

Rear-End Vehicles Collision Detection in Internet of Vehicles Using Explainable AI and Machine Learning

Soumya Ranjan Mahanta¹, Mrutyunjaya Panda^{2,*}

¹ M. Tech Student, Department of Computer Science and Applications, Utkal University, Vani Vihar ,
Bhubaneswar, Odisha, India

² Associate Professor, Department of Computer Science and Applications, Utkal University, Vani Vihar ,
Bhubaneswar, Odisha, India

Abstract

In recent days, with the high rise in population, increase in possession of multiple vehicles in a family which creates heavy traffic on roadways, vehicle collision is one of the principal causes of death worldwide due to road accidents. Internet of vehicles (IoV) has become a salient innovation of intelligent transportation system (ITS), which is attracting attentions of researchers and scientists recently to assist the drivers to have a safe driving experience. With the tremendous opportunities in today's autonomous world, explainable artificial intelligence playing a remarkable character to address the censorious facet of the IoV with a detailed explanations while detecting rear end collision of vehicles in IoV to have quality of services and overall safety. Previous research applied machine learning and deep learning algorithms for the detection of rear end collisions in IoV but have limitations in not using explainable artificial intelligence (xAI) techniques which would dramatically enhance the interpretability and performance acceptance of the IoV overall system. This paper uses LIME and Captum's DeepLift attribution xAI methods to understand the inner details of the model prediction in IoV scenario. The IoV datasets are collected from Kaggle for the proposed research. The performance of the proposed xAI based machine learning approaches using several Boosting algorithms, Random Forest and Gated Recurrent Neural Network approaches, for rear end collision detection in IoV are measured with extensive simulations in this research. The experimental results show that the proposed work is accurate in detecting the rear end collision of vehicles in IoV environment with detailed explanations which shows its effectiveness, transparency and trustworthiness in improved public safety in IoV environments.

Keywords: Collision Detection; Internet of Vehicles; Explainable AI; DeepLift; LIME; SHAP; Random Forest; XGBoost; CatBoost; Gated Recurrent Neural Network Unit

1. Introduction

The internet of vehicle (IoV) engenders intelligent vehicles [1] exactly in the same way how the Internet of Things (IoT) [2] accomplish intelligence in smart building, smart health, Smart energy and transportation systems. In IoV scenario, vehicles are considered to be intelligent and allowed to signal each other through smart mobile phones or so wirelessly with the help of road side units [3]. There are four different communication networks are used in an IoV environment including: Vehicle-to-Vehicle (V2V), Vehicle-to-Road (V2R), Vehicle-to-Human (V2H) and Vehicle-to-Sensor (V2S) for communication between public networks and vehicles. It is also envisioned that by 2025, almost all types of vehicles that will be sold globally will be connected, either by smart-phone or through

embedded integration technology [4]. While moving on roadways, most of the times, either due to poor environmental situations, vehicle conditions, brake applying mechanism, and/or due to rough driving during lane change, overtaking etc. which overall affect the public welfare to a great extent [5]. It is observed that a large amount of around 60%-70% of total accidents are due to rear-end collision of vehicles where vehicle hit you from the back resulting heavy traffic congestion and may affect the passing by commuters to injuries or even to death. Most of the cases, rear-end collision happened by the teenagers or the early young mass who lack experiences in understanding the danger of accident well in advance or possibly not able to apply the brake in appropriate time. This may be remedied with proper traffic management, better road

infrastructure and awareness about over-speeding. At the same time, it is also pointed out that such situations can be avoided if the driver can get the warning sign 0.5 second to 1.5 second ahead [6]. Looking into this very pertinent social issue, even though many researches are carried out in this area, still IoV has not yet explored fully.

Further, apart from enhanced road safety, in-car amusement, services through online cyberspace such as social networking, online gaming, real time traffic navigation system, IoV could be immensely beneficial in vehicle monitoring, lane monitoring, and possible accident identification by using 5G Communication, Cloud Computing and Big Data Analytic [7-8].

Driving the vehicle in multi-lane roadways and Lane Changing is a basic driving behaviour in both lateral and longitudinal movements under different driving scenarios for safe driving and to have better traffic safety of people and materials [9].

Many of the past research fail to spot the effectiveness of the proposed intelligent algorithms in comparison to traditional machine learning algorithms which makes it little hard to come to a conclusion while finding the best solution for rear-end collision detection in IoV environment. Further, there are no explanations found on why to take a decision on the intelligent algorithms in the literature using Explainable Artificial Intelligent (xAI) methods. So, in order to alleviate those limitations, our research considers applying xAI methods such as: LIME, SHAP and Anchor Explainer for better explanation to the end-users to have better understanding of the situations at hand and take appropriate and efficient decisions. Intelligent algorithms with Random Forest, XGBoost, CatBoost and deep learning strategies were adopted to have a most accurate classifiers with better explanations, so that they can unveil the possible impacts of several impact factors such as: number of lanes, environmental conditions, driver condition etc. on detecting the rear-end collision of vehicles in IoV.

Our research contributions are as follows.

- This may be the first work of applying explainable AI (xAI) techniques on rear-end collision detection in IoV using machine learning approaches.
- Anchor Tabular LIME and DeepLift SHAP Explainer used on the IoV dataset to unveil the hidden explanations in the dataset for better interpretability.

- Feature Engineering techniques using Pearson correlation with feature importance are applied in the IoV dataset to perform data visualization

- Several machine learning algorithms such as Random Forest, XG Boost, Cat Boost with deep learning based Gated Recurrent Neural Network (GRU) methods are employed to understand the most accurate the rear-end collision detection in IoV environment

- The performance comparison in terms of accuracy, Precision, Recall, F-Measure and Time taken to build the model amongst the classifiers is conducted to obtain the accurate classifier with high interpretability in detecting rear-end collision detection in IoV.

The rest of the paper is organized as follows. Section 2 discusses about the related research in the field of IoV and explainable AI. The methodologies adopted in this research is presented in Section 3 followed by the datasets used are presented in Section 4. Section 5 illustrates the experimental set-up followed by experimental finding with discussions in Section 6. Finally, we conclude with future scope of research in Section 7.

2. Related Research

Nekovee and Bie [10] proposes an adaptive inter vehicle collision avoidance protocol by highlighting the usefulness of effective V2V (Vehicle-to-Vehicle) communication network in order to prevent rear end collisions effectively and in reliable manner.

Kedareswar and Krishnamoorthy [11] develops an embedded intelligent system using an accelerometer that is interfaced to ARM Cortex M0 microcontroller that can alert the immediately following vehicle looking into the braking intensity of the vehicle through rear LED display and transmitting a message through IR communications. This way, the driver of immediately following vehicle gets some idea to either press the brake or decelerate the speed of the vehicle and in return the rear end collisions of the vehicles is avoided.

Chiroma et al. [12] presented a survey on usefulness of big data analytics in IoV , convenes the possible research opportunities in this infant research area and finally proposes to use Apache Server, Cloud based IoV environments for data analytics.

Razzaq et al. [13] illustrates the need of advanced autonomous vehicles technologies with a driver assistance system for collision detection and

prevention. In this, they use a Fuzzy Inference system coupled with VOMAS agent in the NetLogo simulation environment and demonstrate its effectiveness in driving through vehicle automated system so that collision could be avoided to a greater extent.

Aditya et al. [14] designed an efficient deep learning model using ResNext on GTACrash synthetic data by comparing the several pre-trained deep learning models like VGG 16, ResNet50 etc. and found that their approach performs well with an accuracy of 91% with low computational overhead.

Teimouri and Ghatee [15] uses TOPSIS multi-criteria selection in a Random Forest classifier on vehicles trajectory data of 100 car's database from Virginia tech transportation institute to detect rear-end vehicle collision in an IoV environment. They compared with the other classifiers such as Bayesian Networks, MLP neural network, Decision tree and support vector machines etc. and concludes that random forest classifier outperforms other classifiers with 88.4% accuracy in detecting driving warning situations and 94.7% for detecting safe situations.

