Improving driver safety and trip quality through real-time anomaly detection with multi-model learning in taxi services

Nesar Ahmad Bahin[1,†], Rahim Ahmed Druba[1,*,†], Gulzat Turken[1,†] and Vladislav Karyukin[1,†]

[1] *Al-Farabi Kazakh National University, Al-Farabi Avenue 71, 050040, Almaty, Kazakhstan*

Abstract

This study presents a system for Real-time anomaly detection and trip rating in taxi driving. It uses vehicle telemetry data such as speed, acceleration, yaw rate, steering angle, and GPS. Unlike computer vision-based or infrastructure-heavy methods, it only relies on onboard sensors. This makes the system scalable and low cost. To test it, we built a synthetic dataset in Python using latitude and longitude from routes between cities in Kazakhstan. About 10% of the data had anomalies such as sudden braking, speeding, unsafe turns, and route changes. We used four models: Random Forest, Extra Trees, Gradient Boosting, and Multi-Layer Perceptron. With soft voting, the models reached high accuracy. Precision, recall, and F1-scores were close to the best across all anomaly types. A regression model was also used to rate trip quality. It achieved an R^2 of 0.996 and a mean absolute error of 0.043. This result shows it can capture how unsafe driving affects passenger experience. Results showed it works well across different traffic and environmental conditions. The trip rating model gives each trip a score between 1 and 10. The score is based only on telemetry data, so it avoids passenger bias and manual scoring. This allows fair and consistent safety evaluation and helps fleet operators monitor and improve service quality.

Keywords

real-time anomaly detection, telematics data, ensemble machine learning, trip quality rating[1]

1. Introduction

At present, the growth of taxi and ride-hailing services has increased the need for safe and reliable trips. Driver safety and trip quality are important for customer satisfaction, fleet efficiency, and meeting regulations. Traditional monitoring methods, such as manual checks or reports after incidents, are often too slow and limited for large fleets.

Recent advances in vehicle data collection create new opportunities for Real-time monitoring. Telemetry features, including speed, acceleration, yaw rate, and GPS, can reveal how a vehicle is being driven. Yet many existing detection methods have limits. They are hard to scale, depend on cameras or Internet of Vehicles systems, and can be affected by changes in traffic or the environment.

This study proposes a scalable system that uses telemetry data and ensemble machine learning to detect unsafe driving. The system identifies risky actions, such as harsh braking, sudden stops, acceleration, speeding, tight turns, sudden turns, and route changes, without relying on cameras or heavy infrastructure. Several models, including Random Forest, Extra Trees, Gradient Boosting, and Multi-Layer Perceptron, are combined with soft voting to improve accuracy and reliability.

To train and test the system, a simulated urban dataset was created. It includes normal trips along with injected unsafe behaviors under different traffic and road conditions. This setup allows the models to learn the difference between safe and unsafe driving. The final goal or motive is to connect the system to a fleet dashboard that gives operators Real-time alerts and insights. This research

[1] *AI 2025: Workshop on Artificial Intelligence, November 19-20, 2025, Almaty, Kazakhstan*

* Corresponding author.

† These authors contributed equally.

nesarahmadbahin1@gmail.com (N. A. Bahin); rahimdruba125@gmail.com (R. A. Druba); turken.gulzat@gmail.com (G. Turken); vladislav.karyukin@gmail.com (V. Karyukin)

0009-0006-1357-5892 (N. A. Bahin); 0009-0001-2177-9960 (R. A. Druba); 0000-0003-4981-514X (G. Turken); 0000-0002-8768-0349 (V. Karyukin)

presents a practical and scalable framework for continuous monitoring of driver behavior in taxi fleets. It supports safer driving and better passenger experiences through ensemble learning methods applied to vehicle telemetry data.

