Accepted version

To appear in *Proc. 2023 19th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* Montreal, Canada, 2023.

Analyzing Inter-Vehicle Collision Predictions during Emergency Braking with Automated Vehicles

Joseba Gorospe*, Shahriar Hasan†, Mir Riyanul Islam†,
Arrate Alonso Gómez*, Svetlana Girs†, Elisabeth Uhlemann†
*Electronics and Computer Science Department, Mondragon Unibertsitatea, Arrasate, Spain
†School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden
Corresponding author: J. Gorospe (e-mail: jgorospe@mondragon.edu)

***Abstract*—Automated Vehicles (AVs) require sensing and perception to integrate data from multiple sources, such as cameras, lidars, and radars, to operate safely and efficiently. Collaborative sensing through wireless vehicular communications can enhance this process. However, failures in sensors and communication systems may require the vehicle to perform a safe stop or emergency braking when encountering hazards. By identifying the conditions for being able to perform emergency braking without collisions, better automation models that also consider communications need to be developed. Hence, we propose to employ Machine Learning (ML) to predict inter-vehicle collisions during emergency braking by utilizing a comprehensive dataset that has been prepared through rigorous simulations. Using simulations and data-driven modeling has several advantages over physics-based models in this case, as it, e.g., enables us to provide a dataset with varying vehicle kinematic parameters, traffic density, network load, vehicle automation controller parameters, and more. To further establish the conditions for inter-vehicle collisions, we analyze the predictions made through interpretable ML models and rank the features that contribute to collisions. We also extract human-interpretable rules that can establish the conditions leading to collisions between AVs during emergency braking. Finally, we plot the decision boundaries between different input features to separate the collision and non-collision classes and demonstrate the safe region of emergency braking.**

I. INTRODUCTION

To address the challenges of modern transportation systems and improve traffic safety and efficiency, Connected and Automated Vehicles (CAVs) offer a promising solution. Through advanced sensor fusion technologies and Vehicle-to-Vehicle (V2V) communications, CAVs can enable cooperative awareness, paving the way for safe, efficient, and Intelligent Transportation Systems (ITSs). Specifically, automated vehicles can employ Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) to closely follow a front vehicle and promote fuel and road efficiency while increasing traffic flow. In addition, CAVs can form platoons with a group of tightly-coupled vehicles called Following Vehicles (FVs) that follow the lateral and longitudinal movements of a Lead Vehicle (LV) using onboard sensors and V2V communications.

The benefits that CAVs can endorse are closely tied to the inter-vehicle distances between the vehicles [1]. For instance, shorter gaps allow higher fuel efficiency by minimizing aerodynamic drags and also increase road throughput. However, this creates a challenge when emergency braking is necessary due to, e.g., encountering a road hazard. Short gaps, high speed, and the requirement for strong deceleration to minimize the stopping distance for evading the risk add to the challenge. Furthermore, wireless communications, a key enabling technology of CAV systems, can experience transient communication outages due to path loss, multipath fading, interference, etc., [2]. These outages can result in time-varying communication delays, further complicating safety management in CAVs. Hence, CAV systems must be designed and operated to avoid inter-vehicle collisions and hazards, as well as transition to a known safe state in the event of any failures or malfunctions.

To this end, this paper proposes a data-driven approach to predict inter-vehicle collisions during emergency braking by automated vehicles. Our approach trains ML models during the design phase to learn the input parameter values that lead to collisions. This knowledge can then be used during runtime to adjust the input parameter values and prevent collisions in the event of an emergency. To prepare the datasets for our study, we conducted extensive simulations of emergency braking scenarios, varying parameters such as vehicle dynamics, network load, traffic density, speed, message frequency, controllers, inter-vehicle gaps, etc., and recorded the collision and non-collision instances. These simulation parameters form the feature vector, which we use to train several ML models, including black-box and interpretable models. We evaluate the performance of these models in classifying emergency braking events as either collision or non-collision cases. While the black-box models are evaluated for the sake of comparison, the main focus of this paper is on the interpretable models, which provide explanations for the predictions and reasons for collisions. We analyze these models by ranking features, generating decision boundaries, and extracting rules for collisions. The feature ranking allows us to identify the significance of different parameters in causing collisions during emergency braking, while the decision boundaries help us understand the input parameter values leading to the prediction of collisions. Additionally, the rules extracted from Decision Tree (DT) models give us the conditions that lead to collision predictions.