Ma, Yu and Liu [16] applies several machine learning model for an increased rear-end collision probability (RCP) of each vehicle in an IoV framework with an unbalanced dataset and then concludes that Multi-Layer Perceptron (MLP) model with Generalized Pareto Distribution (GPD) is the best amongst all other methods in minimizing the risk associated with freeway rear-end collision.

Muzammel et al. [17] proposes to design an ensemble of two convolutional neural networks (CNNs) with feature descriptors to detect blind-spot in real-time rear end collisions of the vehicles in IoV scenario. Further, they use two pre-trained neural networks (Resnet 50 and Resnet 101) on several vehicle detection dataset for feature fusion and to improve the performance of R-CNN with a false alarm of approximately 3%.

Muzahid et al. [18] performed a critical review on sustainable planning procedures to investigate multiple vehicle cooperation and collision avoidance (MVCCA) strategies using AI enabled conceptual framework in Autonomous vehicles to avoid rear-end collision detection with issues pertaining to actual implementation of driverless automated traffic systems.

Choi et al. [19] proposes multiple classifier using both audio and video data obtained from vehicle dashboard cameras to detect the possibility of a car crash which may be useful in calling for roadside assistance automatically to ensure immediate rescue of the affected ones.

Xin et al. [20] advocates the importance of rear-end collision detection in smart cities to avoid traffic accidents and concludes that deep convolutional neural network model outperforms others traditional methods like Honda, Berkeley and multi-layer perception neural-network-based algorithms in accurate prediction of rear-end vehicle collisions.

Chen et al. [21] proposes Dedicated Short Range Communication (DSRC) method in place of previously used on-board sensor to reduce the rear-end collisions and other traffic accidents in autonomous cars where Monte Carlo simulation using Bayesian method is adopted for the effectiveness in dynamic road safety environments.

Chen et al. [22] proposes Emergency steering method to consider vehicle lateral stability, emergency manoeuvre selection and Vehicle dynamics, Tire Slip Angle and Segal lateral force model instead of Emergency braking only to avoid rear-end collision in a Monte Carlo simulation.

3. IoV Dataset

The publicly available IoV dataset is collected to perform the experiments in rear-end collision detection [23-24]. Following are some of the procedural details of this dataset generation and collection in an IoV environment. In Table 1, parameter setting used for vehicle mobility for IoV simulation environment is as presented. Here, a simulation area of 1500meter x 1500 meter is created with one-way or three-way lane roads are made. There are three vehicle speed variations such as maximum speed, minimum speed and vehicle average speed in kmph in a freeway mobility environment presented. There are approximately 30 to 180 numbers of vehicles are proposed to be on road for understanding the rear-end collision scenario. In Table 2, the features which are considered to be influential in rear-end collisions are mentioned. There are five features considered in this dataset such as: (i) Human feature where Status of the driver, Gender and Age are considered, (ii) Weather condition, Level of traffic, Light situation, Nature of the road surface falls under nature of

environment, (iii) Rest of the on-road vehicles is represented through the Velocity and Distance between vehicles, (iv) The Vehicle itself feature is through its Braking capability, Sliding property and

Steering etc. and finally (v) number of lanes in the road. The quantization of these influential factors is shown in Table 3.

Table 1: IoV simulation Environment for vehicle mobility parameter selection

Size of simulation area	1500 m X 1500 m
Type of Lane	One-way three-lanes
Minimum speed (m/s)	50 km/h
Normal speed (m/s)	90 km/h
Maximum speed (m/s)	140 km/h
Vehicle mobility model	Freeway mobility
Number of nodes (vehicles)	30 to 180 numbers

Table 2: Influential Features of IoV dataset

Feature	Classes
Human feature	Status of the driver, Gender and Age
Nature of the Environment	Weather condition, Level of traffic, Light situation, Nature of the road surface
Rest of the on-road vehicles	Velocity, Distance between vehicles
The Vehicle itself	Braking capability, Sliding property, Steering
Road	Number of lanes

Table 3: Quantization of Influential Features of rear-end vehicle collisions

Feature		Feature status	Variable/Index value
Number of Lanes		Single, Multiple Lanes	1 is single, 2 is multiple
Status of driver		Normal, Abnormal	1 is normal, 2 is abnormal Case-1: : if the vehicle is moving without any issue or passes without any hitches, it is considered normal with index value 1 Case-2: When the vehicle is stopping due to because of any reason like Brake, Dwell, close up etc., it comes to a stop, so it returns a value 2 being abnormal
Nature of the environment		Normal, Abnormal	1 is normal, 2 is abnormal
Velocity of Vehicle		Low, normal, over speeding	1 is over-speeding, 0.65 is normal, 0.35 is low
Distance between vehicles		Less than 5 m, less than 10 m, less than 20 m	1 for less than 5 m, 0.5 for less than 10 m, 0 for less than 20 m

Braking capability		Available or Not available	<p>1 for available, 0 for not available</p> <p>Case-1: The vehicle is moving, but with zero break application, Index value is 0</p> <p>Case-2: Hard pressing of Break has index value of 1</p> <p>Case-3: While willing to change from one lane to other or already changed the lane, no need to apply break, so index value is 0</p> <p>Case-4: When there is a moderate break applied, the index value is 1</p> <p>Case-5: While changing the lane depends on the present road scenario, it may or may not apply break, so it is a probability distribution which ultimately returns index value of 1</p>
Collusion		Rear-end collision occurs or not	1 for collision, 0 for safe

4. Proposed Methodologies

In this section, details of the machine learning and Explainable AI (xAI) methods used for the experiments are discussed.

(i) Model Interpretability using xAI

Model interpretability techniques, sometimes referred as Explainable AI (xAI) aim to shed light on the decision-making process of machine learning models. The Explainable AI (xAI) methods provide insights into the importance and influence of different features or components on the model's predictions. By using Captum and DeepLift, a deeper understanding of the model's behavior and the factors driving its decisions can be gained by the users.

● DeepLift Explainer

DeepLift introduced by Shrikumar et al. [25] works by computing the difference between feature activations at the current input and a reference input. This difference is then propagated through the model to calculate feature attributions. By analyzing these attributions, the impact of individual features on the model's predictions can be identified, which might be beneficial for model validation, debugging, and building trust in AI systems. These methods help validate the model's behavior, identify potential biases or issues, and explain its predictions to stakeholders. In domains such as healthcare, finance, or autonomous driving, Image Annotation where interpretability is crucial, need to meet regulatory requirements and ensuring transparency, Deep Lift is found its place in decision process.

Overall, Captum's DeepLift attribution method empowers practitioners to interpret and explain the decision-making process of complex machine learning models, enabling greater trust, validation, and understanding of model behavior in critical domains.

● **Integrated Gradients (IG) Explainer**

Integrated Gradients (IG), an Explainable AI technique which is popularly used in Deep neural Networks intends to find the correlations between the features of the dataset with its machine learning model predictions [26]. It is one of the feature attribution methods where an importance score is being assigned to each input attribute by approximating the integral of the gradients of the learned model corresponds to each input. Apart from understanding the feature importance, Integrated Gradients explainer is found useful in identifying skewness in the data with model's performance. IG explainer has been found of immense benefits to the end users in terms of better explanations, ease of implementation, trustworthy and computationally efficiency.

● **Tree SHAP Explainer**

SHAP (SHapley Additive exPlanation) is a game theoretic approach to explain the output of any machine learning model [27]. The goal of SHAP is to explain the prediction for any instance x_i as a sum of contributions from its individual feature values. Tree SHAP is a fast and exact method to estimate SHAP values for Decision Trees based models and ensembles of trees, under several different possible assumptions about feature dependence.

