Aggregating Telematics Data for Road Safety Analysis

Simone Paradiso[a, *], Apostolos Ziakopoulos[a], George Yannis[a]

[a]National Technical University of Athens, Department of Transportation Planning and Engineering, 5 Iroon Polytechneiou Street, GR-15773, Athens, Greece

(Contact: simone_paradiso@mail.ntua.gr, apziak@central.ntua.gr, geyannis@central.ntua.gr)

Abstract

Spatial data play a key role in road safety analysis, with OpenStreetMap (OSM) offering valuable geospatial insights. This study presents an improved method for aggregating telematics data from a smartphone app into OSM entities, such as nodes and edges, to enhance road safety spatial analysis. The dataset used in the study consists of several trips where each instance is characterized by geospatial coordinate corresponding to driver locations, along with features related to driver dynamics. Telematics data was aggregated to edges using the nearest edge approach and to nodes with a buffer method, introducing inconsistencies. This refined approach constrains data points to edges connected to the buffer origin node, reducing node variance and creating a dataset with lower variance, resulting in a more structured representation. An Autoencoder was tested on the produced database, with the loss curve and a PCA variance distribution indicating a more consistent and meaningful dataset.

Keywords: *Road Safety, Telematics Data, OpenStreetMap, Spatial Analysis, Data Aggregation, Autoencoder curve, Principal Component Analysis.*

1. Introduction

According to estimates from the World Health Organization (WHO), road safety remains a critical public health concern, with approximately 1.19 million fatalities resulting from road crashes worldwide. Road traffic injuries are the leading cause of death among children and young adults aged 5 to 29 years and rank as the 12th leading cause of death globally (WHO, 2023). While road safety is a critical global issue, focusing on Europe, in 2022, approximately 20,600 fatalities occurred due to road crashes across the European Union, a 3% increase from 2021 as traffic volumes returned to pre-pandemic levels (Road Safety in the EU, 2023).

In road safety analysis, spatial data is essential for understanding the factors contributing to hazardous conditions (Ziakopoulos & Yannis, 2020). A widely used resource in this domain is OpenStreetMap (OSM), a free, editable global map created by volunteers and released under an open-content license (OpenStreetMap, 2025). Traditionally historical road crash data have been used as main indicator to measure road safety outcomes. Over the past decades, researchers have been exploited Surrogate Safety Measures (SSMs) as proxy measurements which can complement or substitute crash data, helping to address the challenge posed by the crash data rarity (Nikolaou, Ziakopoulos, et al., 2023). Moreover, crash-focused data are useful to explore the risk given a crash, but they do not support research exploring the risk of a crash itself when the risk is defined as the probability that exposure to a hazard (crash or being on a road) leads to a negative consequence (Tarko, 2018). Recently researchers integrated SSMs with spatial data for road safety analysis, as in

(Nikolaou, Dragomanovits, et al., 2023) who exploited road geometry data and SSMs to investigate various aspects of road crashes on motorway segments.

The increasing availability of telematics data, captured through smartphone apps, has introduced new opportunities for enhancing road safety analysis. However, their applications in spatial analysis remain under-researched in the current literature. Several studies have exploited telematics data to derive Safety SSMs. Telematics data have been investigated both for their correlation with crashes (Stipancic et al., 2021) and for their potential to substitute crash data, when it comes to road safety outcomes, when combined with spatial information (Nikolaou et al., 2025), .

Telematics is most used in insurance to predict claim frequency, improving upon classical models, with intense use of machine learning and regression techniques (Boylan et al., 2024). Some studies have explored the potential of integrating telematics data into structured graph to enhance the analysis of driving behavior and road safety (Stipancic et al., 2019).

Furthermore, thanks to telematics and the latest technological advancements, insurance companies have developed Usage-Based-Insurance (UBI) schemes. These are schemes where the insurance rate is affected by their driving behaviour instead of traditional auto insurance pricing factors only, such as driving experience, vehicle type, etc., promoting improvements in society by encouraging better driving behavior among the population, leading to long-term crash reductions and environmental benefits through reduced fuel consumption and emissions (Ziakopoulos et al., 2022).

This study presents an approach to aggregating telematics data to the graph generated by querying OSM via OSMnx, which is a Python library that downloads and analyzes street networks for anywhere in the world from OSM. With a single line of code, you can query data by bounding box, address, or place name, and specify network types such as drive, walk, bike, and more (Boeing, 2017). The output is a graph representation of the street network. Such graphs are accompanied by two datasets: one for the nodes and another for the edges. The proposed approach focuses on presenting a clearer method for aggregating telematics data points to a single node or edge in the OSM-derived graph.

An Exploratory Data Analysis (EDA) was conducted to gain insights into different aggregation methods. Techniques such as Principal Component Analysis (PCA) and an Autoencoder (AE) were utilized to further explore the data, aligning with the recent surge in machine learning and deep learning models in road safety (Silva et al., 2020). By combining the insights gained from the dataset variance, the explained variance from PCA, and the ease of input reconstruction from the AE, a conclusion was drawn.