The rest of the paper is structured as follows: Section II introduces our system model, including different controllers and their communication settings. After that, Section III reviews the related works that employ ML models for collision

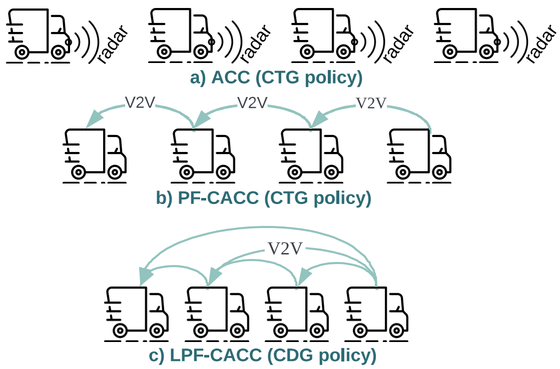

Fig. 1: Communication structure with different controllers.

avoidance. Next, we describe the simulation settings and scenarios in Section IV that are used to prepare the datasets. The training and evaluation of different ML models, including the interpretation criteria of the model predictions, are described in Section V. The evaluation results are then presented in Section VI. Finally, Section VII concludes the paper.

II. System Model

We have identified two primary requirements for predicting collisions in automated vehicles. Firstly, predictions must be made based on real-time information such as experienced communication quality, inter-vehicle gaps, speed, and other relevant factors. Secondly, ML models must be trained using accurate data and safeguarded against data tampering by potential attackers. To address these requirements, we assume that pre-trained ML models are installed in the vehicles and that these models can receive updates from trusted sources. Since these models are installed in the vehicles and can generate predictions during runtime, they serve as an onboard prediction tool. In order to predict collisions in the event of an emergency braking, we use a binary classification model to approximate a mapping function f for input variables described in Table I. The class labels TRUE and FALSE denote collision and non-collision, respectively. Additionally, our prediction tool generates collision avoidance rules, including instructions such as speed, inter-vehicle gaps, deceleration rates, and emergency braking strategies that must be followed.

Whether a vehicle acts on the predictions or instructions provided by the onboard prediction tool in a distributed or centralized manner depends on the type of controller the vehicle uses. Fig. 1 illustrates three different control strategies: ACC, Predecessor-Following CACC (PF-CACC), and Leader-Predecessor Following CACC (LPF-CACC). In ACC, a vehicle follows a desired gap with its predecessor using its onboard radar sensors to measure the relative distance and speed. When V2V communications are added to ACC, it becomes CACC. There are several types of CACC controllers. For instance, in PF-CACC shown in Fig. 1, a vehicle computes its acceleration based on information obtained from its immediate predecessor through V2V communications. Both ACC and PF-CACC use a Constant Time Gap (CTG) policy, where the inter-vehicle gaps change with the change in the speed of the vehicles. When a *string of vehicles* uses ACC or PF-CACC, the onboard prediction tool is used in a distributed manner. However, in a centralized *platoon of CAVs* using LPF-CACC, the onboard prediction tool instructions regarding speed, inter-vehicle gaps, deceleration rates, etc., are communicated to the following vehicles by the leading vehicle. In LPF-CACC, the vehicles form a platoon where each vehicle receives information from both its predecessor and the LV, as depicted in Fig. 1. Moreover, a Constant Distance Gap (CDG) policy is followed in LPF-CACC, i.e., the inter-vehicle gaps remain constant despite speed changes. Note that in the rest of the paper, the term *platoon* denotes a group of vehicles using the LPF-CACC controller, which maintains the CDG policy. In contrast, a group of vehicles using the ACC or PF-CACC controller is regarded as a *vehicle string* that uses the CTG policy.

In order to evaluate the performance of our proposed collision prediction approach across different automated vehicle scenarios, we prepared three separate datasets in this study by simulating the ACC, PF-CACC, and LPF-CACC controllers. By using multiple datasets with different controllers, we can better understand the impact of controller choice on collision prediction accuracy and investigate the potential benefits of using a specific controller in reducing collision risk.

III. Related Works

Recent research has shown promise for using ML models in collision prediction. For instance, Lin *et al.* investigate the performance of various ML models for predicting the risk of accidents within an intersection in [3]. In [4], Chen *et al.* study the prediction of rear-end collisions in the internet of vehicles using a Back-Propagation Neural Network (BPNN). Ribeiro *et al.* use the Long Short-Term Memory (LSTM) network to predict collisions between a motorcycle and a vehicle in an intersection [5]. Wang *et al.* propose a Rear-end Collision Prediction Mechanism (RCPM) using deep learning to predict collisions [6]. However, the use of black-box ML models in these works lacks the ability to explain predictions, which is crucial for sensitivity analysis and understanding the factors driving predictions.

