The use of Graph Neural Networks for Clustering in Road Safety Analysis

Simone Paradiso*[1], Apostolos Ziakopoulos[1], George Yannis[1]

[1] 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)

SHORT SUMMARY

Road safety analysis aims to reduce traffic crashes and enhance transportation systems' efficiency. Graph theory, and its related mathematical framework, offers tools to model complex road networks and analyze various dynamics such as traffic flow, routing, crashes correlation, etc. By representing road networks as graphs and integrating telematics data onto the graph, it becomes possible to capture behaviors on the network. Graph Neural Networks (GNNs) provide a robust framework for analyzing such data and enabling network-level inference. The study presents an approach to identify node clusters in a network using edge features. Specifically, selecting an area within the Athens metropolitan area, it compares a clustering on node features with a clustering based on node embeddings generated by a GNN that incorporates the attention mechanism. By leveraging this mechanism, edges features are not left out from the analysis and the clustering algorithm shows better overall performance in terms of cluster quality.

Keywords: Attention Mechanism, Clustering, Graph Neural Networks, Intersections.

1. INTRODUCTION

Road crashes are the leading cause of death of children and youth, and they typically strike during our most productive years, causing huge health, social and economic harm. Since 2010, deaths from road crashes have fallen slightly to 1.19 million per year (WHO, 2023).
Econometrics models have been widely used to predict crashes and their severity, with the advent of big data, the use of Machine Learning (ML) and Artificial Intelligence (AI) have been rapidly increased (Ziakopoulos & Yannis, 2020). ML techniques such as Classification and Regression Trees (CART) or Support Vector Machines (SVM), and Deep Learning (DL) techniques such as Artificial Neural Networks (ANNs) are very common in road safety field as shown in (Silva et al., 2020).
The ANNs' architecture has evolved through the years leading to develop more sophisticated models as Convolutional Neural Network (O'Shea & Nash, 2015) used mostly in computer vision or Recurrent Neural Networks primarily used for sequential data (Salehinejad et al., 2018). Both have been used and continue to be applied in the road safety context (Mowen et al., 2022), (Yuan et al., 2019). However, since many relationships in science and engineering can be represented as graphs, researchers developed the GNNs, extending neural network methods to graph-structured data (Scarselli et al., 2009).
The present study aims to explore the use of GNNs to analyze infrastructure data on a road network. Using a GNN model, a different data representation was obtained and analyzed, with results compared to the analysis conducted on the raw data. Employing GNNs generates node representations from a graph that capture their structural or relational properties. This process, known as embedding, helps ML models process the nodes more effectively for tasks like classification or clustering (Xu, 2021).
Indeed, a clustering algorithm was applied to the node embeddings to identify partitions of node characteristics and compared with a simpler clustering approach using raw node features, thereby demonstrating how the embedding improves clustering performance.

2. METHODOLOGY

The methodological approach is based on analyzing telematics data coming from a smartphone application developed by (OSeven), that records driver behavior using smartphone hardware sensors. The dataset consists of anonymous trips in compliance with standing Greek and European personal data protection legislation (GDPR), with trip coordinates and periodic flags (1 or 0) indicating the presence of harsh events or speeding. This dataset defines a coordinate bounding box used to extract geometry features from OpenStreetMap (OSM) (OpenStreetMap), a free, editable global map created by volunteers and released under an open-content license.
From the defined graph in OSM, node and edge features have been stored in two different datasets, with the considered nodes being "true" edge endpoints (i.e., intersections or dead-ends) (Boeing, 2024). The telematics data have been aggregated to the OSM data by summing or averaging, depending on the feature's nature, over all the points falling within a 50 meters buffer, coherent with precedent works in the field (Erramaline et al., 2022).
Regarding the edges, the closest edge to each single point in the raw OSeven dataset was identified and then the values, per edge, have been aggregated as done for the nodes. The number of trips per node and per edge have been calculated as well. It is worth mentioning that not all edges and nodes have telematics data, as it depends on the trip paths.
At this stage, geometric and telematics data for each edge in the network, including features for both its initial and final nodes, have been collected.
The research approach involves clustering the nodes, hence the endpoints of the network.

Clustering is a type of unsupervised ML technique that groups objects into clusters, such that objects within each cluster are more similar to each other than to those in other clusters. The present research implements the K-Means algorithm, described in detail in (Steinley, 2006), due to its efficiency and because it involves examining the similarity of points within a cluster relative to all other points in the same cluster (Steinbach et al., 2000).

