The Need of Advanced Driver-Assistance System's Development based on an Analysis of Road Accidents

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

Taking a look at the numbers of road accidents using Machine Learning techniques and configuring predictions based on historic data, the paper aims to emphasize the need of a method to decrease numbers of accidents. 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. A particular interest of this Article is underlining the importance of ADAS in the automotive field and how it can benefit the drivers.

Keywords: *Machine Learning, ADAS, Road accidents, Automotive Industry*
DOI: 10.24818/issn14531305/27.3.2023.02

1 Introduction

This research study is focusing on analyzing the evolution of road accidents throughout the years and emphasizing the need of an Advanced Driver-Assistance System (ADAS), as part of the Infotainment division. On one side it is focusing on how hardware components can work to make features that benefit the driver in terms of safety. On the other hand, the paper applies Machine Learning (ML) models with usage of a well-known predictor, namely Random Forest, to car accidents dataset. By leveraging machine learning algorithms to analyze historical accident data, insights can be gained regarding the need of ADAS's further development.

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 technology development with the boost in hardware and software capabilities alltogether. This has made possible for new technologies to emerge in the Automotive Industry, making easier for the manufacturers to include features that can prevent disaster and alert the driver on certain imminent dangers.

Introduction of ADAS has 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 spotlight in recent years among the Euro NCAP (European New Car Assessment Program) summits, waiting for special regulations and conformities to be taken in support of its advancement.

2.2 ADAS Car Components

The ADAS we know today consists of various mechanism 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 dif-

ferent environment variables surrounding the car.
These hardware components give us features like Rear-view camera parking assistance, Pedestrian detection, Lane Departure Warning (LDW), Traffic sign recognition (TSR). The usage of hardware components for different ADAS-related features can be seen in Figure 1.

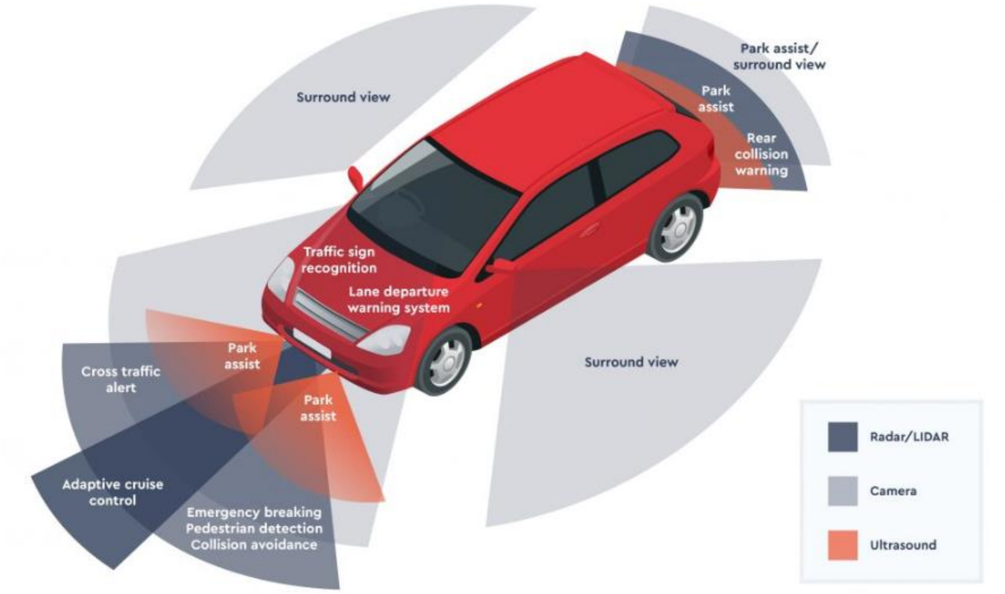

Fig. 1. ADAS Hardware Component Usage [2]