Qian [7] proposes a two-phase approach for vehicle collision prediction. The first phase uses LightGBM, a gradient-boosting framework based on tree learning algorithms, to predict collisions. If a collision is predicted, the second phase uses the K-nearest neighbor model to determine the time of the predicted collision. However, the author does not provide any interpretation of the predictions in this work. In [8], Ferni *et al.* consider a dataset obtained through simulations with features such as speed, acceleration, distance, braking force, and packet error rate to identify collisions during braking in a platoon. The authors then use Logic Learning Machine (LLM) to generate a set of rules to find safety regions during braking in a platoon. Mongelli *et al.* [9] extended the prior work [8] to conduct intelligibility analysis on string stability in a platoon through knowledge extraction and sensitivity analysis of collision avoidance during braking. *String stability* refers to the ability of a control system to dampen disturbances or perturbations propagating from the LV to the FVs in a string of vehicles. The use of interpretable ML models and the explanation of predictions in the works [8] and [9] represent a great strength. The authors demonstrate how this approach can help devise rules for collision avoidance and maintain string stability in vehicle platoons. However, the simulations conducted in these works do not account for the time-varying communication delays caused by changing neighboring traffic densities. Instead, packet error rates were used to characterize the communication medium. In contrast, our work simulates a variety of parameters that affect communication quality and collisions during emergency braking. Additionally, we examine different types of controllers and analyze the factors contributing to collisions when using these controllers.

IV. Simulations for Dataset Generation

This section outlines the controllers and emergency braking strategy employed for the simulations, followed by a description of the simulation scenarios used to generate the datasets. The simulations in this paper are carried out using Platoon-SAFE [10], a simulation tool for evaluating platoon safety during, e.g., cruising and emergency braking, using realistic modeling of wireless communications. In the simulations, the Physical (PHY) and Medium Access Control (MAC) layer parameters follow the IEEE 802.11p specifications [11]. In addition, the free space path loss model ($\alpha = 2$) and the Nakagami-m fading model ($m = 1.86$) are used to simulate the path loss and fading effects, respectively.

A. Control Laws and Parameters

The control laws proposed by Ioannou and Chien [12], Ploeg *et al.* [13], and Rajamani *et al.* [14] are used to represent ACC, PF-CACC, and LPF-CACC, respectively. Table I lists the parameter values, such as CTGs and CDGs, speeds, packet size, and beacon intervals used for simulations. In addition, for readers' convenience, we express various CTGs in meters for different speeds in Table II.

B. Emergency Braking Strategy

In the simulations carried out in this paper, the Synchronized Braking (SB) strategy proposed in [15] is used during the emergency braking maneuver. With the SB strategy, the LV broadcasts Hazard Warning (HW) messages upon encountering a road hazard and instructs the FVs to wait for a τ_{wait} period before braking. The rationale behind waiting before braking is to synchronize the braking actions of the vehicles to mitigate the effects of communication delays. Studies have shown that braking hard and starting to brake as soon as a message is received can lead to collisions, particularly in scenarios where communication delays are high [2]. Therefore, a delay-aware emergency braking strategy, e.g., SB, is better in terms of collision avoidance and minimizing the stopping distance of the LV since a higher deceleration can be used. However, collisions may still happen with the SB strategy if the experienced communication delays are significantly higher than the predicted τ_{wait} period. The onboard prediction tool can also instruct the vehicles to pursue the τ_{wait} period that is required to avoid potential collisions.

C. Simulation Scenarios for Generating the Dataset

In this study, we conducted rigorous simulations to generate datasets for training and evaluating the ML models. We systematically varied the parameters listed in Table I that affect communication delays and collision events during emergency braking. Each simulation run changed only one parameter while keeping others constant, and the resulting data was used to create three separate datasets based on the control policy used in the vehicles, i.e., *ACC*, *PF-CACC*, and *LPF-CACC*. Within each dataset, all vehicles in the same vehicle string or platoon utilized the same controller. Moreover, to simulate the periodic beacons and HW messages during cruising and emergency braking, we employed Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs), respectively [16]. Upon emergency braking, if the inter-vehicle gap between any pair of vehicles at zero speed is greater than zero, it is considered a non-collision case. In order to generate different emergency braking scenarios, we also vary the τ_{wait} period, which is required for SB.

In summary, we generated three different datasets for three different controllers, i.e., ACC, PF-CACC, and LPF-CACC, in which the emergency braking is performed with the SB strategy. In these datasets, the independent variables (features) are the parameters presented in Table I with their corresponding values. Moreover, to specify the collision and non-collision cases, a binary variable (TRUE or FALSE) is considered as the dependent variable (label) across the three datasets. In particular, after preprocessing the raw simulation data, the final dataset for the ACC controller contains 4794 labeled instances with four features, and the datasets for both PF-CACC (459 instances) and LPF-CACC (948 instances) have ten features each. The reason for the smaller number of features in the ACC dataset is that vehicles using the ACC controller do not rely on V2V communications and instead use onboard sensors.

V. ML Models for Collision Prediction and Its Interpretation

This section outlines the ML models used for collision prediction and the criteria used to evaluate the models and interpret their predictions.

A. ML models for Collision Prediction

In this paper, we evaluate the effectiveness of different ML models in predicting collisions during emergency braking. To this end, we assess the performance of five commonly used and relatively simple models: Decision Tree (DT) and Random Forest (RF), which are interpretable models, and Logistic Regression (LR), Support Vector Classifier (SVC), and Multi-layer Perceptron (MLP), which are black-box models. To emphasize interpretability, DT models with different parameters are used for different datasets to interpret the prediction of

TABLE I: Input parameters for simulations and dataset generation.