The K-means algorithm requires **K**, the number of clusters, as input. Common methods to determine the optimal K include the elbow method and the silhouette score. The elbow curve plots the Within-Cluster Sum-of-Squares (WCSS), or inertia, which measures the variance within each cluster, against a range of K values (Cui, 2020).
The silhouette score (Shahapure & Nicholas, 2020), ranging from -1 to 1, is calculated for each data point in the dataset. A score near to +1 means the data point is correctly clustered, near to 0 suggests potential overlap, and near to -1 indicates wrong cluster.
The average score provides information about whether the clusters are well-separated or overlapping.

K-Means has been applied to the raw features of the nodes and to the embeddings found by the means of a GNN, which is an extension of existing neural network methods suitable for data represented in graph domains. The two clustering approaches are compared using the WCSS and Silhouette Index calculated for the chosen value of K. Additionally, the Davies-Bouldin Index (DBI) (Davies & Bouldin, 1979) and the Calinski-Harabasz Index (CHI) (Calinski & Harabasz, 1974) were used to provide further insights and comparative evaluation of the two methods.
GNNs aim to encode the underlying graph structured data using the topological relationships among the nodes, instead of squashes the graph data into vectors (Scarselli et al., 2009). There has been an evolution of GNN which led the academic world to models such as Graph Convolution Networks (GCN) which involves the convolution operation (Kipf & Welling, 2017) or Graph Attention Networks (GAT) introduced by (Veličković et al., 2018). The latter was considered for this study to obtain the embeddings. The GAT model is a novel neural network architecture leveraging attention mechanisms.
A simple neural network with two GAT layers has been defined following the official related documentation of PyTorch library (GATConv, 2024).
A GAT layer operates with self-attention:

$$\boldsymbol{x}_i' = \alpha_{ii}\boldsymbol{\Theta}_s\boldsymbol{x}_i + \sum_{j\in N_i} \alpha_{ij}\,\boldsymbol{\Theta}_t\boldsymbol{x}_j$$

Where:

- N_i: Neighbors of node i.
- α_{ij}: Learnable attention coefficient between nodes i and j, involving edge features, computed as:

$$\alpha_{ij} = \frac{e^{(LeakyReLU(\boldsymbol{a}_s^T\boldsymbol{\Theta}_s\boldsymbol{x}_i + \boldsymbol{a}_t^T\boldsymbol{\Theta}_t\boldsymbol{x}_j + \boldsymbol{a}_e^T\boldsymbol{\Theta}_e\boldsymbol{e}_{i,j}))}}{\sum_{k\in N_i\cup i} e^{(LeakyReLU(\boldsymbol{a}_s^T\boldsymbol{\Theta}_s\boldsymbol{x}_i + \boldsymbol{a}_t^T\boldsymbol{\Theta}_t\boldsymbol{x}_k + \boldsymbol{a}_e^T\boldsymbol{\Theta}_e\boldsymbol{e}_{i,k}))}}$$

- $\boldsymbol{a}_s^T, \boldsymbol{a}_t^T, \boldsymbol{a}_e^T$ are learnable vectors defining attention scores.
- $\boldsymbol{\Theta}_s, \boldsymbol{\Theta}_t, \boldsymbol{\Theta}_e$ are learnable weight matrices that working on the node features $\boldsymbol{x}_i$, $\boldsymbol{x}_j$ and the edge features $\boldsymbol{e}_{i,j}$.

Thanks to the attention mechanism each neighbor of a node gets a different weight (α_{ij}), based on the importance of their features.

The model was trained in a self-supervised way to obtain the embeddings, through a contrastive loss function, inspired by prior work in self-supervised learning such as the ones from (Chen et al., 2020) and (Oord et al., 2019).
For each node, the function calculates the cosine similarity between the node's embedding and those of its neighbors, as well as between the same node's embedding and the embeddings of randomly sampled non-neighbors.
Once the two similarity values are computed, they are passed through the exponential function, after having been scaled by a temperature parameter controlling the sharpness of the function. The values are summed respectively over the neighbors and over the non-neighbors and the ratio between the first sum and the sum of the two calculated sums is then passed through the logarithm function. The average over all the nodes is finally computed to obtain the final loss.
Across 10 epochs, the training data was split into subgraphs using the PyG NeighborLoader and trained using the defined function above with the Adam optimizer to calculate total loss.

