Scientific Journal of Silesian University of Technology. Series Transport
Zeszyty Naukowe Politechniki Śląskiej. Seria Transport

Volume 127 **2025**

p-ISSN: 0209-3324

e-ISSN: 2450-1549

DOI: https://doi.org/10.20858/sjsutst.2025.127.8

Journal homepage: http://sjsutst.polsl.pl

Article citation information:
Le, K.G. Improving road safety: supervised machine learning analysis of factors influencing crash severity. *Scientific Journal of Silesian University of Technology. Series Transport*. 2025, **127**, 129-153. ISSN: 0209-3324. DOI: https://doi.org/10.20858/sjsutst.2025.127.8

Khanh Giang LE[1]

IMPROVING ROAD SAFETY: SUPERVISED MACHINE LEARNING ANALYSIS OF FACTORS INFLUENCING CRASH SEVERITY

Summary. Road traffic crash severity is shaped by a complex interplay of human, vehicular, environmental, and infrastructural factors. While machine learning (ML) has shown promise in analyzing crash data, gaps remain in model interpretability and region-specific insights, particularly for the UK context. This study addresses these gaps by evaluating supervised ML models – Decision Tree, Support Vector Machine (SVM), and LightGBM – to predict crash severity using 2022 UK accident data. The research emphasizes interpretability through SHapley Additive exPlanations (SHAP) to identify critical factors influencing severity outcomes. Results demonstrate that LightGBM outperforms other models in predictive performance, with police officer attendance at the scene, speed limits, and the number of vehicles involved emerging as pivotal determinants of severity. The analysis reveals that higher speed limits and single-vehicle collisions correlate with severe outcomes, while police presence may mitigate accident severity. However, the study acknowledges limitations, including dataset constraints. By integrating ML with post-hoc interpretability techniques, this work advances actionable insights for policymakers to prioritize road safety interventions, such as optimizing enforcement strategies and revising speed regulations. The findings underscore the potential of interpretable ML frameworks to enhance understanding of crash dynamics and inform targeted safety measures, contributing to global efforts to reduce traffic-related fatalities and injuries.

---

[1] Faculty of Civil Engineering, University of Transport and Communications, Hanoi, Vietnam. Email: gianglk@utc.edu.vn. ORCID: https://orcid.org/0000-0002-6295-8578

Keywords: road traffic crash, prediction, machine learning, classification, severity

1. INTRODUCTION

The severity of road traffic crashes is influenced by a complex interplay of factors such as human behavior, vehicle characteristics, road conditions, and environmental factors [1]. Additionally, poor road design and maintenance significantly contribute to traffic accidents, compromising overall road safety [2]. There are various approaches to analyzing traffic crashes, such as historical crash data analysis, crash site analysis, safety surrogate measures, crash reconstruction, safety effectiveness evaluation, and crash prediction models. Among these, leveraging historical crash data for crash analysis, assessment, and prediction remains one of the most widely adopted methods. Understanding the nature of traffic crashes, identifying the key factors influencing their severity, and developing accurate crash prediction models are essential steps toward building a safer and more efficient transportation system [3].

Accurate crash predictions are essential for understanding the main causes of road traffic crashes and devising effective solutions to minimize their impact [4]. This involves analyzing a vast accident database covering various factors such as road users, vehicles, roadways, and environment [5]. Statistical and artificial intelligence models are employed to examine the interactions between these factors, with artificial intelligence models gaining popularity due to their ability to handle large datasets and identify complex interactions [6]. Machine learning, a data-driven approach and a branch of artificial intelligence, plays a key role in data analysis and decision-making, enabling computers to learn and make decisions autonomously with minimal human intervention [3].

Nowadays, machine learning has been widely applied in many fields, including road safety. In this field, it has been utilized for various purposes such as identifying crash-prone road locations [7], assessing injury severity [8], analyzing the role of road users in crashes [9], evaluating the role of road types in crashes [10], exploring the mechanism of crashes with autonomous vehicles [11], evaluating the impact of factors like alcohol consumption [12] and environmental conditions [13]. There are various machine learning models, including supervised, unsupervised, semi-supervised, and reinforced learning categories [14]. This study focuses specifically on supervised machine learning models, as they have shown promise in predicting crash severity and identifying contributing factors [15]. Accurately predicting crash severity aids in the timely management of traffic safety and the implementation of effective strategies [1]. Supervised learning is further divided into regression and classification methods [16].

Classification in machine learning is a form of supervised learning where the dataset consists of both input features and corresponding class labels. The model is trained on this labeled dataset to recognize patterns and predict the class of new instances. Classification methods are particularly effective in handling large-scale data and serve as a key data mining technique for categorizing information into distinct groups while extracting meaningful insights. By grouping datasets with similar characteristics, classification enables the development of predictive models that accurately assign class labels. In essence, classification involves determining the most appropriate category for each data instance based on learned patterns [27].

Classification methods have been widely applied in crash studies. In order to classify collisions into three categories – fatal, non-fatal, and Property Damage Only (PDO) – the study

[15] used Deep Neural Networks (DNN) and tree-based classifiers. While the Decision Tree (DT) and Random Forest (RF) performed well for other categories, the results indicated that DNN was more accurate in predicting fatal crashes.

Similarly, in order to categorize accident severity, the study [17] compared the performance of five machine learning classifiers – K-Nearest Neighbor (KNN), Multilayer Perceptron, DT, Support Vector Machine (SVM), and Naïve Bayes – against the traditional Logistic Regression (LR). According to their results, the Multilayer Perceptron, KNN, and DT performed better than the others. Additionally, they found that two factors that significantly affect class prediction are traffic control and ground surface temperature.

The study [18] observed that DT, KNN, SVM, evolutionary algorithms, and Artificial Neural Networks (ANN) are frequently used in safety models. The study [19] found that the linear regression model's goodness-of-fit and prediction accuracy were comparatively low, and it was insufficient in explaining the impact of most variables. Additionally, traffic management organizations can prevent or mitigate secondary accidents by using the back-propagation neural network model to forecast the time interval between primary and secondary incidents.

In recent years, the Light Gradient Boosting Machine (LightGBM) is a cutting-edge tree-based ensemble learning method known for its high predictive accuracy, rapid training speed, and efficient memory usage, making it particularly suitable for research involving large datasets [20]. Although previous research has extensively utilized machine learning algorithms for crash analysis, a significant gap remains in providing clear explanations of how these models work and the factors influencing their predictions. While traditional statistical models often rely on predefined assumptions, machine learning models offer a more flexible approach that does not require predetermined relationships between variables, making them more suitable for crash analysis.

Despite their superior accuracy, machine learning-based classifiers face challenges in transparency and interpretability. In the context of road safety, where decisions can range from preventing minor accidents to saving lives, understanding how these models make predictions is crucial. By shedding light on the factors influencing the model's predictions and classification results, decision-making processes can be improved, and our understanding of road safety can be deepened. Moreover, there is a lack of comprehensive, data-driven crash severity analysis specifically addressing road systems in the United Kingdom (UK), which underscores the need for further research in this area.

This study aims to fill this gap by investigating the effectiveness of several supervised machine learning models, including Decision Tree, Support Vector Machine, and LightGBM in predicting crash severity within the UK context. There has been no study evaluating the effectiveness of these three algorithms simultaneously in predicting crash severity. By investigating different machine learning algorithms for crash severity identification, comparing their results, applying post-hoc techniques to interpret machine learning models and their predicted classes, and highlighting the variables affecting crash severity in the UK context, this study introduces new components. The findings of this study will contribute to the improvement of road safety by providing better predictive tools and deeper insights into the factors that affect crash severity.

The machine learning workflow defines the specific work steps in implementing machine learning. However, depending on each requirement and each project, there are different machine learning workflows [21]. This study is implemented, including the following six basic steps, as shown in Fig. 1.

- ❖ Data collection: Gather data from various sources, ensuring quality and representativeness.

