The Development of the Advanced Driver-Assistance System by Analyzing the Road Accidents

Diana GHEORGHE
Bucharest University of Economic Studies, Romania
gheorghe1diana16@stud.ase.ro

This paper aims to emphasize the need for a method to decrease the number of accidents by examining the number of road accidents using Machine Learning techniques and configuring predictions based on historical data. Machine learning techniques have shown great potential in analyzing large-scale datasets related to road accidents. By leveraging these techniques, researchers have been able to identify key contributing factors, such as driver behavior, road conditions, and vehicle characteristics, which play a crucial role in accident occurrence. Through the analysis of historical accident data, machine learning models can effectively predict the likelihood of future accidents and identify high-risk areas, enabling proactive measures to be implemented. ADAS systems provide real-time information and assist drivers in making informed decisions while driving, thereby mitigating potential risks. This article's particular interest is underlining the importance of ADAS in the automotive field and how it can benefit drivers.

Keywords: *Machine Learning, Random Forest, PSO, ADAS, Road Accidents, Automotive Industry*
DOI: 10.24818/issn14531305/29.2.2025.06

1 Introduction

This research study analyzes the evolution of road accidents throughout the years and emphasizes the need for an Advanced Driver-Assistance System (ADAS) as part of the Infotainment division.

On one side, it focuses on how hardware components can work to create features that benefit the driver in terms of safety. On the other hand, the paper applies Machine Learning (ML) models using a well-known predictor, Random Forest, to the car accidents dataset. By leveraging machine learning algorithms to analyze historical accident data, insights can be gained regarding the need for further development of ADAS.

The author has taken into consideration facts from the past years, current behavior, and future predictions of the market.

2 Advanced Driver-Assistance System (ADAS)

2.1. What is ADAS

Millions of people are involved annually in car accidents. Whether it is due to speed, drowsiness, lack of concentration, or environmental distress, it is a global problem. Recent years have witnessed an immense technological development with the boost in hardware and software capabilities. This has made it possible for new technologies to emerge in the Automotive Industry, making it easier for manufacturers to include features that can prevent disaster and alert the driver to certain imminent dangers.

Introduction of ADAS started around the year 1948, when a modern Cruise Control was developed by the American engineer Ralph Teetor [1]. It has since advanced and become an essential part of the modern automotive industry. Moreover, it has come to the spotlight in recent years among the Euro NCAP (European New Car Assessment Program) summits, waiting for special regulations and conformity to be taken in support of its advancement.

2.2 ADAS Car Components

The ADAS we know today consists of various mechanisms of data collection, including RADAR, LiDAR, and cameras. RADARs, which use radio waves, are

combined with LiDAR's laser-like light measurement system to determine distance and angle of different environmental variables surrounding the car.
These hardware components give us features like Rear-view camera parking assistance, Pedestrian detection, Lane Departure Warning (LDW), and Traffic sign recognition (TSR). The usage of hardware components for different ADAS-related features can be seen in Figure 1.

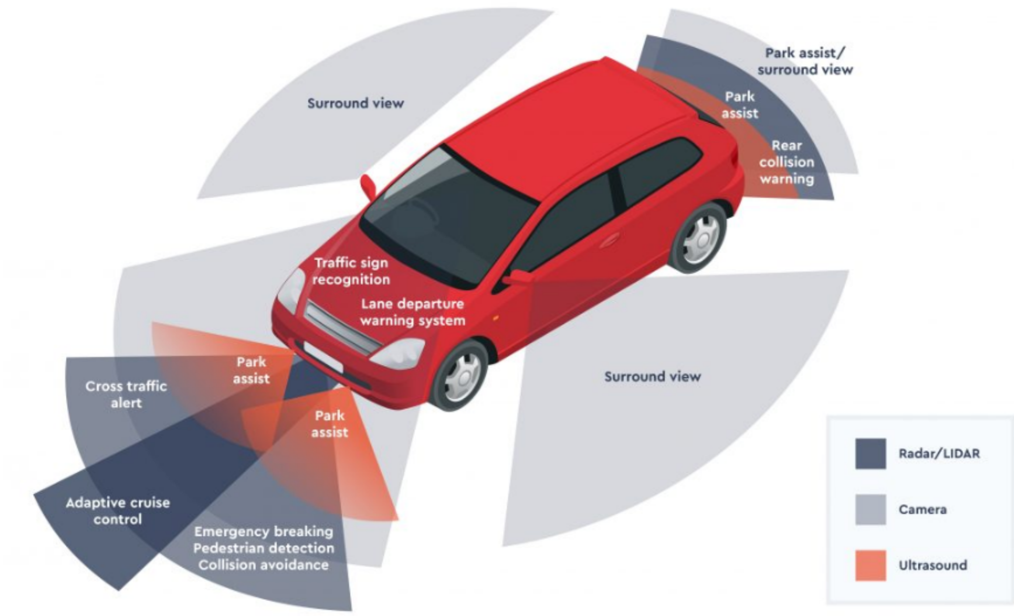

Fig. 1. ADAS Hardware Component Usage [2]

Data collection and processing are possible with the help of high-performance processors such as Samsung's newly developed Exynos V9 for Automotive usage, a processor model that includes an octa-core CPU, GPU, and high RAM. The development of processors specifically used in Automotive underlines the growth and potential lying in this industry and displays a glimpse at the future improvements that can be achieved.

