Assessing and Explaining Collision Risk in Dynamic Environments for Autonomous Driving Safety

Richa Nahata[1], Daniel Omeiza[2], Rhys Howard[1], and Lars Kunze[1]

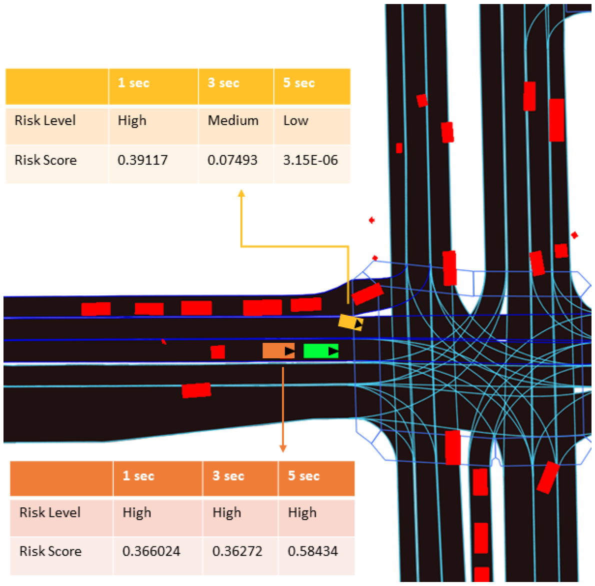
***Abstract*— Autonomous vehicles operating in dynamic environments are required to account for other traffic participants. By interpreting sensor information and assessing the collision risk with vehicles, cyclists, and pedestrians, near-misses and accidents can be prevented. Moreover, by explaining risk factors to developers and engineers the overall safety of autonomous driving can be increased in future deployments.**

In this paper, we have designed, developed, and evaluated an approach for predicting the collision risk with other road users based on a planar 2D collision model. To this end, we have trained interpretable machine learning models to classify and predict the risk of collisions on a range of features extracted from sensor data. Further, we present methods for inferring and explaining the factors mostly contributing to the risk. Using counterfactual inference, our approach allows us to determine the factors which highly influence the risk and should in turn be minimised. Experimental results on real-world driving data show that collision risk can be effectively predicted and explained for different time horizons as well as different types of traffic participants such as cars, cyclists and pedestrians.

I. INTRODUCTION

Highly automated driving (HAD) is considered as the future of intelligent road mobility [1]. A study from the Insurance Institute for Highway Safety states that even though HADs account for some human error, it still cannot prevent 2/3rds of all accidents unless it can account for more complex, prediction-based scenarios [2]. Traffic safety evaluation is one of the most important processes in analyzing transportation system performance. Traditional methods like statistical models and before–after comparisons have many drawbacks, such as limited time periods, sample size problems, and reporting errors [3].

Mahmud et al. [5] summarise recent advancements in metrics applied to quantify risks to driving vehicles. However, most of these metrics are far too simple for real world traffic conditions. They often make assumptions such as the vehicles are on the same lane, ignore overtaking or lane change, assume constant speed or are highly intensive and attainable only in a simulation environment. On the other hand, while there are more generic metrics such as Crash Index [3], Planar TTC and Looming [6], they have not been tested extensively on real world data to determine their usefulness in risk assessment. This paper aims to bridge the gap by applying multifaceted risk quantification techniques to the largest set of prediction data for autonomous vehicles currently available [4]. The methods described in this paper are very versatile and can be applied to any autonomous driving dataset with sufficient information about other road users. The risk is predicted at different time instants in the future to give the vehicle sufficient time to avoid a collision. Most of the past work in this area defines thresholds and formulates classification problems whereas our work extends this to regression models so that we understand the risk of other traffic participants at greater detail. Apart from the quantitative results of the risk assessment provided in this paper, there are also qualitative algorithmic explanations provided to gain an intuitive understanding of the problem to enable developers and engineers to identify important features and increase the situational awareness of the autonomous vehicle.