The structure of the paper is as follows: After this Introduction, the Main Text includes the Methods section presenting the processing of data along with the chosen models. In the Results section the models are analyzed, and performances are compared. In the Discussion the findings are interpreted, and an overview of the results is provided; finally, the study is summarized and some takeaways are highlighted in the Conclusion section.

2. Main Text

2.1 Methods

The methodology of the paper is centered on the analysis of telematics data obtained from a smartphone application developed by OSeven Telematics (OSeven, 2025), that records driver behavior using smartphone hardware sensors.

The OSeven app collects highly disaggregated in space and time. They are stored in the backend cloud server, and signal processing, Machine Learning (ML) algorithms, Data fusion and Big Data algorithms are used to transform raw data into driving behavior indicators. This is achieved by using state-of-the-art technologies and procedures and operating in compliance with standing Greek and

European personal data protection legislation (GDPR) (Kontaxi et al., 2022). The overall flow system is illustrated in ***Figure 1***.

Figure 1: *OSeven data flow system.*

The dataset used in the current study consists of anonymous trips. Each record in the dataset is characterized by a geospatial coordinate corresponding to a driver's location at a per-second frequency (1Hz), along with various features related to the driver dynamics and behavior. The variables used, including those derived from the dataset, are listed in the first column of **Table 1**.

The smartphone hardware sensors to collect the data involve the use of an accelerometer, gyroscope, magnetometer, and GPS, while data fusion techniques are provided by iOS and Android with nine degrees of freedom models (Yaw, Pitch, Roll), gravity, and linear acceleration (Kostopoulos et al., 2024). Studies found that young drivers are more probable to choose the telematics-based insurance policy compared to older ones, raising concerns that the data may be skewed toward a younger demographic (Tselentis et al., 2018).

Half of the features—SpeedingFlag, Mobile_usage, Harsh_acc, and Harsh_brk—are originally binary variables indicating whether the respective event occurred. Event_intensity reflects the intensity of harsh events on a scale from 1 to 3. Trips_count and Points_count represent the number of trips and individual data points associated with the spatial entity. Finally, smoothedSpeed indicates the vehicle's speed at a specific coordinate point.

Given the coverage of telematics, a bounding box was used to extract geometric features from OSM, resulting in a structured graph. ***Figure 2*** illustrates a zoomed-in portion of this graph.
From the defined graph in OSM, node and edge features were stored in two separate datasets, with the considered nodes being the "true" endpoints of edges (i.e., intersections or dead-ends) (Boeing, 2024).

GeoPandas library (GeoPandas, 2025) was used to aggregate telematics to the data, allowing spatial operations on geometric types.

Common geometric types used to represent spatial data in GeoPandas are:

- Point: Represents a single location in space. In our case study each record of the telematics data corresponds to a Point, as does each record in the Node dataset.
- LineString: Represents a sequence of connected points forming a line. In our case study, the Edge dataset has a geometric representation defined as a LineString.
- Polygon: Represents a closed shape defined by a sequence of points (with the first point being the same as the last), forming an area. When creating a buffer around a Point, the result will be a polygon geometry.

Moreover, GeoPandas provides two spatial-join functions, both utilized in this study, to merge two geometric objects based on their spatial relationship:

- GeoDataFrame.sjoin(): the function performs a spatial join based on specific spatial relationships (e.g., intersects, within, contains, etc.) between geometries. It is used in order to join based on the spatial relations of geometries (e.g., points within polygons).
- GeoDataFrame.sjoin_nearest(): the function performs a spatial join based on proximity. It joins geometries from one GeoDataFrame to the nearest geometry in another GeoDataFrame.

While the first one was used to aggregate telematics data to the nodes after creating a buffer around the node coherently with what has been done in the field (Erramaline et al., 2022), (Stipancic et al., 2019), the second one was used to aggregate telematics data to the edges.

Aggregation is the process of combining things or amounts into a single group or total.
Therefore, aggregating telematics data to spatial entities means combining individual telematics observations into summarized information organized by spatial features (like roads, intersections, zones).

In the OSeven dataset, each telematics instance includes geographic coordinates, making it possible to link the data points to the corresponding geometric entities. This can be thought of as overlapping the telematics points onto the spatial network, and then aggregating them — for example, by summing, averaging, or applying other operations — per spatial entity.

The aggregation to the edges is straightforward, performed by applying a spatial nearest-neighbor join, as mentioned before, between the telematics data points and the edges. In this procedure, each point is linked to the nearest edge based on the Euclidean distance between the two geometries, enabling the aggregation per single edge.