2. Related works

Prior studies have primarily emphasized computer vision–based driver monitoring. For instance, Hassan et al. [1] employed Vision Transformer (ViT) and Swin Transformer architectures to detect drowsiness using datasets such as MRL, NTHU-DDD, and CEW, achieving over 99% accuracy. Similarly, Gupta et al. [2] developed CPS-based ITS models using deep learning to recognize facial behaviors like eye closure and yawning, reporting accuracies of around 94%. While effective, these approaches rely on visual inputs, which makes them sensitive to lighting conditions, occlusions, and privacy concerns. Hu et al. [3] proposed a real-time taxi anomaly detection framework (TAPS) based on vehicle trajectory prediction, which identifies anomalous driving routes by comparing predicted and actual trajectories in real time.

Beyond vision, clustering-based methods have been used for general pattern recognition in complex data and later applied to analyze and group driver behaviors. Costa et al. [4] explored hierarchical clustering methods for pattern recognition in complex data. Zhao et al. [5] proposed a clustering-based framework to analyze individual travel mode choice behavior using GPS trajectories, showing how hierarchical methods reveal behavioral dynamics. Moavinis et al. [6] worked on detecting unusual driving paths using DBSCAN clustering with Hausdorff distance and an SVM classifier, tested on real data from Beijing and Cyprus with good accuracy. Liu et al. [7] introduced GeoDMA, a deep auto-encoder model combined with geographical partitioning to detect abnormal driving maneuvers from GPS data, capturing both temporal and spatial dependencies to improve detection accuracy. Collectively, these works demonstrate the usefulness of clustering for grouping driving behaviors, but they are not designed for Real-time anomaly detection in fleet environments.

Complementary advances include IoV-based driver identification. Gheni et al. [8] integrated CNNs with multi-head attention within an Internet of Vehicles framework, reporting a 99.95% accuracy in identifying drivers, though requiring heavy IoT-cloud infrastructure. Likewise, Fu et al. [9] analyzed GPS traces from 5,700 taxis to detect speeding behaviors, employing spatial autocorrelation measures such as Moran's I to reveal clustering patterns of risky driving events. Another study [10] focused on detecting unusual ride-hailing drivers using deep transfer inverse reinforcement learning, combining transfer learning and inverse reinforcement learning to model each driver's habits and identify abnormal behavior even with limited driving data.

However, most existing methods either rely on visual inputs or need complex infrastructure, which makes them hard to use in large fleet systems. Camera-based systems have limits due to lighting, occlusions, and privacy issues, while IoV-heavy methods need costly infrastructure. There is still a gap in building lightweight, real-time models that use telematics data to detect unsafe driving. Our work fills this gap by combining telematics features like speed, acceleration, yaw rate, and GPS with ensemble learning to find different types of anomalies, including route deviations. Unlike past studies that stop at detection, our framework also includes a trip rating model that gives each trip a safety score based only on telemetry. This makes it possible to evaluate trips objectively and provide consistent feedback for fleet monitoring.

3. Proposed methodology

3.1. Dataset description

A synthetic dataset was created in Python using latitude and longitude from routes between cities in Kazakhstan in GeoJSON format. It reflects real taxi driving under different traffic and environmental conditions. The dataset has spatial-temporal data (lat, lon, time) and kinematic variables (speed, acceleration, heading, yaw rate, steering angle, jerk).

External factors like traffic, road type, road condition, and time of day were added. Anomalies were injected logically, for example, sudden braking combines negative acceleration, speed drop, and jerk changes. Speeding was simulated by exceeding speed limits. Table 1 shows dataset statistics: central values, ranges, and notes for each feature. About 10% of events are anomalous. To show normal vs anomalous behavior, a 2D PCA projection was used (Figure 1).