- Data preprocessing: This step involves cleaning the data, handling missing values, normalizing or standardizing features, and performing feature selection or engineering to enhance the dataset's quality. Proper preprocessing ensures that the machine learning model receives meaningful and well-structured input.
- Training the model:
  - Once the data is preprocessed, a machine learning model is selected and trained using historical data.
  - The model learns patterns from the data through optimization techniques.
  - The choice of algorithm depends on the problem type (classification, regression, clustering, etc.) and dataset characteristics.
- Evaluating model:
  - After training, the model's performance is assessed using validation techniques such as cross-validation, accuracy scores, precision-recall analysis, or other evaluation metrics.
  - If the model does not meet the desired performance criteria, further improvements are required. This is a crucial step to ensure the model generalizes well to unseen data.
- Improving model (if needed):
  - If the evaluation results indicate poor performance, model optimization techniques are applied.
  - This may include adjusting hyperparameters, using more advanced algorithms, gathering more training data, or feature engineering.
  - The cycle of training, evaluation, and improvement continues iteratively until a satisfactory performance is achieved.
- Using Model:
  - Once the model achieves acceptable accuracy and reliability, it is deployed for real-world applications.
  - The model is integrated into a system where it makes predictions on new data and provides insights or automated decisions.
  - Continuous monitoring and updating may be required to maintain model effectiveness over time.

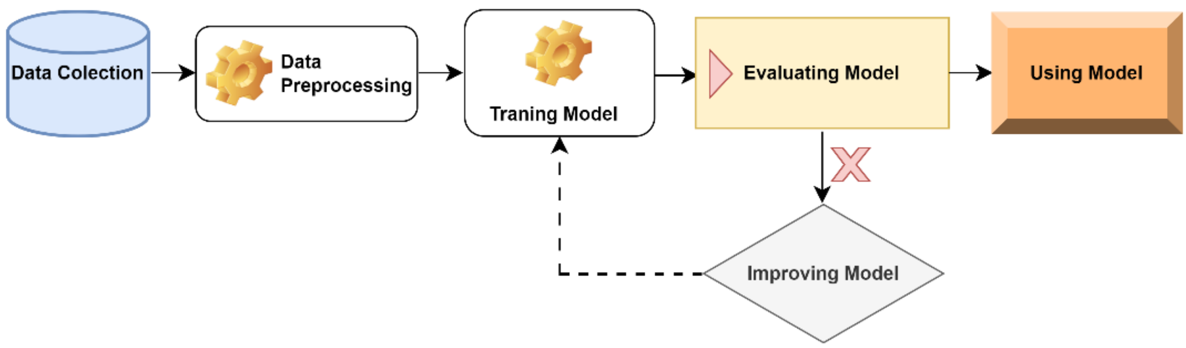

Fig. 1. Machine learning workflow

The remainders of the paper are arranged as follows. Section 2 presents study area and data. Section 3 presents research methods. Section 4 illustrates results and discussions. In Section 5, conclusions are presented.

2. RESEARCH AREA AND DATA COLLECTION

2.1. Research Area

The total area of the United Kingdom is 244,376 square kilometers, with an estimated population of nearly 67.6 million people in 2022. The road infrastructure spans approximately 422,100 kilometers of paved roads, with 396,700 kilometers located in Great Britain (comprising England, Scotland, and Wales) and an additional 25,500 kilometers in Northern Ireland. Great Britain had 40.8 million licensed automobiles in total in 2022. The main means of transport in the UK include cars, buses, coaches, vans, taxis, motorcycles, pedal cycles, and other vehicles. Vehicles are driven on the left in the UK, and drivers are legally required to stay in the left lane on multilane carriageways unless overtaking or turning right. Speed limits in the UK range from 20 mph (32 km/h) to 70 mph (113 km/h).

In the UK, road systems are grouped into five categories, including: Motorways; A roads: Major transport links within or between areas; B roads: Connect different areas and link A roads to smaller roads; Classified unnumbered roads (C roads): Smaller roads linking unclassified roads to A and B roads; Unclassified roads: Local roads for local traffic, making up 60% of the UK's road network.

2.2. Data Collection

This study utilizes a dataset from data.gov.uk, which includes records of road traffic accidents reported by the UK's Department for Transport in 2022 [22]. The original dataset comprises 106,004 records, including the target variable accident_severity and several independent variables related to accident_reference, road conditions, environmental factors, and vehicle involvement. Key features considered in the dataset include number_of_vehicles, number_of_casualties, day_of_week, first_road_class, first_road_number, road_type, speed_limit, junction_detail, junction_control, light_conditions, weather_conditions, road_surface_conditions (RSC), did_police_officer_attend_scene_of_accident (PASA), trunk_road_flag, urban_or_rural_area, and special_conditions_at_site (SCAS). Descriptive statistics for these independent variables are presented in Tab. 1.

Tab. 1

Descriptive statistics of the independent variables

Statistics / Variables	Mean	Standard deviation	Standard error	Minimum	Maximum
Number_of_vehicles	1.825	0.688	0.002	1.000	16.00
Number_of_casualties	1.278	0.699	0.002	1.000	16.00
Day_of_week	4.169	1.940	0.005	1.000	7.000
First_road_class	4.222	1.465	1.000	6.000	0.004
First_road_number	784.8	1576	4.841	0.000	9176
Road_type	5.252	1.704	0.005	1.000	9.000
Speed_limit	35.96	14.21	0.043	20.00	70.00
Junction_detail	4.016	12.83	0.039	0.000	99.00
Junction_control	1.714	2.502	0.007	-1.00	9.000
Light_conditions	2.010	1.689	0.005	-1.00	7.000

Weather_conditions	1.636	1.851	0.005	1.000	9.000
RSC	1.346	0.972	0.002	-1.00	9.000
SCAS	0.242	1.345	0.004	-1.00	9.000
Urban_or_rural_area	1.323	0.468	0.001	1.000	3.000
PASA	1.481	0.766	0.002	1.000	3.000
Trunk_road_flag	1.725	0.787	0.002	-1.00	2.000

3. METHODS

Figure 2 illustrates the process of training the model in this study. This process has three main steps, including 3 steps.

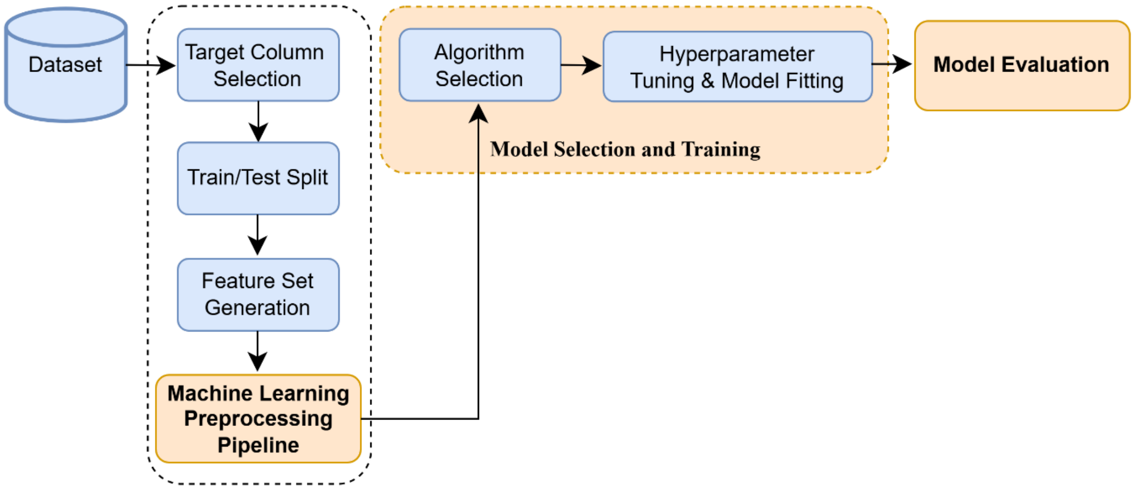

Fig. 2. The process of training model

Step 1: Machine learning preprocessing pipeline

Before model training, data must be preprocessed to ensure its quality and suitability for machine learning algorithms. This process includes several key stages:

- Handling missing data: Records with incomplete data are either removed or imputed using the mean, median, or mode of the respective feature [28].
- Categorical data encoding: Categorical variables are converted into numerical representations using either Label Encoding or One-Hot Encoding [29].
- Target variable selection: Identifying the dependent variable for classification.
- Train/Test split: The dataset is divided into training (80%) and testing (20%) sets using 10-fold cross-validation to ensure robustness in evaluation [30].
- Feature set generation: Selection of the most relevant attributes from the dataset.