2.3 Levels of ADAS

As ADAS features imply various fabrication and maintenance costs, they are present in a wide range of cars in various degrees. The Euro NCAP launched, in 2020, new ADAS ratings adapted to the technology's possibilities [3].
The Adaptive Cruise Control, which considers the acceleration and braking on highways, is a representative of an ADAS Level 1 vehicle, as it acts like an assistant to the driver. There is no moment in which the driver can fully let go of their assignments.
An ADAS Level 2 consists of a mild automation, a semi-automated process where the driver can trust the car at parking or driving through slow-moving traffic. However, even here, the driver must also assist the automation system at any moment.
Level 3 offers more automation, and the driver can leave the control of the car. However, this automation is conditional, and the system will announce to the driver when he should take over the car again.
Level 4 is the last level where a driver is needed. Although it operates autonomously in most cases, if there are extreme conditions where the driver's input is needed, it will send signals and wait for an immediate answer. If not provided, the car functionality is shut down, and the car is locked.
Level 5 is the highest that can be achieved and implies full automation, thus no

necessity for a driver. The role of the person can now switch to that of a passenger's and no driving license should be required for this level. So far, no car has reached this ADAS level as it is not sustained by current technology and by marketing strategies.
Achieving full automation means that there will be no need for a driver and thus, no need for various features currently present in the infotainment structure. This will lead to a restructuring in the whole industry chain, from manufacturing, production sites, to marketing and management levels.
There are various controversies about marketing vehicles as having higher ADAS levels than they have. Recent years have shown that there is a great need for regulations and analysis in this field.
However, whether we are talking about a level 1 or a level 5 rated ADAS featured car, there is considerable potential for improvement towards achieving the prevention of road accidents.
This article further focuses on finding the Automotive areas that can be enhanced and provides an analysis on the number of accidents that happened in a span of 10 years and applies Machine Learning (ML) models to predict the variation of the data.

3 Prediction of Vehicle Accidents

In the past years, there has been substantial research in the field of prediction, and various Machine Learning (ML) and Deep Learning (DL) techniques have been proposed [4]. The paper uses a hybrid approach, involving ML, DL, and Evolutionary techniques, which are among the most successful forecasting methods [5] [6].
For the study, the work environment is the PyCharm program with Python serving as a programming language. The usage of different libraries, such as pandas, seaborn, matplotlib, is present throughout the code development.
This study involves a set of data from a span of 10 years, with vehicle accidents that happened in the United Kingdom [7]. The dataset consists of three subsets and more than four million rows: the accidents that happened with their descriptions, the vehicle and driver's criteria, and the casualties' labels.
The variables comprising this dataset, detailed in Table 1, are multifaceted and can be categorized into several distinct groups: accident circumstances, vehicle-related information, and casualty-specific data. Accident circumstances and conditions encompass factors such as weather and lighting at the time of the incident, while casualty-related details include severity and demographic attributes. The vehicle category captures characteristics such as the driver's age, experience, and the vehicle's engine capacity. This structured categorization facilitates a systematic analysis of the contributing factors, enabling a comprehensive examination of their collective impact on the outcome under investigation.

Table 1. Dataset Variables

Accident Circumstances	**Vehicle**	**Casualty**
Accident Index	Accident Index	Accident Index
Police Force	Vehicle Reference	Vehicle Reference
Accident Severity	Vehicle Type	Casualty Reference
Number of Vehicles	Towing and Articulation	Casualty Class
Number of Casualties	Vehicle Manoeuvre	Sex of Casualty
Date	Vehicle Location-Restricted Lane	Age of Casualty
Day of Week	Junction Location	Age Band of Casualty
Time	Skidding and Overturning	Casualty Severity
Road Type	Hit Object in Carriageway	Pedestrian Location
Speed limit	Vehicle Leaving Carriageway	Pedestrian Movement

Accident Circumstances	**Vehicle**	**Casualty**
Junction Detail	Hit Object off Carriageway	Car Passenger
Junction Control	1st Point of Impact	Bus or Coach Passenger
2nd Road Class	Was Vehicle Left Hand Drive	Pedestrian Road Maintenance Worker
2nd Road Number	Journey Purpose of Driver	Casualty Type
Pedestrian Crossing-Human Control	Sex of Driver	Casualty IMD Decile
Pedestrian Crossing-Physical Facilities	Age of Driver	Casualty Home Area Type
Light Conditions	Age Band of Driver	
Weather Conditions	Engine Capacity	
Road Surface Conditions	Vehicle Propulsion Code	
Special Conditions at Site	Age of Vehicle (manufacture)	
Carriageway Hazards	Driver IMD Decile	
Urban or Rural Area	Driver Home Area Type	

This study employs a Random Forest model to analyze the influence of six independent variables: driver age, engine capacity, light conditions, road type, weather conditions, and road surface conditions, on casualty severity as the dependent variable. After establishing the baseline model, the Particle Swarm Optimization (PSO) algorithm is applied to optimize its hyperparameters, enhancing predictive performance. The resulting hybrid model is then used to reassess feature importance, allowing for a comparative analysis of how variable significance shifts post-optimization. This approach not only refines the model's accuracy but also provides deeper insights into the key factors impacting casualty severity under improved computational conditions.