Data collection and processing is 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 laying 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 fabricating and maintenance costs, they are present in a wide range of cars in various degrees. The Euro NCAP has launched, in 2020, new ADAS ratings adapted to the technology's possibilities [3].
The Adaptive Cruise Control, that takes into account the acceleration and braking on highways driving are 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 its 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-motioned 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 the driver when he should manage the car again.
Level 4 is the last level where a driver is needed. Although it operates autonomous 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 the one 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 of a driver and thus, no need for various features currently present in the infotainment structure. This will lead to a restructuration in the whole industry chain, from manufacturing, production sites, to marketing and managing levels.
There are various controversies on marketing vehicles as having higher ADAS levels than they have in reality. Recent years have shown that there is a great need of regulations and analysis towards this field.
However, whether we are talking about a level 1 or a level 5 rated ADAS featured car, there is a considerate potential of improvement towards achieving the prevention of road accidents.
This article further focuses to find 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, various Machine Learning (ML) and Deep Learning (DL) techniques being proposed [4]. The paper uses a hybrid approach, involving ML, DL and Evolutionary techniques, being among the most successful forecasting methods [5] [6].
For the study the work environment is PyCharm program with Python serving as a programming language. The usage of different libraries such as pandas, seaborn, matplotlib are 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 is consisting of three subsets and more than four million rows: the accidents that happened with their descriptions, the vehicle and driver's criteria and the casualty's labels.
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 analysis have been performed on the available labels.

3.1 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.

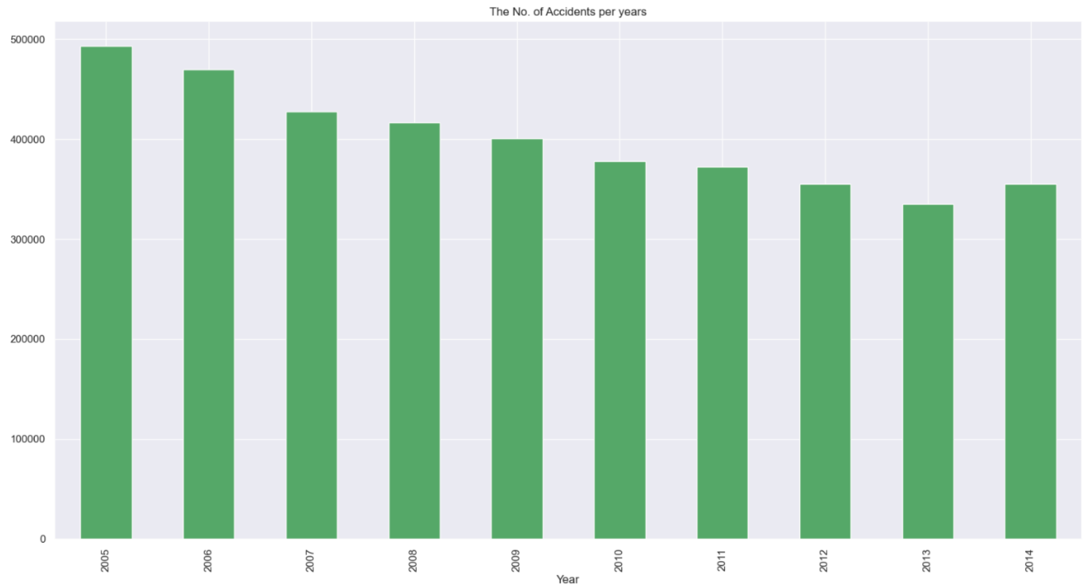

Fig. 2. Number of accidents per year

Taking a look at the count of accidents grouped by Year, it can be seen that there has been registered a decrease, yet small, and with tendencies of growing towards the more recent years.