For model specific approximations, we consider Deep SHAP method to improve the sample efficiency of model-agnostic estimations of SHAP values. In these model agnostic approximations, for a small number of inputs, low computational evaluations of the original model are required to obtain similar approximation accuracy in comparison to the other existing approaches such as: LIME (Local Interpretable Model-Agnostic Explanations). LIME interprets individual model predictions based on locally approximating the model around a given prediction. LIME is a local linear explanation model which uses additive feature attribution method for simple explanations.

The main reason of choosing xAI methods lies in following: (i) getting local accuracy for a better explanation model with (ii) missingness represents the features missing from the original input do not have any impact on the model and then (iii) consistency of the model that represents even though the model changes then the inputs attribution should either increase or no change at all but should not decrease.

In Deep SHAP (DeepLIFT + Shapley values) method, two optional assumptions such as: feature independence and model linearity are used for simple computational evaluations of the expected values as defined below.

- $E[f(y)]$ is the predicted value from the base value when features to the current output $f(x)$ is unknown.
 - In this, S is the non-zero indexes in y' and S' is the set of features not in set S .
 - Since SHAP uses conditional probabilities to define simplified inputs with input mapping, $h_x(\bar{y}) = y_S$, where y_S has missing values for features not in the set S .
 - Here, $f(x(y')) = f(h_x(y')) = E[f(y) | y_S]$, Since most models cannot handle arbitrary patterns of missing input values, $f(y_S)$ is approximated with $E[f(y) | y_S]$
- For novel Deep SHAP based xAI method, the following properties are considered.
- For SHAP explanation model simplified input mapping where $f(h_x(y')) = E[f(y) | y_S]$
 - The above may be approximated expectation over $y_{S'} | y_S$ as $E_{y_{S'} | y_S}[f_y]$
 - Feature independence can be assumed through $E_{y_{S'}}[f_y]$
 - Finally, assume model linearity from $f([y_S, E[y_{S'}]])$

Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. More details can be found from [28-29].

(ii) Machine Learning Classifiers

● **Random Forest Classifier**

Random forests [30] is a very powerful ensemble machine learning algorithm which combines the predictions from several decision trees with uniform distribution in the forest, to obtain a result. The generalization error in Random Forest classifier is calculated as the probability of error that is predicted

by the classifier on a new data. As this is an ensemble of classifiers, this error is largely dependent on the correlation and strength of the individual trees in the forest ensemble. This way, it becomes computationally efficient as it does not require data splitting and several predictive model retraining. Sometimes, randomness is imputed to minimize the correlation but preserving the same strength to obtain an improved accuracy, which makes it a clear choice over AdaBoost (Adaptive Boosting). Several other measures for which Random Forest is suitable for analysis are as outlined below.

- It's robustness to deal effectively with outliers and noise in data.
- It's faster than bagging or boosting, as retraining is not required
- It gives useful data visualization in terms of error, strength, correlation and variable importance.
- It's quite simple and can easily be parallelized.

● **XGBoost**

XGBoost (eXtreme Gradient Boosting algorithm) is an optimized distributed gradient boosting algorithm which is an extreme version of the previous gradient boosting algorithm [31]. As XGBoost uses regularization methods for its classification task, it is also called as a regularized form of the existing gradient-boosting tree algorithm. XGBoost performs better when both numeric and nominal data are present in the dataset. Since XGBoost uses ensemble method, it is useful in selecting the feature importance from selected features [32], which will also be useful in understanding the inner details of

the dataset. It is efficient in terms of speed and scalability in comparison to existing boosted tree algorithm. Even though, it works well for imbalanced datasets, more attentions are needed to tune the training algorithm so as to avoid the misclassification rate in datasets having skewed class distribution.

● **CatBoost**

CatBoost [33] is a categorical boosting algorithm that effectively deals categorical attributes in comparison to other machine learning algorithms. In CatBoost, the weight assigned to categorical features at the output column during training process makes it a most accurate classifier while dealing categorical attributes [34]. Apart from this, following are some of the key characteristics for choosing CatBoost classifier for analysis such as: high performance in terms of accuracy, automatic handling of categorical features, robust with no extensive hyper-parameter tuning, more generalized classifier with low chance of overfitting and more importantly it is user friendly. Furthermore, it presents low computational complexity of $O(XT)$, where X is the number of subsets permutations and T denotes the number of base decision tree models. Finally, since CatBoost performs in a permutation-driven approach where it trains the model on a subset of data while calculating residuals on another subset, it prevents target leakage and overfitting, which makes it a promising ML algorithms in real life scenarios. A comparative study of several boosting algorithms with Random Forest is presented in Table 4.

Table 4: Difference amongst Boosting and Bagging Algorithms

Algorithms	Gradient Boosting	AdaBoost	XGBoost	CatBoost	Random Forest
Year	1999	1998	2014	2017	2006
Handling Categorical Variables	Pre-processing required	No	No	Automaticall y handles categorical variables	No
Speed/Scalability	Moderate	Fast	Fast	Moderate	Fast/better

Algorithms	Gradient Boosting	AdaBoost	XGBoost	CatBoost	Random Forest
Memory Usage	Moderate	Low	Moderate	High	High
Regularization	NO	No	Yes	Yes	Yes
Parallel Processing	No	No	Parallel in node building within each tree	Yes	Yes
GPU Support	No	No	Yes	Yes	Yes
Feature Importance	Available	Available	Available	Available	available

From the Table 4, it is quite clear that XGBoost and CatBoost are the best ones in comparison to Gradient Boosting and AdaBoost algorithms for their GPU support, regularization and parallel processing capabilities. At the same time, while comparing XGBoost and CatBoost with Random Forest, a bagging classifier, it seems they are quite similar as both are tree based learners but different in terms of algorithm and architectural point of view, which in turn have some effect on their performances in classification. While regularization and GPU support is possible in all the three models, numeric to categorical data conversion is needed in XG Boost and Random Forest as a pre-processing task. While CatBoost and random Forest are parallelize the overall ensemble model after building trees, XG Boost is a sequential model, but it parallelizes the node building within each tree to generate a faster tree. It is also noticed from Table 1 that all the three algorithms XG Boost, Cat Boost and Random Forest uses multiple CPU/ GPU support during their training and we need to set proper hyper-parameter setting in all of them to understand the training speed, they all perform equally with no practical significant of one over the other. Sometimes, it is seen that hyper-parameter tuning in XGBoost and Cat Boost are little bit complex than that of Random Forest, which makes random forest a little edge over the others. While

considering the classification accuracy comparison, Random Forest is considered as one of the fastest classifier in comparison to the other at the cost of overfitting, but over-fitting may be taken care of by cleaning the data through efficient pre-processing techniques. However, smaller tree creation depending upon the node's similarity score (i.e. it does not create the tree to the full depth) in case of XGBoost helps it to overcome the over-fitting issue and present it as a suitable classifier in real life scenario. In case of CatBoost algorithms, they build symmetric or balanced trees unlike XGBoost, and using early stopping criteria, CatBoost can stop building further trees once over fitting occurs.

(iii) Deep Learning Models

In this section, deep learning based Gated Recurrent Neural Network Model (GRU) is discussed.

● Gated Recurrent Neural Network Model (GRU)

A Gated Recurrent Neural Network Model (GRU) model is an extension of traditional recurrent neural network (RNN) model such as simple Long Short-Term Memory (LSTM) networks proposed by Cho et al. in 2014 [35] to process sequential data with improved memory requirements and classification accuracy. GRU model vanish the gradient vanishing/exploding problems that occurs in

traditional RNN model, making it a good choice and a comparable one to that of Random Forest classifier. The GRU model uses two gating mechanism such as the Reset gate and the update gate to update the hidden state of the neural network each time in order to control the information inflow and outflow in the neural network. In this information flow, where the reset gate throws the number of past hidden state that are to be forgotten, the update gate calculate the number of new input that are required to update the hidden state and then finally, the output of the GRU model is found based on the updated hidden state. A GRU model is presented in Figure 1 [36].The procedures adopted in creating a GRU model are outlined below.