Data manipulation has been performed to indicate the best outcome possible. The categorical values have been changed to numerical, and the null value-containing rows have been removed.

Before applying the ML model, a series of analyses have been performed on the available labels.

Exploratory Analysis

Exploratory analysis provides valuable insights into the dataset, helps identify issues related to data quality or anomalies, and guides further steps in the analysis and modeling process.

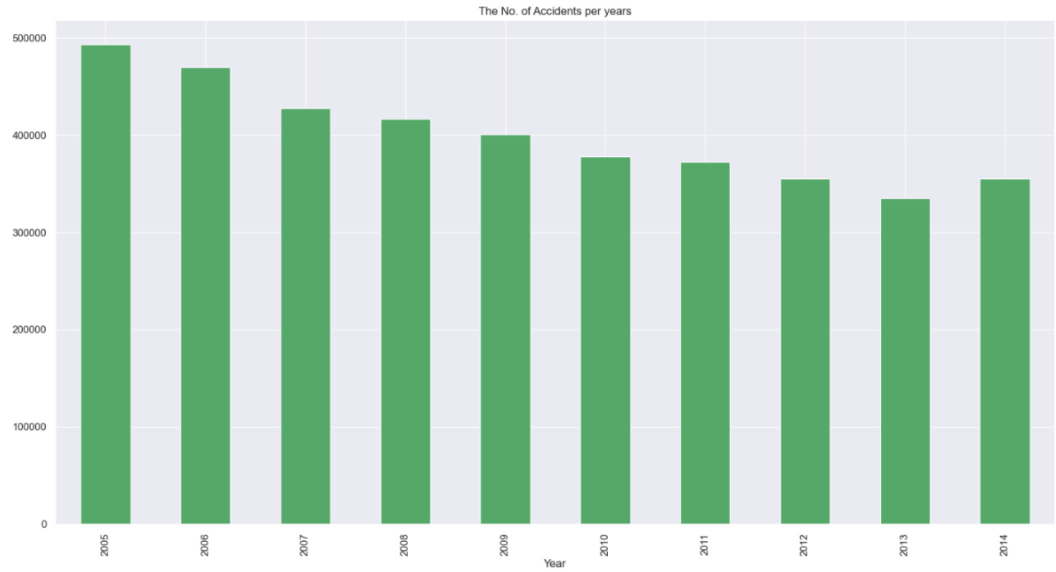

Fig. 2. Number of accidents per year

Looking at the count of accidents grouped by Year, there has been a decrease, yet small, and with a tendency to grow towards the more recent years.

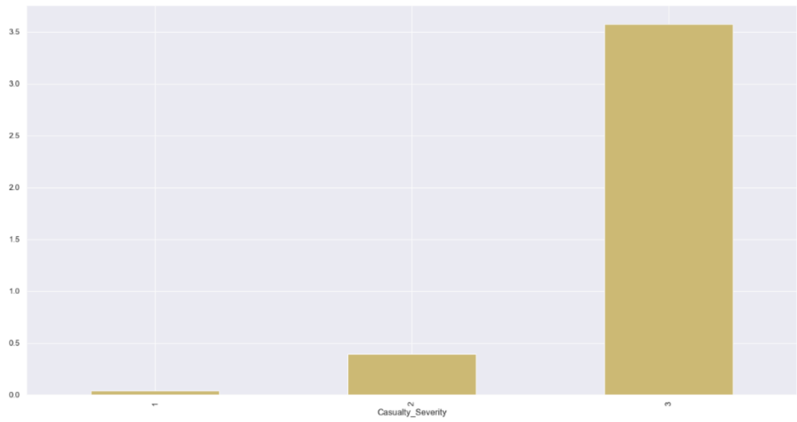

Fig. 3. Number of accidents based on Casualty Severity

Being categorized, from left to right, as Fatal, Serious, and Slight, the chart above (Figure 3) reveals that the severity of the casualty increases as the severity gets lighter.

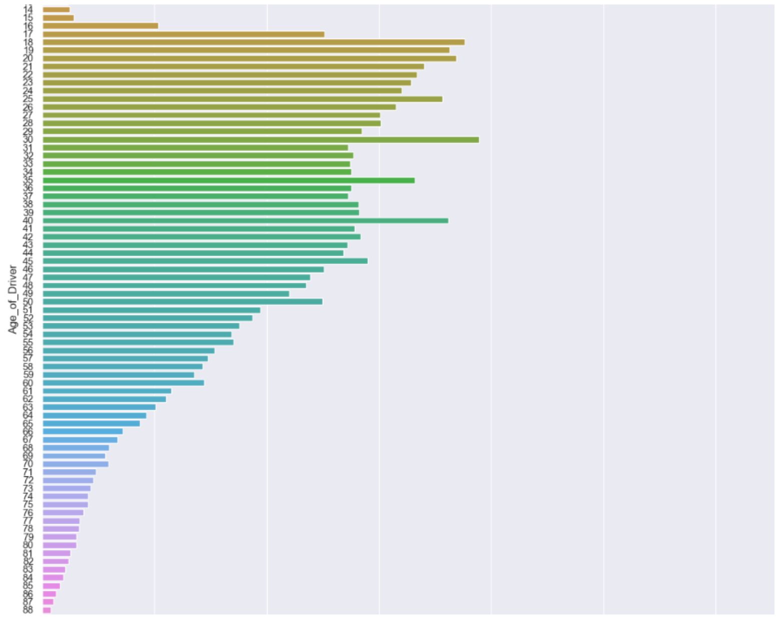

Fig. 4. Number of accidents based on the Age of the Driver

Figure 4 displays how the age of the driver also plays an important role, as the main accidents are clustered in the gap 17-62 years, with significant spikes at the beginning of each new group of age group: the twenties, thirties, and forties.