Aggregation to the nodes is firstly achieved by performing a spatial join between the telematics data points and a 50-meter buffer zone created around each node in the network, following the literature (Petraki et al., 2020; Stipancic et al., 2019). The radius was stated to be helpful to avoid large, continuous overlaps, thus maintaining realistic conditions for the research. However, this simple approach requires further analysis.

The buffer approach has a key drawback: it can distort the representation of the origin node if multiple nodes are present within the buffer.
If multiple nodes lie within the buffer zone, issues in associating telematics data to the correct node might arise. This happens because the buffer includes nodes that are geographically close, however some telematics points are not directly relevant to the origin node but instead they affect other nodes within the buffer.

Figure 2 provides an example of a buffer containing six nodes (i.e. intersections). The blue node represents the origin node from which the 50-meter buffer was generated, while the green and red dots are the telematics data points.
The green dots represent points that fall on the edges directly connected to the origin node hence directly relevant to the origin node. The red dots are points within the buffer that are not located on edges directly connected to the origin, hence they do not have impact on the origin node, as they are not associated with its specific traffic flow.

A further complication arises when choosing to aggregate telematics points solely by assigning them to their nearest node. This approach could lead to a case where some points are further than the buffer radius from the nearest intersection and thus falling outside the buffer and not being considered influencing for the node.

Therefore, a mixed, more structured approach, is necessary to overcome these limitations.

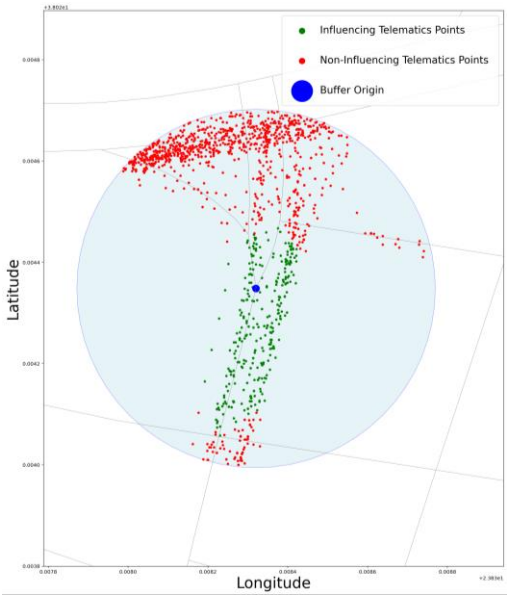

Figure 2: *Telematics Data Points and Buffers (Example).*

In addition, the simple buffer approach can create "isolated" nodes characterized by telematics data, while the outgoing edges from these nodes lack telematics. This occurs when the buffer covers several edges, however the edges directly connected to the origin node have no associated telematics data, while other edges within the buffer do, as shown in ***Figure 3***.

The figure illustrates the case where the main roads, which carry the traffic flow, are not connected to the origin node, whereas the roads connected to the buffer are actually empty due to the network structure.

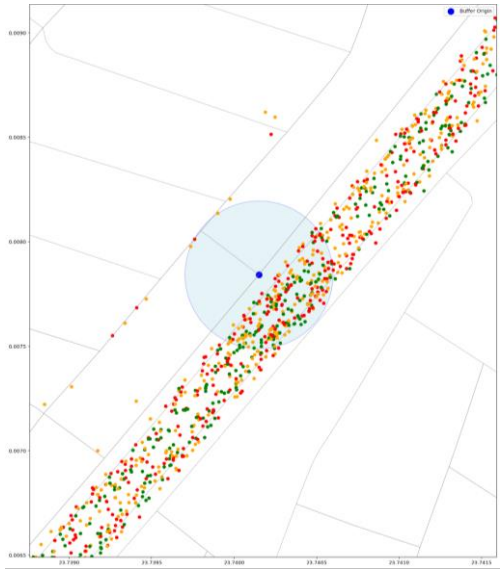

Figure 3: *Isolated Node with Connected Edges Lacking Telematics Data (Example).*

Based on the previous observations, the buffer approach was refined by adding an additional constraint: telematics data points must not only fall within the buffer but also lie on the edges connected to the origin node. The following assumptions were made:

- Within 50 meters of the origin buffer node, the driver influences its characteristics. Beyond this distance, there is no effect.

- ❖ The telematics points that influence the node must be located on the edges that intersect at the origin buffer node.
- ❖ A point p is located on the edge e, if and only if e is the nearest edge to p.

In **Figure 2**, the green points represent those points that satisfy these assumptions.

The redefined approach leads to a new aggregated dataset, compared to the buffer-only approach. The two datasets were analyzed by first comparing their internal standard deviation. A lower standard deviation indicates data points more concentrated around the mean and hence an increased homogeneity in the dataset, vice versa a higher standard deviation, which indicates a more heterogeneous dataset.