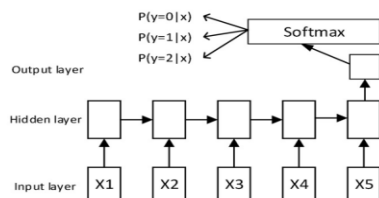


Figure 1. Structure of a GRU model

As shown in Figure 1, the basic structure of a Gated Recurrent Unit (GRU) contains multiple neurons in the input layer (X_1, X_2, \dots), where the number of such input neurons is same as the size of the input feature space and the number of output neurons ($P(y=0|x)$ or $P(y=1|x)$ etc.) corresponds to the outer feature space. Here, the GRU model contains fully connected layers followed by a ReLU activation function. The output of the model is passed through a softmax activation to obtain probabilities. The forward method specifies the forward pass of the model. The main function of the GRU model lies in the hidden layer containing memory cells which is shown in Figure 2 [38]. From Figure 2, it can be seen that there are two gates Reset Gate and Update gate in the memory cell which are responsible for cell's status change and maintenance.

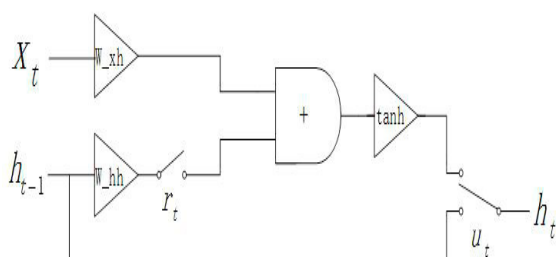


Figure 2: GRU memory cell structure

In this Figure 2, r_t and u_t are represented as reset gate and update gate whereas X_t and h_{t-1} are the input neurons and the output value of the memory cell from previous step. Further, the switch shown as reset gate is open when h_{t-1} doesnot have any impact on the input sequence of neurons X_t , otherwise the reset gate is closed indicating that h_{t-1} has some affect on the current input neuron sequence X_t . At the same time, the purpose of the update gate switch is used to check whether there is an update on input is needed or we can ignore the input sequence X_t . Accordingly, the update gate switch is short circuited or closed making an open connection with input but a connection between h_t and h_{t-1} . This way the gradient propagates in a reverse way solving the gradient vanishing problem effectively. It is to be noted here that, both reset gate and update gate uses sigmoid function as their activation function. The GRU model process the input sequence element wise and finally deliver the output once the last element from the input sequence is successfully processed. The weight and bias update of GRU model is exactly similar to that of conventional feed forward artificial neural network structure. Here, Cross-Entropy Loss function is used for multi-class classification tasks with ADAM optimizer with a learning rate of 0.01. In this methodology, Integrated Gradients attribution method is used to create instances of the model for training which helps in adding the attributing feature importance to the model's predictions.

5. System Modeling and Experiments

Experimental setup:

The experimental setup for the proposed research is shown in Figure 3. In this, it can be seen that Input Internet of Vehicles (IoV) dataset collected from Kaggle to detect rear end vehicle collision detection is applied to the machine learning and deep learning classifier for its performance. The dataset is divided into 80% training and the rest 20% for testing purposes. Next, suitable xAI methods such as DeepLift SHAP and Anchor Tabular LIME explainers are used to obtain the hidden explanations about the prediction output obtained from the classifier model. These explanations are useful in understanding the details about how the classifiers took the decision, so

as to build a transparent, interpretable and trustworthy model accepted by the users.

The deep learning algorithm architectures were implemented on Google Colab (Colaboratory, 2023) using Python libraries, particularly Keras and Tensor-

flow. The descriptive statistics along with feature correlations are also presented. Explainable AI using DeepLift SHAP is used to understand the interpretability of decision taken.

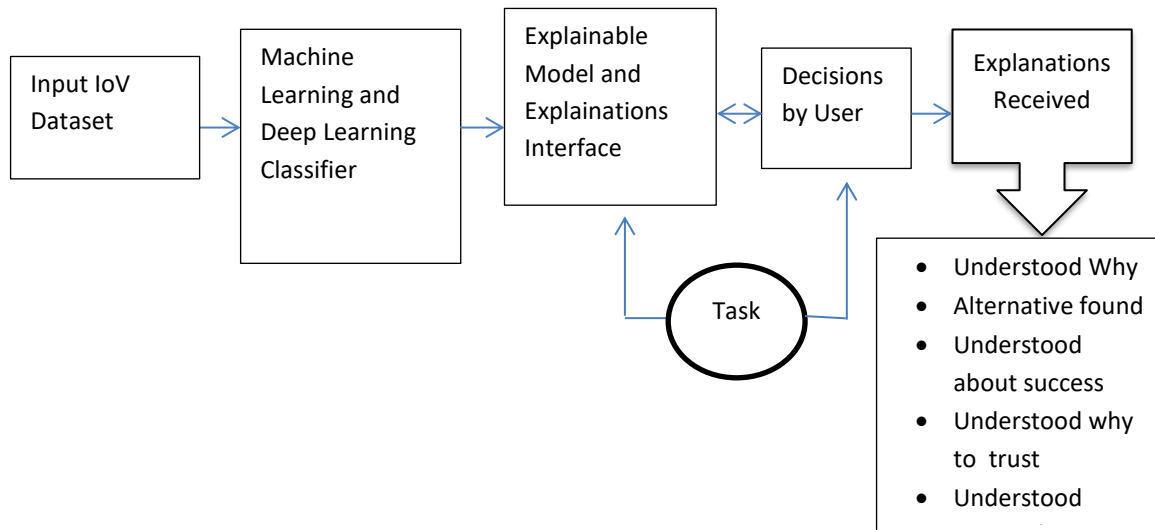


Figure 3: Proposed Experimental Framework

The correlations and feature importance are presented to get more inner details of the data for making better and transparent predictions. The experimental results of the propose Gradient Boosting, Random Forest and GRU deep learning model were compared with the other existing algorithms. The algorithms were run several times. The simulation performance of the algorithms was measured based on accuracy, precision, recall, F-measure and time taken to build the model.

6. Results and Discussions

In this section, model explanations using DeepLift explainable AI (xAI) techniques are discussed. The results obtained DeepLift explainable AI is shown in Figure 4, where the impact of the six features (i.e. status of driver, Number of Lanes, Distance between vehicles etc.) on model output prediction whether collision occurs or not are shown through mean Shapley values. At the same time, the importance of these features on model performance are shown Like: while Status of driver has maximum feature importance of approx. of 50.54%, velocity of vehicle has the lowest importance with approximately 1 %,

while predicting the possible vehicle collision in IoV scenario. The Pearson Feature Correlation amongst several IoV features using DeepLift SHAP is shown in Table 5 and the same is also shown in Figure 5. From Figure 4, one can observe that all the features equally contributing to both the class (Class 0 for no collision and Class 1 are for collision occurred) as red and blue are evenly present in all the features. Further, Status of driver is having highest significance in the IoV based collision detection followed by number of lanes and distance between vehicles. The velocity of the vehicle has the lowest significance in detection collisions in an IoV environment.

Another way to represent the feature importance map is shown in Figure 6 using the DeepLift SHAP explainer, with its SHAP values. Here, the red dot indicates more significance in comparison to blue dot. For Status of Driver, more red dots in positive x-axis implies positive impact on the model predictions, whereas the blue dots in positive x-axis shows the low feature importance in vehicle collision detection. At the same time, blue dots in the left side of x-axis have no feature importance in detecting the vehicle collision.

