Ward, S. Worrall, and E. Nebot. 2015. An unsupervised approach for inferring driver behavior from naturalistic driving data. *IEEE Trans. Intell. Transp. Syst.* 16, 6 (2015).

[2] L. Bergasa, J. Nuevo, M. Sotelo, R. Barea, and M. Lopez. 2006. Real-time systemformonitoring driver vigilance. *IEEE Trans. on Intell. Trans. Sys.* 7, 1 (2006).

[3] R Bhattacharyya, R Senanayake, K Brown, and M Kochenderfer. 2020. Online Parameter Estimation for Human Driver Behavior Prediction. *arXiv* (2020), 1–6. https://doi.org/arXiv:2005.02597v1

[4] G. Castignani, T. Derrmann, R. Frank, and T. Engel. 2015. Driver behavior profiling using smartphones: A low-cost platform for driver monitoring. *IEEE Intell. Transp. Syst. Mag.* 7, 1 (2015).

[5] R Chandra, U Bhattacharya, T Mittal, A Bera, and D Manocha. 2020. CMetric: A Driving Behavior Measure using Centrality Functions. *arXiv* (2020), 1–8. https://doi.org/arXiv:2003.04424v2

[6] J. Engelbrecht, M. Booysen, J. van Rooyen, and F. Bruwer. 2015. Survey of smartphone-based sensing in vehicles for intelligent transportation system applications. *IET Intell. Transp. Syst.* 9, 10 (2015).

[7] D Farooq and S Moslem. 2020. Evaluation and Ranking of Driver Behavior Factors Related to Road Safety by Applying Analytic Network Process. *Periodica Polytechnica Transp Eng* 48, 2 (2020), 189–195. https://doi.org/10.3311/PPtr.13037

[8] F. Feng, S. Bao, J.R. Sayer, C. Flannagan, M. Manser, and R. Wunderlich. 2017. Can vehicle longitudinal jerk be used to identify aggressive drivers? An examination using naturalistic driving data. *Accid. Anal. Prev.* 104 (2017).

[9] J Ferreira, E Carvalho, BV Ferreira, and et.al. 2017. Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS ONE* 12, 4 (2017), 1–16.

[10] C. Goh, J. Dauwels, N. Mitrovic, M. Asif, A. Oran, and P. Jaillet. 2012. Online map-matching based on Hidden Markov model for real-time traffic sensing applications. In *15th IEEE Intl. Conf. Intell. Transp. Sys. (ICITS)*.

[11] J. Goncalves, J.S. Goncalves, R. Rossetti, and C. Olaverri-Monreal. 2014. Smartphone sensor platform to study traffic conditions and assess driving performance. In *Proc. 17th Int. IEEE Conf. Intelligent Transportation Systems (ITSC)*.

[12] F. Guo, S. Klauer, J. Hankey, and T. Dingus. 2010. Near crashes as crash surrogate for naturalistic driving studies. *Transp. Res. Rec.* 2147 (2010).

[13] B. Higgs and M. Abbas. 2015. Segmentation and clustering of car-following behavior: Recognition of driving patterns. *IEEE Trans. Intell. Transp. Syst.* 16, 1 (2015).

[14] J. Hong, B. Margines, and A. Dey. 2014. A smartphone-based sensing platform to model aggressive driving behaviors. In *Proc. 32nd Annu. ACM Conf. Human Factors in Computing Systems (CHI)*.

[15] M Khan and Lee S. 2019. A Comprehensive Survey of Driving Monitoring and Assistance Systems. *Sensors* 2574, 19 (2019), 1–32. https://doi.org/10.3390/s19112574

[16] N. Koutroumanis, G. Santipantakis, A. Glenis, C. Doulkeridis, and G. Vouros. 2019. Integration of Mobility Data with Weather Information. In *Proc. EDBT/ICDT 2019 Joint Conference (EDBT/ICDT)*.