	1 sec	3 sec	5 sec
Risk Level	High	Medium	Low
Risk Score	0.39117	0.07493	3.15E-06

	1 sec	3 sec	5 sec
Risk Level	High	High	High
Risk Score	0.366024	0.36272	0.58434

Fig. 1: Assessing and explaining collision risk in dynamic environments. An instance of the Lyft Level5 dataset [4] used for training our risk prediction models. The green rectangle represents the ego vehicle, the yellow rectangle is the agent whose risk value is predicted relative to the ego vehicle. The orange rectangle represents another different agent behind the ego vehicle. The tables show the ground truth risk prediction values for 1 second, 3 seconds, and 5 seconds. The black triangle points in the direction the vehicle is moving. In Section V-C, we provide explanations for the risk predicted for the yellow agent.

[1] Richa Nahata, Rhys Howard, and Lars Kunze are with the Oxford Robotics Institute, Dept. of Engineering, Science, University of Oxford. Email: lars@robots.ox.ac.uk

[2] Daniel Omeiza is with the Dept. of Computer Science, University of Oxford. Email: daniel.omeiza@linacre.ox.ac.uk

The paper makes the following contributions:

- a learning-based approach to dynamic risk assessment based on abstracted sensor information and a planar collision risk metric;
- a novel approach for inferring and explaining risk factors in autonomous driving; and
- an experimental evaluation and discussion of quantitative and qualitative results on real-world driving data.

The remainder of the paper is structured as follows. In Section II, we discuss related work on risk assessment and methods for explanation generation. Our learning-based approach to dynamic risk assessment is described in Section III, and in Section IV, we present two methods for generating explanations from learnt risk models. In Section V, we present and discuss quantitative and qualitative results from experiments using real-world driving data. Section VI concludes the paper.

II. RELATED WORK

A. *Risk Assessment*

There are different categories of metrics defined to estimate the risk posed to a vehicle by another vehicle. Li et al. [7] discusses various threat assessment techniques for risk prediction which include time, kinematics, statistics and potential field-based metrics. In this paper, we focus on time-based metrics to estimate risk in an elegant and efficient manner. It also makes it easier to account for different attributes and develop explainable models. Lefèvre et al. [8] also summarises different risk assessment techniques under the brackets of physics-based, maneuver-based, and interaction-aware models. We focus on physics-based models because, as explained in the paper, they allow for efficient computation of risk and short-term collision prediction. Wardziński [9] defines a four level risk scale where the lowest level is 'no risk', the next level is 'acceptable safe' followed by 'hazardous situation' and the highest level is an 'accident'. While the author realised the importance of detecting the threat early on, only the minimum distance between vehicles was used to estimate the risk level. Likewise there are metrics which only work in the one dimensional case of rear-end collisions such as TTC (Time-to-Collision), THW (Time Headway), TTR (Time-to-React), Safe Distance Model for minimum safe distance between human driven and driverless vehicles [10], Unsafe Density [5] and many more. These fail to generalise well to more practical situations and often contain overly simplified assumptions. Most of these metrics also only work on the binary classification scale as they generally have a threshold defined whereas our work extends to regression models. While Vasconcelos et al. [11] does define an accident prediction model for three-leg and four-leg priority intersections, it requires access to reliable accident records based on which a new model is defined for each place. This is not scalable, as even though it might work well for the particular locations mentioned in the paper, it is not easily extendable to the entire world, especially in areas where traffic monitoring is not done as extensively and previous records are hard to acquire. Moreover, a number of these metrics have been tested only in simulation environments and not on real world datasets. This greatly limits their usage as while they do perform well in simulations, real cars, more often than not, do not behave like an ideally simulated car. There are suggestions that the way to get over the errors in real traffic situations is for vehicles to communicate their planned routes to other vehicles and cooperate in route planning [9]. However, this may cause concerns about security and privacy, as users of autonomous vehicles may not be comfortable letting every car on the road to acquire their destination information.