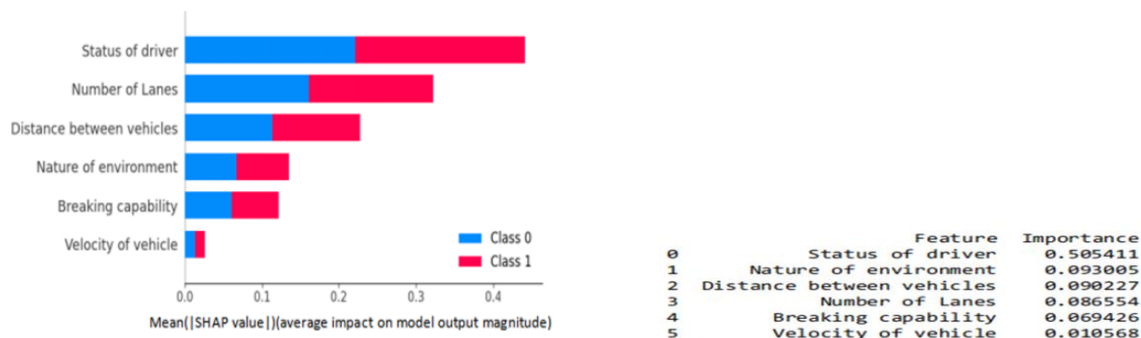


Figure 4: Explainable AI with DeepLift SHAP method for determining variable of importance

Table 5: Pearson Feature Correlation amongst several IoV features using DeepLift SHAP

	Number of Lanes	Status of driver	Nature of environment	Velocity of vehicle	Distance between vehicles	Breaking capability	Collusion
Number of Lanes	1.00	0.11	0.03	0.11	0.24	-0.02	0.51
Status of driver	0.11	1.00	-0.15	-0.48	0.50	0.12	0.67
Nature of environment	0.03	-0.15	1.00	0.10	0.14	-0.14	0.14
Velocity of vehicle	0.11	-0.48	0.10	1.00	-0.37	-0.08	-0.26
Distance between vehicles	0.24	0.50	0.14	-0.37	1.00	-0.11	0.54
Breaking capability	-0.02	0.12	-0.14	-0.08	-0.11	1.00	0.25
Collusion	0.51	0.67	0.14	-0.26	0.54	0.25	1.00

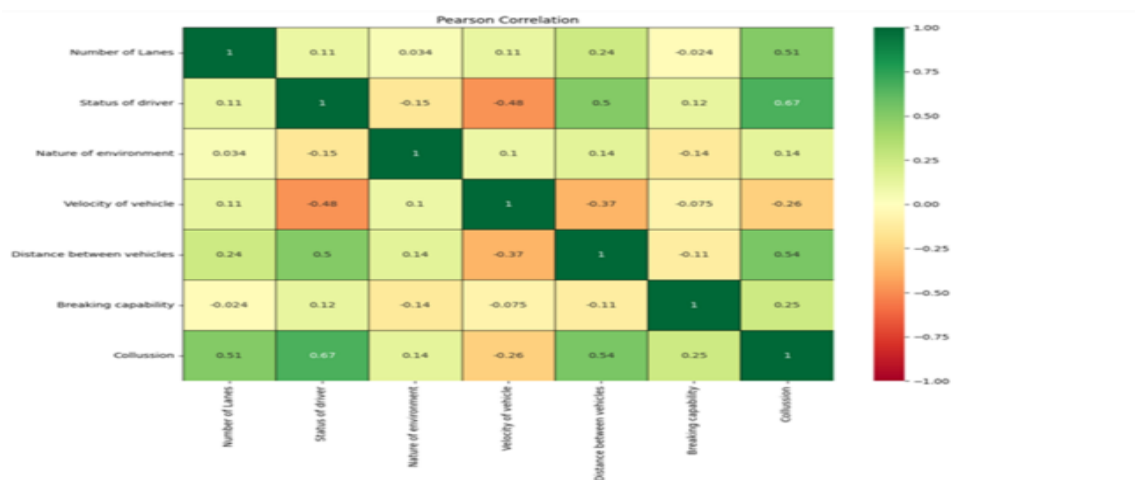


Figure 5: Pearson Feature correlation amongst all features in IoV dataset

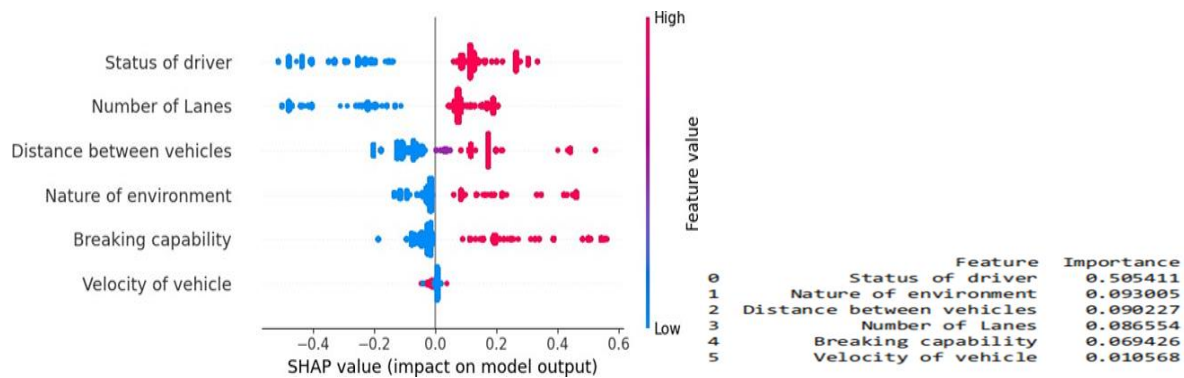


Figure 6: Summary Plot with SHAPley values for Random Forest model output

Accordingly, DeepLift SHAP applied to CatBoost, Light GBM and XGBoost classifiers and the obtained results are shown in Figure 7, Figure 8 and Figure 9 respectively. Next, the xAI using LIME (Local Interpretable Model-agnostic Explanations) anchor tabular explainer is used in all the classifiers to understand their impact on the respective models prediction. Figure 10 shows the interpretation of Random Forest classifier's prediction using LIME explainer. Now, we used LIME as local interpretable xAI method to understand the explainability of the prediction model outcome and feature relevance analysis. In this, two class level with value 1 and 0 are represented as whether vehicle collision occurs or not respectively. Only one sample from testing dataset is used for explanations. Next, LIME is used to show the explanations for the model developed using Random Forest algorithm on the IoV dataset and the results obtained are presented in Figure 10. Figure 10 reveals that when the driver is driving in a normal state in a two lane road with the minimum distance between the vehicles are 20 meter, nature of environment is normal even though the driver drives the vehicle at an over speed condition, the random forest classifier predicts the rear-end vehicle collision

100% accurately with following explanations: no collision (0.00), normal status of driver (1), distance between vehicle (0), normal environment (1), number of lanes (2) and moderate vehicle speed (1) in comparison to their respective thresholds as mentioned left side of the detailed feature values.

Similarly, all other classifiers such as Light GBM, XG Boost and Cat Boost uses LIME tabular explanations for better explanations to the model predictions which are shown in Figure 11, Figure 12 and Figure 13 respectively.

From all the above discussions using DeepLift Shap and LIME xAI methods, it is observed that Random forest is the best choice classifier in comparison to boosting model (i.e. XG Boost, CatBoost and Linear GBM) with better explanations and accurately prediction of the rear end vehicle collision detection in an IoV environment.

Next, simulations are conducted to compare the prediction performance of Random Forest Classifiers with Deep Neural network models such as: GRU, RNN and LSTM models. The comparison of these models based on the experimental results obtained is shown in Table 6.