3. RESULTS AND DISCUSSION

The area covered by telematics data is mostly the north-east of the Athens metropolitan area. After aggregating telematics data per node and per edge, K-means was applied to the node features, shown in Table 1:

Table 1: Node features

Features	Description
Street_Count	Number of streets connected to the intersection
SmoothenedSpeed	Average speed of vehicles near the node
SpeedingFlag	Count of speeding events near the node
Mobile_usage	Number of instances of phone usage near the node
Harsh_acc	Number of harsh acceleration events near the node
Harsh_brk	Number of harsh braking events near the node
Trips_count	Number of trips recorded near the node

To choose K, the elbow curve and silhouette score for different values of K were plotted.
The analysis of the elbow curve on the node features does not have a clear inflection point preventing straightforward decisions.

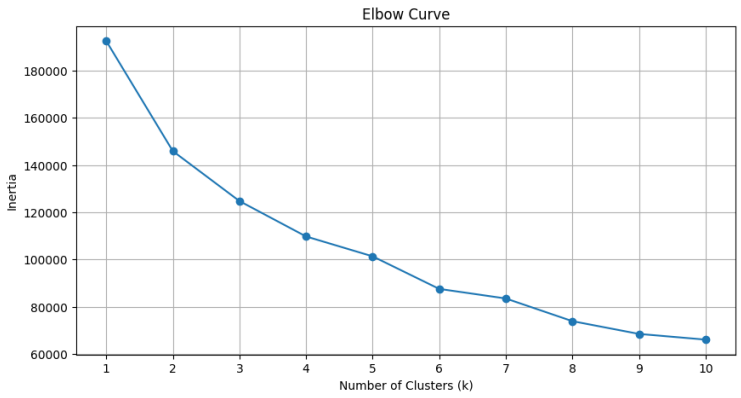

Figure 1: Elbow curve on node features.

The Silhouette score calculated for different K values shows the highest score at K=2, for a score of 0.58.

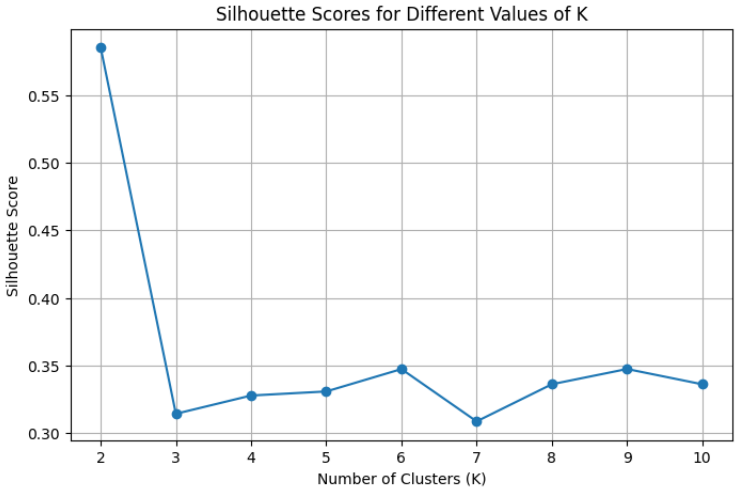

Figure 2: Silhouette score for different K on node features.

Based on the silhouette, K was set to 2 to obtain well-separated clusters, yielding an inertia of 145909, compared to 192556 for K=1, representing the dataset as a single cluster. This corresponds to a reduction in inertia of approximately 20%, which is quite considerable. After the algorithmic execution, results are shown below in Figure 3.

