Unsupervised Learning for Detection of Rare Driving Scenarios

Dat Le[1], Thomas Manhardt[2], Moritz Venator[2], Johannes Betz[1]

***Abstract*— The detection of rare and hazardous driving scenarios is a critical challenge for ensuring the safety and reliability of autonomous systems. This research explores an unsupervised learning framework for detecting rare and extreme driving scenarios using naturalistic driving data (NDD). We leverage Deep Isolation Forest (DIF), a novel anomaly detection algorithm that combines neural network-based feature representations with Isolation Forests (IFs), to identify non-linear and complex anomalies. Data from perception modules, capturing vehicle dynamics and environmental conditions, is preprocessed into structured statistical features extracted from sliding windows. The framework incorporates t-distributed stochastic neighbor embedding (t-SNE) for dimensionality reduction and visualization, enabling better interpretability of detected anomalies. Evaluation is conducted using a proxy ground truth, combining quantitative metrics with qualitative video frame inspection. Our results demonstrate that the proposed approach effectively identifies rare and hazardous driving scenarios, providing a scalable solution for anomaly detection in autonomous driving systems. Given the study's methodology, it was unavoidable to depend on proxy ground truth and manually defined feature combinations, which do not encompass the full range of real-world driving anomalies or their nuanced contextual dependencies.**

I. Introduction

Advanced Driver Assistance Systems (ADAS) have greatly improved road safety and driver comfort by handling challenging situations and avoiding potential hazards. A significant challenge for these systems is determining when to intervene. Although certain situations, such as lane departure, clearly require warning or feedback, it can be challenging to identify scenarios that deviate from normal driving conditions (Fig. 1).

Traditionally, detecting driving anomalies has relied heavily on expert domain knowledge and manual analysis of driving data. Numerous rule-based approaches have been proposed for driving anomaly detection, including abnormal driving behaviors [1]–[4], risky driving scenarios [5]–[7], and monitoring road conditions [8]–[10]. This process is labor-intensive, time-consuming, and challenging to scale as data volumes grow. Exhaustively listing all hazardous scenarios is impractical, and manual analysis often misses subtle or complex patterns, limiting anomaly detection. Exhaustively listing all hazardous scenarios is impractical, and manual analysis often misses subtle or complex patterns, limiting anomaly detection. In contrast, data-driven methods using machine learning excel at detecting non-linear and complex anomalies and can effectively handle diverse data features, making them suitable for dynamic driving environments. However, these models often require extensive labeled datasets for training and evaluation, which is impractical in real-world scenarios due to the scale of the dataset and the rarity of certain anomalies.

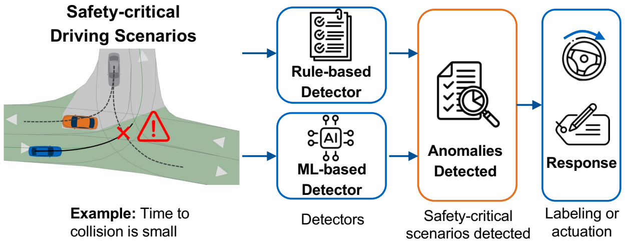

Fig. 1. Illustration of the pipeline for detecting safety-critical driving scenarios, showcasing the integration of ML-based and rule-based detectors to identify anomalies and respond effectively.

To address these limitations, we propose an automated framework leveraging unsupervised learning to detect driving anomalies without relying on labeled data. These anomalies include a variety of types, ranging from point anomalies (e.g., sensor errors) to contextual anomalies and collective patterns in driving scenarios [11]. Our study primarily focuses on rare, extreme, and abnormal events, such as sensor malfunctions and risky maneuvers. Unlike traditional rule-based approaches, which rely on pre-defined conditions [3], [12], our method identifies non-intuitive patterns that cannot easily be categorized. A key challenge in this work is evaluating unsupervised learning models when labeled data is unavailable. As mentioned earlier, creating labeled data is both time-consuming and impractical, especially for extreme driving scenarios. To address this, we propose the use of a proxy ground truth set consisting of approximate annotations derived from domain knowledge. This proxy set allows us to evaluate our unsupervised learning framework effectively without labeled data. Additionally, we utilize t-distributed Stochastic Neighbor Embedding (t-SNE) [13], a dimensionality reduction technique, to visualize high-dimensional data in 2D, making it easier to interpret detected anomalies. The main contributions of our study are:

- We provide a novel unsupervised learning framework using Deep Isolation Forest (DIF) [14] for detecting anomalies in driving and street scenarios.
- We provide the implementation of a robust data preprocessing pipeline, including selecting and engineering

[1] D. Le and J. Betz are with the Professorship of Autonomous Vehicle Systems, TUM School of Engineering and Design, Technical University Munich, 85748 Garching, Germany; Munich Institute of Robotics and Machine Intelligence (MIRMI), {johannes.betz}@tum.de

[2] T. Manhardt and M. Venator are with CARIAD SE, In-Campus Allee 22, 85053 Ingolstadt, Germany.

features from multimodal driving data (e.g., vehicle bus signals, object detection, lane detection).

- We provide a robust evaluation and visualization of our anomaly detection framework by providing driving anomalies using proxy ground truth and t-SNE.