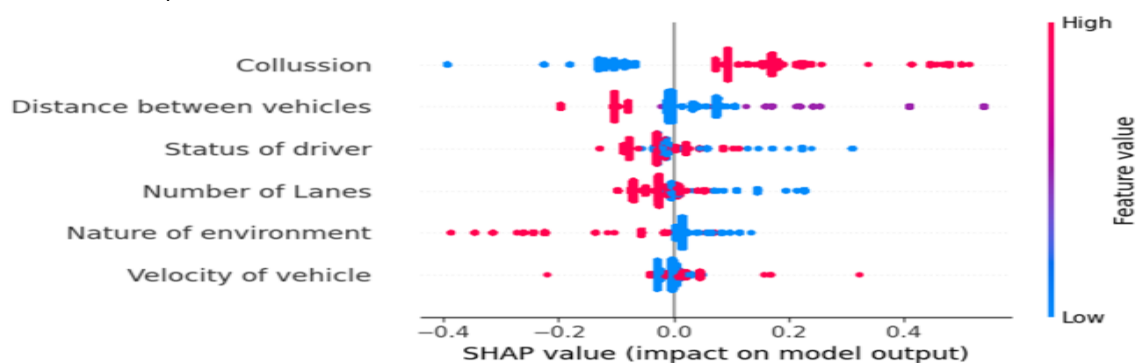


Figure 7: Summary Plot of SHAP value showing its impact on CatBoost model

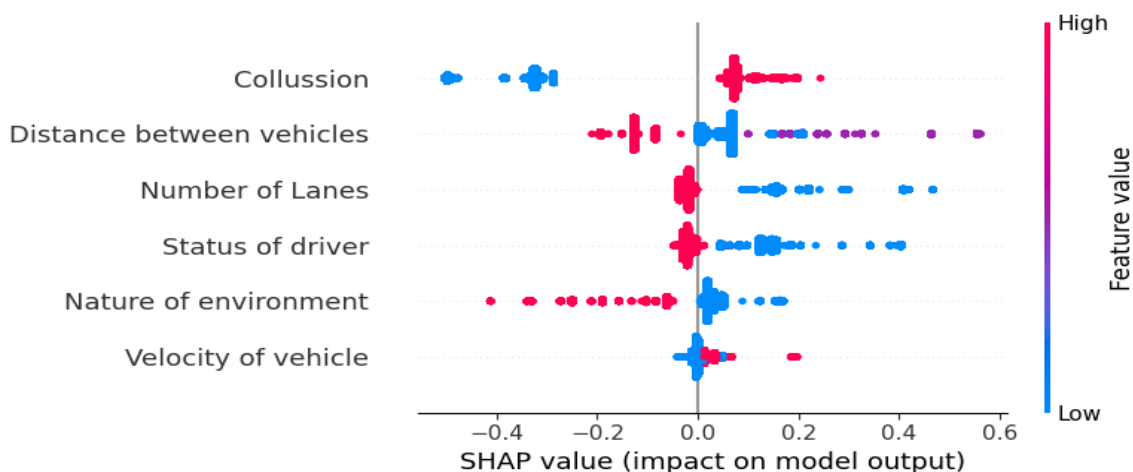


Figure 8: Summary Plot of SHAP value showing its impact on Light GBM model

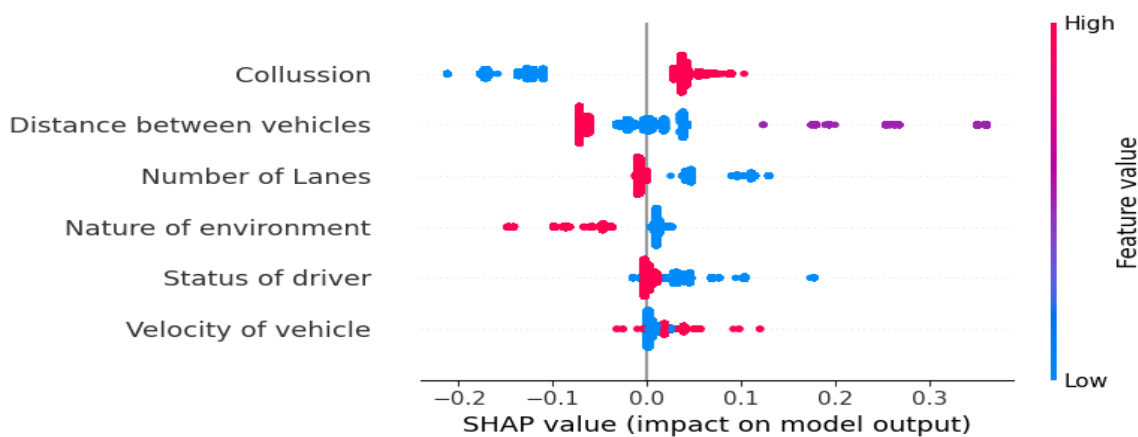


Figure 9: Summary Plot of SHAP value showing its impact on XGBoost model

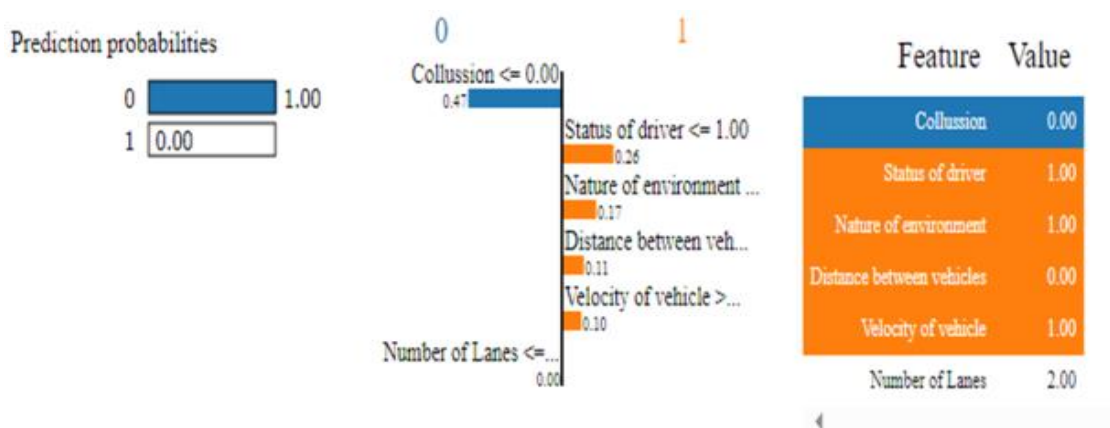


Figure 10: Interpretation of Random Forest classifier's prediction using LIME

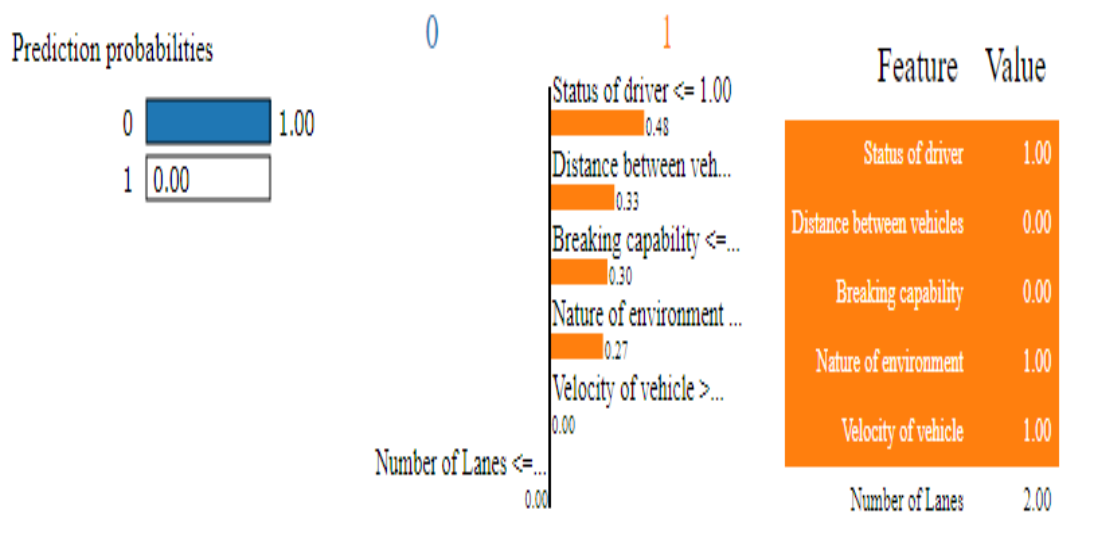


Figure 11: Interpretation of Light GBM classifier's prediction using LIME

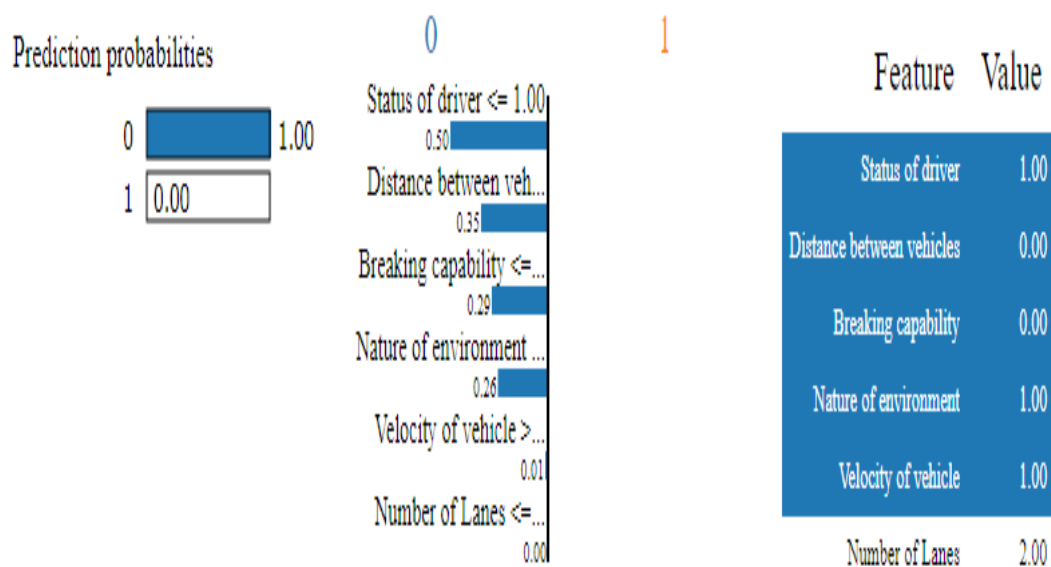


Figure 12: Interpretation of XG Boost classifier's prediction using LIME

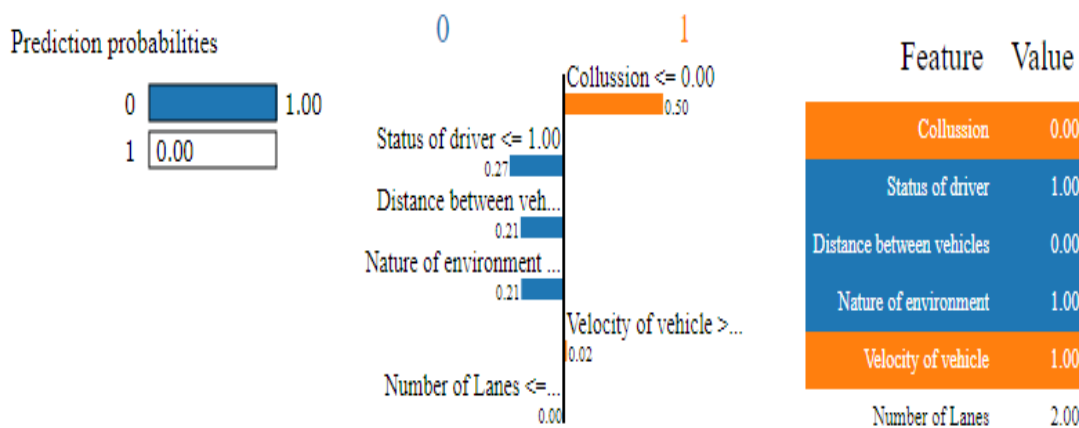


Figure 13: Interpretation of CatBoost classifier's prediction using LIME

Table 6: Comparison of Random Forest with GRU based deep learning Models

Comparison of Classifiers	precision		recall		F1-score		Model Building time in Seconds	
	GRU	RF	GRU	RF	GRU	RF	GRU	RF
Class-0, No Collision	1.00	1.00	1.00	1.00	1.00	1.00	312.57	37.25
Class-1, Collision	1.00	1.00	1.00	1.00	1.00	1.00		
accuracy	----	----	1.00	1.00	1.00	1.00		
macro avg	1.00	1.00	1.00	1.00	1.00	1.00		
weighted avg	1.00	1.00	1.00	1.00	1.00	1.00		

From Table 6, it is clear that Random Forest classifier and GRU based deep learning method performs equally well in terms of accuracy, precision, recall and f-measures, which is quite interesting, but random forest is little faster in building the model taking only 37.25 seconds in comparison to 312.57 seconds by GRU deep learning model. However, more simulation experiments might be needed to observe their performances in a more complex environment.

7. Conclusions and Future Scope

This research discusses the applicability of explainable AI with several types of explanations with example in an IoV environment. Further, xAI framework with Deep SHAP and LIME are used for post-hoc interpretation of the black box AI model to enhance the understanding of the predicted model for more transparency and acceptance. Bagging and boosting tree models are used on a publicly available IoV dataset for prediction of the rear end vehicle