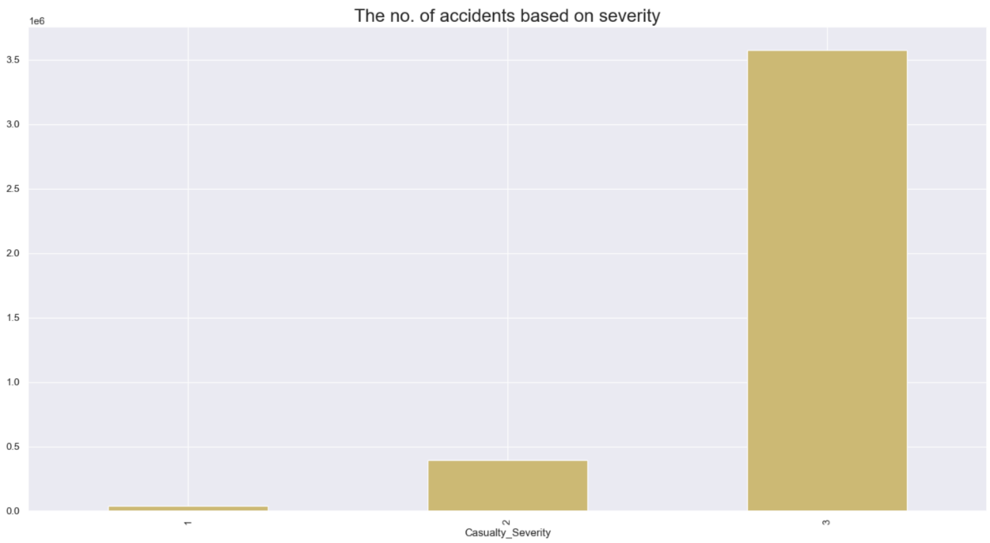

Fig. 3. Number of accidents based on Casualty Severity

Being categorized, from left to right as Fatal, Serious and Slight it can be observed in the chart above (Figure 3) that the severity of the casualty has a significant increase as the severity gets lighter.

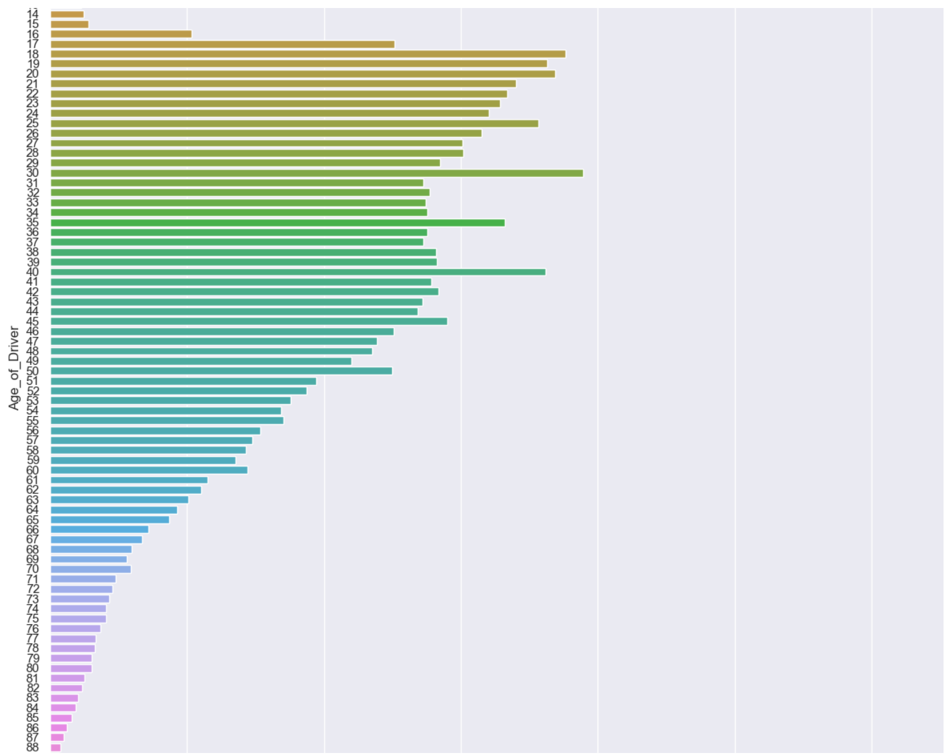

Fig. 4. Number of accidents based on Age of Driver

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

Impact of environmental conditions on the road accidents have been explored further taking into account Light Conditions, Weather Conditions, Road Surface factors and Road Type.