II. Related Work

A. Driving Anomaly Detection

Anomaly detection is the process of identifying unexpected events that deviate from the norm. The types of anomalies range from point anomalies and collective anomalies to more complex contextual anomalies. In automated driving, Breitenstein et al. [11] provided a systematization of anomalies for visual perception, with the categories structured by detection complexity. Heidecker et al. [15] even extend the categorization of anomalies for perception concerning camera, LiDAR, and RADAR sensor modalities.

Studies have proposed anomaly detection approaches in particular problems by setting thresholds. Malta et al. [16] proposed an anomaly detection method based on vehicle speed and brake pedal thresholds. The scenarios are considered dangerous when the mean velocity is above a certain threshold, and the brake pedal pressure is high. Zhao et al. [17] used acceleration and steering wheel information to identify aggressive driving behavior. They set the thresholds based on the angles of the steering wheel. These established rule-based models are only effective for simple cases. When the driving environment is complicated, they may underestimate risk when the thresholds are not met or overstate risk even when drivers control the car properly.

An alternative approach is to detect anomalies using machine learning algorithms. Hofmockel et al. [18] and Matousek et al. [19] implement Isolation Forest to detect anomalous driving behaviors such as aggressive maneuvers, drowsiness, and tailgating, using raw vehicle sensor data via the Controller Area Network (CAN-bus). The model shows its effectiveness across different benchmarks and strong scalability but fails to detect hard anomalies that are difficult to isolate in non-linear separable data space. Chen et al. [3] introduced an SVM-based method to classify the types of abnormal driving behaviors. The study utilized orientation data from a smartphone. However, due to the complexity and variability of driving anomalies, creating a comprehensive classification-based solution that accurately identifies all anomalous scenarios is challenging and time-intensive. Clustering-based methods have also been employed for driving anomaly detection. Zheng et al. [20] proposed an unsupervised clustering technique utilizing smartphone accelerometer sensor data. The outliers on the clustering map were considered anomalies. Still, identifying meaningful clusters becomes problematic as the dimensionality of the feature space grows.

B. Isolation Forest and Its Extensions

Isolation Forest (IF) [21] is widely used for anomaly detection due to its simplicity, efficiency, and scalability. It performs well across various benchmarks and is particularly suited for large datasets [22].

The core assumption of IF is that anomalies, being ”few and different”, are easier to isolate from the rest of the data than normal instances. The algorithm constructs an ensemble of binary decision trees, known as isolation trees, built by recursively applying random splits on the data until all instances are isolated. Anomalies typically result in shorter average path lengths within these trees. One major limitation of IF is that it employs a linear, axis-parallel isolation approach, where each split considers only one feature at a time, making it hard to effectively detect complex anomalies in non-linear separable data spaces.

Several extensions of IF have been proposed to address this. SCIF [23] introduces a non-axis-parallel branching criterion by employing optimal slicing hyperplanes rather than single-feature splits. EIF [24] also uses hyperplanes but with random slopes and intercepts, enabling more diverse partitions. To mitigate the empty branching issue in EIF, Lesouple et al. [25] select splitting thresholds from the range of projected values along the hyperplane direction. A probability-based method from Tokovarov et al. [26] also refines split selection by finding more effective splitting values than random choices. These extensions generally improve performance by using non-axis-parallel or heuristic partitions. However, the main issue is that they still rely on shallow, linear isolation techniques. While these methods can approximate non-linear partitions through recursive splits, isolating hard anomalies (as shown in Fig. 2) often requires many splits, resulting in long path lengths, which means lower anomaly scores and are, therefore, not considered anomalies. This prevents true anomalies from being detected, leading to high false negative errors.

Deep Isolation Forest (DIF) [14] addresses this limitation by introducing a novel approach, which integrates non-optimized deep neural networks (DNNs) with Isolation Forests (IFs). The randomly initialized DNNs project the original data into random representation spaces. (an example shown in Fig. 2). Then, IFs use simple axis-parallel cuts to identify anomalies in these new data spaces. The main goal of the randomly initialized DNNs is to better detect complex anomalies that are not easily isolated in the original non-linear data space.

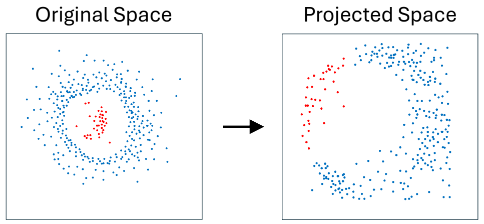

Fig. 2. Synthetic data example showing hard anomalies in the original space (left) and their representation in the projected space by DIF (right).