B. *Explanations in Risk Assessment*

Blackbox AI models are being deployed in different domains to predict risk. As the consequences of the outcomes of these models are grave in critical domains, their decision making process needs to be transparent to relevant stakeholders [12]. In response to this, explainable approaches to risk prediction have been adopted in credit lending [13] and healthcare [14]. Despite the increasing attention in explainable autonomous driving, and risk analysis, explainable risk prediction still seems to be under-explored. In a related work, Yu et al. [15] assessed the subjective risk level of different driving maneuvers using a Multi-Relation Graph Convolution Network (MRGCN), a Long Short-Term Memory Network, and attention layers. The use of scene graphs allows for explainable intermediate representation of driving scenes. As a further step, interpretbility and intelligibility need to be considered all through the learning and prediction process in order to enhance transparency and accountability [16]. Hence, we apply interpretable models (tree-based) with high intelligibility (natural language explanation) in risk prediction and classification tasks.

III. DYNAMIC RISK ASSESSMENT

A. *Problem Statement*

The aim of this risk assessment is to quantify the risk posed to the autonomous (ego) vehicle by other road users (agents) present in the environment at any instant. This is used to predict the risk of collision at various time horizons in the future so that the vehicle can be notified of the approaching danger ahead of time (see Figure 1).

B. *Risk Metrics*

1) Planar TTC: TTC at an instant t is defined as 'the time that remains until a collision between two vehicles would have occurred if the collision course and speed difference are maintained' [5]. While TTC is often used for risk calculations, it has many limitations. It can only be used for rear-end collisions and thus fails to model a real world environment in which collision can occur in any direction. Moreover, this definition of TTC implies that only if the speed of the following vehicle is larger than that of the leading vehicle, a collision will occur and ignores any potential conflicts due to acceleration or deceleration changes. Thus, there is a need for a Modified Time to Collision (MTTC) which takes these factors into account to provide a more comprehensive risk assessment. Ward et al. [6] extends TTC to general traffic scenarios by combining the planar TTC with looming. Planar TTC is calculated by assuming constant acceleration in contrast to the constant speed assumption.

Consider d_{ij} to be the distance between the closest points of the ego vehicle (p_i) and the agent (p_j) and $\dot{d_{ij}}$ and $\ddot{d_{ij}}$ to be its first and second derivatives respectively. If

$$T_1 = \frac{-d_{ij}}{\dot{d_{ij}}} \tag{1}$$

is the first order TTC where the closure rate is omitted, and

$$\Delta = \dot{d}_{ij}^2 - 2\ddot{d_{ij}} d_{ij} \tag{2}$$

is the discriminant of the second order case, then the formula to calculate the planar TTC (T_2) is as follows:

$$T_2 = \begin{cases} T_1 & \text{if } \ddot{d_{ij}} = 0 \\ \frac{\dot{d_{ij}}}{\ddot{d_{ij}}} & \text{if } \Delta < 0 \\ \min(\frac{-\dot{d_{ij}} \pm \sqrt{\Delta}}{\ddot{d_{ij}}}) & \text{if } \min(\frac{-\dot{d_{ij}} \pm \sqrt{\Delta}}{\ddot{d_{ij}}}) \geq 0 \\ \max(\frac{-\dot{d_{ij}} \pm \sqrt{\Delta}}{\ddot{d_{ij}}}) & \text{if } \min(\frac{-\dot{d_{ij}} \pm \sqrt{\Delta}}{\ddot{d_{ij}}}) < 0 \end{cases} \tag{3}$$

T_1 and T_2 are capped to 30 seconds as values above these are much larger than the horizon we want to predict the collision in and this might skew the data. When Δ is negative, T_2 is defined as the time of closest approach as there are no real roots. Moreover, negative values of TTC indicate that there is no risk for collision and hence they are all treated the same.