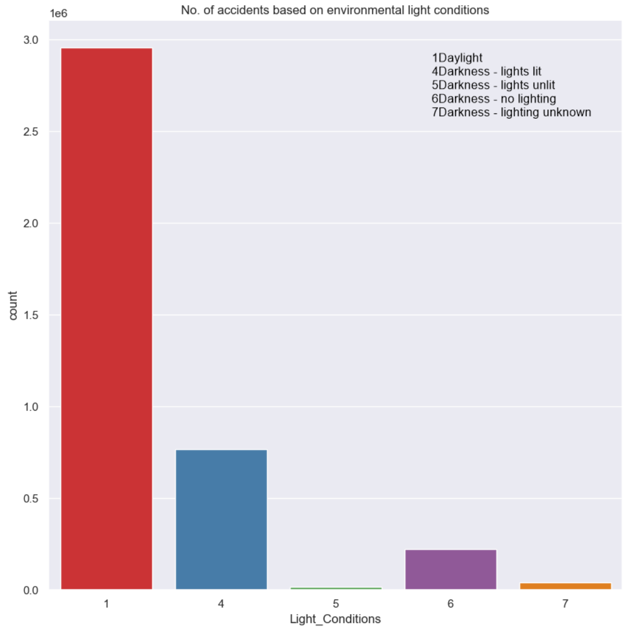

Fig. 5. Number of accidents based on Light Conditions

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

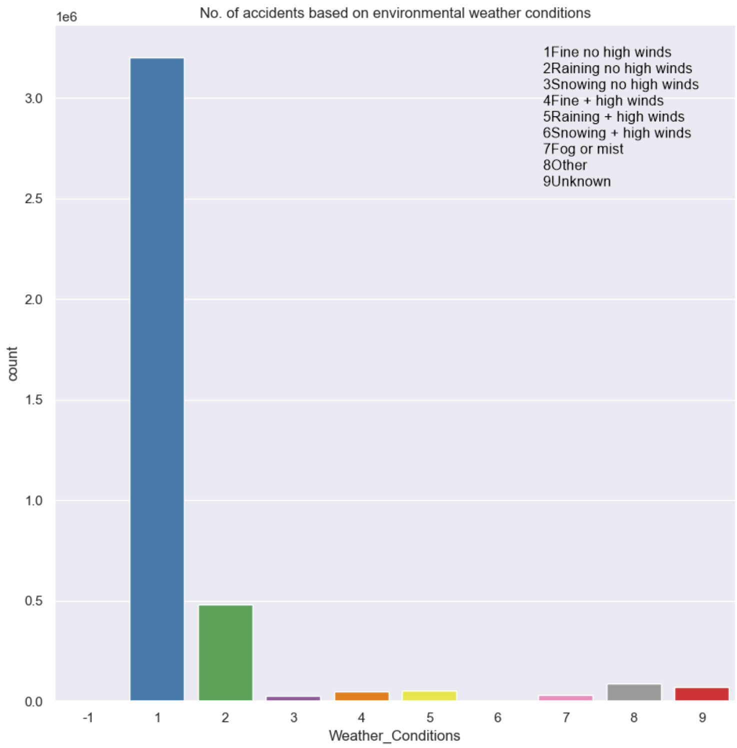

Fig. 6. Number of accidents based on Weather Conditions

Referring to weather conditions, with no high winds, there still is a large number of accidents happening in mild weather conditions, Figure 6.