Parameter $[Identifier]$	Value	Parameter $[Identifier]$	Value
Controllers $[C_i]$	ACC, PF-CACC, LPF-CACC	CTGs with ACC $[CTG_{ACC}]$	0.3, 0.4, 0.5, 0.6, 0.8, 1.0, 1.2 s
CTGs with PF-CACC $[CTG_{CACC}]$	0.2, 0.3, 0.4, 0.5 s	CDGs with LPF-CACC $[CDG_{CACC}]$	3, 4, 5, 8 m
String size $[N_s]$ or platoon size $[N_p]$	7, 8, 10, 12	No. of neighboring vehicles $[N_h]$	100, 200, 300, 400, 500
Beacon interval (platoon) $[f_p]$	50, 100 ms	Beacon interval (neighbors) $[f_h]$	10, 20, 30, 40 ms
DENM Interval $[f_{denm}]$	50, 100 ms	Packet size $[B]$	100, 500 B
Leader speed $[\dot{x}_{LV}]$	80, 100 kmh^{-1}	Deceleration rate $[\ddot{x}]$	–6, –8 ms^{-2}

TABLE II: Various CTGs (s) expressed in meters.

CTGs (s)	0.2	0.4	0.5	0.6	0.8	1.0	1.2
80 kmh^{-1}	6.44	10.88	13.1	15.32	19.76	24.2	28.64
100 kmh^{-1}	7.55	13.11	15.89	18.67	24.23	29.79	35.35

TABLE III: Confusion matrix and different classification performance metrics.

Total Instance P + N		Prediction	
		Positive	Negative
True	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Precision: $\frac{TP}{TP+FP}$; Recall: $\frac{TP}{TP+FN}$

collisions. Specifically, DT models with eight, two, and seven leaves are used for ACC, PF-CACC, and LPF-CACC datasets, respectively. In order to assess the performance of the ML models, we adopt the widely used holdout method, where 75% of the dataset (*training dataset*) is used to train the model, and the remaining 25% (*test dataset*) is used for evaluating the ML models.

B. Evaluation Criteria of ML models

In order to evaluate the ML models, we employ the classification metrics *precision* and *recall*, which are illustrated with the help of a confusion matrix in Table III. Precision measures the percentage of True Positive (TP) predictions out of all positive predictions, while recall measures the percentage of TP predictions out of all actual positive instances. We focus on these two metrics because detecting the positive class, i.e., collisions, is more critical than detecting the negative class, i.e., non-collisions. Moreover, recall is of greater significance as a higher recall value indicates fewer False Negative (FN) results. In safety-critical systems such as CAVs, misclassifying collisions as non-collisions (FN) is more hazardous than misclassifying non-collisions as collisions (FP).

C. Interpreting ML Model Predictions

Classifier models such as SVM and Convolutional Neural Network (CNN) are known to perform well but are often less interpretable, making it unclear how their internal algorithm leads to extract predictions. In contrast, DT and RF are simpler and generate rules that can be more easily read and understood with domain knowledge, making them interpretable and useful for extracting knowledge from the model. This is especially important for understanding the impact of specific input parameters on the model outcome [17]. It is worth noting that while LR also produces interpretable predictions in the form of mathematical equations, understanding these equations requires both expertise in mathematics and knowledge in the ML domain. To this end, this paper uses DT or RF (depending on the recall value and explainability) to interpret the predictions.

1) Feature Importance: We choose an interpretable model, such as DT or RF, by evaluating its performance in terms of precision and recall on the test dataset and considering its explainability and usability. This model is then used to rank the features listed in Table I, enabling us to determine which input parameters have a more significant impact on inter-vehicle collisions during emergency braking.

2) Rule Extraction: We extract a set of human-interpretable rules by analyzing the decision path of a DT, which defines the combinations of input parameter values that lead to the prediction of collisions during emergency braking. These rules can be used to develop guidelines for designing safe CAV systems, e.g., the authors in [8] extract such rules using LLM. Additionally, the rules can be used to develop a decision support system that can warn the driver or automatically trigger an emergency braking system if the input parameters fall within the range of values that are likely to result in collisions.

3) Decision Boundaries: In order to identify the parameter values that lead to collisions during emergency braking, we employ decision boundaries that separate collision and non-collision classes. The most important parameters from Table I are selected according to the feature ranking and plotted in different combinations for each type of controller. The objective is to visually identify the safe regions of the input parameters where the parameter values do not lead to collisions.

VI. Results Evaluation

A. Evaluation of ML models in Predicting Collisions

Table IV presents the performance of various ML models in classifying collisions and non-collisions for ACC, PF-CACC, and LPF-CACC datasets; the best results in terms of precision and recall are highlighted in bold. The results in Table IV show that DT achieves the highest recall value for the ACC dataset, although its precision is lower than that of RF and MLP. On the other hand, for the PF-CACC and LPF-CACC datasets, RF and MLP exhibit better performance in terms of precision and recall compared to DT. Specifically, both RF and MLP achieve the same precision and recall for the LPF-CACC dataset, but MLP has a slightly higher recall than RF for the PF-CACC dataset.