The impact of environmental conditions on road accidents has been explored further, considering Light Conditions, Weather Conditions, Road Surface factors, and Road Type.

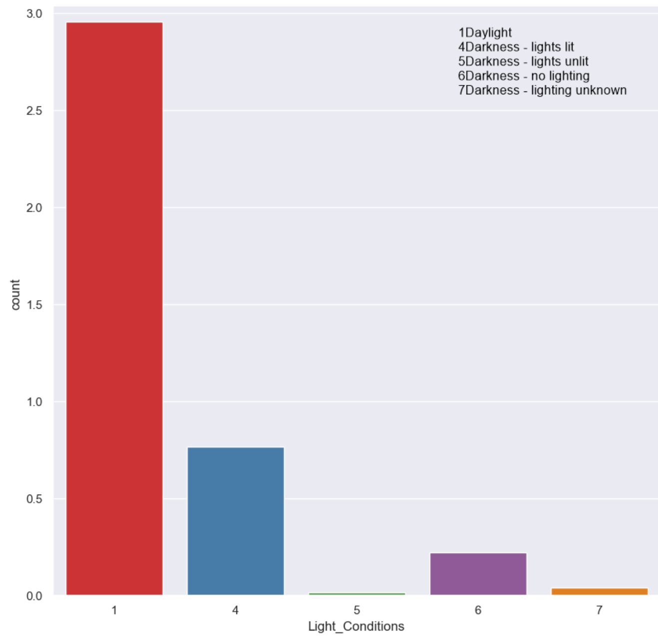

Fig. 5. Number of accidents based on Light Conditions

Focusing on the light conditions at the time of impact, most cases have occurred in daylight or the darkness with lights lit, Figure 5.

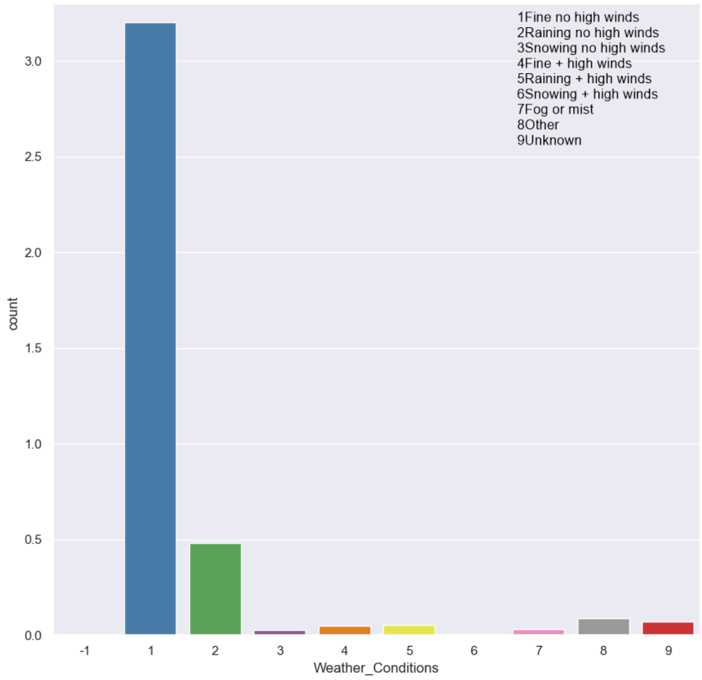

Fig. 6. Number of accidents based on Weather Conditions

Referring to weather conditions, with no high winds, there still is many accidents happening in mild weather conditions, Figure 6.

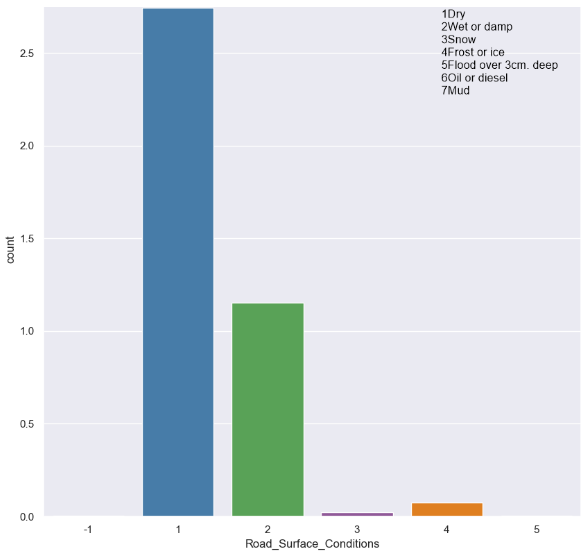

Fig. 7. Number of accidents based on Road Surface

Furthermore, the Road Surface Conditions display the same behavior, as seen in Figure 7, with no extreme conditions, but with most cases registered for dry or wet/damp roads.