Another tool used to compare the two datasets is Principal Component Analysis (PCA), a technique used to reduce the dimensionality of a dataset, while preserving as much 'variability' (i.e. statistical information) as possible (Jolliffe & Cadima, 2016). PCA aims to create new variables that are linear functions of those ones in the original dataset, that successively maximize variance and that are uncorrelated with each other. This problem translates to solving an eigenvalue problem.
Once these new variables (principal components) are identified, their importance can be defined by the explained variance which indicates how much of the data's variability is persevered by each principal component. This information might be used for dimensionality reduction tasks, to reduce the number of features.

Additionally, the Proportion of Variance Explained (PVE) for a single principal component m is calculated as follows:

$$PVE_m = \frac{EV_m}{Total\ variance} = \frac{\frac{1}{n}\sum_{i=1}^{n} {z_{im}}^2}{\sum_{j=1}^{p}\frac{1}{n}\sum_{i=1}^{n} {x_{ij}}^2} = \frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p} \emptyset_{jm}\, x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n} {x_{ij}}^2}, \qquad (\mathbf{1})$$

Where:

- n is the number of observations.
- p is the number of original features.
- x_{ij} is the value of the j-th original feature for the i-th observation.
- $\emptyset_{jm}$ is the loading of the j-th original variable on the m-th principal component, similar to a weight of the variable in forming the principal component.
- z_{im} is the score of the i-th observation on the m-th principal component (i.e., the projection of the observation onto that component). It shows how much the observation matches along the principal component.

The denominator is the total variance in the dataset aggregated across all the features and observations, whereas the numerator is the Explained Variance (EV) for the m-th principal component, indicating how much variance is captured by that principal component.

By examining the PVE across the principal components, insights into the data complexity can be gained. For instance, if one component explains most of the variability, the dataset has a simple structure with the data being concentrated along a single feature (Shlens, 2014), providing a reasonable characterization of the complete data set. On the other hand, if the PVE is evenly distributed across the components, it indicates a more homogeneous and complex structure of the dataset.

Deep learning, a part of the broad field of Artificial Intelligence (AI), which focuses on the development of intelligent machines that have the ability to achieve goals like humans do (Sze et al., 2017), offers various tools, one of them is the Autoencoder (AE) (Chen & Guo, 2023). An AE takes an input vector X and then maps it to a hidden representation Z using a deterministic mapping process:

$$Z = f_\theta(X)$$

The latent representation Z, or the hidden representation, is then mapped back into a reconstruction vector $\hat{X}$, with the same shape of X. The mapping is performed using a similar transformation:

$$\hat{X} = g_{\emptyset}(Z)$$

Where f_{θ} and $g_{\emptyset}$ are two neural networks, called respectively encoder and decoder. The parameters θ and $\emptyset$ refer to the learnable parameters in these networks.

The goal of an AE is indeed to learn an efficient, compressed representation of data. The encoder is forced to map the input X to a more compact representation, into a lower-dimensional latent space Z. The latent representation Z is reconstructed by the decoder. This process enables dimensionality reduction while preserving the information of the input data.
Mapping the input into a lower-dimensional space through AEs enables effective dimensionality reduction, with a key advantage: unlike PCA, AEs can capture non-linear relationships between input features, providing more flexibility (Michelucci, 2022).

In the present analysis, the goal is to train an AE using the same architecture and hyperparameters on the two datasets built through different aggregation approaches. This aims to perform feature extraction and evaluate how well the model reconstructs the original inputs across the two datasets, by comparing the reconstruction errors and analyzing the loss curves, thereby assessing the capability of the autoencoder in learning a meaningful representation of the data in the latent code (Berahmand et al., 2024).
The reconstruction error is calculated in terms of Mean Squared Error (MSE)(Michelucci, 2022):

$$L_{MSE} = MSE = \frac{1}{M}\sum_{i=1}^{M} |x_i - \hat{x_i}|^2, \qquad (\mathbf{2})$$

Where the symbol $|\cdot|$ indicates the norm of the vector which is the difference between the true value x_i and predicted value $\hat{x_i}$ for the i-th observation in the dataset. M is the number of observations in the training dataset.

2.2 Results

For this study, map data from OSM were extracted through OSMnx and processed. The data were collected by selecting a bounding box defined by the minimum and maximum latitude and longitude of the Central Unit Athens region, with an additional margin of 0.03 degrees. This results in two key datasets: an edge dataset, consisting of 100914 edges, and a node dataset, containing 63236 nodes.

Telematics data were provided by OSeven Telematics, covering the last four months of 2024 within the previously defined area. After cleaning the dataset, this resulted in 10970875 per-second geospatial data points. The four-month collection period limits the ability to analyze seasonal effects, though weekly trends may still be examined.