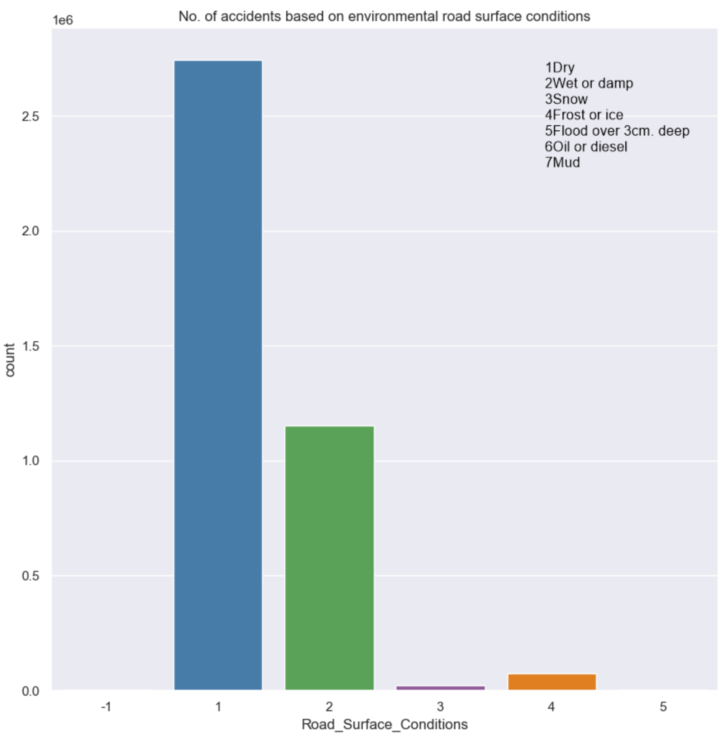

Fig. 7. Number of accidents based on Road Surface

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

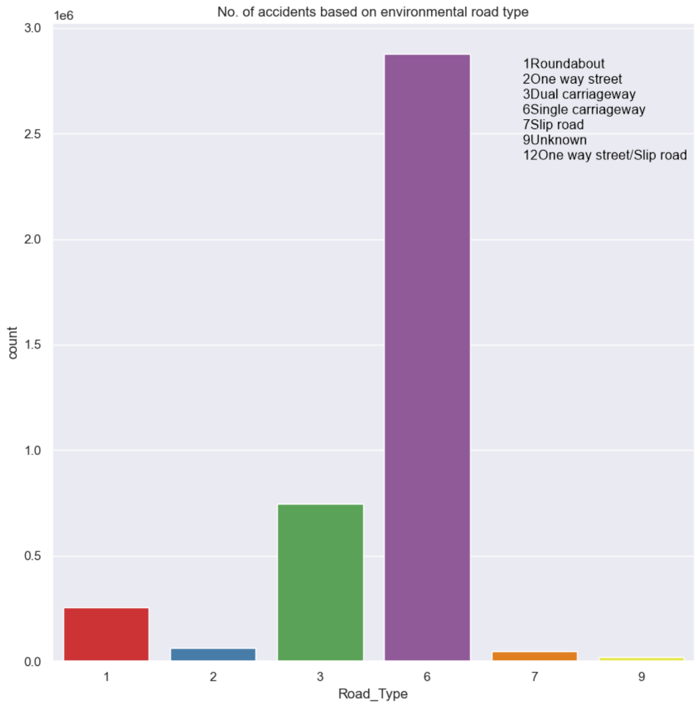

Fig. 8. Number of accidents based on Road Type

Considering the Road Type in Figure 8, to be observed that the most cases are registered on single carriageway, followed by dual carriageways, with almost a quarter of the cases mentioned beforehand.
The Analysis implicates that, in certain cases, the weather, road state, light conditions were not extreme, therefore the avoidance of accidents could have been higher. This implicates 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 in reality, perform 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 reach 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 on, Figure 9. This stops the ensemble from generating same errors and biases [9].