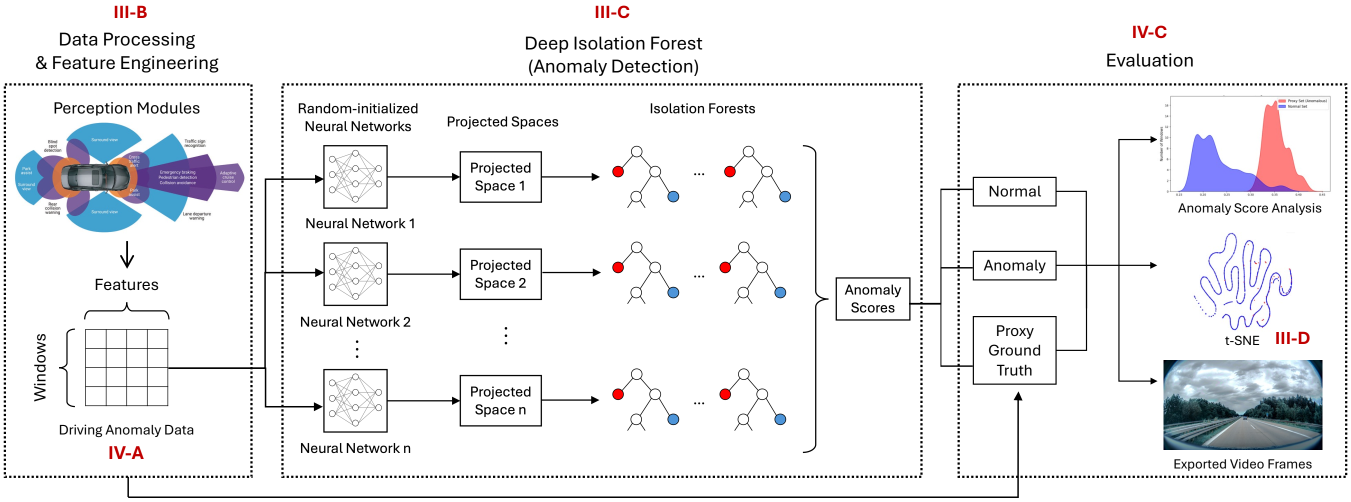

Fig. 3. Overview of the Framework Workflow: The figure depicts the flow of *vehicle bus signals* and *perception signals* (derived from *perception modules*) through processing and feature engineering steps, followed by anomaly detection using Deep Isolation Forest. The model takes Driving Anomaly Data as input, a multivariate tabular dataset where rows represent windows and columns represent features. The output consists of anomaly scores for each window. A threshold is defined to classify windows as anomalies if their anomaly scores exceed the threshold, while those below it are considered normal. Finally, the detection results are evaluated using proxy ground truth set derived from the Driving Anomaly Data.

III. Methodology

A. Workflow

The overall workflow of our framework is illustrated in Fig. 3. We utilize naturalistic driving data (NDD), consisting of *vehicle bus signals* and *perception outputs*. Vehicle bus signals are sensor data capturing internal vehicle dynamics, such as speed, acceleration, and yaw rate. Perception outputs, derived from environmental perception modules like object detection, blindness detection, and lane detection, provide contextual information about the surrounding environment. These data, stored as a multivariate time series, is processed and feature-engineered into a multivariate tabular format, referred to as Driving Anomaly Data (DAD), and used as input for unsupervised anomaly detection. Our study primarily employs Deep Isolation Forest (DIF) [14], an advanced anomaly detection algorithm that integrates neural networks, and Isolation Forest (IF) [21] to capture complex feature interactions and detect non-linear patterns. For unsupervised evaluation, we created a rule-based proxy dataset, considered as ground-truth anomalies, and employed a three-pronged evaluation strategy: i) analyzing the anomaly score distributions of the proxy and normal sets, comparing the top 100 highest-score anomaly segments with randomly selected normal segments, ii) visualizing data and reducing dimensionality using t-SNE [13], and iii) exporting video frames of detected anomalies for perceptual evaluation.

B. Data Processing and Feature Engineering

We utilized 100 hours of naturalistic driving data (NDD) collected from multiple measurements recorded during test drives on public roads in Europe. The dataset is multimodal, real-world, and extracted from perception systems, comprising modules such as CAN bus, blindness detection, object detection, and lane detection. The data is structured as a multivariate time series, where rows represent frames (sampled at approximately 10 Hz), and columns correspond to perception module signals. The first step was to identify and select relevant signals that capture vehicle kinematics and environmental factors. For vehicle kinematics, we selected speed as a fundamental and essential signal. For environmental factors, we included road-type conditions, weather, and time-to-collision (TTC) with detected vehicles.

We employed a feature aggregation approach combining sliding window segmentation and feature extraction. Sliding window segmentation divides the time series data into overlapping windows of fixed duration, allowing for the representation of localized temporal behavior. In this study, we used a window size of 6 seconds with a step size of 3 seconds, resulting in a 50% overlap between consecutive windows. Within each window, we extract features to summarize the behavior of the signals. The extracted features, described in Table I, range from simple statistical measures to complex derived metrics that quantify higher-order driving behaviors and risks. Continuous features were standardized (mean of zero and unit variance) to maintain comparable scales across the dataset. Categorical features were encoded into binary indicators using one-hot encoding, ensuring compatibility with the anomaly detection models. In order to mitigate biases caused by temporal continuity, the windows were randomly sampled across different time series measurements. This ensures that the final multivariate tabular dataset represents a diverse range of driving scenarios, minimizing dependence on sequential patterns and improving model generalizability.

C. Deep Isolation Forest

Deep Isolation Forest (DIF) [14] is a novel hybrid algorithm that combines random neural network-based representations with axis-parallel isolation trees to enhance anomaly detection by creating non-linear partitions in transformed data spaces. The architecture of DIF is illustrated in Fig. 3. First, the random initialized (i.e., non-optimized) neural networks project the original data into multiple random representation spaces. These random representation ensemble is defined as:

$$\Omega(D) = \{\chi_u \subset \mathbb{R}^d \mid \chi_u = \phi_u(D; \theta_u)\}_{u=1}^{n} \tag{1}$$

where χ_u is the u-th random representation, ϕ_u is the u-th neural network that maps the original input data D into a new d-dimensional space (with d controlling the richness of the representation and chosen based on the task or model design), n is the ensemble size (number of neural networks), and θ_u are the randomly initialized network weights.