Table 1
Statistical Summary of Driving Features

Feature	Mean (± Std)	Range	Notes
Speed (km/h)	58.7 ± 19.4	5 – 120	Adjusted by road, traffic, and time of day
Acceleration (m/s^2)	-0.12 ± 1.42	-9 – 6	Includes realistic braking and acceleration events
Yaw Rate (deg/s)	3.8 ± 6.2	0 – 60	Normalized with higher peaks in anomalies
Steering Angle (deg)	7.1 ± 6.8	0 – 50	Correlated with yaw rate and speed
Jerk (m/s^3)	0.03 ± 0.47	-5 – 4	Spikes during anomalies (e.g., sudden_brake)
Anomalous Events (%)	~10% of data points	Distributed almost evenly	Covers six anomaly categories

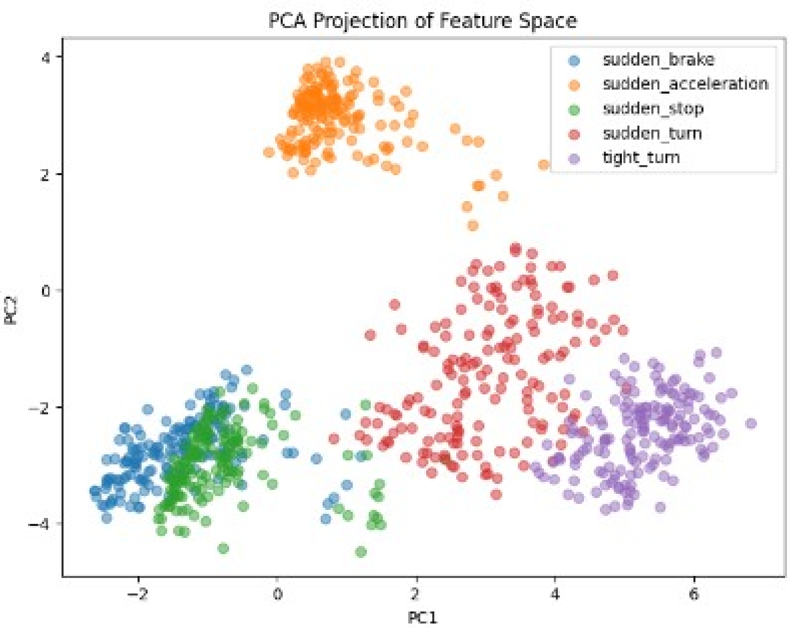

Figure 1: PCA Projection of five types of anomalies.

Figure **2** Below shows the correlation heatmap of driving and context features. Kinematic variables are strongly related, while behavioral events show moderate ties with continuous measures. The feature relationships reflect realistic driving patterns. Figure 2. Correlation heatmap among selected driving features. Columns include time, latitude, longitude, speed, speed_limit, acceleration, heading, yaw_rate, steering_angle, jerk, traffic_condition, road_condition, road_type, time_of_day, sudden_brake, sudden_acceleration, sudden_stop, sudden_turn, tight_turn, and speeding.