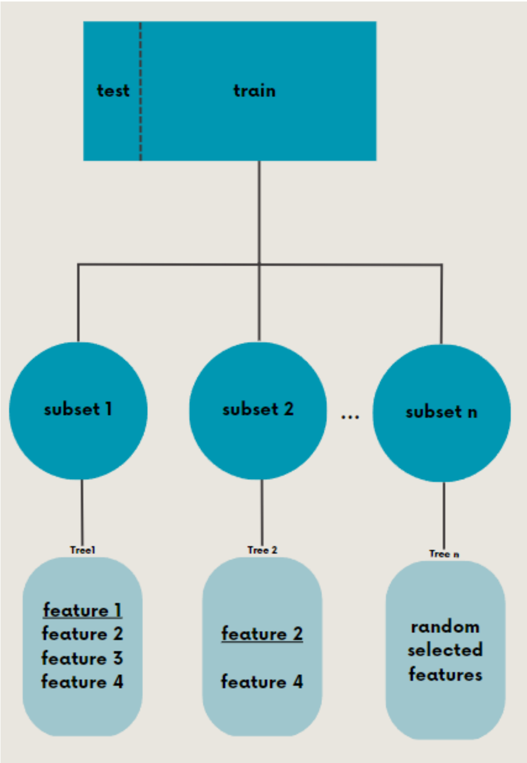

Fig. 9. Random Forest spread of decision trees and data

In the upcoming analysis this paper uses the Random Forest Regressor model based on a series of independent and dependent variables, with the main focus on the gravity of the casualty.
Therefore, the dependent variable is considered to be, 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 module train_test_split of sklearn.model_selection library, while considering a 20% test size and remaining 80% training size.

With the module Random Forest Regressor from 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 looking at the first decision tree, Figure 10.