Step 2: Model selection and training

After preprocessing, selecting the appropriate algorithm is crucial. The choice of algorithm depends on the problem type and the characteristics of the dataset. The following steps are followed:

- Algorithm selection: Various classification algorithms, such as Decision Trees, Random Forest, and Support Vector Machines (SVM), are selected.

- Hyperparameter tuning & Model fitting: Optimization of model parameters using grid search and cross-validation to enhance prediction accuracy and prevent overfitting [30].

Step 3: Model evaluation

The trained models are evaluated on the test dataset using multiple performance metrics, including [31]:
- Accuracy: Measures the proportion of correctly classified instances.
- Precision, recall, and F1-score: Evaluate the model's ability to correctly predict each accident severity class.
- Confusion matrix: Used to analyze misclassification rates across different severity levels.

3.1. Data Pre-Processing

In this stage, it's necessary to remove missing data, eliminate irrelevant attributes, label all data, encode features, and subsequently extract features, reducing the dataset while ensuring the quality of the dataset.

❖ Removing unnecessary features: Unnecessary features, such as accident_index and accident_reference, are removed to reduce redundancy and improve model efficiency because they have no impact on the prediction results of the traffic accident severity model.

❖ Handling missing data: The collected dataset does not contain any missing values.

❖ Categorical encoding: Each attribute was categorized as either categorical or numerical, depending on its inherent nature. In this dataset, categorical data has been encoded into numerical form to facilitate analysis using supervised machine learning algorithms, shown in Tab. 2. Typically, accident severity is divided into three classes and coded as follows: 1 for fatal; 2 for serious; and 3 for slight. Fig. 3 illustrates the distribution of accident severity.

❖ Target variable selection: accident_severity is the target variable for classification.

❖ Feature set generation: Selection of the most important features from the dataset.

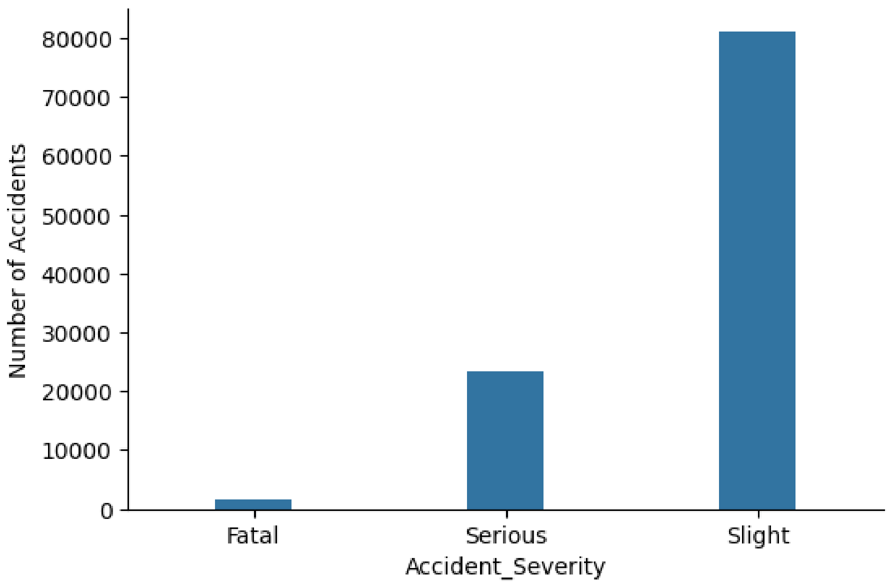

Fig. 3. Distribution of accident severity

Tab. 2

Describe accident features and encode categorical variables

Variables	Description	Encoded variables
Accident_severity	The severity level of the accident	1=Fatal, 2=Serious, 3=Slight
Day_of_week	The day of the week the accident occurred	1=Sunday, 2=Monday, 3=Tuesday, 4=Wednesday, 5=Thursday, 6=Friday, 7=Saturday
First_road_class	The type of road where the accident occurred	1=Motorway, 2=A (major) road, 3=A (minor) road, 4=B road, 5=C road
Number_of_vehicles	The number of vehicles involved in the accident	Numerical data
Number_of_casualties	The number of injured or deceased individuals	Numerical data
First_road_number	The road number where the accident occurred	Numerical data
Road_type	The road layout	1=Roundabout, 2=One way street, 3=Dual carriageway, 6=Single carriageway, 7=Slip road, 12=One way street
Speed_limit	The speed limit	20, 30, 40, 50, 60, 70 are the only valid speed limits on public highways
Junction_detail	The type of junction where the accident occurred	1=Roundabout, 2=Mini-roundabout, 3=T or staggered junction, 5=Slip road, 6=Crossroads, 7=More than 4 arms (not roundabout), 8=Private drive or entrance
Junction_control	The traffic control at the junction	1=Authorized person, 2=Auto traffic signal, 3=Stop sign, 4=Give way or uncontrolled
Light_conditions	The lighting conditions at the time of the accident	1=Daylight, 4=Darkness - lights lit, 5=Darkness - lights unlit, 6=Darkness - no lighting, 7=Darkness – lighting unknown
Weather_conditions	The weather conditions at the time of the accident	1=Fine no high winds, 2=Raining no high winds, 3=Snowing no high winds, 4=Fine + high winds, 5=Raining + high winds, 6=Snowing + high winds, 7=Fog or mist
Road_surface_conditions	The condition of the road surface	1=Dry, 2=Wet or damp, 3=Snow, 4=Frost or ice, 5=Flood over 3cm deep, 6=Oil or diesel, 7=Mud

Special_conditions_at_site	Special conditions at the accident scene	0=None, 1=Auto traffic signal-out, 2=Auto signal part defective, 3=Road sign, 4=Roadworks, 5=Road surface defective, 6=Oil or diesel, 7=Mud
Trunk_road_flag	Whether the accident occurred on a trunk road	1=Trunk (Roads managed by Highways England), 2=Non-trunk
Urban_or_rural_area	The area where the accident occurred	1=Urban, 2=Rural
Did_police_officer_attend_scene_of_accident	Whether a police officer attended the scene	1=No, 2=Yes

Handling numerous features can significantly impact model performance due to the exponential increase in training time and the heightened risk of overfitting. Consequently, certain redundant or unnecessary features were eliminated to streamline the model and improve its functionality [23]. After preprocessing the data, it is necessary to select the most important features for training the model [24].

The Random Forest algorithm was utilized to identify the most influential features based on their correlation with accident severity. This process ensures that only the most relevant attributes are retained for model training. Random Forest, an ensemble learning method, constructs multiple decision trees and aggregates predictions. The following steps were used to identify key features for accident severity prediction [25]:

Step 1: Random Forest fundamentals: Build an ensemble of decision trees to predict accident severity. A Random Forest consists of *M* decision trees. Each tree *m* is trained as follows:

[1] Bootstrap sampling:
- Randomly select *N* samples with replacement from the training set (*N*: total samples).
- This creates a subset D_m for tree m.

[2] Random feature selection:
- At each node, select *d* features randomly from *D* total features *(d ≤ D).*
- Typical choices: $d = \sqrt{D}$ (classification).

[3] Tree construction:
- Split nodes using Gini impurity (for classification).
- Stop when reaching maximum depth or minimum samples per node.

[4] Aggregation of predictions:
- Classification: Majority voting across all trees.
- The formula used for classification in a Random Forest is:

$$\hat{y} = argmax_c \sum_{m-1}^{M} 1(f_m(x) = c) \quad (1)$$

Where:
$\hat{y}$ is the final predicted class label for input x.
c is one of the possible classification labels.
$argmax_c$ is the operator that finds the value of c (the classification label) with the highest total votes. In other words, this is the label most frequently predicted by the trees in the forest.
m is the index of the decision tree in the forest, ranging from 1 to *M*.
M is the total number of trees in the Random Forest.