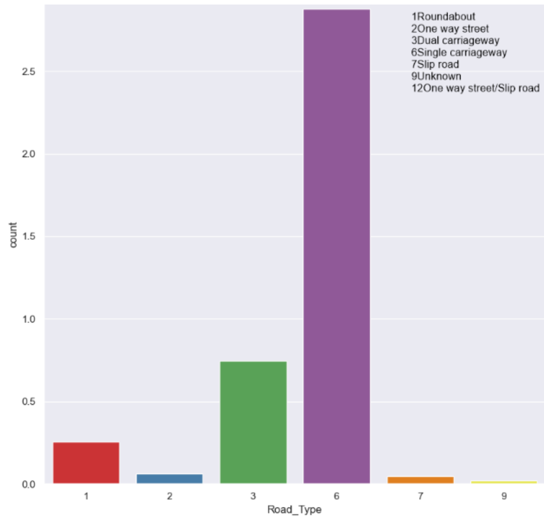

Fig. 8. Number of accidents based on Road Type

Considering the Road Type in Figure 8, it is observed that most cases are registered on single carriageways, followed by dual carriageways, with almost a quarter of the cases mentioned beforehand.

The Analysis implicates that, in certain cases, the weather, road state, and light conditions were not extreme, therefore, the avoidance of accidents could have been higher. This implies that ADAS might play a big role in the future development of traffic monitoring.

4 Random Forest Regressor for data prediction

The Random Forest Regressor is an ensemble of decision trees made to combat the biases that can occur when using a singular decision tree [8]. One decision tree, on its own, can provide an overfitting result, meaning that its prediction can be very high for the data provided, but it performs poorly outside its dataset.

A Random Forest consists of multiple decision trees, each tree being an individual. The process called Bagging allows the model to select randomly, for each tree, a subset of the training data on which to perform. Moreover, each decision tree from the forest gets a random set of features on which to perform the algorithm (Figure 9). This stops the ensemble from generating the same errors and biases [9].

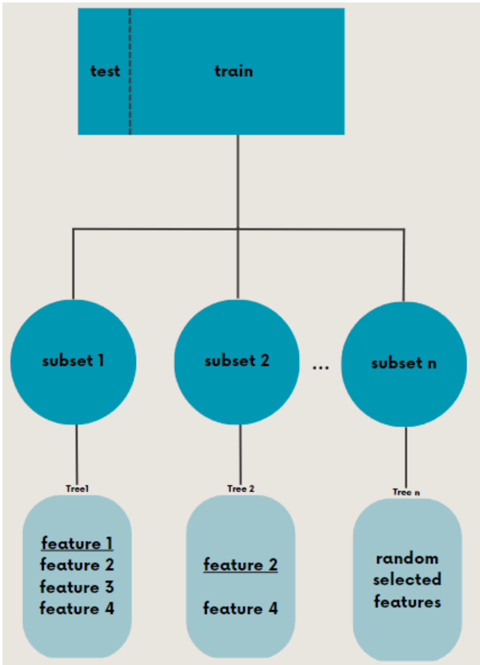

Fig. 9. Random Forest spread of decision trees and data

In the upcoming analysis, this paper uses the Random Forest Regressor model based on a series of independent and dependent variables, with the focus on the gravity of the casualty.
Therefore, the dependent variable is, from the dataset, the Casualty Severity. As independent variables, the author has taken into consideration the following: Age of Driver, Engine Capacity (CC), Light Conditions, Road Type, Weather Conditions, and Road Surface Conditions.
The split between train and test sets has been obtained with the help of the module train_test_split of sklearn.model_selection library, while considering a 20% test size and a remaining 80% training size.
With the module Random Forest Regressor from the sklearn library, the forest has been instantiated with 50 decision trees and a random state of 42 and has been trained on the previously split subsets.
Having a maximum depth of 48, there is a need to set the depth lower for better graphical visualization of the tree. Hence, the depth has been set to 3, obtaining, while shown in the first decision tree, Figure 10.