The Python function sjoin_nearest() was used to identify the nearest edge for each telematics data point in the OSeven dataset. The nearest spatial join gives as output a dataset with more rows than the original dataset, as some data points are equidistant to more than one edge. Therefore, each point is associated with all of its nearest edges, reflecting all possible connections between data points and their nearest edges.
Under the assumption that the point has the same influence on all of its nearest edges, the dataset was left unchanged. This means that no differentiation was made among the multiple edges associated with the telematic data point, as its influence was assumed to be equal across all of them. Based on a unique key constructed from the OSM indexes in the edge database, the features were aggregated as presented in **Table 1** resulting in a dataset of 49908 edges characterized by telematics data.

The Python function sjoin was used to identify the telematics data points falling within a 50 meters buffer originating around each node of the graph. The output is a dataset in which each telematics data point may appear multiple times, depending on the number of buffers it falls within. The features were aggregated again as presented in **Table 1**. This approach yields a dataset of 34886 nodes, hereafter referred to as DF1.
By using the novel approach with the additional constraint previously described a second dataset of 31924 nodes is generated, hereafter referred to as DF2.

The non-telematics features were dropped from both DF and DF2, hence containing the same telematics features, shown below in **Table 1**:

Table 1: *Features in the Datasets*

Feature	Description
smoothenedSpeed	Average speed of the influencing points.
SpeedingFlag	Total count of influencing points flagged for speeding.
Mobile_usage	Total count of influencing points flagged for mobile phone usage.
Harsh_acc	Total count of influencing points flagged for harsh acceleration.
Harsh_brk	Total count of influencing points flagged for harsh braking.
Event_intensity	Average intensity of the harsh event (acceleration or braking).
Trips_count	Total number of unique trips among the influencing points.
Points_count	Total number of unique points among the influencing points.

DF1 and DF2 were evaluated in terms of standard deviation, the results are presented below in **Table 2**.

Table 2: *Comparative Standard Deviation of Features Across the Two Datasets*

Feature	Standard Deviation in DF1	Standard Deviation in DF2
smoothenedSpeed	11.76	10.26
SpeedingFlag	59.14	30.90
Mobile_usage	50.07	26.64
Harsh_acc	3.68	1.96
Harsh_brk	2.09	1.32
Event_intensity	0.81	0.75
Trips_count	108.79	54.80
Points_count	1291.68	657.37

A reduction in the standard deviation for each feature was achieved with the new approach, leading to less dispersed data (Dodge, 1999).

OSM provides the edge types in the form of column ‘highway’. The edge types were manually encoded into three categories: rural, urban, and service. The node type was determined based on the most frequent edge type among those connected to each node. This work was useful to understand the difference between the entire dataset, the Urban scenario and the Rural scenario within the study area, by simply filtering the original entire dataset. **Figure 4** shows the reduction obtained in terms of standard deviation per each dataset.

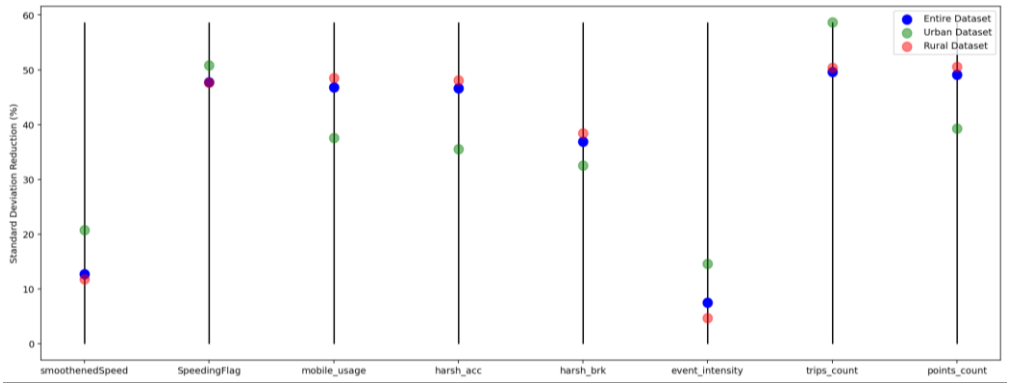

Figure 4: *Standard Deviation Reduction Across Different Scenarios.*

The chart above illustrates how the dispersion of the data with our approach decreased for every feature. However, smoothenedSpeed, SpeedingFlag, Event_intensity and Trips_count benefit more from the in the urban scenario, whereas the remaining features in the rural scenario.

PCA was used to understand how the data variance is distributed across the first 6 principal components. In **Figure 5** the results are presented.