$f_m(x)$ is the prediction function of tree *m* for input *x*.
$1(f_m(x) = c)$ is the indicator function, which has the following values:

- 1, if tree *m* predicts that *x* belongs to class *c*;
- 0, otherwise.

The sum $\sum_{m-1}^{M} 1(f_m(x) = c)$ counts the number of trees in the forest that predict *x* belongs to class *c*.

Step 2: Feature importance calculation
[1] Gini importance
(a) Gini impurity at node *t*:

$$G(t) = 1 - \sum_{i=1}^{C} p_i^2 \tag{2}$$

Where:
C: Number of classes (e.g., fatal, serious, minor).
p_i: Proportion of samples in class *i* at node *t*.

(b) Gini reduction for feature x_j at node *t*:

$$\Delta G(x_j, t) = G(t) - \left(\frac{N_{left}}{N} G(t_{left}) + \frac{N_{right}}{N} G(t_{right})\right) \tag{3}$$

Where:
N_{left}, N_{right}: Number of samples in left/right child nodes.
N: Number of samples at parent node *t*.

(c) Total Gini importance for feature x_j:

$$GI(x_j) = \frac{1}{M} \sum_{m=1}^{M} \sum_{t \in T_m} \Delta G(x_j, t) \tag{4}$$

Where:
T_m: Set of nodes in tree *m*.

[2] Threshold for feature selection
(a) Sort features: Rank features by descending importance:

$$I_1 \geq I_2 \geq \cdots \geq I_n \tag{5}$$

(b) Cumulative importance calculation:

$$C_k = \frac{\sum_{i=1}^{k} I_i}{\sum_{i=1}^{n} I_i}, k = 1,2, \dots, n \tag{6}$$

(c) Threshold identification: In study [26], an 80% threshold was proposed. However, in my research, selecting an 80% threshold would eliminate many important features. Therefore, a 90% threshold was chosen as it better suits the objectives of this study.

Find the smallest k such that:

$$C_k \geq 0.9 \tag{7}$$

Retain the top k features.

Where:
I: The importance score of each feature.
C_k: The normalized cumulative sum up to the current feature.

3.2. Decision Tree

A Decision Tree is a hierarchical model that classifies data by recursively splitting it into subsets based on feature values. Each internal node represents a decision rule, each branch corresponds to an attribute value, and each leaf node represents a class label. At each node, the algorithm selects the best feature to split the data using a criterion such as Gini Index, Entropy, or Information Gain. The tree continues to grow until a stopping criterion is met (e.g., max depth, minimum samples per node) [32]. In this study, the Classification and Regression Trees (CART) algorithm was applied to segment the data and construct a tree that maximizes the homogeneity of the dependent variable's values within the nodes [33].

❖ Impurity measure (Gini index):
In classification problems, CART commonly uses the Gini index to evaluate the impurity of a node. The Gini index (G) is defined as shown in Equation (2). The Gini Index reaches its minimum (zero) when all samples in the node belong to a single class, indicating a pure node.

❖ Gini decrease:
To decide the best split at each node, the algorithm calculates the reduction in impurity from splitting the node into two child nodes. This reduction, quantified as the Gini decrease (ΔG), is computed as shown in Equation (3). A higher ΔG indicates a more effective split in terms of class separation.

❖ Detailed workflow for accident severity prediction:
Step 1: Initialize the root node: Start with the entire dataset at the root node.
Step 2: Evaluate splitting criteria for each feature
- For every feature and possible threshold value, compute G for the potential split.
- Calculate ΔG using the formula mentioned above.

Step 3: Select the best split and partition the data
- Choose the feature and threshold with the highest ΔG (greatest impurity reduction).
- Split the dataset into two child nodes based on this decision rule.

Step 4: Recursive splitting
- For each child node, repeat Step 2 and Step 3.
- Continue this process recursively until a stopping condition is met.
- Stopping conditions: The maximum tree depth is reached; The minimum number of samples per leaf is reached; No further impurity reduction is possible.

Step 5: Assign class labels to leaf nodes
- Once no further splits are made, assign a class label to each leaf node.
- The label is typically determined by the majority class of the samples within that leaf.

- For accident severity prediction, each leaf node will be labeled as fatal, serious, or slight, depending on which severity level is most prevalent among the samples in that node.

3.3. Support Vector Machine (SVM)

SVM is a classification algorithm that finds the optimal hyperplane to separate different classes. Fig. 4 illustrates the process of SVM. The optimal hyperplane in SVM is the one that maximizes the distance from both classes. SVM aims to achieve this by evaluating various hyperplanes that best classify the labels, and then selecting the one with the greatest margin from the data points [34].

❖ Hyperplane equation:

The decision boundary in SVM is defined by a hyperplane, which can be expressed as:

$$w^T x + b = 0 \tag{8}$$

Where: w is the weight vector; w^T is the transpose of the weight vector; x is the feature vector; b is the bias term.

Data points are classified based on the sign of the value $w^T x + b$.

❖ Margin and optimal hyperplane:

The margin is the distance between the hyperplane and the nearest data points from any class, known as support vectors. SVM aims to maximize this margin, which can be mathematically formulated as:

$$Margin = \frac{2}{\|w\|} \tag{9}$$

Where: $\|w\|$ is the norm of the weight vector w, representing the magnitude of this vector.

Maximizing the margin is equivalent to minimizing $\|w\|^2$.

❖ Optimization problem:

For a linearly separable case, SVM finds the optimal hyperplane by solving the following constrained optimization problem. The objective function is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \tag{10}$$

subject to:

$$y_i(w^T x_i + b) \geq 1,\ \forall i \tag{11}$$

Where: y_i represents the class label for the data point x_i, where y_i is typically +1 or −1.

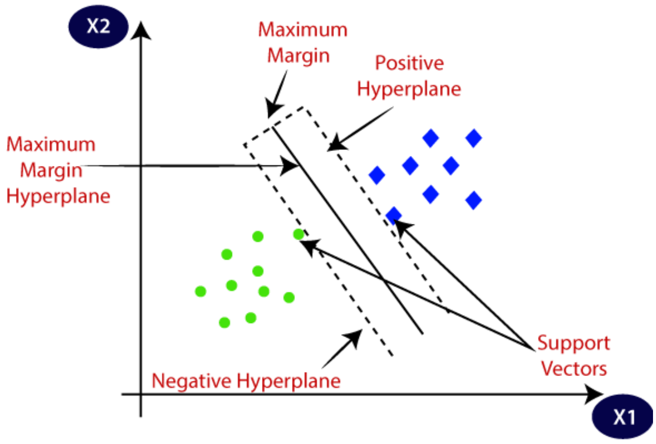

Fig. 4. The process of SVM [35]

3.4. LightGBM

LightGBM is an effective and efficient open-source gradient boosting framework designed for machine learning. It excels at handling large datasets while maintaining high performance in speed and memory efficiency. LightGBM utilizes gradient boosting, a technique that merges multiple weak learners, typically decision trees, to form a robust predictive model. One potential drawback of LightGBM is its sensitivity to hyperparameters. While LightGBM offers various hyperparameters for fine-tuning model performance, selecting the optimal values can be challenging and may require extensive experimentation [20].

❖ LightGBM is a gradient boosting framework that uses decision trees. For a multi-class classification problem (3 classes), the objective function at iteration *t* is defined as:

$$L(t) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \quad (12)$$

Where:
y_i: The actual label of sample *i* (fatal/serious/slight).
$\hat{y}_i^{(t-1)}$: The accumulated prediction from previous trees.
f_t: The decision tree added at iteration *t*.
$\Omega(f_t)$: The regularization function to prevent overfitting:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (13)$$

Where:
T: Number of leaves in the tree.
w_j: Output value at leaf j.
γ, λ: Hyperparameters that control the complexity of the tree.

This formulation ensures that LightGBM optimizes the model by minimizing loss while maintaining regularization to prevent overfitting.