collision detection. Feature significance and dependence plots for SHAP as a global interpretability method along with LIME, as a local interpretability method are discussed, which added more insights to the decision making process. Several Recurrent neural network variants such as GRU, RNN and LSTM models are also used for accurate detection of rear end collision of vehicles in an IoV environment which were then compared with the Random Forest classifier. It is observed through simulations that Random Forest outperform all the Boosting classifier with better interpretability and at par with the most recent GRU based deep neural network model. In future, the aim is to explore further possibilities in a more complex IoV environment with better xAI methods for better

predictability, explainability and trustworthiness of the machine learning and deep learning models.

Statements and Declarations:

We declare that the submitted manuscript is our original work which has been read and approved by all named authors.

Competing Interests:

We declare that there are no known conflicts of interest associated with this publication. Further, it is to confirm that we have not received any financial support from any sources for this research work.

References

- [1] Feki, M. A., et al., The internet of things: the next technological revolution, Computer (2): 24-25, 2013.
- [2] Ok, J.-S., et al., A Survey of Industrial Internet of Things Platforms for Establishing Centralized Data-Acquisition Middleware: Categorization, Experiment, and Challenges, Scientific Programming, Volume 2021, Article ID 6641562, 11 pages, 2021. <https://doi.org/10.1155/2021/6641562> 2021
- [3] Labriji, I., et al., Mobility aware and dynamic migration of MEC services for the Internet of Vehicles, IEEE Transactions on Network and Service Management, 18(1), 570–584, 2021.
- [4] M.M. Rathore, H. Son, A. Ahmad, A. Paul, Real-time video processing for traffic control in smart city using Hadoop ecosystem with GPUs, Soft Comput. 1–12, 2017.
- [5] Wang, X.-S., et al., Analysis of risk factors for suburban highways using hierarchical negative binomial model. Zhongguo Gonglu Xuebao, 27(1), 100–106, 2014.