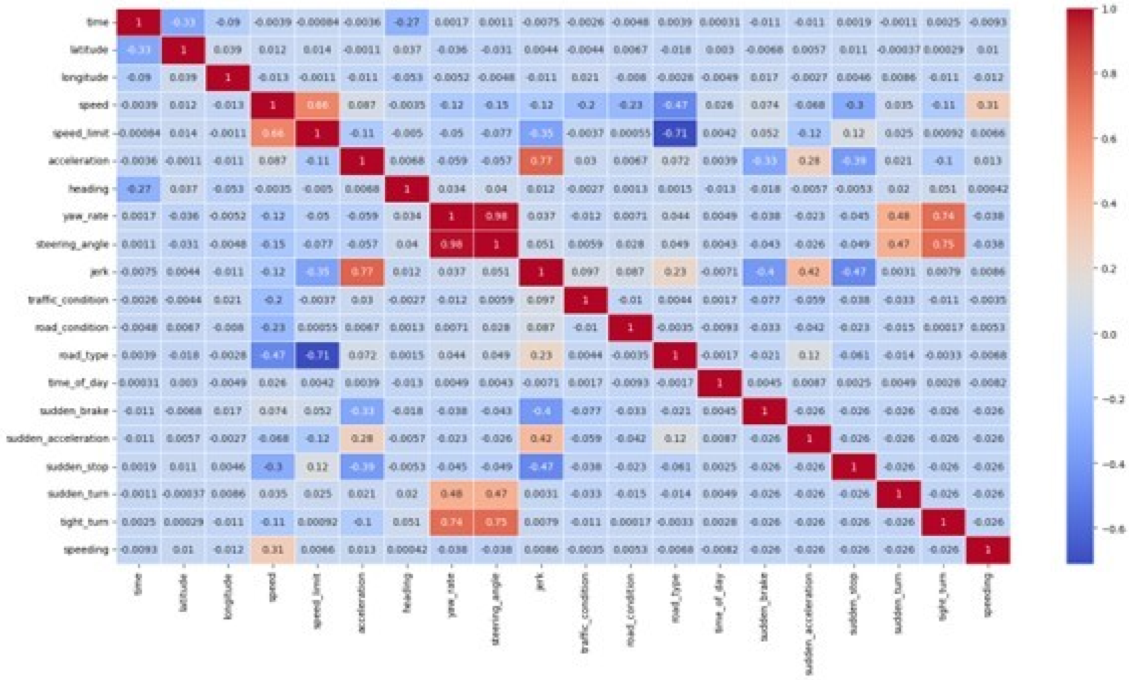

Figure 2: Correlation heatmap of the dataset

3.2. Route deviation calculation

The shortest (crow-flight) geodesic distance d between the vehicle's current position (φ,λ) and the closest projected point on the planned route polyline is used to measure deviation. This distance is computed using the haversine formula, which accounts for the Earth's spherical shape and provides an accurate great-circle distance:

$$d=2R\cdot\arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\varphi}{2}\right)+\cos(\varphi 1)*\cos(\varphi 2)*\left(\sin^2\left(\frac{\Delta\lambda}{2}\right)\right)}\right) \tag{1}$$

where:

- R=6,371,000m is the mean radius of the Earth;
- $\Delta\varphi=\varphi_2-\varphi_1$ is the difference in latitude;
- $\Delta_\lambda=\lambda_2-\lambda_1$ is the difference in longitude.

This formula ensures the distance and reflects true geodesic separation rather than simple Euclidean projection.

A route deviation event is flagged whenever $d>T$. This threshold can be tuned depending on the allowed corridor around the planned path (e.g. T =150m for city driving, larger for highways). Figure 3 shows this concept: if the distance between the planned route (orange) and taken route (blue) exceeds the threshold, here, the anomaly is flagged as a route deviation. Then, by comparing the vehicle's instantaneous location with the nearest route segment, the system tolerates small GPS drifts while reliably detecting meaningful off-route behavior.

Figure 3: Simulation of route deviation on map.

3.3. Ensemble learning with soft voting

To improve anomaly detection, an ensemble of four classifiers was used: Random Forest (RF), Extra Trees (ET), Gradient Boosting (GB), and a Multi-Layer Perceptron (MLP) neural network. Instead of a single model, the ensemble combines outputs using soft voting.

In soft voting, each classifier gives a probability for each class (anomaly or normal), not just a label. These probabilities show the model's confidence. The ensemble averages the probabilities across all models, and the class with the highest average is chosen as the final prediction.

Formally, let:

- K = number of classes (in this case, $k = 2$: *normal* or *anomaly*);
- J = number of classifiers (here, $J = 4$);
- $p_{j,k}(x)$ = probability assigned by classifier j to class k for input sample x.

The final predicted class $\hat{y}(x)$ is given by:

$$\hat{y}(x) = argmax_{k \in \{1,\ldots,K\}} \left(\frac{1}{J} * \Sigma_{j=1}^{J} p_{(j,k)}(x) \right) \tag{2}$$

This formula works as follows:

1. Each classifier M_j computes the probability of the input x belonging to class k;
2. These probabilities are summed across all classifiers;
3. The sum is divided by the number of classifiers J, giving the mean probability for each class;
4. The class k with the highest averaged probability is selected as the ensemble's decision.