❖ Loss function for multi-class classification: For a 3-class classification problem, the Cross-Entropy Loss is used:

$$l(y_i, \hat{y}_i) = -\sum_{c=1}^{3} y_{i,c} \log(p_{i,c}) \tag{14}$$

Where:
$y_{i,c}$: Equals 1 if sample i belongs to class c, otherwise 0.
$p_{i,c}$: The predicted probability for class c, computed as:

$$p_{i,c} = \frac{\exp(\hat{y}_{i,c})}{\sum_{k=1}^{3} exp(\hat{y}_{i,k})} \tag{15}$$

This formulation ensures that the model optimizes the predicted probabilities for multi-class classification.

3.5. Model Evaluation

The aim of constructing a predictive model is to ensure its accuracy when applied to new, unseen data. This is achieved through the use of statistical techniques, wherein the training dataset is meticulously chosen to gauge the model's efficacy on novel and unexplored data. A fundamental approach to validating the model involves partitioning a segment of the labeled data, which is reserved for assessing the model's ultimate performance. Maintaining the statistical integrity of the data during this split is crucial. It necessitates that both the training and test datasets possess similar statistical properties to the original data to prevent bias in the trained model. In this study, the labeled dataset was divided into an 80% training set and a 20% testing set. The efficacy of each model was sequentially evaluated to compare their performance regarding metrics such as confusion matrix, sensitivity, and specificity for accident severity. The model's performance was assessed using various criteria derived from the confusion matrix. This matrix provides a range of evaluation metrics, including accuracy, which represents the proportion of correct predictions and is computed as follows [36]:

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{16}$$

Where:
TP (True Positives): The number of samples correctly predicted as the positive class.
TN (True Negatives): The number of samples correctly predicted as the negative class.
FP (False Positives): The number of samples incorrectly predicted as the positive class (actually negative).
FN (False Negatives): The number of samples incorrectly predicted as the negative class (actually positive).

Precision, defined as the ratio of correctly identified positive cases to the total predicted positive cases, is calculated as follows:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{17}$$

Recall, or sensitivity, is the ratio of correctly identified actual positive cases to the total actual positive cases, and it is calculated as follows:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{18}$$

The F1 score, which measures the balance between precision and recall, is computed as follows:

$$F1 \text{ score} = 2 \text{ x } \frac{\text{Recall x Precision}}{\text{Recall + Precision}} \tag{19}$$

Moreover, other various evaluation metrics are utilized to assess the performance of each classifier model, including Cohen's Kappa, Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Relative Absolute Error (RAE), and Root Relative Squared Error (RRSE). To analyze and draw conclusions, we adapt the confusion matrix, which compares actual results (rows of the table) with model predictions (columns of the table). This allows us to scrutinize each algorithm by examining the number of instances correctly or incorrectly predicted.

Cohen's Kappa is a statistical measure that evaluates the accuracy of a classification model by comparing the level of agreement between the model's predictions and actual values, adjusting for the possibility of random agreement, is computed as follows:

$$\text{Cohen's Kappa} = \frac{P_0 - P_e}{1 - P_e} \tag{20}$$

Where:
P_0 is the observed agreement (the proportion of correctly classified instances).
P_e is the expected agreement due to chance.
Cohen's Kappa values range from [-1, 1]:
Cohen's Kappa > 0.8: Almost perfect agreement.
Cohen's Kappa = 0.6–0.8: Strong agreement.
Cohen's Kappa < 0.4: Weak agreement.

Mean Absolute Error (MAE) measures the average absolute error between predicted values and actual values, is computed as follows:

$$\text{MAE} = \frac{1}{2}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{21}$$

Where:
y_i is the actual value.
$\hat{y}_i$ is the predicted value.
n is the number of samples.

A lower MAE indicates a more accurate model.

Mean Squared Error (MSE) measures the average squared error between actual and predicted values, penalizing larger errors more than MAE, is computed as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{22}$$

A lower MSE indicates a better model. Since MSE squares the errors, it is more sensitive to large deviations. The squared unit of MSE makes interpretation less intuitive compared to MAE.

Root Mean Squared Error (RMSE) is the square root of MSE, bringing the unit back to the original scale of the target variable. RMSE is more sensitive to large errors than MAE, is computed as follows:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{23}$$

Relative Absolute Error (RAE) compares the total absolute error of a model to the total absolute error of a simple baseline model (predicting the mean of actual values), is computed as follows:

$$\mathrm{RAE} = \frac{\sum|y_i - \hat{y}_i|}{\sum|y_i - \bar{y}|} \tag{24}$$

Where $\bar{y}$ is the mean of actual values.
If RAE < 1: The model performs better than a simple mean predictor.
If RAE > 1: The model performs worse than predicting the mean.

Root Relative Squared Error (RRSE) is a normalized version of RMSE, comparing the model's performance to a baseline model that predicts the mean of actual values), is computed as follows:

$$\mathrm{RRSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{25}$$

If RRSE < 1: The model is better than a simple mean predictor.
If RRSE > 1: The model is worse than a simple mean predictor.

3.6. Explaining the Model Using SHAP (Shapley Additive Explanations)

SHAP is a method that can explain predictive models both at an overall level and on a per-instance basis [37]. SHAP is based on game theory, where each feature in the model is considered a player contributing to the final outcome [38]. SHAP is widely recognized as a consistent approach for determining feature importance. Tree-SHAP, a variant of SHAP optimized for decision tree models, was utilized in this study. The Shapley value is computed using the following equations:

$$\mathrm{f}(\mathrm{y}') = \phi_0 + \sum_{i=1}^{N}\phi_i y_i' \tag{26}$$

Where:
f represents the explanation model.
N is the maximum size of the feature set.
ϕ_0 represents the expected output of the model when no features are included.
$\phi_i \in R$ denotes the feature attribution for feature *i*.
y' represents the simplified input feature representation used in SHAP calculations.
y_i' is an individual element of y', indicating whether a specific feature *i* is included in the current subset.

To compute the contribution of each feature, the Shapley formula is applied:

$$\Phi_i = \sum_{S\subseteq\{1,\ldots,p\}\setminus\{i\}} \frac{|S|!(p-|S|-1)!}{p!}[g_x(S\cup\{i\}) - g_x(S)] \quad (27)$$

Where:

$$g_x(S) = E[g(x)|x_S] \quad (28)$$

Where:
S represents a subset of input features.
x is the vector of feature values for a specific instance that needs to be explained.
p is the total number of features in the model.
$g_x(S)$ is the value function, which expresses the expected model output when only using the subset *S*.

4. RESULTS AND DISCUSSIONS

4.1. Results from Selecting Features

Selecting the most important features helps reduce model complexity while maintaining high performance, optimizing the prediction accuracy of traffic accident severity. Since feature selection is crucial for model performance, it is essential to analyze the importance of input variables before building a predictive model [24]. Identifying relevant features allows the model to focus on the most influential factors while discarding redundant or less significant attributes. This study examines the relationships between selected features and accident severity to enhance prediction accuracy and model efficiency.

In this study, the Random Forest algorithm was employed to identify the most influential features related to accident severity. This approach ensures that only the most relevant attributes are retained for model training.