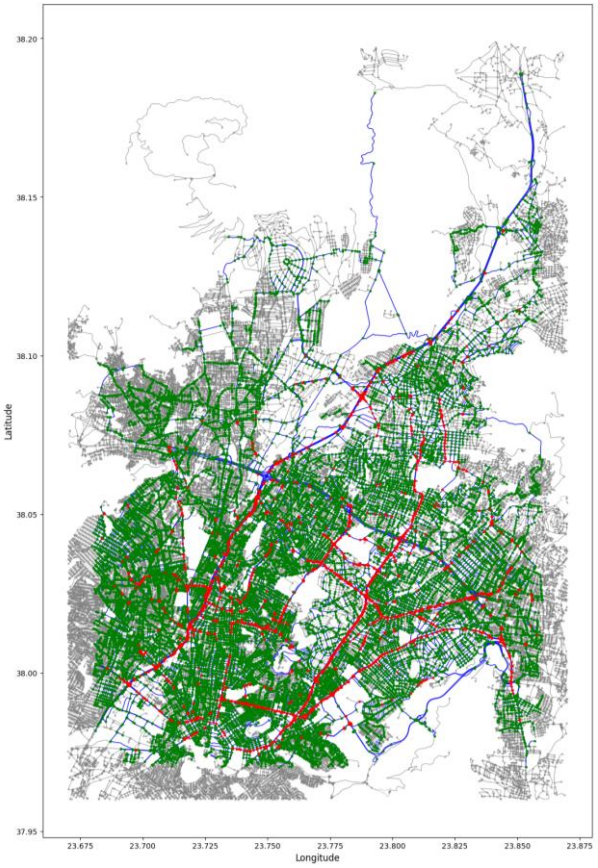

Figure 3: Node clusters.

Grey edges and nodes are entities for which telematics was not available. Blue edges are edges with associated telematics, green and red nodes are nodes with associated telematics divided by color referring to the two different clusters.

At this stage, the GNN was used to involve the features of the edges connected to each node, besides the road network topology, employing multi-head attention mechanisms instead of a single attention head. The edge features are the same listed in Table 1, excluding street_count and supplemented by four additional features:

- Edge length.
- Two binary features were derived via one-hot encoding from a three-category variable indicating road type (service, urban, rural).
- A binary oneway column shows if vehicles can go in only one direction or in both directions.

The elbow curve and silhouette score were analyzed first, and based on these results, K-Means was applied to the embeddings generated by the GNN.
As in the previous paragraph, the elbow curve does not have a clear elbow, but the scale of the inertia is significantly reduced, as displayed in Figure 4.

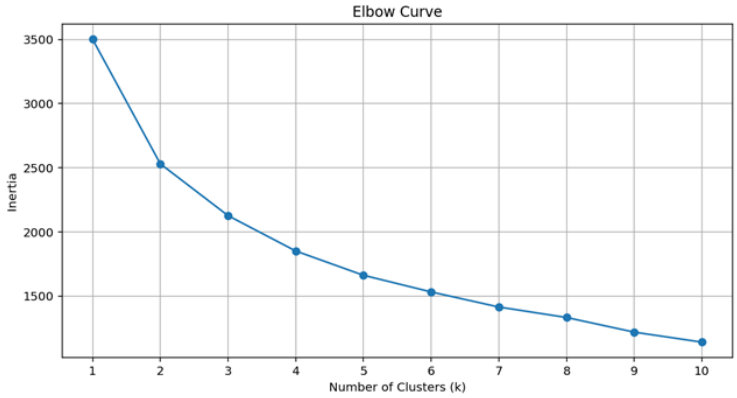

Figure 4: Elbow curve on node embedding features.

The average silhouette score calculated for several K also exhibits a better behavior in the partition of the data. At K=2 there is a silhouette peak, as in the previous paragraph.

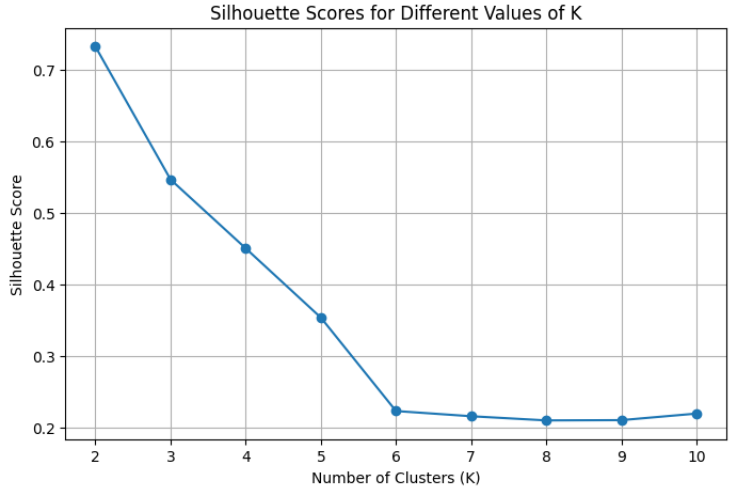

Figure 5: Silhouette score for different K on node embedding features.