Each representation χ_u is assigned with t isolation trees (iTrees), and a forest $\Gamma = \{\tau_i\}_{i=1}^{T}$ containing $T = n \times t$ iTrees is constructed. An iTree τ_i is essentially a binary tree. An ensemble of iTrees constitutes an Isolation Forest (IF), which iteratively partitions the data using random splits to isolate anomalies. An iTree consists of a subset of data starting at the root node and proceeding to an isolated leaf node by recursively splitting the data into smaller subsets until each sample is isolated. This concept assumes that anomalies require fewer splits to be separated from other normal samples. IF then assigns an anomaly score by calculating the average path length needed to isolate a sample across all iTrees. For each sample, the path length is defined as the number of edges traversed from the root node to the isolated leaf node. Since abnormal samples are expected to require fewer splits to be isolated, they tend to have a shorter average path. The anomaly score for a given sample is calculated as follows:

$$s(x, m) = 2^{-\frac{E(h(x))}{c(m)}}, \tag{2}$$

where $s(x, m)$ is the anomaly score for sample x, $h(x)$ is the path length of x in a single iTree, $E(h(x))$ is the average path length of x across all iTrees, and $c(m)$ is the normalization factor for a dataset of size m. The normalization factor $c(m)$ is given by:

$$c(n) = 2H(m-1) - \frac{2(m-1)}{m}, \tag{3}$$

where $H(i)$ is the i-th harmonic number, approximated as:

$$H(i) \approx \ln(i) + 0.5772156649, \tag{4}$$

with 0.5772156649 being the Euler-Mascheroni constant. The anomaly score $s(x, m)$ ranges from 0 to 1, where values closer to 1 indicate higher anomaly likelihood.

D. t-SNE

t-distributed Stochastic Neighbor Embedding (t-SNE) [13] is an unsupervised non-linear dimensionality reduction technique that visualizes high-dimensional data by mapping it to a low-dimensional space, typically 2D or 3D. Non-linear dimensionality reduction means that the algorithm allows us to separate data that cannot be separated by a straight line. Unlike Principal Component Analysis (PCA) [27], which is a linear dimensionality reduction technique suited for data with a linear structure, t-SNE is a non-linear method designed to preserve pairwise similarities between data points in a lower-dimensional space. While PCA aims to maximize variance by preserving large pairwise distances, t-SNE focuses on maintaining small pairwise distances to better capture the local structure of the data. t-SNE finds similarity measures between pairs of instances in both high-dimensional and low-dimensional spaces, then optimizes these similarities as follows:

i) Pairwise Similarities in High-Dimensional Space For high-dimensional data points x_i, t-SNE computes pairwise similarities p_{ij} based on Gaussian probabilities:

$$p_{ij} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \tag{5}$$

Here, the variance σ_i for point x_i is adjusted to match a user-defined perplexity, ensuring adaptive scaling based on local density.

ii) Pairwise Similarities in Low-Dimensional Space For the low-dimensional embeddings $\{y_i\}$, similarities are computed using a Student-t distribution:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}, \tag{6}$$

where q_{ij} represents the probability of similarity between the low-dimensional points y_i and y_j, and the heavy-tailed nature of the Student-t distribution allows better representation of pairwise relationships in the reduced space.

iii) Optimization The algorithm then maps the high-dimensional data points to a lower-dimensional space, aiming to preserve these pairwise similarities. This is done by minimizing the Kullback-Leibler (KL) divergence [28] between the high-dimensional probabilities p_{ij} and low-dimensional probabilities q_{ij}:

$$\mathrm{KL}(P\|Q) = \sum_{i \neq j} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right), \tag{7}$$

where $P = \{p_{ij}\}$ and $Q = \{q_{ij}\}$ are the high-dimensional and low-dimensional probability distributions, respectively. The cost function is minimized using gradient descent, refining the embedding to align the two distributions. The cost is minimized using gradient descent, which iteratively updates the low-dimensional embeddings $\{y_i\}$. The gradient of the KL divergence with respect to y_i is given by:

$$\frac{\partial \mathrm{KL}(P\|Q)}{\partial y_i} = 4 \sum_{j \neq i} (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}. \tag{8}$$

Here, p_{ij} and q_{ij} are the high-dimensional and low-dimensional similarities, respectively, and the term $\left(1 + \|y_i - y_j\|^2\right)^{-1}$ accounts for the heavy-tailed nature of the Student-t distribution. This gradient is used in each iteration of gradient descent to refine the embeddings. This optimization process gradually refines the lower-dimensional embedding until it stabilizes, creating clusters and subclusters of similar data points in the lower-dimensional space. These clusters can then be visualized to uncover the structure and relationships within the original high-dimensional data.

IV. Experiments & Results

A. Dataset

Our dataset comprises *vehicle bus signals* and *perception outputs*, as described in Section III-A. After processing and feature engineering, the final dataset, referred to as the Driving Anomaly Data (DAD), is used as input for the anomaly detection model. DAD is a multivariate tabular dataset, where each row represents a single 6-second window, and each column corresponds to a specific feature. The rows are randomly sampled to avoid sequential biases, and the extracted features are detailed in Table I. These features were selected based on their ability to capture three key aspects: i) vehicle driving kinematics, ii) environmental factors, and iii) driving behavior. While there are numerous features related to driving, this study focuses on the most essential ones for unsupervised anomaly detection.