2) Looming: The drawback of MTTC is that it assumes that the vehicles are on the same collision course which might not always be the case. Thus, looming, as introduced in [6], is used to check if the vehicles actually reach the point of intersection at the same time or they simply pass one before another. To calculate looming, seven test points are chosen on the vehicle as shown in Figure 2. The loom points are biased to the front of the vehicle as predicting the likelihood of collision with this part of the vehicle is more useful to the driver than the end of the vehicle. Then the linear velocity of the loom point ($\bar{\mathbf{v}}_\mathbf{i}$) is calculated as follows:

$$\bar{\mathbf{v}}_\mathbf{i} = \mathbf{v_i} + (\mathbf{p_i} - \mathbf{p_c}) \times \omega_\mathbf{i} \tag{4}$$

Where $\mathbf{v_i}$ is the ego vehicle velocity, $\mathbf{p_i} - \mathbf{p_c}$ is the displacement of the loom point ($\mathbf{p_i}$) from the vehicle center of rotation ($\mathbf{p_c}$) and $\omega_\mathbf{i}$ is yaw rate of the vehicle.

The vector sum of vehicle velocity and the linear velocity due to the yaw of the vehicle about its centre gives the linear velocity of the loom point. Thus, the loom rate (angular velocity of the loom point) is calculated as follows:

$$\dot{\theta} = \frac{(\mathbf{p_j} - \mathbf{p_i}) \times \bar{\mathbf{v}}_\mathbf{i} + (\mathbf{p_j} - \mathbf{p_i}) \times \mathbf{v_i}}{\|\mathbf{p_i} - \mathbf{p_j}\|^\mathbf{2}} \tag{5}$$

where $\mathbf{p_j}$ is the vector position of the agent.
This gives rise to fourteen loom rates corresponding to the left loom rates (named alpha1 to alpha7) and the right loom rates (named beta1 to beta7) of the seven loom points. This calculation helps to determine if the vehicles are on a collision course.

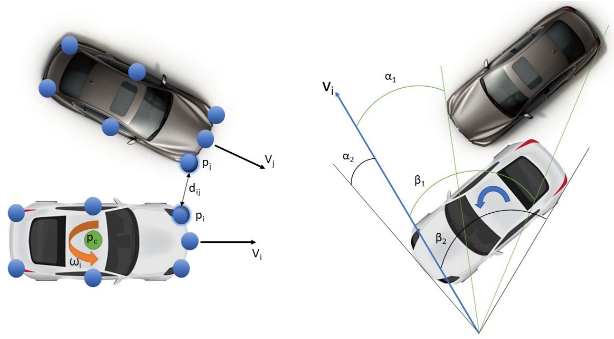


Fig. 2: Some Attributes of the Feature Vector. The blue circles denote the loom points. The right diagram shows the first four angles corresponding to the loom rates. In the right diagram, the rightmost point of the object is moving clockwise relative to the observer and the leftmost point is moving anticlockwise. Thus, the observer's field of vision is filled increasingly by the object and the object is looming.

C. Feature Extraction

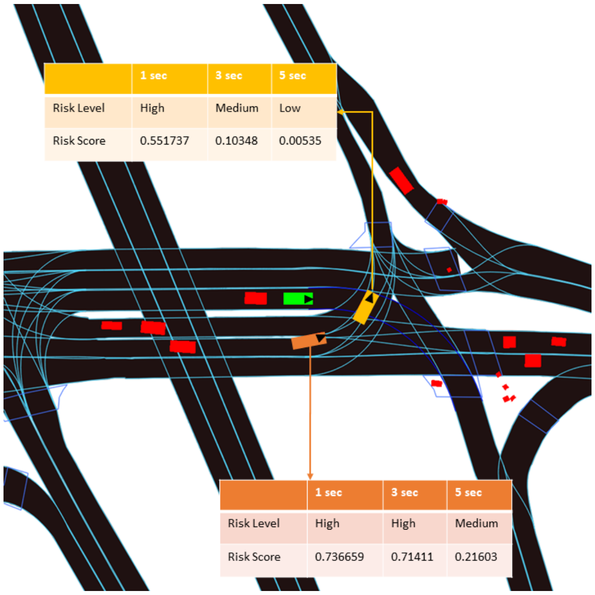
In addition to the metrics discussed above, we extract several other features. The relative distance between the ego vehicle and the agent is calculated by extracting their current positions and taking the L^2 norm of their difference. The agent velocity and the ego velocity are calculated by iterating through the frames and averaging its changing position over time. The acceleration is calculated in a similar way by averaging its instantaneous velocity over time. The relative velocity and the relative acceleration are calculated by taking the difference in each of the two dimensions, followed by its L^2 norm. The angular velocity of the ego is calculated by averaging its changing yaw over time. The relative yaw is the difference between the yaw of the agent and the ego. The target position of the ego vehicle is also included in the feature vector as it gives a sense of the direction the ego vehicle aims to move towards. The type of agent (e.g. car, cycle, or pedestrian) is also included.