TABLE IV: Precision and recall of different ML models. The highest values are highlighted.

Classifiers	ACC		PF-CACC		LPF-CACC	
	Precision	Recall	Precision	Recall	Precision	Recall
LR	0.570	0.958	0.913	0.792	0.430	0.889
SVC	0.597	0.986	0.913	0.792	0.424	0.933
DT	0.486	**0.993**	0.913	0.792	0.472	0.933
RF	0.903	0.910	**0.923**	0.906	0.953	0.911
MLP	**0.944**	0.944	0.891	**0.925**	**0.956**	**0.956**

One possible explanation for why DT performs better with the ACC dataset could be that this dataset has fewer features, as the communication-related features in Table I do not apply to the ACC controller. Due to the simplicity of the dataset, DT can efficiently classify between collisions and non-collisions by simple decision boundaries. In contrast, the PF-CACC and LPF-CACC datasets include communication parameters, resulting in more complex relationships between input parameters, which could explain why RF and MLP perform better than DT in these datasets.

In conclusion, choosing an ML model that suits the dataset characteristics and allows for interpretability is crucial. Table IV shows that interpretable models such as DT and RF exhibit good recall values. For safety-critical systems, these models are preferred over black-box models like MLP as they offer better insights into the decision-making process.

B. Interpreting the predictions of DT or RF

1) Feature Importance: Figs. 2, 3, and 4 depict the feature importance rankings for the ACC, PF-CACC, and LPF-CACC datasets, respectively. The feature importance for the ACC dataset is ranked using DT, which exhibits the highest recall value in Table IV. In contrast, for the PF-CACC and LPF-CACC datasets, RF is used to rank the feature importance since it shows higher precision and recall than DT, as shown in Table IV.

Fig. 2 shows that deceleration rate and time gap are the most crucial factors during emergency braking situations when the ACC controller is engaged during cruising. This is because vehicles using ACC rely only on their onboard sensors, which are subject to detection, processing, and actuation delays. Consequently, a longer inter-vehicle gap is necessary when the preceding vehicle undergoes sudden deceleration to ensure a timely response to the braking and avoid a collision. However, when communication is added on top of ACC in the PF-CACC algorithm, the experienced communication delays become the dominant factor that dictates the likelihood of collisions. This finding is illustrated in Fig. 3, which shows that the beacon interval of neighboring vehicles is the most significant feature identified by the RF classifier. The number of neighboring vehicles and their beacon frequency largely determine the data density. As a result, the vehicles in the string experience higher communication delays in scenarios with dense data traffic due to a higher channel-busy ratio. Fig. 4, which presents the feature importance for the LPF-CACC dataset, also highlights the substantial impact of neighboring vehicles

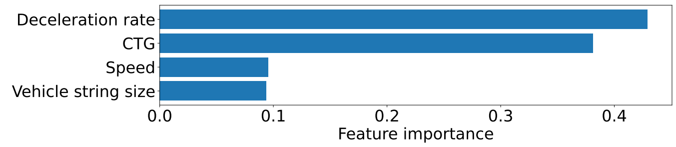

Fig. 2: Feature ranking for ACC dataset using DT.

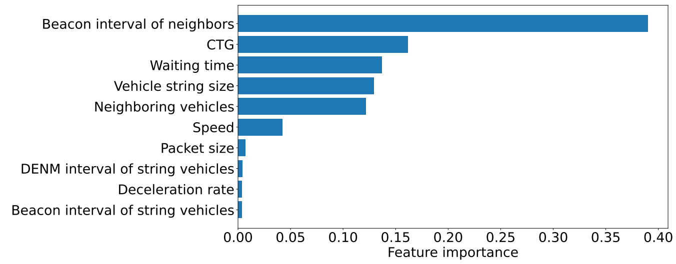

Fig. 3: Feature ranking for PF-CACC dataset using RF.

and their beacon intervals on the communication quality and the likelihood of collisions. In addition, Fig. 3 demonstrates that the time gap and waiting time before braking using the SB strategy significantly influence inter-vehicle collisions during emergency braking when PF-CACC is employed during cruising. Similar results can also be observed with the LPF-CACC dataset, as depicted in Fig. 4. Nevertheless, the inter-vehicle gap remains the most crucial factor when using the LPF-CACC algorithm. This is because the inter-vehicle gaps with the LPF-CACC algorithm are considerably shorter than with the PF-CACC algorithm, increasing the demand for attainable communication quality. In LPF-CACC, if there is a temporary communication outage between an ego vehicle and the lead vehicle, the maintained short inter-vehicle gaps can potentially lead to collisions in the event of emergency braking. Moreover, as shown in Fig. 4, platoon length is also important when operating with the LPF-CACC algorithm. The rationale is that the rear vehicles in a platoon experience longer delays due to path loss and fading effects, which increases with an increased distance between the transmitter (the LV) and the receiver (an ego vehicle).