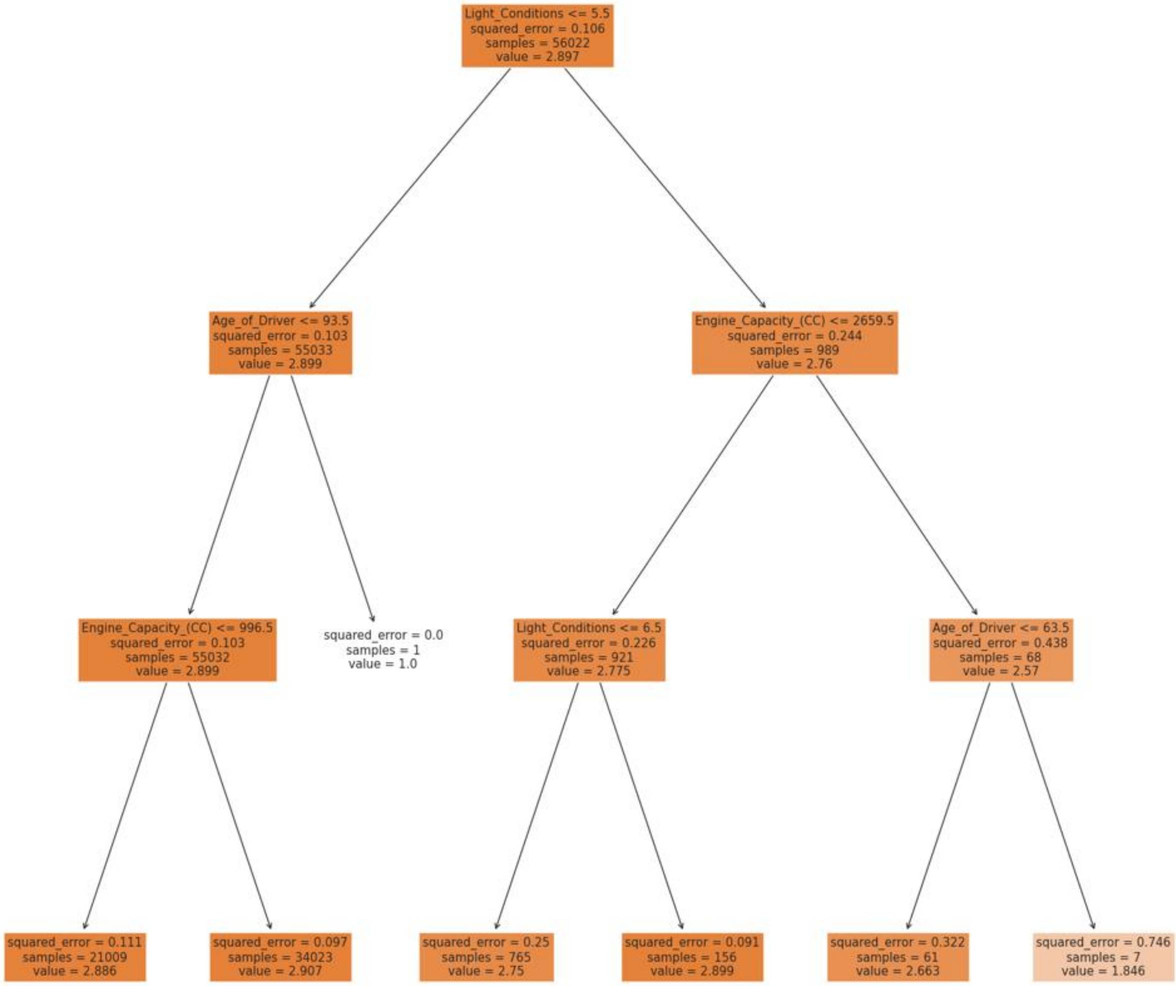

Fig. 10. First decision tree with a set depth

As it 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 resulted graph is shown in Figure 12.

```
Variable: Engine_Capacity_(CC) Importance: 0.49
Variable: Light_Conditions     Importance: 0.41
Variable: Road_Type            Importance: 0.07
Variable: Weather_Conditions   Importance: 0.03
Variable: Age_of_Driver        Importance: 0.0
Variable: Road_Surface_Conditions Importance: 0.0
```

Fig. 11. Importance of variables calculation

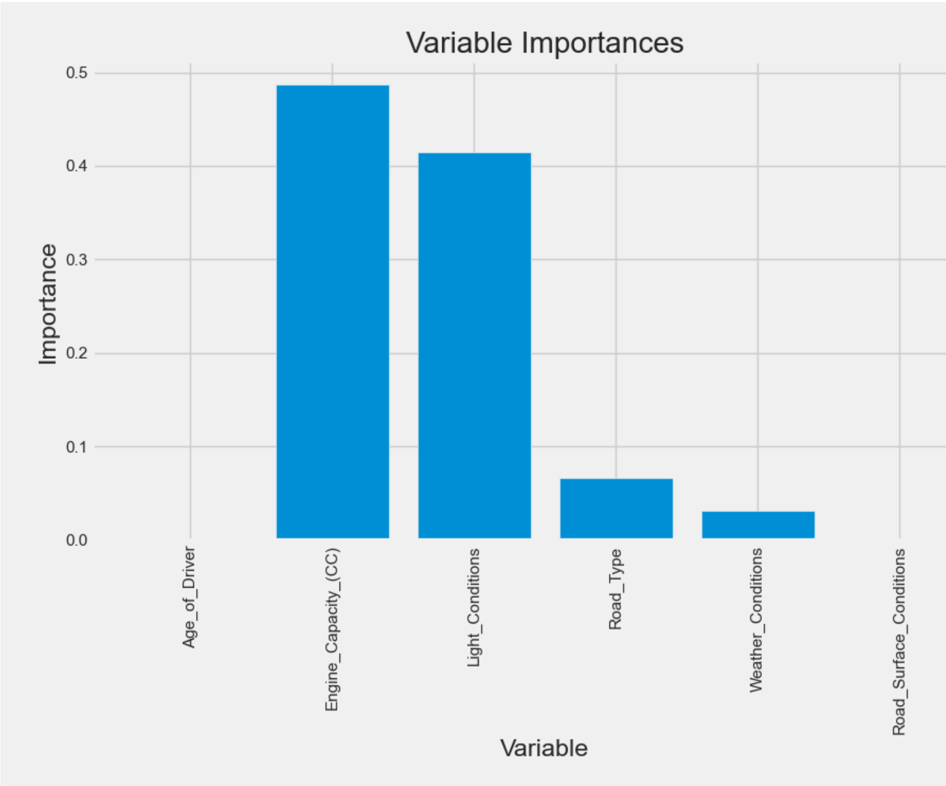

Fig. 12. Importance of variables graph

The forest's predict 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 which are inspired by the natural evolution and behavior of organisms. The core concept that lays 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 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 commonly used SI techniques, PSO (Particle Swarm Optimization).