Tab. 3

Feature importance and cumulative contribution

No	Feature	Importance (*I*)	Cumulative (C_k)	Result
1	first_road_number	0.3010	0.3010	Important features selected for predicted model
2	day_of_week	0.1528	0.4538	
3	junction_detail	0.0669	0.5208	
4	speed_limit	0.0630	0.5838	
5	light_conditions	0.0495	0.6333	
6	number_of_vehicles	0.0480	0.6813	
7	weather_conditions	0.0436	0.7249	
8	first_road_class	0.0420	0.7669	
9	number_of_casualties	0.0408	0.8077	
10	road_surface_conditions	0.0397	0.8474	
11	did_police_officer_attend_scene_of_accident	0.0372	0.8846	

12	road_type	0.0358	0.9204	
13	junction_control	0.0306	0.9510	Less important features
14	trunk_road_flag	0.0175	0.9685	
15	urban_or_rural_area	0.0166	0.9851	
16	special_conditions_at_site	0.0149	1.0000	

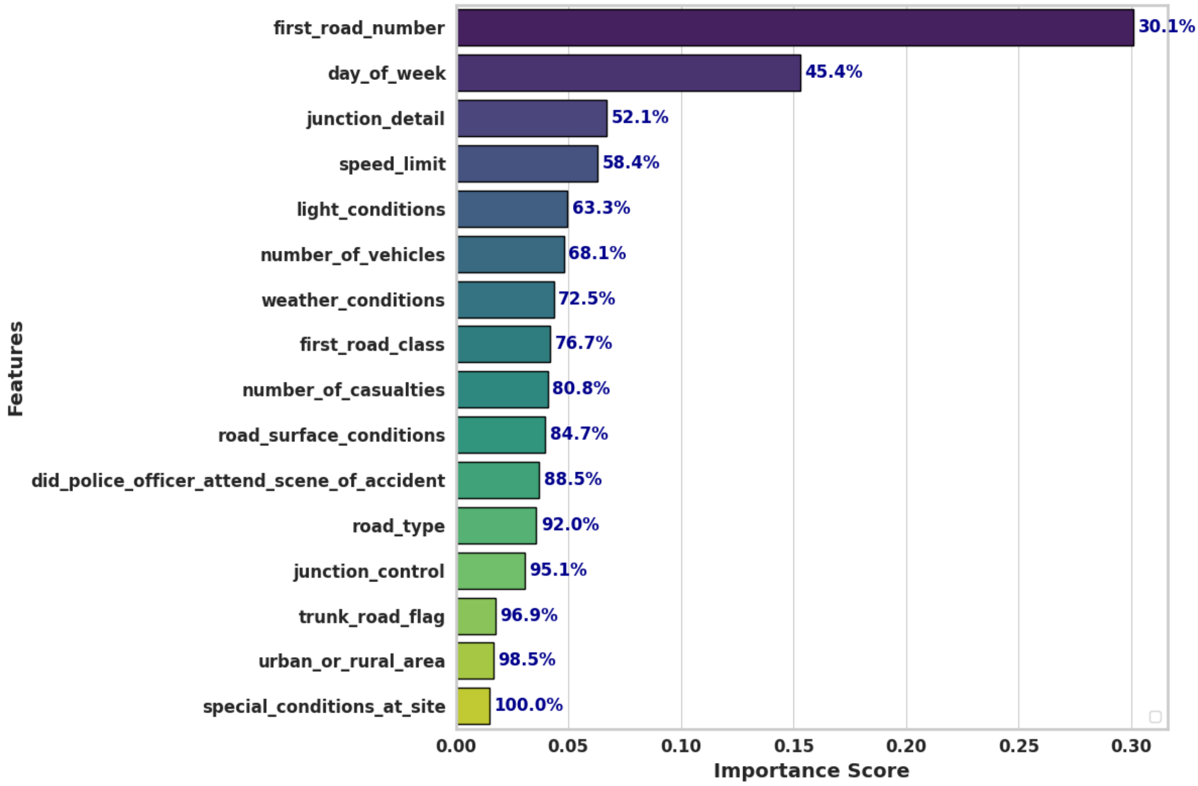

Fig. 5. The importance of accident features

Figure 5 illustrates the importance of accident features, highlighting the key predictors of accident severity. Moreover, Tab. 3 presents the feature importance scores derived from the Random Forest model. The first road number was identified as the most significant predictor, followed by the day of the week and junction details. These features exhibited high importance values, indicating their substantial impact on accident severity. According to Tab. 3, the first 12 features have a cumulative importance of 92.04%, exceeding the 90% threshold, and are therefore chosen as the most critical features in the model. The remaining features (junction_control, trunk_road_flag, urban_or_rural_area, special_conditions_at_site) contribute less than 10% to the cumulative importance and are considered to have a lower impact on the prediction model.

4.2. Results from the comparison between the predictive models

In this study, a total of 106,004 traffic accident instances from the UK in 2022 were analyzed to extract critical insights into accident severity. The dataset was divided into 80% for training and 20% for testing, with three accident severity levels: slight, serious, and fatal. Several machine learning algorithms, including Decision Tree, LightGBM, and SVM, were tested to determine the most effective model for predicting accident severity.

The statistical results are summarized in Tab. 4, showing that LightGBM demonstrated the highest accuracy of 76.50% compared to Decision Tree (66.30%) and SVM (76.30%). This shows that the LightGBM model can classify accident severity levels more accurately than the other two models. Although SVM achieved similar accuracy to LightGBM, its precision was lower at 65.70%, while LightGBM reached 70.10%. This means that when the LightGBM model predicts a case as belonging to a certain severity level (e.g., "serious accident"), the probability of that prediction being correct is higher. High precision is especially important in applications requiring immediate action. For example, if the model is used to alert about serious accidents, a higher precision helps reduce the number of false alarms (false positives). Besides, LightGBM has the highest Recall and F1-score, demonstrating that this model has the most accurate classification ability and the best balance among the criteria compared to the other models.

Moreover, the MAE value reflects the average deviation between the predicted and actual values. The lower the MAE value, the more accurate the model. The results show that LightGBM achieves the lowest MAE (0.250), outperforming Decision Tree (0.358) and SVM (0.253). The MSE and RMSE values heavily penalize large errors, helping to assess the stability of the model. LightGBM continues to show superiority with MSE = 0.281 and RMSE = 0.530, while Decision Tree has the highest errors (MSE = 0.403, RMSE = 0.632). The MSE value of LightGBM was the lowest among the three models, further validating its superior predictive capability. RAE and RRSE compare the model’s performance to the baseline (usually predicting the mean value). A value of RAE < 1 or RRSE < 1 indicates that the model performs better than the baseline. LightGBM achieves RAE = 0.650 (35% better than baseline), but RRSE = 1.128 shows that the model is still 12.8% worse than the baseline when measured by RMSE. Notably, Decision Tree (RAE = 0.929, RRSE = 1.346) barely exceeds the baseline, reflecting its limitations in capturing data patterns. The overall results confirm that LightGBM is the optimal model, balancing accuracy (low MAE, MSE) and the ability to reduce large errors (low RMSE). To evaluate the accuracy of the model, indicators such as MAE, MSE, etc., are crucial. However, the study [39] does not use these indicators to assess the model's accuracy. Study [39] shows that the Decision Tree model is more effective than the LightGBM model.

Tab. 4

Assessing various algorithms based on metrics

Algorithms / Metrics	LightGBM	Decision Tree	SVM
Accuracy	0.765	0.663	0.763
Precision	0.701	0.648	0.657
Recall	0.765	0.663	0.763
F1-score	0.670	0.655	0.662
Cohen’s Kappa	0.022	0.044	0.000
MAE	0.250	0.358	0.253
MSE	0.281	0.403	0.285
RMSE	0.530	0.632	0.534
RAE	0.650	0.929	0.655
RRSE	1.128	1.346	1.136

4.3. Analysis of LightGBM Model Results

Table 5 illustrates the performance metrics of the LightGBM model in predicting three accident severity levels (Fatal, Serious, Slight). The model achieves an overall accuracy of 76.40%, indicating a reasonable general classification capability. Moreover, a confusion matrix, structured with three rows and three columns, was generated to outline the classification results for three distinct classes, including Fatal, Serious, and Slight accidents, depicted in Tab. 6. The main diagonal, displaying values (2, 82, 16110) denotes correct predictions, while the remaining entries in the table signify incorrect predictions. Tab. 6 shows that the LightGBM model predicts more correctly the slight class than the other two classes, with the number of correctly predicted cases up to 16,110. Furthermore, in terms of recall for the slight class, it indicates that 99.00% of slight injuries were accurately identified as positive.

Tab. 5

Evaluation metrics of the predicted values for the three classes of the LightGBM model

Accuracy		**0.764**	
Value	**Precision**	**Recall**	**F1-score**
Fatal	0.14	0.01	0.01
Serious	0.45	0.02	0.03
Slight	0.77	0.99	0.87

Tab. 6

Confusion matrix of the LightGBM model

		Predicted Condition		
		Fatal	**Serious**	**Slight**
Actual Condition	**Fatal**	2	19	322
	Serious	6	82	4573
	Slight	6	81	16110

4.4. Explaining the LightGBM Model Using SHAP

❖ SHAP chart analysis and interpretation

The SHAP chart provides insights into the influence of each feature on the model's predictions, as shown in Fig. 6.