[17] N. Lin, C. Zong, M. Tomizuka, P. Song, Z. Zhang, and G. Li. 2014. An Overview on Study of Identification of Driver Behavior Characteristics for Automotive Control. *Math. Probl. Eng.* 2014 (2014).

[18] T. Lofstedt, P. Brynolfsson, T. Asklund, T. Nyholm, and A. Garpebring. 2019. Gray-level invariant Haralick texture features. *PLoS ONE* 14, 2 (2019).

[19] G. Meiring and H. Myburgh. 2015. A review of intelligent driving style analysis systems and related artificial intelligence algorithms. *Sensors* 15, 12 (2015).

[20] E. Ohn-Bar and M. Trivedi. 2016. Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Trans. Intell. Veh.* 1, 1 (2016).

[21] J. Paefgen, F. Kehr, Y. Zhai, and F. Michahelles. 2012. Driving behavior analysis with smartphones: Insights from a controlled field study. In *Proc. 11th Int. Conf. Mobile and Ubiquitous Multimedia (MUM)*.

[22] B. Porat. 1994. *Digital processing of random signals: Theory and methods.* Prentice Hall, Englewood Cliffs, NJ, USA.

[23] J. Przybyla, J. Taylor, J. Jupe, and X. Zhou. 2015. Estimating risk effects of driving distraction: A dynamic errorable car-following model. *Transp. Res. C* 50 (2015).

[24] W. Rongben, G. Lie, T. Bingliang, and J. Lisheng. 2004. Monitoring mouth movement for driver fatigue or distraction with one camera. In *Proc. 7th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.

[25] F. Sagberg, G.F. Bianchi, and J. Engstrom. 2015. A review of research on driving styles and road safety. *Hum. Factors* 57, 7 (2015).

[26] J Sun, J Xu, R Zhou, and et.al. 2018. Discovering Expert Drivers from Trajectories. In *34th IEEE Intl Conf on Data Eng 2018 (ICDE'18), Paris, France.* 1332–1335. https://doi.org/10.1109/ICDE.2018.00143

[27] Z. Sun and X.J. Ban. 2013. Vehicle classification using GPS data. *Transp. Res. C* 37 (2013).

[28] X. Tang. 1998. Texture information in run-length matrices. *IEEE Trans. Im. Proc.* 7, 11 (1998).

[29] S. Theodoridis and K. Koutroumbas. 2008. *Pattern Recognition* (4th ed.). Academic Press.

[30] S Ullah and D-H Kim. 2020. Lightweight Driver Behavior Identification Model with Sparse Learning on In-Vehicle CAN-BUS Sensor Data. *Sensors* 5030, 20 (2020), 1–21. https://doi.org/10.3390/s20185030

[31] W. Wang, J. Xi, A. Chong, and L. Li. 2017. Driving style classification using a semisupervised support vector machine. *IEEE Trans. Human–Mach. Syst.* 47, 5 (2017).

[32] W. Wang, J. Xi, and D. Zhao. 2019. Driving style analysis using primitive driving patterns with Bayesian nonparametric approaches. *IEEE Trans. on Intell. Trans. Sys.* 20, 8 (2019).

[33] Josh Warren, Jeff Lipkowitz, and Vadim Sokolov. 2019. Clusters of driving behaviour from observational smartphone data. *IEEE Intell. Trans. Sys. Mag.* 11, 3 (2019).

[34] H Wu, W Sun, and B Zheng. 2017. A fast trajectory outlier detection approach via driving behavior modeling. In *ACM Conference on Information and Knowledge Management 2017 (CIKM'17), Singapore.* 837–846.

[35] Y Yao, X Zhao, Y Wu, Y Zhang, and J Rong. 2019. Clustering driver behavior using dynamic time warping and hidden Markov model. *J Intel Transp Systems* 25, 3 (2019), 249–262. https://doi.org/10.1080/15472450.2019.1646132

[36] T. Zhang, R. Raghu, and L. Miron. 1996. BIRCH: An efficient data clustering method for very large databases. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data. Canada (SIGMOD)*.