For the four-model ensemble used in this study (RF, ET, GB, and MLP), the final decision rule can be expressed as:

$$\hat{y}(x) = argmax_{k \in \{0,1\}} \left(\frac{1}{4} * \left(p_{RF}, k(x) + p_{ET}, k(x) + p_{GB}, k(x) + p_{MLP}, k(x) \right) \right) \tag{3}$$

where each term $pModel,k(x)$ represents the probability assigned by a specific model to class k.

Anomaly detection was strengthened using a mix of four classifiers: Random Forest (RF), Extra Trees (ET), Gradient Boosting (GB), and a Multi-Layer Perceptron (MLP). Rather than relying on one model, their outputs were combined through soft voting. With soft voting, each model assigns a probability to each class, anomaly or normal, showing how confident it is. The ensemble then averages these probabilities, and the class with the highest mean is selected as the final result.

3.4. Training setup of predictive models

In this work, we treated the problem as predicting several driving anomalies at once, rather than just one. The features we picked, came from vehicle motion and the surrounding environment, speed, acceleration, heading, yaw rate, steering angle, jerk, and also things like traffic, road type, road condition, and time of day. The targets were five types of anomalies: sudden_brake, sudden_acceleration, sudden_stop, sudden_turn, and tight_turn. We split the data, 80% for training, 20% for testing, so the models could be checked on new trips.

To get better, more stable predictions, we used a soft-voting ensemble, combining a few different classifiers, instead of relying on just one. Base classifiers are:

- Random Forest (RF): Utilized for its ability to handle non-linear feature interactions and reduce overfitting through bagging;
- Extra Trees (ET): Similar to Random Forest but introduces additional randomness to enhance variance reduction.
- Gradient Boosting (GB): Sequentially builds weak learners, optimizing prediction by reducing residual errors;
- Multi-Layer Perceptron (MLP): A feedforward neural network capable of capturing complex, non-linear relationships between features and targets.

The ensemble was put inside a MultiOutputClassifier, so it could predict all five anomalies at once. It learned from the training data, figuring out how the features matched the target labels. Once trained, we tested it on the hold-out set and calculated precision, recall, and F1-score for each anomaly, to see how well it worked across the board. The ensemble, by combining different models, was able to take advantage of each one's strengths, giving more reliable predictions overall. Finally, we saved the trained model with joblib, ready for future use or deployment.

3.5. Trip rating model

Beyond anomaly detection, trip ratings were predicted using a Random Forest Regressor trained on trip features (Trip length, Sudden_brake Count, Sudden_acceleration Count, Sudden_turn Count, Tight_turn Count, Speeding Violation Count, Sudden_stop Count).

The rating formula:

$$\text{TripScore} = 10 - \frac{\sum_i \left(\text{violation}_i\right)}{\text{trip\_length}} \cdot 9 \tag{4}$$

The formula yields ratings on a 1–10 scale, where higher scores indicate safer and smoother trips. This normalizes penalties by trip length, ensuring fairness (two violations in 2 km penalize more heavily than in 30 km).

Evaluation metrics: R^2, MAE, and MSE confirmed strong predictive accuracy.

4. Experiments and results

Experiments used taxi trip data from several provinces in Kazakhstan: Astana, Almaty, Aktobe, Karaganda, Shymkent, and Pavlodar. Longitude and latitude sequences were turned into datasets with kinematic features (speed, acceleration, yaw rate, steering angle, jerk) and contextual factors (traffic, road type, road condition, time of day). This covers diverse driving situations and improves generalizability. The anomaly detection model was trained on five event types: sudden_brake, sudden_acceleration, sudden_stop, sudden_turn, and tight_turn. It performed reliably on test data, with precision, recall, and F1-scores near optimal for all classes. The detailed evaluation metrics are summarized in Figure 4, showing consistent performance across all anomaly types.

	precision	recall	f1-score	support
sudden_brake	0.99	1.00	0.99	84
sudden_acceleration	1.00	0.99	0.99	78
sudden_stop	1.00	0.99	0.99	93
sudden_turn	1.00	1.00	1.00	75
tight_turn	1.00	1.00	1.00	84
micro avg	1.00	1.00	1.00	414
macro avg	1.00	1.00	1.00	414
weighted avg	1.00	1.00	1.00	414
samples avg	1.00	1.00	1.00	414

Figure 4: Classification report of anomaly detectors.