+ The vertical axis ranks features in descending order of impact;
+ The horizontal axis represents SHAP values, indicating how each feature affects the predicted accident severity;
+ The color gradient further enhances interpretation, with red indicating higher feature values and blue representing lower values, corresponding to the encoding scheme used in the dataset.

In this study, accident severity is encoded on a decreasing scale: 1 = Fatal, 2 = Serious, 3 = Slight. This means that higher predicted values (closer to 3) indicate a lower accident severity, whereas lower predicted values (closer to 1) suggest more severe accidents. Consequently, a positive SHAP value pushes predictions towards lower severity (Slight accidents), while a negative SHAP value drives predictions towards higher severity (Fatal or Serious accidents).

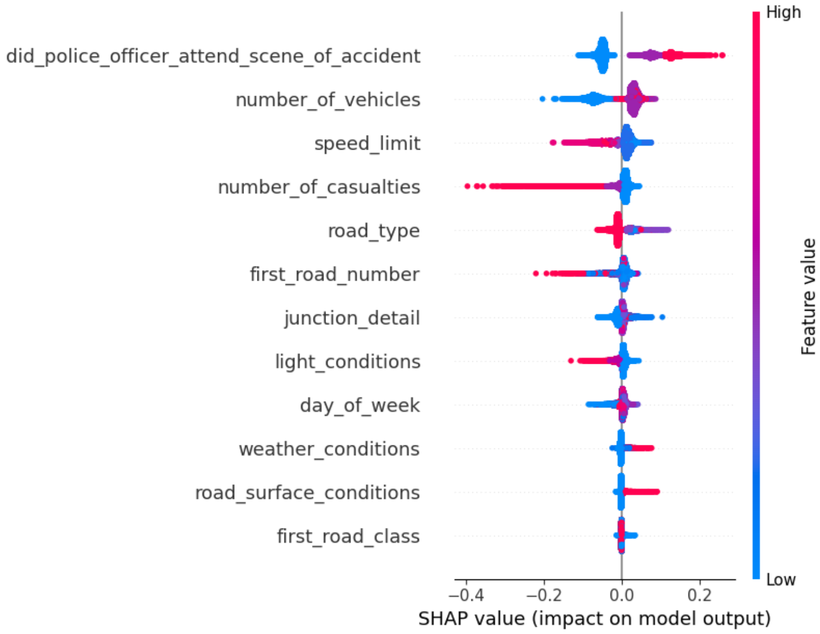

Fig. 6. The LightGBM model explanation using SHAP

❖ Key findings

The SHAP analysis reveals that the "did_police_officer_attend_scene_of_accident" feature has the strongest influence on accident severity predictions, followed by "number_of_vehicles" and "speed_limit". These findings highlight key factors that impact the severity of road accidents, as analyzed in the following feature-by-feature breakdown. The application of SHAP analysis techniques to gain a deeper understanding of the relationship between input variables and the predictive model is crucial. However, the study [39] only focuses on a comparative analysis of machine learning techniques but does not evaluate the relationship between input variables and the predictive model.

The presence of a police officer at the scene significantly affects accident severity. When a police officer is present (coded as 2 = Yes, represented in red), the SHAP values are generally positive, meaning that the model predicts a less severe accident (Slight). Conversely, when a police officer is absent (coded as 1 = No, shown in blue), the SHAP values become negative, pushing predictions toward higher severity (Fatal or Serious). This finding suggests that police presence might contribute to improved accident response, reducing the likelihood of fatal or serious outcomes.

The number of vehicles involved in an accident also plays a crucial role. When more vehicles are present (higher values, red), the SHAP values are positive, indicating a tendency toward less severe accidents. On the other hand, when fewer vehicles are involved (lower values, blue), the SHAP values turn negative, suggesting an increased likelihood of Fatal or Serious accidents. This pattern aligns with real-world observations, where single-vehicle accidents, particularly those involving high speeds or poor road conditions, tend to result in greater severity.

The speed limit at the accident location strongly influences accident severity. Lower speed limits (blue) are associated with positive SHAP values, meaning accidents are more likely to be classified as Slight. Conversely, higher speed limits (red) correlate with negative SHAP values, pushing predictions towards Fatal or Serious accidents. This finding supports the well-documented relationship between speed and accident severity: higher speeds increase the force of impact, leading to more severe outcomes. Study [39] mentions that the speed limit characteristic affects the severity of accidents; however, it does not specify how it influences them.

Several additional features influence accident severity but with a lower impact compared to the top-ranked factors. These include light conditions, day of the week, weather conditions, road surface conditions, and first road class. While they contribute to the model's predictions, their effects are less pronounced, suggesting that external environmental factors, although important, may not be the primary determinants of accident severity compared to police presence, vehicle count, and speed limit.

5. CONCLUSIONS

This study investigated the application of supervised machine learning models to predict road accident severity in the UK context, utilizing a 2022 dataset comprising 106,004 accident records. Three algorithms – Decision Tree, SVM, and LightGBM – were evaluated, with LightGBM emerging as the most effective model, achieving 76.5% accuracy and demonstrating superior precision (70.1%), recall (76.5%), and error metrics (MAE = 0.250, MSE = 0.281).

To enhance model interpretability, SHAP analysis was employed to identify key factors influencing crash severity. The results highlighted police officer attendance, speed limits, and the number of vehicles involved as significant determinants. Notably, police presence at the scene was associated with reduced severity, while higher speed limits and single-vehicle collisions correlated with an increased likelihood of fatal or serious outcomes.

By integrating machine learning with post-hoc interpretability techniques, this study provides actionable insights for policymakers to enhance road safety. The findings emphasize the importance of optimizing enforcement strategies and revising speed regulations. Ultimately, this research highlights the potential of interpretable machine learning frameworks to improve the understanding of crash dynamics and support targeted interventions, contributing to global efforts to reduce traffic-related fatalities and injuries.

Despite its contributions, this study has several limitations. First, the dataset was restricted to UK accidents in 2022, limiting its generalizability to other regions or time periods. Second, key variables such as driver behavior and vehicle-specific details were absent, potentially omitting critical predictors of severity. Therefore, future research should incorporate multi-year, multi-region datasets to capture temporal and geographical variability.

References

1. Gan Jing, Linheng Li, Dapeng Zhang, Ziwei Yi, Qiaojun Xiang. 2020. „Alternative Method for Traffic Accident Severity Prediction: Using Deep Forests Algorithm". *Journal of advanced transportation* 1: 1257627. ISSN: 2042-3195.
2. Madushani J. S., R.K. Sandamal, D.P.P. Meddage, H.R. Pasindu, P.A. Gomes. 2023. „Evaluating expressway traffic crash severity by using logistic regression and explainable & supervised machine learning classifiers. *Transportation Engineering* 13: 100190. ISSN: 2666-691X.