For vehicle driving kinematics, we use the *relative speed range* as the primary feature, as it effectively reflects variations in speed over the window while accounting for proportional changes, making it more robust than *speed range*. For example, braking from 200 km/h to 180 km/h is more common and less likely to be considered anomalous than braking from 30 km/h to 10 km/h. With *speed range*, both cases have a value of 20 km/h, making it impossible to distinguish anomalies. However, with *relative speed range*, the values are normalized to $\frac{200-180}{200} = 0.1$ and $\frac{30-10}{30} = 0.67$, respectively, thus making it easier to quantify how anomalous each case is.

For environmental factors, we use mode features derived from categorical signals, such as *weather severity* and *road-type conditions*. These categorical features are converted into binary values using one-hot encoding to integrate seamlessly with continuous features. Driving behavior is captured through *lane-keeping quality*, aggregated from three lane boundary safety signals (left, middle, and right of ego's lane). The feature is categorized into three levels: good (all lane boundaries are safe), bad (one lane boundary is unsafe), and worst (all lane boundaries are unsafe). To estimate *collision risk*, we derive a feature based on *time-to-collision* (TTC) with detected vehicles and their *lateral position* relative to the ego vehicle. A scenario is considered risky if the TTC is less than 2 seconds and the lateral position is less than 2.2 meters. The resulting collision risk feature ranges from 0 (not risky) to 1 (high risk), providing a continuous measure of risk level.

Given the dataset's large size and unlabeled nature, manual annotation is impractical. Instead, we create a rule-based proxy set to serve as a ground truth for unsupervised evaluation. Using pre-defined heuristic rules, we filter the dataset to extract a subset representing proxy anomalies. Examples of these proxy rules include:

- **Extreme speed variations**: extremely high relative speed range.
- **Unusual signal combinations**: severe rain on dry road, severe rain with severe sun ray, severe sun ray with blur image.
- **Risky events**: bad lane keeping under severe rain, bad lane keeping with high relative speed range, high relative speed range on wet or snow-covered road, high collision riskiness.

These rules are manually defined based on the extracted features and do not capture all possible anomalies in the dataset. However, they provide an effective way to annotate and filter data for evaluation or to inject synthetic anomalies. The size of the proxy set can be adjusted by modifying the thresholds defined for each signal in heuristic rules. These thresholds determine the criteria for selecting proxy anomalies. For instance, time-to-collision values below a critical threshold, lateral acceleration exceeding an upper threshold, or significant change in speed during adverse weather are considered anomalous. By modifying these signal-specific thresholds in the heuristic rules, we can adjust the number of proxy anomalies, with stricter thresholds resulting in fewer anomalies and relaxed thresholds capturing more.

B. Implementation Details

This section introduces the details of our approach's implementation. DIF uses 50 representations ($r = 50$) and six isolation trees per representation ($t = 6$), with a subsampling size of 256 ($n = 256$) for each isolation tree.

DIF processes tabular data using a fully connected multilayer perceptron (MLP) network with two hidden layers of 500 and 100 units, respectively, and a *tanh* activation function. The network outputs representations with 20 dimensions, with optional skip connections and dropout applied based on configuration. Weights in the network are initialized with a normal distribution (mean 0, standard deviation 1), and the network processes data in batches of 64 samples. Competing IF used for comparison in the experiments employ 300 trees (matching the ensemble size of DIF, 50×6) with the same subsampling size of 256. The training was conducted on an NVIDIA A100 GPU using CUDA for acceleration. All anomaly detection algorithms, including DIF, were implemented in Python, with IF from the scikit-learn package used as a baseline. The unsupervised nature of the method allowed it to take the entire unlabeled dataset as input for training, producing predictions for the same dataset. Additionally, the contamination level (the expected proportion of anomalies) was manually set to guide the model.

C. Evaluation

Evaluating unsupervised algorithms is inherently more challenging than supervised methods because the absence of labeled data makes it difficult to measure performance directly. In this study, we use a proxy set (described in section IV-A) as ground truth anomaly annotations for the evaluation. We propose four evaluation approaches. The first

TABLE I

KEY FEATURES OF SIGNAL CHARACTERISTICS.

Aspects	Features	Values	Type
Vehicle Kinematics	Relative Speed Range	$\frac{max(speed)-min(speed)}{max(speed)}$	Continuous
Environmental factor	Rain severe Sunray severe Camera image blurriness severe Road-type conditions	Severe, Normal Severe, Normal Severe, Normal Dry, Wet, Snow-covered	Categorical Categorical Categorical Categorical
Driving Behavior	Lane Keeping Quality Time-to-collision Riskiness	Good, Bad, Worst $\frac{1}{TTC} \cdot \max(0, \frac{2.2-\lvert LateralPosition\rvert}{2.2})$	Categorical Continuous