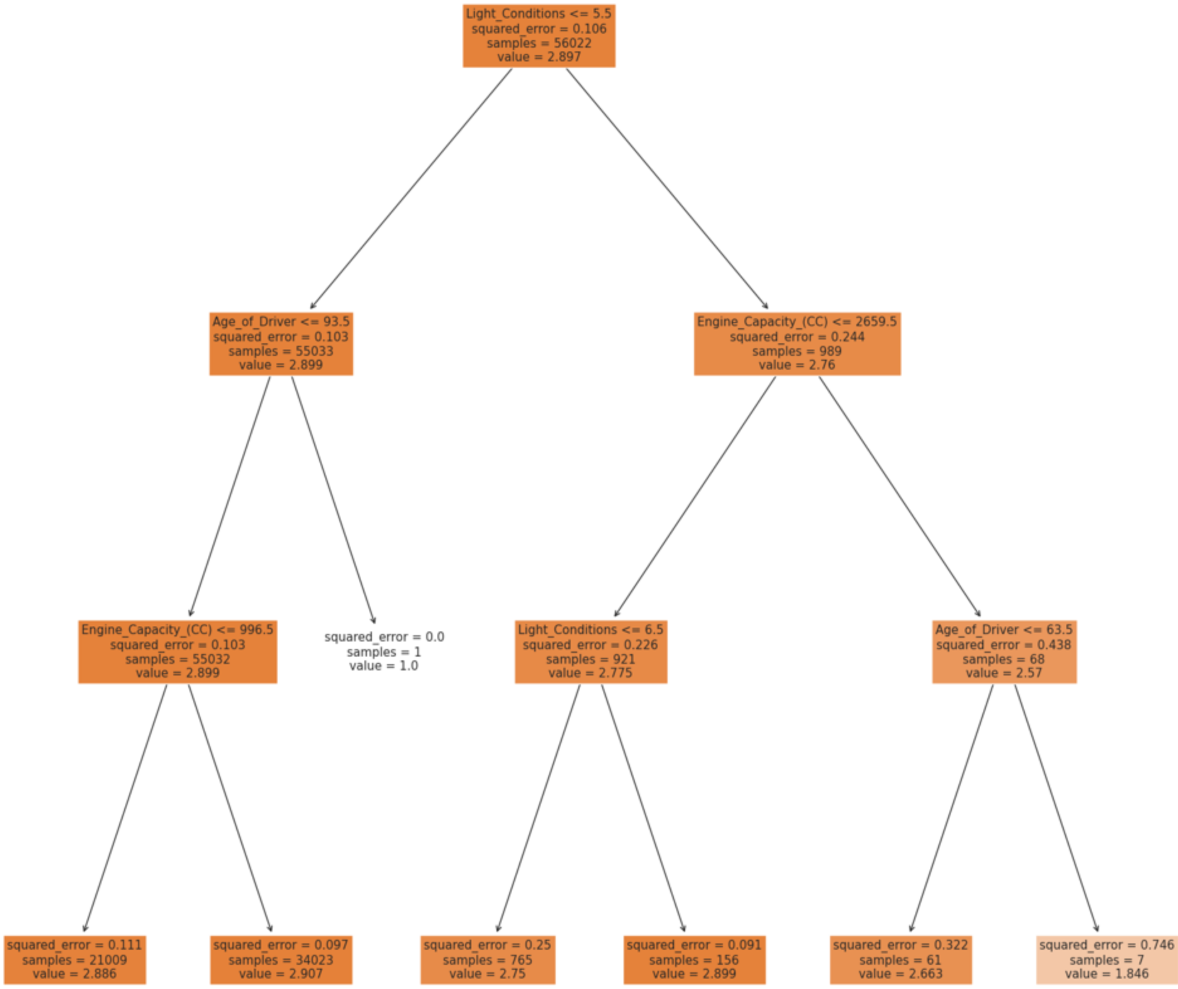

Fig.10. First decision tree with a set depth

As can be observed by looking at the root node in the first decision tree, the split has been made considering the value of X to be set at 5.5 or less, having a square root error of 0.106, meaning the distance from the regression line to the set of split data points. As shown in the graph, the number of samples taken into the root node is 56022, with a prediction of 2.897.
Further, the author focuses on the importance of the considered labels, as follows: "Engine_Capacity_(CC)" and "Light_Conditions" present a higher importance compared to the rest. At the other end, "Age_of_Driver" and "Road_Surface_Conditions" have a null importance, meaning that in further development of this process, those latter variables could be removed with no implication for the accuracy of the prediction.
The calculations can be seen in Figure 11, and the resulting graph is shown in Figure 12.

```
Variable: Engine_Capacity_(CC) Importance: 0.45
Variable: Age_of_Driver        Importance: 0.34
Variable: Road_Surface_Conditions Importance: 0.07
Variable: Weather_Conditions   Importance: 0.05
Variable: Light_Conditions     Importance: 0.04
Variable: Carriageway_Hazards  Importance: 0.03
Variable: Urban_or_Rural_Area  Importance: 0.02
```

Fig. 11. Importance of variable calculation

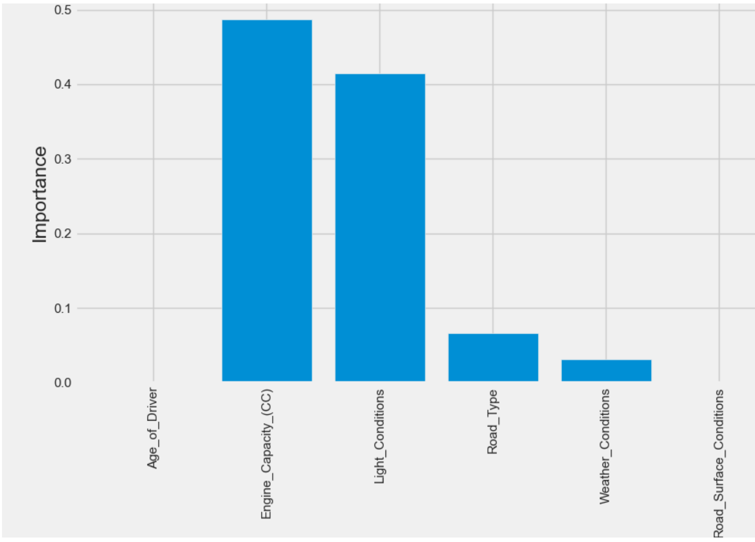

Fig. 12. Importance of variables graph

The forest's prediction method has been used on the test data, obtaining a Mean Absolute Error of 0.2 and an Accuracy of 90.88%. The Accuracy indicates that the model used fits the purpose.
Further analysis and calibrations will give an even better outcome and will serve as a base for the usage of other ML techniques along with DL and Evolutionary methods.