The choice of K was once again set to 2, for a silhouette score of 0.73 and a clustering inertia of 2530, whereas the inertia for K=1 is 3501, corresponding to a reduction in inertia of 28%.
The inertia of the last approach is an order of magnitude lower compared to the previous clustering, indicating more coherent and tightly packed groups of elements.

Figure 6 shows the clusters identified after obtaining the node embedding and color considerations from the previous paragraph are also applied here.

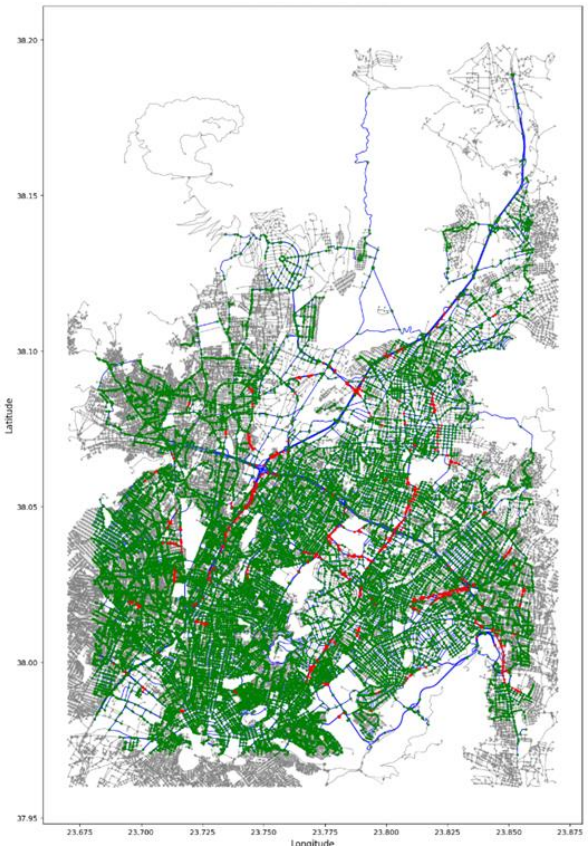

Figure 6: Node embedding clusters.

The groups identified by both clustering methods exhibit similar characteristics upon comparison. Green nodes are associated with safer behaviors, whereas the red ones are characterized by less safe events but also a higher number of associated trips. However, the clustering based on the embeddings tends to polarize the nodes, shrinking the less safe cluster while expanding the other one.

The cluster sizes are presented in Table 2 for both clustering approaches: the first applied directly to the raw, scaled features of the dataset, referred to as *Simple Clustering*; and the second performed on the embeddings generated by the GNN, referred to as *Embeddings Clustering*.

Table 2: Clustering differences

Clustering type	Labels	Node numbers
Simple Clustering	Safer nodes	24776
Simple Clustering	Riskier nodes	2732
Embeddings Clustering	Safer nodes	26587
Embeddings Clustering	Riskier nodes	921

Furthermore, the clustering performance was evaluated using the WCSS and the Silhouette Index for K=2, as well as the DBI and the CHI for K=2. The numerical comparison of these metrics is presented in Table 3.

Table 3: Clustering Performance

Index	Simple Clustering value	Embeddings Clustering value
WCSS	145909	31409
Silhouette	0.58	0.73
DBI	1.41	0.98
CHI	8794	10551

The WCSS provides insights into the compactness of the clusters, indicating that the embeddings improve intra-cluster cohesion. The Silhouette Index assesses the quality of clustering at the point level, focusing on cohesion and separation for each individual point and then averaging them. The DBI is a centroid-based method that compares the distances of points within each cluster to the cluster's centroid, as well as the distances between the centroids of different clusters. The CHI measures the dispersion of the clusters relative to the overall centroid and the spread of the points within each cluster.
Focusing on different aspects of clustering quality, the indices offer a holistic view of the clustering performance, collectively suggesting superior overall performance of the embeddings-based approach.