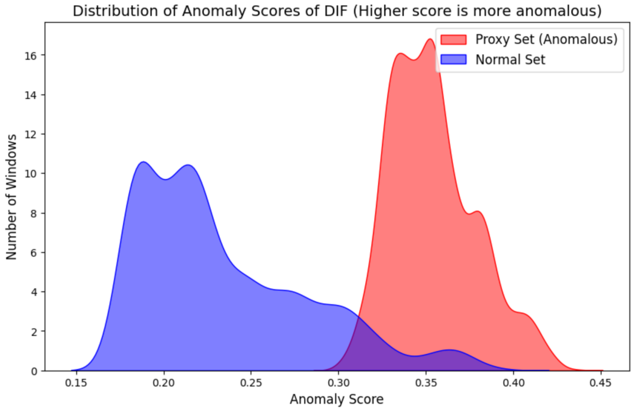

Fig. 4. Anomaly scores distribution for the proxy ground truth set (anomalous) compared to the random sample set (normal).

analysis compares the distribution of the anomaly scores for events in the proxy set and normal sets. The second analysis examines the top 100 anomalous events based on their anomaly scores and evaluates how many of them overlap with events in the proxy set. The third analysis provides dimensionality reduction and visualization of the dataset to inspect anomalous data points interactively. Finally, we export some video data frames of the events with the highest anomaly scores to inspect whether they are meaningful anomalous scenarios perceptually.

1) Anomaly Scores Distribution: The distribution of anomaly scores for the proxy set is compared against a normal set (randomly sampled) of equal size. In particular, we compare the anomaly score distributions between two sets: events in the proxy ground truth (expected to be anomalous) and events randomly sampled from the remaining dataset (expected to be normal). In our experiment, we identified 550 events using heuristic rules to define the proxy ground truth, representing approximately 2% of the dataset. We then randomly sampled 550 additional events from the presumed normal set to compare their anomaly score distributions. Fig. 4 illustrates the anomaly score distributions for both sets. It shows that events in the proxy ground truth set tend to have higher anomaly scores than those in the normal set. This higher anomaly score distribution for the proxy set indicates that the model effectively distinguishes anomalous events from normal ones.

2) Top Anomalies Analysis: The top 100 scenarios (i.e., events) with the highest anomaly scores are extracted. These scenarios are classified into two sets: those overlapping with the proxy ground truth (anomalous) and those outside it (normal). A summary table, as in Table II, shows the proportion of these high-scoring anomalies aligning with the proxy labels. The majority of the scenarios in the Top 100 group overlap with the proxy ground truth set, while only a small percentage are classified as normal. In contrast, the Random 100 group contains a much higher percentage of scenarios from the normal set. This shows that our model effectively identifies most of the anomalies in the proxy set.

TABLE II

ASSIGNMENT OF THE TOP 100 HIGHEST-SCORING ANOMALY EVENTS DETECTED BY DIF AND IF INTO THE NORMAL AND PROXY SETS.

Events	Normal set	Proxy set
Top 100 of DIF	16	84
Top 100 of IF	35	65
Random 100	88	12

3) t-SNE: To further evaluate the separation between anomalous and normal scenarios, we applied t-SNE to project the high-dimensional feature space into a two-dimensional visualization. Fig. 5 presents the t-SNE visualizations of the dataset for OC-SVM [29] (left), Isolation Forest (middle), and Deep Isolation Forest (right). Red points represent detected anomalous scenarios, while blue points indicate normal scenarios. The results illustrate that OC-SVM performs poorly; Isolation Forest struggles with some challenging local outliers, whereas Deep Isolation Forest outperforms both by effectively detecting these anomalies.

4) Perceptual Evaluation: Lastly, frames corresponding to extreme anomalies are exported for human inspection. This step provides a qualitative assessment of the model's performance, allowing domain experts to verify whether the identified anomalies represent genuinely interesting or rare scenarios. Given the time and resource-intensive nature of perceptual evaluations, we concentrated only on the most anomalous segments identified by the model. These segments