For overall trip assessment, a regression model was built using features such as *trip length, sudden_brake, sudden_acceleration, sudden_turn, tight_turn, speeding,* and *sudden_stop* to predict a *trip rating*. The model yielded strong predictive accuracy (Testing R^2 = 0.996, MAE = 0.043). These outcomes reflect the logical dataset generation process, where anomalies were logically injected with cross-feature consistency, enabling the models to capture true behavioral patterns under realistic conditions. The performance of the trip rating model is presented in Figure 5, and the relationship between predicted and actual trip ratings is further illustrated in Figure 6 through scatter plots for both training and testing sets.

```
Training R2 Score: 0.9994428490850715
Testing R2 Score: 0.9958165945221744
MAE: 0.042696033730162467
MSE: 0.015246776788035092
```

Figure 5: Performance of Trip Rating Model.

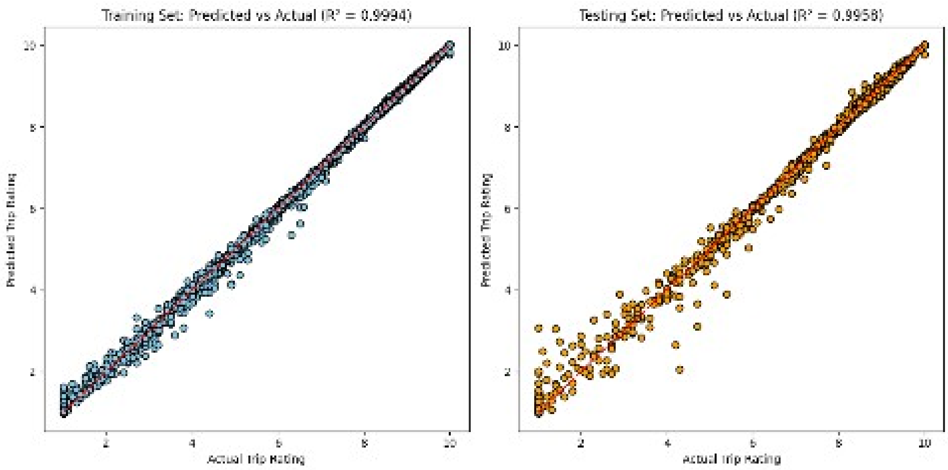

Figure 6: Scatter Plot of Predicted vs. Actual Trip Ratings.

5. Future implementations

Future work will focus on installing the system in real taxi fleet management platforms. Running the anomaly detection framework on live GPS and telematics data will enable real-time feedback for drivers and better fleet monitoring. The trip rating model can also be extended by adding passenger feedback and contextual factors such as weather, road work, or accident histories. Testing with larger datasets and more diverse regions will help confirm reliability and generalizability.

6. Discussion and conclusion

The results show that ensemble learning is effective for detecting unsafe driving patterns from sensor data. Each model contributed to consistent accuracy, and the soft voting setup gave stable performance across different anomaly types. The regression model also proved useful, with a near-perfect R^2 score, linking trip safety to measurable driving features. Together, these outcomes confirm that basic telemetry data carries enough information to monitor driving behavior without extra hardware or passenger input.

The novelty of this study lies in combining ensemble learning with basic telematics data to detect multiple types of unsafe driving without using cameras or heavy IoT setups. In addition, the proposed trip rating model introduces a new way to assess driving quality directly from sensor data, removing the need for passenger input or manual evaluation. Together, these features make the framework practical for real-time fleet monitoring and fair trip assessment.