5 Evolutionary Algorithms

Evolutionary Algorithms (EA) consist of a set of methods that are inspired by the natural evolution and behavior of organisms. The core concept that lies as the foundation of these algorithms is the problem-solving method "trial and error", found in the natural evolution of organisms.
The algorithms belonging to EAs are grouped in several categories, the most important ones being: Genetic algorithms, Evolutionary strategies, Evolutionary programming, Genetic programming, and Swarm Intelligence (SI). Nowadays, the most successful approaches are of a hybrid type, that is, algorithms that combine classical, ML, and DL techniques, with EAs.
In this paper, we report a hybrid method, involving RFs and one of the most used SI techniques, PSO (Particle Swarm Optimization).
To optimize the Random Forest model's performance, Particle Swarm Optimization (PSO) was employed to fine-tune key hyperparameters, including the number of estimators (n_estimators), maximum tree depth (max_depth), minimum samples per leaf (min_samples_leaf), and the number of features considered for splitting (max_features). This approach ensured an efficient and automated search for the optimal parameter configuration, enhancing the model's predictive accuracy while mitigating overfitting.
Particle Swarm Optimization (PSO) is a population-based optimization algorithm inspired by the collective behavior of bird flocks or fish schools. It is commonly used to solve optimization problems, particularly in the field of computational intelligence and machine learning [10].
The basic idea behind PSO is to create a swarm of particles that move through a search space to find the optimal solution. Each particle represents a potential solution to the problem and moves within the search space by adjusting its position based on its own experience and the experiences of neighboring particles.
The PSO algorithm, considering its implementation steps, is described as follows [11]:
Initialization: Initialize a population of particles randomly within the search space. Each particle has a position and a velocity.
The initialization of the swarm for PSO has been achieve by randomly generating starting positions (swarm_position) for each particle

across a 4-dimensional search space (representing 4 hyperparameters), setting initial velocities (swarm_velocity) to zero, and preparing storage for each particle's personal best position (swarm_best_position) and corresponding fitness (swarm_best_fitness), which will be updated iteratively during the optimization process. The initialization ensures diverse exploration of the hyperparameter space while maintaining tracking of individual and collective best solutions.

The fitness function quantifies the quality of the particle's position concerning the optimization problem being solved. The evaluation method takes training and testing datasets along with a set of hyperparameters as inputs, then performs the following operations: (1) converts the continuous hyperparameter values to appropriate integer formats for Random Forest implementation, (2) instantiates a Random Forest Regressor with these parameters, (3) trains the model on the provided training data, (4) makes predictions on the test set, and (5) calculates and returns the R-squared coefficient as the fitness metric. The R-squared value serves as the optimization criterion, with higher values indicating better model performance that the PSO algorithm will seek to maximize during the hyperparameter tuning process. The function handles four key hyperparameters: number of estimators, maximum tree depth, minimum samples per leaf, and maximum features considered for splits.

Updates of personal best and global best are handled as the key update steps in a PSO algorithm, where each particle's personal best position (swarm_best_position) and fitness (swarm_best_fitness) are updated if its current position yields a better solution (higher fitness value), while simultaneously checking and updating the swarm's overall best solution (global_best_position and global_best_fitness) to ensure the optimization process tracks and converges toward the best-found solution across all particles.

Further, the update of the velocity and position of each particle based on its current velocity, personal best position, and global best position has been implemented. The new velocity determines the direction and magnitude of movement, while the new position reflects the updated location of the particle in the search space.

The script implements the core Particle Swarm Optimization (PSO) velocity and position update equations, where each particle's movement is determined by balancing its inertia (weighted by 0.8), cognitive attraction to its personal best (weighted by 1.5 with randomization), and social attraction to the swarm's global best (weighted by 1.5 with randomization), then updates the particle's position accordingly to iteratively converge toward an optimal solution.

The algorithm terminates when a stopping criterion is satisfied. This could be a maximum number of iterations, reaching a desired fitness value, or a predefined tolerance level.

The algorithm helped improve the previous method by obtaining an R-squared of 0.038, as seen in Figure 13.

```
Best Hyperparameters: [ 11.69236774  13.26841184 -14.2911886    5.39289143]
Best R-squared: 0.03860930806010465
Model's R-squared on Testing Data: 0.03841531010966903
```

Fig. 13. Obtained results after optimization

PSO aims to strike a balance between exploration (searching a wide area of the search space) and exploitation (narrowing down to promising regions). The particles communicate and share information about the best positions found so far, allowing for collective learning and convergence towards the optimal solution.

6 ADAS key features usage

By looking at the importance of variables with the PSO-improved Random Forest, it is concluded that the most important features extracted from the analyzed data are the Vehicle Type and Vehicle Maneuver, as well as seen earlier, Engine Capacity and Age of Driver, equally at 0.14 (Figure 14).

```
Variable: Vehicle_Type         Importance: 0.33
Variable: Vehicle_Manoeuvre    Importance: 0.3
Variable: Age_of_Driver        Importance: 0.14
Variable: Engine_Capacity_(CC) Importance: 0.14
Variable: Weather_Conditions   Importance: 0.03
Variable: Road_Surface_Conditions Importance: 0.03
Variable: Special_Conditions_at_Site Importance: 0.02
```

Fig. 14. Variable importances after optimization

Several ADAS key features can prevent or reduce the Casualty Severity, which we are focusing on in this article, by controlling the resulting important features.
Considering a high Engine Capacity, ADAS could intervene with its Automatic Emergency Braking System (AEB), considered to be efficient in reducing frontal collisions by up to 50% (National Highway Traffic Safety Administration).
Adding the Adaptive Cruise Control (ACC) is estimated, by the Insurance Institute for Highway Safety (IIHS), to reduce speed-related crashes by 20%.
Furthermore, considering the Driver Monitoring Systems (DMS), any distraction of the driver, common among young drivers, or even drowsiness, which is common among older groups of drivers, could be detected, preventing the collisions related to this cause. This feature, as it has been said by the European New Car Assessment Programme (Euro NCAP), can prevent distraction-related accidents by up to 40%.