3. Mokoatle M., D. Vukosi Marivate, P. Michael Esiefarienrhe Bukohwo. 2019. „Predicting road traffic accident severity using accident report data in South Africa". In: *Proceedings of the 20th Annual International Conference on Digital Government Research*: 11-17. 18-20 June 2019, Dubai, United Arab Emirates.
4. Bokaba Tebogo, Wesley Doorsamy, Babu Sena Paul. 2022. „A Comparative Study of Ensemble Models for Predicting Road Traffic Congestion". *Applied Sciences* 12(3): 1337. ISSN: 2076-3417.
5. Assi Khaled, Syed Masiur Rahman, Umer Mansoor, Nedal Ratrout. 2020. „Predicting Crash Injury Severity with Machine Learning Algorithm Synergized with Clustering Technique: A Promising Protocol". *International journal of environmental research and public health* 17(15): 5497. ISSN: 1660-4601.
6. Khattak Afaq, Hamad Almujibah, Ahmed Elamary, Caroline Mongina Matara. 2022. „Interpretable Dynamic Ensemble Selection Approach for the Prediction of Road Traffic Injury Severity: A Case Study of Pakistan's National Highway N-5". *Sustainability* 14(19): 12340. ISSN: 2071-1050.
7. Anderson Tessa K. 2009. „Kernel density estimation and K-means clustering to profile road accident hotspots". *Accident Analysis & Prevention* 41(3): 359-364. ISSN: 0001-4575.
8. Jamal Arshad, Muhammad Zahid, Muhammad Tauhidur Rahman, Hassan M. Al-Ahmadi, Meshal Almoshaogeh, Danish Farooq, Mahmood Ahmad. 2021. „Injury severity prediction of traffic crashes with ensemble machine learning techniques: A comparative study". *International journal of injury control and safety promotion* 28(4): 408-427. ISSN: 1745-7300.
9. Komol Md Mostafizur Rahman, Md Mahmudul Hasan, Mohammed Elhenawy, Shamsunnahar Yasmin, Mahmoud Masoud, Andry Rakotonirainy. 2021. „Crash severity analysis of vulnerable road users using machine learning". *PLoS one* 16(8): e0255828. ISSN: 1932-6203.
10. Le Khanh Giang, Ho Thi Lan Huong, Do Van Manh, Tran Quang Hoc. 2024. „Applying a two-step cluster algorithm in traffic accident data analysis". *Transport and Communications Science Journal* 75(4): 1673-1687. ISSN: 2615-9554. DOI: https://doi.org/10.47869/tcsj.75.4.16.
11. Chen Hengrui, Hong Chen, Ruiyu Zhou, Zhizhen Liu, Xiaoke Sun. 2021. „Exploring the mechanism of crashes with autonomous vehicles using machine learning". *Mathematical problems in engineering* 2021(1): 5524356. ISSN: 1024-123X.
12. Van der Wall H.E.C., R.J. Doll, G.J.P. van Westen, I. Koopmans, R.G. Zuiker, J. Burggraaf, A.F. Cohen. 2020. „The use of machine learning improves the assessment of drug-induced driving behaviour". *Accident Analysis & Prevention* 148: 105822. ISSN: 0001-4575.
13. Lee Jonghak, Taekwan Yoon, Sangil Kwon, Jongtae Lee. 2020. „Model evaluation for forecasting traffic accident severity in rainy seasons using machine learning algorithms: Seoul city study". *Applied Sciences* 10(1): 129. ISSN: 2076-3417.
14. Sarker I.H. 2021. „Machine learning: Algorithms, real-world applications and research directions". *SN computer science* 2(3): 160. ISSN: 2661-8907.
15. Chakraborty Meghna, Timothy J. Gates, Subhrajit Sinha. 2023. „Causal analysis and classification of traffic crash injury severity using machine learning algorithms". *Data science for transportation* 5(2): 12. ISSN: 2948-1368.

16. Bui Ngoc Dung, Lai Manh Dzung, Tran Vu Hieu, Nguyen Binh T. H. 2019. „Multiple vehicles detection and tracking for intelligent transport systems using machine learning approaches”. *Transport and Communications Science Journal* 70(3): 214-224. ISSN: 2615-9554. DOI: https://doi.org/10.25073/tcsj.70.3.29.
17. Cigadem A., O. Cevher. 2018. „Predicting the severity of motor vehicle accident injuries in Adana-Turkey using machine learning methods and detailed meteorological data”. *International Journal of Intelligent Systems and Applications in Engineering* 6(1): 72-79. ISSN: 2147-6799.
18. Silva Philippe Barbosa, Michelle Andrade, Sara Ferreira. 2020. „Machine learning applied to road safety modeling: a systematic literature review”. *Journal of traffic and transportation engineering (English edition)* 7 (6): 775-790. ISSN: 20957564.
19. Wang Junhua, Boya Liu, Ting Fu, Shuo Liu, Joshua Stipancic. 2019. „Modeling when and where a secondary accident occurs”. *Accident Analysis & Prevention* 130: 160-166. ISSN: 0001-4575.
20. Ke Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu. 2017. „LightGBM: a highly efficient gradient boosting decision tree”. *Advances in neural information processing systems* 30: 3146-3154. ISSN: 1049-5258.
21. Bonaccorso G. 2018. *Machine Learning Algorithms: Popular algorithms for data science and machine learning*. Packt Publishing Ltd. ISBN: 1789347998.
22. Road Safety Data. Available at: https://data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safetydata.
23. Rella Riccardi Maria, Filomena Mauriello, Sobhan Sarkar, Francesco Galante, Antonella Scarano, Alfonso Montella. 2022. „Parametric and Non-Parametric Analyses for Pedestrian Crash Severity Prediction in Great Britain”. *Sustainability* 14(6): 3188. ISSN: 2071-1050.
24. Heinze Georg, Christine Wallisch, Daniela Dunkler. 2018. „Variable selection–a review and recommendations for the practicing statistician”. *Biometrical journal* 60(3): 431-449. ISSN: 0323-3847.
25. Menze Bjoern H., B. Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, Fred A. Hamprecht. 2009. „A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data”. *BMC bioinformatics* 10: 1-16. ISSN: 1471-2105.
26. Yang Jianjun, Siyuan Han, Yimeng Chen. 2023. „Prediction of traffic accident severity based on random forest”. *Journal of Advanced Transportation* 1: 7641472. ISSN: 0197-6729.
27. Hussain S. 2017. „Survey on current trends and techniques of data mining research”. *London Journal of Research in Computer Science and Technology* 17(1): 11. ISSN: 2514-8648.
28. García Salvador, Julián Luengo, Francisco Herrera. 2015. *Data preprocessing in data mining*. Switzerland: Springer International Publishing. ISBN: 978-3-319-10247-4.
29. Pedregosa F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas. 2011. „Scikit-learn: Machine learning in Python”. *Journal of machine Learning research* 12: 2825-2830. ISSN: 1532-4435.
30. James Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani. 2013. *An introduction to statistical learning*. New York: springer. ISBN: 1461471370.
31. Witten I.H., E. Frank, M.A. Hall, C.J. Pal, M. Data. 2005. *Data Mining: Practical Machine Learning Tools and Techniques.* The Netherlands: Elsevier. ISBN: 0123748569.

32. Quinlan J.R. 1986. „Induction of decision trees”. *Machine learning* 1: 81-106. ISSN: 0885-6125.
33. Breiman Leo, Jerome Friedman, Richard A. Olshen, Charles J. Stone. 1984. *Classification and Regression Trees*. Routledge. ISBN: 0412048418.
34. Bishop Christopher M., Nasser M. Nasrabadi. 2006. *Pattern recognition and machine learning*. New York: springer. ISBN: 0387310738.
35. Cortes Corinna, Vladimir Vapnik. 1995. „Support-vector networks”. *Machine learning* 20: 273-297. ISSN: 0885-6125.
36. Hossin Mohammad, Md Nasir Sulaiman. 2015. „A Review on Evaluation Metrics for Data Classification Evaluations”. *International journal of data mining & knowledge management process* 5(2): 501-511. ISSN: 2230-9608.
37. Lundberg Scott M., Su-In Lee. 2017. „A unified approach to interpreting model predictions”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*: 4768-4777. 04.12.2017, Red Hook, NY, United States. ISBN: 9781510860964.
38. Meddage D.P.P., I.U. Ekanayake, A.U. Weerasuriya, C.S. Lewangamage, K.T. Tse, T.P. Miyanawala, C.D.E. Ramanayaka. 2022. „Explainable machine learning (XML) to predict external wind pressure of a low-rise building in urban-like settings”. *Journal of Wind Engineering and Industrial Aerodynamics* 226: 105027. ISSN: 0167-6105.
39. Megnidio-Tchoukouegno Mireille, Jacob Adedayo Adedeji. 2023. „Machine learning for road traffic accident improvement and environmental resource management in the transportation sector". *Sustainability* 15(3): 2014. ISSN: 2071-1050.

Received 01.10.2024; accepted in revised form 10.01.2025