2) Rule Extraction: In order to generate concise and interpretable rules, we use the same DT models as in Table IV, which enables us to extract potential collision rules during an emergency braking. Note that the same DT model can be used to extract the rules for collision avoidance, which we do not present here for brevity. We refer the reader to Table I, where the nomenclature of the input parameters used to represent the

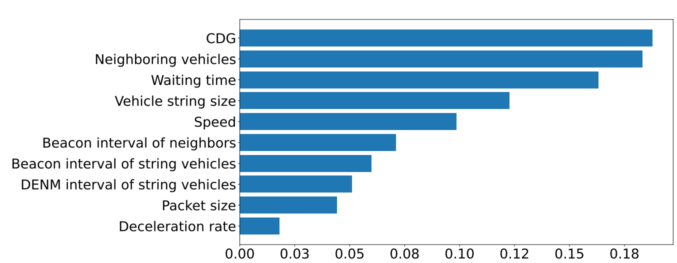

Fig. 4: Feature ranking for LPF-CACC dataset using RF.

rules is provided. Moreover, please refer to Table II in which CTGs expressed in meters are provided.

a) Rules for ACC dataset: Equation (1) outlines the rules governing potential collisions for the ACC dataset, revealing clear relationships between CTGs, deceleration rates, and speed. For example, the rules indicate that collisions may occur if the CTG falls between 0.65 and 0.85 s, the deceleration rate exceeds 7.5 ms^{-2}, and the speed is above 75 ms^{-1}. Similarly, if the CTG is less than 0.35 s and the ACC controller is in use, deceleration rates between 4.5 and 5.5 ms^{-2} can lead to potential collisions. These results suggest that vehicles in a string utilizing an ACC controller should avoid abrupt deceleration to prevent collisions with the preceding vehicle.

$$\begin{aligned}&\text{if } (((\ddot{x} \geq 4.5) \wedge (\ddot{x} \leq 5.5) \wedge (CTG_{ACC} \leq 0.35))\\ &\vee((\ddot{x} \geq 5.5) \wedge (CTG_{ACC} \leq 0.65))\\ &\vee((\ddot{x} \geq 7.5) \wedge (CTG_{ACC} \geq 0.65)\\ &\wedge(CTG_{ACC} \leq 0.85) \wedge (\dot{x}_{LV} \geq 75))) \text{ then collision}\end{aligned} \tag{1}$$

b) Rules for PF-CACC dataset: Equation 2 shows that DT provides only one collision condition for the PF-CACC dataset. The rationale is that the beacon intervals of the neighboring vehicles are significantly more important than the other features, as depicted in Fig 3. Therefore, in this case, DT is overlooking other features as they may not provide additional information beyond what is already captured by the neighboring vehicles' beacon intervals. When a beacon interval of less or equal to 0.01 s is used with the neighboring vehicles, there are collisions in 86.68% of the simulation runs. This clearly shows the impact of neighboring data and road traffic on the experienced communication quality during emergency braking.

$$\text{if } (f_h \leq 0.01) \text{ then collision} \tag{2}$$

c) Rules for LPF-CACC dataset: Equation 3 depicts the rules for the LPF-CACC dataset. Here, we can observe that the waiting time before braking (τ_{wait}) with the SB strategy, platoon size N_p, number of neighboring vehicles N_h, and CDG are the most dominant factor for inter-vehicle collisions. For instance, if the platoon length exceeds eight vehicles and the beacon interval of the neighboring vehicles f_h is less than 0.02 s, a collision may occur. The reason is that with the LPF-CACC controller, a vehicle needs information directly from the LV, and the inter-vehicle gaps are short. As a result, when the vehicles, especially the rear ones, experience transient communication outages with the LV, the maintained short inter-vehicle gaps are not sufficient to respond to the braking of a front vehicle.

$$\begin{aligned}&\text{if } (((N_p \geq 8) \wedge (f_h \leq 0.02))\\ &\vee((N_p \leq 9) \wedge (N_h \geq 200) \wedge (f_{denm} \leq 0.08)\\ &\wedge(\tau_{wait} \geq 0.07) \wedge (CDG_{CACC} \leq 6.5)))\\ &\text{ then collision}\end{aligned} \tag{3}$$

3) Decision Boundaries: In order to plot the decision boundaries, we retrain the DT models using two features, one of which pertains to the inter-vehicle gaps, and the other feature is chosen based on the feature importance shown in Figs. 2, 3, and 4. While only two features may not capture the full complexity of the datasets and lead to a less accurate model, the resultant decision boundaries with two features facilitate more interpretability than with the ten features. Therefore, it is crucial to consider the trade-off between interpretability and model accuracy when such decision boundaries are drawn to separate the target classes. The decision boundaries for the ACC, PF-CACC, and LPF-CACC datasets are depicted in Figs. 5, 6, and 7, respectively. In these figures, the blue region signifies the safe region of emergency braking, and the red region represents the feature values that may potentially lead to collisions.