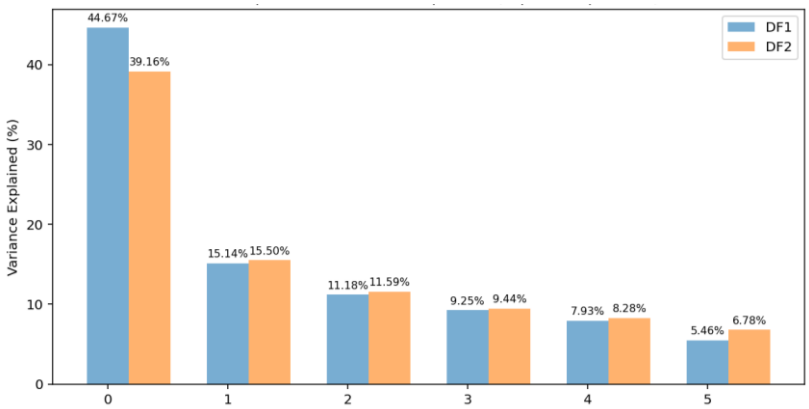

Figure 5: *PCA Explained Variance Comparison (Top 6 Components).*

The explained variance is more evenly distributed across the features for DF2, PCA suggests again DF2 being more homogeneous. Indeed, while for DF1 the first principal component (PC1) explains almost half of the variance in data, PC2, PC3, PC4, and PC5 contribute more evenly in DF2 than in DF1. This aligns with the idea that DF2 is more homogeneous.

PCA has similar effects to autoencoders in terms of dimensionality reduction, however the autoencoder is more flexible than the PCA (Li et al., 2023).
As previously mentioned, an autoencoder with the same architecture and hyperparameters was trained on both DF1 and DF2, to extract features and assess reconstruction quality. The architecture was tested with three different sizes of the encoding layer: 2, 4 and 6. This allows to assess better the reconstruction performance across varying levels of latent space dimensionality.

Following guidance from related literature (Bengio, 2012), the hyperparameters for the training process were fixed as follows: the learning rate for the Adam optimizer was set to 0.005, the batch size was 64, the number of training epochs was 200, and early stopping patience was set to 8.
Additionally, since the main goal is to reconstruct the input data, rather than on generalization to new data, the dataset was not split into training and test set. Instead, all the data were used during the training phase.
Below the results for the three different sizes of the encoding layer are presented:

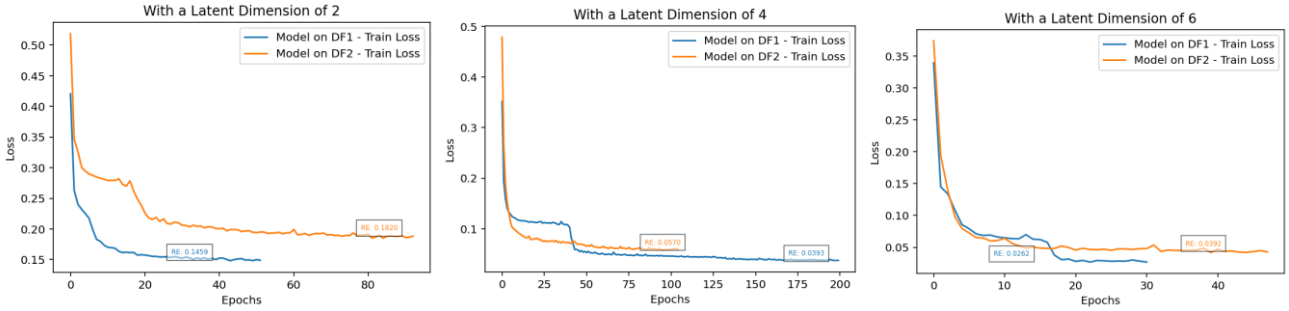

Figure 6: *Reconstruction Error of the Autoencoder across varying Encoding Dimensions.*

The autoencoder outperforms on DF1 across all values of the latent dimension, which represents the compressed input data capturing its essential information. Indeed, the error in the reconstruction between the input and the output is smaller on DF1 than on DF2, hence the encoding of the hidden layer is a better representation of the input data (Li et al., 2023), when DF1 is used as input for the autoencoder.
This indicates that DF1 is more suitable for reconstruction tasks, due to it featuring a simple pattern in the data.

Despite DF1 showing higher variance in the data than DF2, it appears to have more learnable patterns and simpler relationships among the features, which were also seen in the PCA results. On the other hand, DF2 could be simply more diverse and complex, resulting from a better presentation of the node characteristics which has not been skewed by non-influential data points and contains fewer redundant data points.
The novel approach captures real-world variability better, making it harder for the autoencoder to reconstruct it with the same level of precision, DF2 will most likely have more realistic, complex patterns.