5.1 Particle Swarm Optimization (Swarm intelligence)

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 implementations 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, as it can be seen in the following code snippet.

```
# Initialize the swarm
swarm_position = np.random.rand(num_particles, 4) # assuming 4 hyperparameters
swarm_velocity = np.zeros((num_particles, 4))
swarm_best_position = swarm_position.copy()
swarm_best_fitness = np.zeros(num_particles)
```

Evaluation: Evaluate the fitness or objective function value for each particle based on its current position in the search space. In the following code fragment, the fitness function quantifies the quality of the particle's position in relation to the optimization problem being solved.

```
def fitness_function(train_features, train_labels , test_features, test_labels, hy-
perparameters):
    # Instantiate the Random Forest Regressor with the given hyperparameters
    n_estimators=int(round(hyperparameters[0]))
    max_depth_value=int(round(hyperparameters[1]))
    min_samples_leaf=int(round(hyperparameters[2]))
    max_features=int(round(hyperparameters[3]))
    random_forest = RandomForestRegressor(n_estimators=n_estimators,
                                          max_depth=max_depth_value,
                                          min_samples_leaf=min_samples_leaf,
                                          max_features=max_features)
    # Train the model
    random_forest.fit(train_features, train_labels)

    # Make predictions on the testing data
    y_pred = random_forest.predict(test_features)

    # Calculate R-squared
    r_squared = r2_score(test_labels, y_pred)

    # Return R-squared as the fitness value
    return r_squared
```

Update personal best: Update the personal best position and fitness value for each particle. The personal best represents the best solution found by each particle individually up to the current iteration.

Update global best: Identify the particle with the best fitness value among all the particles in the swarm. This particle's position is considered the global best position.

Updates of personal best and global best are calculated in the following lines of code.

```
# Update particle's best position and fitness
if fitness > swarm_best_fitness[i]:
       swarm_best_fitness[i] = fitness
       swarm_best_position[i] = swarm_position[i].copy()
```

```
# Update global best position and fitness
if fitness > global_best_fitness:
      global_best_fitness = fitness
      global_best_position = swarm_position[i].copy()
```

Update velocity and position: In the code fragment below, it is displayed the update of the velocity and position of each particle based on its current velocity, personal best position, and global best position. 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.

```
# Update particle's velocity and position using PSO equations
inertia_weight = 0.8
cognitive_weight = 1.5
social_weight = 1.5
swarm_velocity[i] = (inertia_weight * swarm_velocity[i] + cognitive_weight * np.random.rand() * (swarm_best_position[i] - swarm_position[i]) +social_weight * np.random.rand() * (global_best_position - swarm_position[i]))
swarm_position[i] += swarm_velocity[i]
```

Iteration: Repeat steps 2 to 5 for a specified number of iterations or until a termination condition is met. The particles move through the search space, adjusting their positions based on their velocities and experiences.

Termination: 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 improving the previous method by obtaining an R-squared of 0.024, as seen in Figure 12.

```
Best Hyperparameters: [  9.80831626  54.3493251   13.07769995 -50.06901871]
Best R-squared: 0.024795373176329405
Model's R-squared on Testing Data: 0.02385826743451669
```

Fig. 12. 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 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 the increase of R-squared variable and thus a better explanation of the variation of Casualty Severity data. By predicting the variation of car accidents number 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 Europ 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 Informatic Economics. She works as a Full Stack Software Engineer in the Automotive field, at Harman International. Currently pursuing PhD research focused on 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.