The decision boundary plot in Fig. 5 clearly depicts the regions where collisions and non-collisions occur. For example, at a speed of 100 kmh^{-1}, a CTG of 0.6 s is necessary to prevent collisions, and longer CTGs are needed with higher

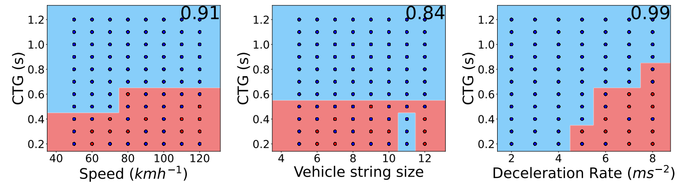

Fig. 5: Decision boundaries for ACC dataset using DT. The top right value indicates the recall of the model.

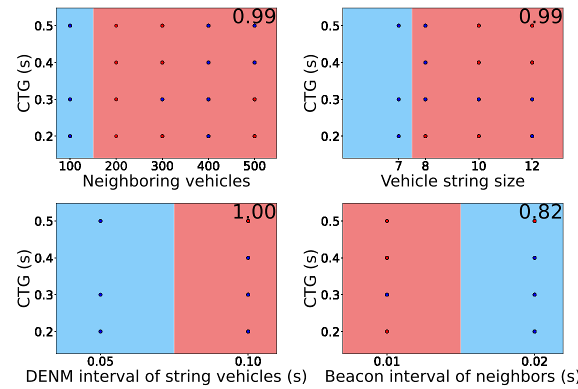

Fig. 6: Decision boundaries for PF-CACC dataset using DT. The top right value indicates the recall of the model.

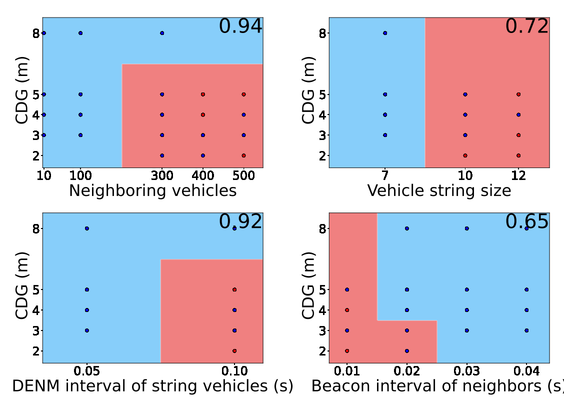

Fig. 7: Decision boundaries for LPF-CACC dataset using DT. The top right value indicates the recall of the model.

deceleration rates for safe emergency braking. Additionally, regardless of the length of the vehicle string, a 0.55 s CTG is necessary when using an ACC controller. Although a single plot from Fig 5 may not depict a complete picture of a safe region for emergency braking, aggregating the decision boundaries from all three plots can yield a more comprehensive and accurate depiction of the safe region. The decision boundaries in Fig. 6 reveal that more neighboring vehicles and higher data density due to shorter beacon intervals result in an unsafe braking region for the vehicle string when using the PF-CACC controller. It is also recommended to use a string length of less than eight because the experienced communication delays amplify while propagating in the downstream direction of the vehicle string. Moreover, Fig. 6 shows that a DENM interval of less than 0.075 s is necessary to ensure safe emergency braking with PF-CACC. Additionally, the decision boundaries shown in Fig. 7 for LPF-CACC indicate a similar impact of neighboring data and road traffic densities as in Fig. 6. An inter-vehicle gap of 6.5 m is required in dense data and road traffic scenarios to remain in the safe braking region when using LPF-CACC. However, longer CDGs may worsen the communication quality between the LV and the rear vehicles due to path loss and fading effects [2]. For the same reason, the decision boundaries reveal that a platoon size should not exceed eight vehicles with LPF-CACC to remain in the safe braking region.