4. CONCLUSIONS

The current work proposes a novel approach for understanding the nature of data structured in graphs. While simple clustering algorithms have been used in road safety (Nitsche et al., 2017), this framework demonstrates the advantages of exploiting graph-based representations for explorative analysis. Using a simple clustering algorithm as baseline, it has been shown how node representations based on features, network topology where the nodes fall and edges characteristics connected to the node, can improve clustering performance compared to the baseline. The study focuses on K-Means due to its efficiency, but other clustering algorithms might be explored. Additionally, investigating different GNNs architecture might provide further insights into graph partitioning, as well as exploring different loss functions, related or not to the contrastive ones. Incorporating traffic and temporal features may further enhance the effectiveness of the approach in real-world applications.

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.

REFERENCES

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

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics - Theory and Methods*, *3*(1), 1–27. https://doi.org/10.1080/03610927408827101

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). *A Simple Framework for Contrastive Learning of Visual Representations* (No. arXiv:2002.05709). arXiv. https://doi.org/10.48550/arXiv.2002.05709

Cui, M. (2020). Introduction to the K-Means Clustering Algorithm Based on the Elbow Method. *Accounting, Auditing and Finance*, *1*(1), 5–8. https://doi.org/10.23977/accaf.2020.010102

Davies, D. L., & Bouldin, D. W. (1979). A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-1*(2), 224–227. IEEE Transactions on Pattern Analysis and Machine Intelligence. https://doi.org/10.1109/TPAMI.1979.4766909

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

GATConv. (2024). *torch_geometric.nn.conv.GATConv—Pytorch_geometric documentation*. https://pytorch-geometric.readthedocs.io/en/2.5.3/generated/torch_geometric.nn.conv.GATConv.html

Kipf, T. N., & Welling, M. (2017). *Semi-Supervised Classification with Graph Convolutional Networks* (No. arXiv:1609.02907). arXiv. https://doi.org/10.48550/arXiv.1609.02907

Mowen, D., Munian, Y., & Alamaniotis, M. (2022). Improving Road Safety during Nocturnal Hours by Characterizing Animal Poses Utilizing CNN-Based Analysis of Thermal Images. *Sustainability*, *14*(19), Article 19. https://doi.org/10.3390/su141912133

Nitsche, P., Thomas, P., Stuetz, R., & Welsh, R. (2017). Pre-crash scenarios at road junctions: A clustering method for car crash data. *Accident Analysis & Prevention*, *107*, 137–151. https://doi.org/10.1016/j.aap.2017.07.011

Oord, A. van den, Li, Y., & Vinyals, O. (2019). *Representation Learning with Contrastive Predictive Coding* (No. arXiv:1807.03748). arXiv. https://doi.org/10.48550/arXiv.1807.03748

OpenStreetMap. *About OpenStreetMap—OpenStreetMap Wiki*. Retrieved December 27, 2024, from https://wiki.openstreetmap.org/wiki/About_OpenStreetMap

OSeven. *Oseven.io*. Retrieved December 27, 2024, from https://oseven.io/

O'Shea, K., & Nash, R. (2015). *An Introduction to Convolutional Neural Networks* (No. arXiv:1511.08458). arXiv. https://doi.org/10.48550/arXiv.1511.08458

Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2018). *Recent Advances in Recurrent Neural Networks* (No. arXiv:1801.01078). arXiv. https://doi.org/10.48550/arXiv.1801.01078

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The Graph Neural Network Model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. IEEE Transactions on Neural Networks. https://doi.org/10.1109/TNN.2008.2005605

Shahapure, K. R., & Nicholas, C. (2020). Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, 747–748. https://doi.org/10.1109/DSAA49011.2020.00096

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

Steinbach, M., Karypis, G., & Kumar, V. (2000). *A Comparison of Document Clustering Techniques*.

Steinley, Douglas. (2006). K-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology*, *59*(1), 1–34. https://doi.org/10.1348/000711005X48266

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). *Graph Attention Networks* (No. arXiv:1710.10903). arXiv. https://doi.org/10.48550/arXiv.1710.10903

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

Xu, M. (2021). Understanding Graph Embedding Methods and Their Applications. *SIAM Review*, *63*(4), 825–853. https://doi.org/10.1137/20M1386062

Yuan, J., Abdel-Aty, M., Gong, Y., & Cai, Q. (2019). Real-Time Crash Risk Prediction using Long Short-Term Memory Recurrent Neural Network. *Transportation Research Record*, *2673*(4), 314–326. https://doi.org/10.1177/0361198119840611

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