7 Counterfactuals for test-proofing ADAS's key features

It is now aimed to find out how many accidents could have been avoided by implementing ADAS's key features as a prevention method.
Several counterfactuals have been generated in the code. It has been taken into consideration the data for Engine capacity and Age of drive, simulating the intervention of ADAS features.

This Python script simulates the potential impact of ADAS on accident severity by applying conditional probability-based reductions: for young drivers (<30 years) with high-powered cars (>2000CC), it models a 30% chance of downgrading fatal accidents (severity 1) to serious (2), while for older drivers (>65 years), it applies a 25% chance of reducing serious accidents (2) to slight (3), otherwise preserving the original severity value, thereby quantifying hypothetical ADAS intervention effects.
By mimicking the intervention of ADAS features into the initial data, the outcomes presented in Figure 15 are obtained.

```
Accidents prevented by severity:
 Casualty_Severity
3   -7046
2    6824
1     222
```

Fig.15. ADAS prevention of accidents

It is to be noted that we only have 222 fatal accidents (labelled as 1) remaining, registering a massive reduction. We then have an increase in the moderate severity (labelled as 2) and a decrease in the slight severity (labelled as 3). The results demonstrate that ADAS technologies effectively downgrade fatal accidents to moderate severity (e.g., via AEB reducing collision impact) while preventing many minor crashes entirely (e.g., through lanekeeping systems avoiding low-speed collisions).

8 Conclusion

The paper successfully applies the Random Forest Regressor to fit the data, obtaining a good prediction accuracy. It further applies hybridization with an Evolutive algorithm, PSO, obtaining an increase of R-squared variable and thus a better explanation of the variation of Casualty Severity data. By predicting the variation of car accidents and the responsible variables for it, the paper opens a path of further research aiming to improve road accidents.

In conclusion, machine learning predictions of car accidents provide a valuable framework for evaluating the role of ADAS technologies in accident prevention. The integration of these predictions with ADAS systems can enhance their capabilities and optimize their performance in real-world driving scenarios. By leveraging machine learning analysis, researchers and industry stakeholders can make informed decisions regarding the design, implementation, and improvement of ADAS technologies to promote safer and more efficient transportation systems.

As an extension of the research, the continuation of exploring various evolutionary algorithms and their hybridization with machine learning models is desired, optimizing the data used in the study, researching the market and trends in the usage of deep learning models, composing a new deep learning model inspired by biology and adapting it to the needs of the study.

References

[1] Hyundai, "The Evolution of Cruise Control," 2018. [Online]. Available: https://www.hyundai.news/eu/articles/stories/the-evolution-of-cruise-control.html.

[2] Group, The Windscreen Company, "Complete Guide to ADAS," 21 02 2023. [Online]. Available: https://www.thewindscreenco.co.uk/adas-guide/complete-guide-to-adas/.

[3] EURO NCAP, "About Euro NCAP," 2023. [Online]. Available: https://www.euroncap.com/en/about-euro-ncap/timeline.

[4] L. Ding, Y. Zou, Z. Zhang, T. Zhu, and L. Wu, "Vehicle Acceleration Prediction Based on Machine Learning Models and Driving Behavior Analysis," Applied Sciences, vol. 12, no. 10, 2022.

[5] C. Cocianu and H. Grigoryan, "Machine Learning Techniques for Stock Market Prediction. A Case Study Of Omv Petrom," Economic Computation and Economic Cybernetics Studies and Research, vol. 50, no. 3, 2016.

[6] C. Cocianu, C. Uscatu, and M. Avramescu, "Improvement of LSTM-Based Forecasting with NARX Model through Use," Electronics, 2022.

[7] Kaggle, "Kaggle," [Online]. Available: https://www.kaggle.com/datasets/silicon99/dft-accident-data.

[8] T. Yiu, "Understanding Random Forest," [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed 2023].

[9] L. Breiman, "Random Forests," Springer, vol. 45, no. 1, 2001.

[10] T. Bäck and P. Schwefel, "An Overview of Evolutionary Algorithms for Parameter Optimization," Evol. Comput., vol. 1, no. 1, 1993.

[11] Eberhart and S. Yuhui, "Particle swarm optimization: developments, applications and resources," Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), vol. 1, 2001.

Diana GHEORGHE graduated from Bucharest University of Economic Studies, Faculty of Economic Informatics. She works as a Full Stack Software Engineer in the Automotive field at Harman International. Currently pursuing PhD research focused on the Analysis of Financial Data using biology-inspired Artificial Intelligence algorithms. The area of interest is focused on different Artificial Intelligence and Machine Learning techniques and their development across the Automotive Industry.