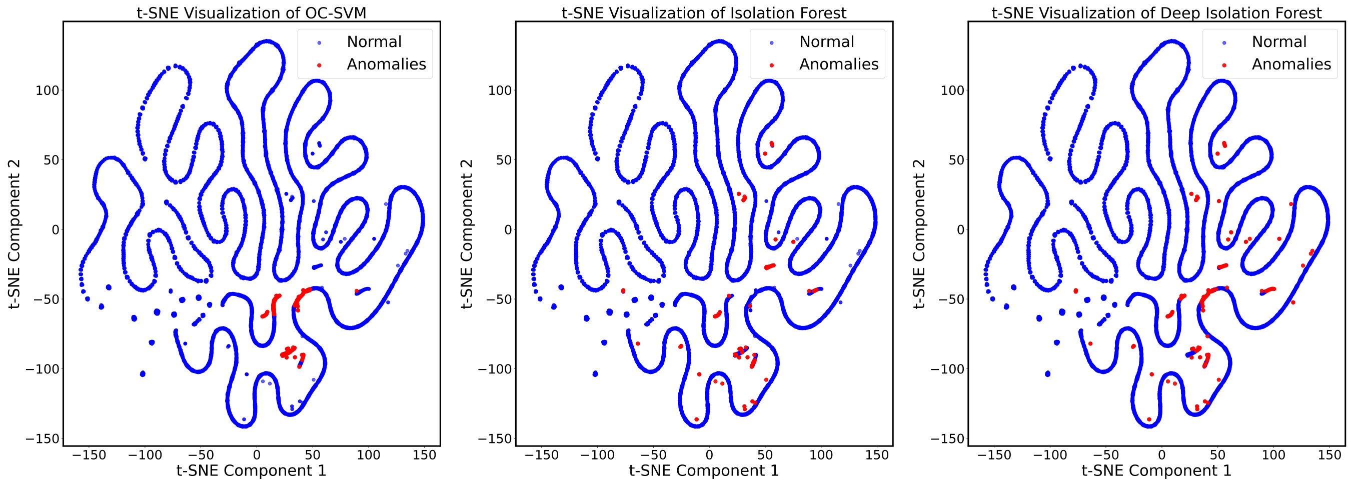

Fig. 5. t-SNE visualization of OC-SVM (left), Isolation Forest (middle), and Deep Isolation Forest (right). Red points represent detected anomalies, while blue points indicate normal data. The figure demonstrates that OC-SVM performs the worst in capturing anomalies, followed by Isolation Forest. Deep Isolation Forest outperforms both by capturing significantly more outliers, highlighting its superior anomaly detection capability.

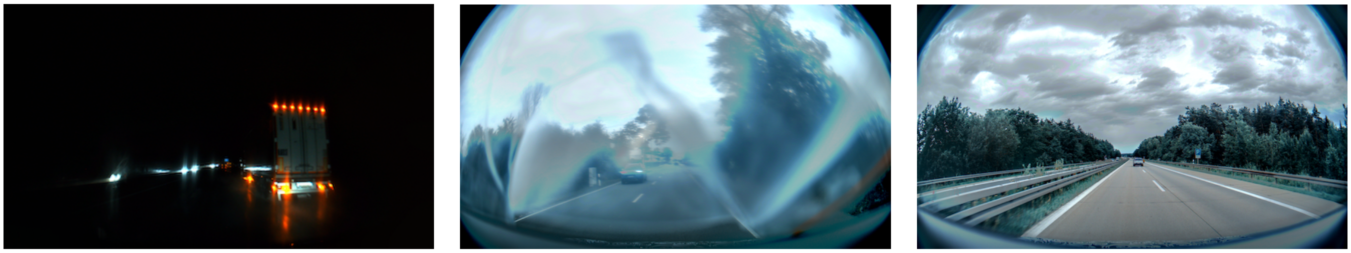

Fig. 6. High speed on slippery roads with unsafe lane boundaries

Fig. 7. Low time-to-collision under blurry image conditions

Fig. 8. Sudden braking triggered by the front vehicle's abrupt lane change without a turn signal

Fig. 9. Exported video frames from anomalous scenarios

were exported as image frames from video data and manually reviewed to determine whether they represent real anomalies, risky situations, or dangerous scenarios. The results are displayed in Fig. 9, including sub-figures representing interesting anomalous events. Fig. 6 depicts high speed on slippery roads with unsafe lane boundaries. Fig. 7 shows a scenario with low time-to-collision under blurry image conditions. Fig. 8 captures sudden braking triggered by a front vehicle's abrupt, unsignaled lane change, illustrating the hazards of unpredictable driving behavior. This perceptual evaluation confirms that the proposed unsupervised approach effectively identifies anomalous events. The high anomaly scores correspond to genuine hazards or rare scenarios, reinforcing the method's utility in detecting meaningful anomalies within the data.

V. Discussion

The experimental results demonstrate that Deep Isolation Forest (DIF) outperforms the baseline Isolation Forest (IF) in anomaly detection. The t-SNE visualization shows that DIF captures more local outliers than IF, and the Top 100 analysis indicates that DIF detects 84% of proxy anomalies, compared to 65% by IF. This improvement is because DIF integrates deep neural networks to transform the data into random representation spaces, enabling it to detect hard anomalies that can only be easily isolated in higher-order subspaces.

However, one challenge is that detected anomalous scenes may not always represent meaningful real-world driving scenarios due to irrelevant feature combinations or noise in high-dimensional data. An advanced feature engineering tool like *Deep Feature Synthesis* (DFS) [30] could address this by automatically generating and prioritizing more relevant features. Another limitation lies in the sensitivity of isolation-based models to parameters like contamination level, which is difficult to estimate in unlabeled, imbalanced datasets. Future work could involve optimization techniques, such as Grid Search [31] or Bayesian Optimization [32], to improve parameter tuning. Additionally, refining the proxy ground truth with more sophisticated rules to capture complex anomalies is crucial for better evaluation. Approaches like rule-based systems augmented with machine learning methods for driver fingerprinting [33] could enhance the accuracy and utility of the proxy ground truth. Lastly, while t-SNE is valuable for interpretation via visualization, it heavily relies on hyperparameters like perplexity and learning rate, making tuning complex and dataset-specific. Future work could focus on hyperparameter optimization techniques [34], [35].

VI. Conclusion

This study presents an unsupervised framework for detecting rare and extreme driving scenarios in naturalistic driving data. By leveraging Deep Isolation Forest (DIF), we address key limitations of traditional rule-based and linear anomaly detection methods, enabling the identification of complex, non-linear anomalies. Our evaluation framework utilizes a proxy ground truth set, allowing effective assessment of the unsupervised model while minimizing the need for time-intensive annotations or synthetic data injection. Despite its strengths, challenges remain in refining feature selection, optimizing hyperparameters, and improving the quality and coverage of the proxy ground truth set. Future work will focus on enhancing feature engineering to capture richer contextual information, automating parameter optimization to improve model performance, and optimizing visualization parameters to interpret anomalous patterns better.