3. Discussion

The present work proposes methods to merge telematics data, collected via a smartphone app, with spatial entities—specifically intersections and roads (referred to as nodes and edges in this study, following OSM terminology).
The aggregation for the roads is straightforward as it involves finding the nearest road per each telematic data point and then summarizing the data at road level. The approach is believed to ensure a reliable representation of the data since it aggregates data based on their proximity and contextual relevance.
The usual aggregation for the intersections is based on buffer zones, however this approach is too simplistic and it lacks the ability to represent accurately the intersection characteristics.
A novel approach has been proposed that builds on the simple buffer method by introducing an additional constraint for better representation. The simple and the novel approach naturally result in two distinct aggregated datasets, from which conclusions about the data were drawn using three different tools: standard deviation metrics, PCA, and an autoencoder. The tools shed light on the data variability, the data overdispersion and data complexity.
The dataset generated with the proposed approach, incorporating both a buffer and an additional constraint, exhibits less variability and lower overdispersion compared to the simpler buffer-based approach. Furthermore, it is more challenging to reconstruct using deep learning techniques, leading to the conclusion that it represents a more realistic dataset with more complex patterns.
These observations, together with the underlying assumption introduced by the additional constraint in the dataset construction, support the interpretation that the dataset more accurately reflects real-world driving behavior and spatial dynamics. It captures subtler relationships and interactions that are more difficult for the proposed models to approximate.

4. Conclusion

Combining telematics data with spatial entities represents a crucial step in spatial analysis, as it enables the contextualization of dynamic driving behavior within the network where it happens, unlocking deeper insights into mobility patterns and road safety.
The present work aims to show perspectives about the aggregation of telematics data to spatial entities, trying to move beyond simplistic methods by utilizing a more structured framework that represent traffic flow within the real-world environment in a refined way. When making key observations on the aggregation of telematics to roads and intersections, the aggregation to the intersections was evaluated using statistical and machine learning tools to assess the effectiveness of the novel 'buffer+constraint' approach compared to the simpler 'buffer' method.
The tools revealed the new approach generate a dataset with lower variance, a more complex structure, and more challenging to reconstruct using deep learning methods, leading to the conclusion that it more faithfully represents real-world conditions.
The limitations might primarily stem from the level of reasoning and detail presented. Future work can improve upon these ideas, considering additional perspectives and methodologies, or maybe expanding the dataset, particularly by extending the temporal coverage to enable seasonal analysis. Particularly, more advanced deep learning techniques such as Variational Autoencoder (Kingma & Welling, 2022) might be used or other useful techniques presented by (Li et al., 2023) could offer a deeper understanding of the data's nature when aggregated, with more robust findings.
Lastly, the entire work is based on the buffer distance which was chosen to be 50 meters for the analysis, however exploring more distances could offer more insights.

5. Acknowledgements

This research is based on work carried out within the IVORY project. The project has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101119590.

6. References-Bibliography

Bengio, Y. (2012). Practical Recommendations for Gradient-Based Training of Deep Architectures. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural Networks: Tricks of the Trade: Second Edition* (pp. 437–478). Springer. https://doi.org/10.1007/978-3-642-35289-8_26

Berahmand, K., Daneshfar, F., Salehi, E. S., Li, Y., & Xu, Y. (2024). Autoencoders and their applications in machine learning: A survey. *Artificial Intelligence Review*, *57*(2), 28. https://doi.org/10.1007/s10462-023-10662-6

Boeing, G. (2017). OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, *65*, 126–139. https://doi.org/10.1016/j.compenvurbsys.2017.05.004

Boeing, G. (2024). *Graph Simplification Solutions to the Street Intersection Miscount Problem* (No. arXiv:2407.00258). arXiv. https://doi.org/10.48550/arXiv.2407.00258

Boylan, J., Meyer, D., & Chen, W. S. (2024). A systematic review of the use of in-vehicle telematics in monitoring driving behaviours. *Accident Analysis & Prevention*, *199*, 107519. https://doi.org/10.1016/j.aap.2024.107519

Chen, S., & Guo, W. (2023). Auto-Encoders in Deep Learning—A Review with New Perspectives. *Mathematics*, *11*(8), Article 8. https://doi.org/10.3390/math11081777

Dodge, Y. (1999). *Premiers pas en statistique*. Springer Paris.

Erramaline, A., Badard, T., Côté, M.-P., Duchesne, T., & Mercier, O. (2022). Identification of Road Network Intersection Types from Vehicle Telemetry Data Using a Convolutional Neural Network. *ISPRS International Journal of Geo-Information*, *11*(9), Article 9. https://doi.org/10.3390/ijgi11090475

GeoPandas. (2025). https://geopandas.org/en/stable/docs.html

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Kingma, D. P., & Welling, M. (2022). *Auto-Encoding Variational Bayes* (No. arXiv:1312.6114). arXiv. https://doi.org/10.48550/arXiv.1312.6114

Kontaxi, A., Ziakopoulos, A., Katrakazas, C., & Yannis, G. (2022). *Measuring the impact of driver behavior telematics in road safety*.