VII. Conclusions and Future Work

This paper proposes a data-driven approach for anticipating collisions among CAVs during emergency braking using machine learning. By simulating a wide range of scenarios and collecting a comprehensive dataset, we have trained several ML models to predict inter-vehicle collisions and evaluated their classification performance. Furthermore, we analyze the predictions of the ML models to identify the parameter values that are most likely to lead to collisions. The analysis demonstrates that ML-based data-driven models have the potential to predict inter-vehicle collisions even under varying wireless channel conditions and communication outage scenarios. Moreover, the evaluation of the ML models indicates that DT and RF, both interpretable classifiers, can predict collisions with high accuracy. Furthermore, understanding the reason behind the predictions provided by these models makes them a practical option for making online predictions in safety-critical systems such as CAVs. The prediction interpretation suggests that when using the ACC controller in a string of vehicles, the inter-vehicle gap and deceleration rate are vital factors in preventing collisions during emergency braking. In contrast, when communication-enabled controllers like PF-CACC or LPF-CACC are used, communication delays induced by data traffic from neighboring vehicles are the most critical factor in emergency braking collisions. Our analysis indicates that shorter platoon or vehicle string sizes and more frequent broadcasting of event-driven messages lead to safer regions for emergency braking. This is because vehicles farther from the leading vehicle in a platoon or vehicle string experience longer communication delays, and higher message frequency may increase the likelihood of receiving a message. Overall, the results presented in this paper demonstrate that ML-based data-driven models can play a crucial role in enhancing the safety of CAVs by predicting collisions and preventing these by taking preemptive measures.

The datasets presented in this paper serve as illustrative examples and can be expanded to cover more complex CAV scenarios. For instance, the study can be extended to explore cases where platoon vehicles operate distributedly, choosing their own controllers based on the experienced communication quality [2]. Moreover, the approach presented in this paper has the potential to be extended beyond collision prediction to also predict and explain the string stability phenomena in automated vehicles, as done by Mongelli *et al.* in [9].

References

[1] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE T-ITS*, vol. 18, no. 5, pp. 1033–1045, 2017.

[2] S. Hasan, S. Girs, and E. Uhlemann, "Characterization of transient communication outages into states to enable autonomous fault tolerance in vehicle platooning," *IEEE OJ-ITS*, vol. 4, pp. 101–129, 2023.

[3] D.-J. Lin, M.-Y. Chen, H.-S. Chiang, and P. K. Sharma, "Intelligent traffic accident prediction model for internet of vehicles with deep learning approach," *IEEE T-ITS*, vol. 23, no. 3, pp. 2340–2349, 2022.

[4] C. Chen, H. Xiang, T. Qiu, C. Wang, Y. Zhou, and V. Chang, "A rear-end collision prediction scheme based on deep learning in the internet of vehicles," *J. Parallel. Distrib. Comput.*, vol. 117, pp. 192–204, 2018.

[5] B. Ribeiro, M. J. Nicolau, and A. Santos, "Using machine learning on v2x communications data for vru collision prediction," *Sensors*, vol. 23, no. 3, 2023.

[6] X. Wang, J. Liu, T. Qiu, C. Mu, C. Chen, and P. Zhou, "A real-time collision prediction mechanism with deep learning for intelligent transportation system," *IEEE Trans. on Veh. Tech.*, vol. 69, no. 9, pp. 9497–9508, 2020.

[7] S. Qian, "Vehicle collision prediction model on the internet of vehicles," in *Proc. WCNA 2022*, pp. 518–530, Springer, Berlin, Germany, 2022.

[8] A. Fermi, M. Mongelli, M. Muselli, and E. Ferrari, "Identification of safety regions in vehicle platooning via machine learning," in *Proc. IEEE WFCS*, pp. 1–4, Imperia, Italy, 2018.

[9] M. Mongelli, E. Ferrari, M. Muselli, and A. Fermi, "Performance validation of vehicle platooning through intelligible analytics," *IET Cyber-Phys. Syst.: Theory Appl.*, vol. 4, no. 2, pp. 120–127, 2019.

[10] S. Hasan, J. Gorospe, S. Girs, A. Alonso Gómez, and E. Uhlemann, "PlatoonSAFE: An Integrated Simulation Tool for Evaluating Platoon Safety," *IEEE OJ-ITS*, 2023 (accepted).

[11] K. Bilstrup, E. Uhlemann, E. Ström, and U. Bilstrup, "On the ability of the 802.11 p mac method and stdma to support real-time vehicle-to-vehicle communication," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, no. 1, p. 902414, 2009.

[12] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. on VT*, vol. 42, no. 4, pp. 657–672, 1993.

[13] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. ITSC'2011*, Washington, DC, USA, Oct 2011, pp 260–265.

[14] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Trans. on Cont. Sys. Tech.*, vol. 8, no. 4, pp. 695–708, 2000.

[15] S. Hasan, A. Balador, S. Girs, and E. Uhlemann, "Towards emergency braking as a fail-safe state in platooning: A simulative approach," in *Proc. IEEE VTC2019-Fall*, pp. 1–5, Honolulu, HI, USA, 2019.

[16] A. Böhm, M. Jonsson, and E. Uhlemann, "Co-existing periodic beaconing and hazard warnings in ieee 802.11p-based platooning applications," in *Proc. ACM VANET '13*, p. 99–102, Taipei, Taiwan, 2013.

[17] S. Wang, Y. Wang, D. Wang, Y. Yin, Y. Wang, and Y. Jin, "An improved random forest-based rule extraction method for breast cancer diagnosis," *Appl. Soft Comput.*, vol. 86, p. 105941, 2020.