At the same time, the analysis points to some limits. The synthetic dataset provided a controlled way to test the system, further real-world conditions might impose more complexity. Other factors like weather condition, and unpredictable driver reactions may affect model performance. This means additional training on real data is needed before large-scale deployment. Still, the results suggest that combining spatial, temporal, and behavioral features is a strong direction for building reliable safety tools.

In practice, the system can give taxi operators a way to track both driving safety and trip quality in real time. Trip ratings based only on sensor data reduce bias and provide fair comparisons across drivers and fleets. This makes the approach not just technically sound but also practical for everyday use. With further validation, it could support safer transport and more transparent service monitoring.

Acknowledgements

We acknowledge Nurislam Kassymbek from the Department of Computer Science, Al-Farabi Kazakh National University, for helpful feedback and advice. We also thank Eshaq Ekhlasi, a student at the same department, for valuable support during this project.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly for grammar and spelling checks, paraphrasing, translating and rewording. After employing these tools, the authors carefully reviewed and edited all content, and take full responsibility for the final version of the manuscript and its accuracy.

References

[1] Hassan, O. F., Ibrahim, A. F., Gomaa, A., Makhlouf, M. A., & Hafiz, B. (2025). Real-time driver drowsiness detection using transformer architectures: A novel deep learning approach. Scientific Reports, 15, 17493. https://doi.org/10.1038/s41598-025-02111-x.

[2] Gupta, B. B., Gaurav, A., Tai Chui, K., & Arya, V. (2024). Deep learning model for driver behavior detection in cyber-physical system-based intelligent transport systems. IEEE Access, 12, 62268–62278. https://doi.org/10.1109/ACCESS.2024.3393909.

[3] Hu, W., Li, M., Kwan, M., Luo, H., & Chen, B. (2023). Real-time taxi spatial anomaly detection based on vehicle trajectory prediction. Travel Behaviour and Society, 34, 100698. https://doi.org/10.1016/j.tbs.2023.100698.

[4] Costa, G., Manco, G., Ortale, R., & Ritacco, E. (2013). Hierarchical clustering of XML documents focused on structural components. *Data & Knowledge Engineering*, *84*, 26-46. https://doi.org/10.1016/j.datak.2012.12.002.

[5] Zhao, P., Bucher, D., Martin, H., Raubal, M. (2020). A Clustering-Based Framework for Understanding Individuals' Travel Mode Choice Behavior. In: Kyriakidis, P., Hadjimitsis, D., Skarlatos, D., Mansourian, A. (eds) Geospatial Technologies for Local and Regional Development. AGILE 2019. Lecture Notes in Geoinformation and Cartography. Springer, Cham. https://doi.org/10.1007/978-3-030-14745-7_5.

[6] Moavinis, A., Gounaris, A., & Constantinou, I. (2023). Detection of anomalous trajectories for vehicle traffic data. CEUR Workshop Proceedings, Vol. 3379. http://ceur-ws.org/Vol-3379/DARLI-AP_2023_9.pdf.

[7] Liu, M., Yang, K., Fu, Y., Wu, D., & Du, W. (2023). Driving maneuver anomaly detection based on deep auto-encoder and geographical partitioning. ACM Transactions on Sensor Networks, 19(2), Article 37. https://doi.org/10.1145/3563217.

[8] Gheni, H. M., AbdulRahaim, L. A., & Abdellatif, A. (2024). Real-time driver identification in IoV: A deep learning and cloud integration approach. Heliyon, 10(7), e28109. https://doi.org/10.1016/j.heliyon.2024.e28109.

[9] Fu, C., Zhou, Y., Xu, C., & Cui, H. (2019). Spatial analysis of taxi speeding event using GPS trajectory data. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC) (pp. 122–127). IEEE. https://doi.org/10.1109/ITSC.2019.8916870.

[10] Liu, S., Wang, Z., Zhang, Y., & Yang, H. (2024). Anomalous ride-hailing driver detection with deep transfer inverse reinforcement learning. Transportation Research Part C: Emerging Technologies, 159, 104466. https://doi.org/10.1016/j.trc.2023.104466.