References

[1] J. Dai, J. Teng, X. Bai, Z. Shen, and D. Xuan, "Mobile phone based drunk driving detection," in *2010 4th International Conference on Pervasive Computing Technologies for Healthcare*, 2010, pp. 1–8.

[2] H. Eren, S. Makinist, E. Akin, and A. Yilmaz, "Estimating driving behavior by a smartphone," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 234–239.

[3] Z. Chen, J. Yu, Y. Zhu, Y. Chen, and M. Li, "D3: Abnormal driving behaviors detection and identification using smartphone sensors," in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2015, pp. 524–532.

[4] J. Yu, Z. Chen, Y. Zhu, Y. Chen, L. Kong, and M. Li, "Fine-grained abnormal driving behaviors detection and identification with smartphones," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2198–2212, 2017.

[5] J. Wahlström, I. Skog, and P. Händel, "Detection of dangerous cornering in gnss-data-driven insurance telematics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3073–3083, 2015.

[6] F. Li, H. Zhang, H. Che, and X. Qiu, "Dangerous driving behavior detection using smartphone sensors," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1902–1907.

[7] C. Ryan, F. Murphy, and M. Mullins, "End-to-end autonomous driving risk analysis: A behavioural anomaly detection approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1650–1662, 2021.

[8] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," 11 2008, pp. 323–336.

[9] C. Yang, A. Renzaglia, A. Paigwar, C. Laugier, and D. Wang, "Driving behavior assessment and anomaly detection for intelligent vehicles," in *2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2019, pp. 524–529.

[10] Z. Liu, M. Wu, K. Zhu, and L. Zhang, "Sensafe: A smartphone-based traffic safety framework by sensing vehicle and pedestrian behaviors," *Mobile Information Systems*, vol. 2016, pp. 1–13, 10 2016.

[11] J. Breitenstein, J.-A. Termöhlen, D. Lipinski, and T. Fingscheidt, "Systematization of corner cases for visual perception in automated driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1257–1264.

[12] T. Chakravarty, A. Ghose, C. Bhaumik, and A. Chowdhury, "Mobidrivescore — a system for mobile sensor based driving analysis: A risk assessment model for improving one's driving," 12 2013, pp. 338–344.

[13] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html

[14] H. Xu, G. Pang, Y. Wang, and Y. Wang, "Deep isolation forest for anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 591–12 604, 2023.

[15] F. Heidecker, J. Breitenstein, K. Rösch, J. Löhdefink, M. Bieshaar, C. Stiller, T. Fingscheidt, and B. Sick, "An application-driven conceptualization of corner cases for perception in highly automated driving," *CoRR*, vol. abs/2103.03678, 2021. [Online]. Available: https://arxiv.org/abs/2103.03678

[16] L. Malta, C. Miyajima, and K. Takeda, "A study of driver behavior under potential threats in vehicle traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 201–210, 2009.

[17] H. Zhao, H. Zhou, C. Chen, and J. Chen, "Join driving: A smart phone-based driving behavior evaluation system," in *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 48–53.

[18] J. Hofmockel and E. Sax, "Isolation forest for anomaly detection in raw vehicle sensor data," 01 2018, pp. 411–416.

[19] M. Matousek, M. Yassin, A. Al-Momani, R. van der Heijden, and F. Kargl, "Robust detection of anomalous driving behavior," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–5.

[20] Y. Zheng and J. H. Hansen, "Unsupervised driving performance assessment using free-positioned smartphones in vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1598–1603.

[21] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*, 2008, pp. 413–422.

[22] R. Bouman, Z. Bukhsh, and T. Heskes, "Unsupervised anomaly detection algorithms on real-world data: how many do we need?" 2023. [Online]. Available: https://arxiv.org/abs/2305.00735

[23] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "On detecting clustered anomalies using sciforest," in *ECML/PKDD*, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:5721991

[24] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, p. 1479–1489, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2019.2947676

[25] J. Lesouple, C. Baudoin, M. Spigai, and J.-Y. Tourneret, "Generalized isolation forest for anomaly detection," *Pattern Recognition Letters*, vol. 149, 06 2021.

[26] M. Tokovarov and P. Karczmarek, "A probabilistic generalization of isolation forest," *Inf. Sci.*, vol. 584, no. C, p. 433–449, Jan. 2022. [Online]. Available: https://doi.org/10.1016/j.ins.2021.10.075

[27] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[28] F. Perez-Cruz, "Kullback-leibler divergence estimation of continuous distributions," in *2008 IEEE International Symposium on Information Theory*, 2008, pp. 1666–1670.

[29] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description*, ser. ODD '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 8–15. [Online]. Available: https://doi.org/10.1145/2500853.2500857

[30] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2015, pp. 1–10.

[31] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[32] P. I. Frazier, "A tutorial on bayesian optimization," 2018. [Online]. Available: https://arxiv.org/abs/1807.02811

[33] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: A new machine learning based authentication scheme," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1417–1426, 2020.

[34] Y. Cao and L. Wang, "Automatic selection of t-sne perplexity," 2017. [Online]. Available: https://arxiv.org/abs/1708.03229

[35] N. Sharma and S. Sharma, *Optimization of t-SNE by Tuning Perplexity for Dimensionality Reduction in NLP*, 09 2023, pp. 519–528.