Kostopoulos, A., Garefalakis, T., Michelaraki, E., Katrakazas, C., & Yannis, G. (2024). Modeling and Sustainability Implications of Harsh Driving Events: A Predictive Machine Learning Approach. *Sustainability*, *16*(14), Article 14. https://doi.org/10.3390/su16146151

Li, P., Pei, Y., & Li, J. (2023). A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, *138*, 110176. https://doi.org/10.1016/j.asoc.2023.110176

Michelucci, U. (2022). *An Introduction to Autoencoders* (No. arXiv:2201.03898). arXiv. https://doi.org/10.48550/arXiv.2201.03898

Nikolaou, D., Dragomanovits, A., Ziakopoulos, A., Deliali, A., Handanos, I., Karadimas, C., Kostoulas, G., Frantzola, E. K., & Yannis, G. (2023). Exploiting Surrogate Safety Measures and Road Design Characteristics towards Crash Investigations in Motorway Segments. *Infrastructures*, *8*(3), Article 3. https://doi.org/10.3390/infrastructures8030040

Nikolaou, D., Ziakopoulos, A., Kontaxi, A., Theofilatos, A., & Yannis, G. (2025). Spatial analysis of telematics-based surrogate safety measures. *Journal of Safety Research*, *92*, 98–108. https://doi.org/10.1016/j.jsr.2024.09.012

Nikolaou, D., Ziakopoulos, A., & Yannis, G. (2023). A Review of Surrogate Safety Measures Uses in Historical Crash Investigations. *Sustainability*, *15*(9), Article 9. https://doi.org/10.3390/su15097580

OpenStreetMap. (2025). *About OpenStreetMap—OpenStreetMap Wiki*. https://wiki.openstreetmap.org/wiki/About_OpenStreetMap

OSeven. (2025). *Oseven.io*. Oseven.Io. https://oseven.io/

Petraki, V., Ziakopoulos, A., & Yannis, G. (2020). Combined impact of road and traffic characteristic on driver behavior using smartphone sensor data. *Accident Analysis & Prevention*, *144*, 105657. https://doi.org/10.1016/j.aap.2020.105657

Road safety in the EU: Fatalities below pre-pandemic levels but progress remains too slow - European Commission. (2023). https://transport.ec.europa.eu/news-events/news/road-safety-eu-fatalities-below-pre-pandemic-levels-progress-remains-too-slow-2023-02-21_en

Shlens, J. (2014). *A Tutorial on Principal Component Analysis* (No. arXiv:1404.1100). arXiv. https://doi.org/10.48550/arXiv.1404.1100

Silva, P. B., Andrade, M., & Ferreira, S. (2020). Machine learning applied to road safety modeling: A systematic literature review. *Journal of Traffic and Transportation Engineering (English Edition)*, *7*(6), 775–790. https://doi.org/10.1016/j.jtte.2020.07.004

Stipancic, J., Miranda-Moreno, L., Saunier, N., & Labbe, A. (2019). Network screening for large urban road networks: Using GPS data and surrogate measures to model crash frequency and severity. *Accident Analysis & Prevention*, *125*, 290–301. https://doi.org/10.1016/j.aap.2019.02.016

Stipancic, J., Racine, E. B., Labbe, A., Saunier, N., & Miranda-Moreno, L. (2021). Massive GNSS data for road safety analysis: Comparing crash models for several Canadian cities and data

sources. *Accident Analysis & Prevention*, *159*, 106232. https://doi.org/10.1016/j.aap.2021.106232

Sze, V., Chen, Y.-H., Yang, T.-J., & Emer, J. S. (2017). Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proceedings of the IEEE*, *105*(12), 2295–2329. https://doi.org/10.1109/JPROC.2017.2761740

Tarko, A. P. (2018). Chapter 17. Surrogate Measures of Safety. In D. Lord & S. Washington (Eds.), *Transport and Sustainability* (Vol. 11, pp. 383–405). Emerald Publishing Limited. https://doi.org/10.1108/S2044-994120180000011019

Tselentis, D. I., Theofilatos, A., Yannis, G., & Konstantinopoulos, M. (2018). Public opinion on usage-based motor insurance schemes: A stated preference approach. *Travel Behaviour and Society*, *11*, 111–118. https://doi.org/10.1016/j.tbs.2018.02.003

WHO. (2023). *Global status report on road safety 2023*. https://www.who.int/publications/i/item/9789240086517

Ziakopoulos, A. (2024). Analysis of harsh braking and harsh acceleration occurrence via explainable imbalanced machine learning using high-resolution smartphone telematics and traffic data. *Accident Analysis & Prevention*, *207*, 107743. https://doi.org/10.1016/j.aap.2024.107743

Ziakopoulos, A., Petraki, V., Kontaxi, A., & Yannis, G. (2022). The transformation of the insurance industry and road safety by driver safety behaviour telematics. *Case Studies on Transport Policy*, *10*(4), 2271–2279. https://doi.org/10.1016/j.cstp.2022.10.011

Ziakopoulos, A., & Yannis, G. (2020). A review of spatial approaches in road safety. *Accident Analysis & Prevention*, *135*, 105323. https://doi.org/10.1016/j.aap.2019.105323