- [6] Wang, X., Liu, J., Qiu, T., Mu, C., Chen, C., & Zhou, P., A real-time collision prediction mechanism with deep learning for intelligent transportation system. *IEEE Transactions on Vehicular Technology*, 69(9), 9497–9508, 2020.
- [7] A Paul, A Ahmad, MM Rathore, S Jabbar, Smartbuddy, Defining human behaviors using big data analytics in social internet of things, *IEEE Wirel. Commun.*, 23 (5), 68–74, 2016.
- [8] M.K. Hassan, A.I. El Desouky, M.M. Badawy, A.M. Sarhan, M. Elhoseny, M. Gunasekaran, EoT-driven hybrid ambient assisted living framework with naïve Bayes-firefly algorithm, *Neural Computing and Applications*, 1–26, 2017.
- [9] Dong-Fan Xie, Zhe-Zhe Fang, Bin Jia, Zhengbing He, A data-driven lane-changing model based on deep learning, *Transportation Research Part C* 106, 41–60, 2019. <https://doi.org/10.1016/j.trc.2019.07.002>
- [10] Nekovee, M., Bie, J., Rear-End Collision: Causes and Avoidance Techniques. In: Naja, R. (eds), *Wireless Vehicular Networks for Car Collision Avoidance*, 2013. Springer, New York, NY. https://doi.org/10.1007/978-1-4419-9563-6_4
- [11] Pradhan Suvendu Kedareswar and Venkatasubramanian Krishnamoorthy, An Intelligent Embedded Diagnostic System on CAN Protocol to Avoid Rear-End Collision of Vehicles, *Indian Journal of Science and Technology*, Vol 8(19), 1-8, 2015. DOI: 10.17485/ijst/2015/v8i19/76700
- [12] Haruna Chiroma, Shafiq M. Abdulhamid, Ibrahim A. T. Hashem, Kayode S. Adewole, Absalom E. Ezugwu, Saidu Abubakar, and Liyana Shuib, Deep Learning-Based Big Data Analytics for Internet of Vehicles: Taxonomy, Challenges, and Research Directions, *Mathematical Problems in Engineering* Volume 2021, Article ID 9022558, 20 pages, Hindawi.
- [13] Razzaq S, Dar AR, Shah MA, Khattak HA, Ahmed E, El-Sherbeeny AM, Lee SM, Alkhaledi K, Rauf HT. Multi-Factor Rear-End Collision Avoidance in Connected Autonomous Vehicles. *Applied Sciences*, 12(3):1049, 2022. <https://doi.org/10.3390/app12031049>
- [14] Aloukik Aditya, Liudu Zhou, Hrishika Vachhani, Dhivya Chandrasekaran, and Vijay Mago. Collision Detection: An Improved Deep Learning Approach Using SENet and ResNext. In 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE Press, 2075–2082, 2021. <https://doi.org/10.1109/SMC52423.2021.9659265>
- [15] Fateme Teimouri, Mehdi Ghatee, A real-time warning system for rear-end collision based on random forest classifier, *Journal of Soft Computing in Civil Engineering*, 4(1), 49-71, 2020
- [16] Ma, X.; Yu, Q.; Liu, J. Modeling Urban Freeway Rear-End Collision Risk Using Machine Learning Algorithms. *Sustainability*, 14, 12047, 2022. <https://doi.org/10.3390/su141912047>
- [17] Muzammel, M.; Yusoff, M.Z.; Saad, M.N.M.; Sheikh, F.; Awais, M.A. Blind-Spot Collision Detection System for Commercial Vehicles Using Multi Deep CNN Architecture. *Sensors*, 22, 6088, 2022. <https://doi.org/10.3390/s22166088>
- [18] Muzahid, A.J.M., Kamarulzaman, S.F., Rahman, M.A. et al. Multiple vehicle cooperation and collision avoidance in automated vehicles: survey and an AI-enabled conceptual framework. *Sci Rep* 13, 603, 2023. <https://doi.org/10.1038/s41598-022-27026-9>
- [19] Jae Gyeong Choi, Chan Woo Kong, Gyeongho Kim, Sunghoon Lim, Car crash detection using ensemble deep learning and multimodal data from dashboard cameras, *Expert Systems with Applications*, 183, 115400, 2021. <https://doi.org/10.1016/j.eswa.2021.115400>
- [20] Wang, Xin; Liu, Jing; Qiu, Tie; Mu, Chaoxu; Chen, Chen and Zhou, Pan, A Real-Time Collision Prediction Mechanism With Deep Learning for Intelligent Transportation System, *IEEE Transactions on Vehicular Technology*, Volume: 69, Issue Number: 9, pp. 9497-9508, Sept. 2020, DOI: 10.1109/TVT.2020.3003933. IEEE.
- [21] Chen, M., Zhan, X., Tu, J. and Liu, M., Vehicle-localization-based and DSRC-based autonomous vehicle rear-end collision avoidance concerning measurement uncertainties. *IEEJ Trans Elec Electron Eng*, 14: 1348-1358, 2019. <https://doi.org/10.1002/tee.22936>
- [22] Chen, M., Zhan, X., Zhang, X. and Pan, W., Localisation-based autonomous vehicle rear-end collision avoidance by emergency steering.

- IET Intell. Transp. Syst., 13: 1078-1087, 2019.
<https://doi.org/10.1049/iet-its.2018.5348>
- [23] Almutairi, M. S., Almutairi, K., & Chiroma, H., Hybrid of deep recurrent network and long short term memory for rear-end collision detection in fog based internet of vehicles. *Expert Systems with Applications*, 213, 119033, 2023.
- [24] Q.Y. Tang, C.X. Zhang, Data Processing System (DPS) software with experimental design, statistical analysis and data mining developed for use in entomological research, *Insect Science*, 20 (2), pp. 254-260, 2013.
- [25] A. Shrikumar et al., Learning important features through propagating activation differences. *ICML 2017*, p. 4844-4866, 2017.
- [26] J. Li, C. Zhang, J. T. Zhou, H. Fu, S. Xia and Q. Hu, "Deep-LIFT: Deep Label-Specific Feature Learning for Image Annotation," in *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7732-7741, Aug. 2022, DOI: 10.1109/TCYB.2021.3049630.
- [27] Lundberg, Scott, Lee, Su-In, A Unified Approach to Interpreting Model Predictions, *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, December 2017, Pages 4768–4777.
- [28] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". *ICML'17: Proceedings of the 34th International Conference on Machine Learning - Volume 70* August 2017 Pages 3145–3153
- [29] Avanti Shrikumar et al. "Not Just a Black Box: Learning Important Features Through Propagating Activation Differences". In: *arXiv preprint arXiv:1605.01713*, 2016.
- [30] Leo Breiman, *Random Forests*, *Machine Learning*, 45, 5–32, 2001.
- [31] C. Hu, J. Yan, and C. Wang, "Advanced Cyber-Physical Attack Classification with Extreme Gradient Boosting for Smart Transmission Grids," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2019-Augus, 2019, DOI: 10.1109/PESGM40551.2019.8973679.
- [32] A. Husain, A. Salem, C. Jim, and G. Dimitoglou, "Development of an Efficient Network Intrusion Detection Model Using Extreme Gradient Boosting (XGBoost) on the UNSW-NB15 Dataset," 2019 IEEE 19th Int. Symp. Signal Process. Inf. Technol. ISSPIT 2019, 2019, DOI: 10.1109/ISSPIT47144.2019.9001867
- [33] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "Catboost: Unbiased boosting with categorical features," *Adv. Neural Inf. Process. Syst.*, vol. 2018-December, Section 4, pp. 6638–6648, 2018.
- [34] Qiao, J.; Qiu, S.; Sun, Y. Combination of Feature Selection and CatBoost for Prediction: The First Application to the Estimation of Aboveground Biomass. *Forests*, 12, 216, 2021. <https://doi.org/10.3390/f12020216>
- [35] Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv 2014*, arXiv:1412.3555
- [36] Guizhu Shena, Qingping Tana, Haoyu Zhanga, Ping Zenga, Jianjun Xu, *Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions for Financial Sequence Predictions*, 8th International Congress of Information and Communication Technology (ICICT-2018), *Procedia Computer Science* 131, 895–903, 2018.