D. Feature Vector Generation

Finally, we combine information from Section III-B and Section III-C to generate a feature vector as input for our learning-based dynamic risk assessment. It includes the following information: T_1, T_2, the fourteen loom rates, relative distance, ego and agent velocity, relative velocity, agent and ego acceleration, relative acceleration, angular velocity of the ego, the target destination of the ego in both the x and the y direction, the relative yaw and the type of agent.

E. Ground Truth

To determine the ground truth (labels for our learning task), the actual relative distance between the ego vehicle and the agent vehicle at future time *t* is extracted.



	1 sec	3 sec	5 sec
Risk Level	High	Medium	Low
Risk Score	0.551737	0.10348	0.00535

	1 sec	3 sec	5 sec
Risk Level	High	High	Medium
Risk Score	0.736659	0.71411	0.21603

Fig. 3: An instance of the Lyft Level5 dataset [4] that visualises a potential left-turn conflict. As before, the ego is depicted in green and two other agents in yellow and orange. The yellow agent is in the direct view of the ego vehicle and has a current high risk value because the relative distance between them is small. However, due to looming, we realise that these cars will never actually meet since the yellow agent will pass the point before the ego vehicle arrives there, which is indicated by its decreasing risk. On the other hand, the orange vehicle, poses a very high risk to the ego as it might actually reach the point of collision at the same time as the ego, once the yellow vehicle has passed. Thus, this helps the ego to prioritize between different agents, to focus on the more risky one (here the orange agent) and to maneuver accordingly.

1) Binary Classification: As applied in [6], 10 metres was used as the threshold for classification. If the future relative distance was less than 10 metres, it was classified as high risk (risk flag = 1) and if it was greater than or equal to 10 metres, it was considered as low risk (risk flag = 0). This is intuitive as the vehicles are not likely to collide when the distance between them is more than 10 metres.

2) Regression: While classification helps us distinguish between high and low risk objects it does not tell us how severe a risk is. Hence, we decided to extend this to a regression problem. To come up with ground truth labels (risk scores), we sampled the probability of the actual distance from a one sided positive Gaussian distribution with zero mean and a standard deviation of five. Two standard deviations correspond to 10 meters and, thus, the majority of the points are included.

3) Prediction times: The ground truth labels for regression and classification were generated at 1, 3 and 5 seconds in the future. Reaction times vary greatly from person to person, and even for the same person it changes based on the time of the day, weather condition and the landscape [17]. A professional driver who is physically fit and trained in high-speed driving might have a reaction time of 0.2 seconds for a given situation, while the average driver may have a slower reaction time of 0.5 seconds, 0.8 seconds or even 1 second [18]. However, in cases where the human driver needs to override and take control, we need more time since human drivers in autonomous vehicles tend to be more disengaged in the task and more overconfident in the automation. They can have a weakened understanding of the operation and status of the automatic system, as well as that of the driving situation the car is in. In the long term, they also could lose the skills required to drive and operate the car safely [1]. Thus, even longer time horizons such as 3 and 5 seconds were included as explained in Fig. 3.

F. Learning Approaches

1) Decision Trees: Decision Trees are a powerful non-parametric supervised learning method often used for prediction. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the feature vectors. Decision trees work for both classification and regression as they can handle both continuous and categorical variables. Another major advantage of using decision trees is that they are interpretable. Thus, faithful explanations in natural language can be provided to enhance intelligibility for different stakeholders. To avoid data overfitting, we resample the data many times and split it into training and validation sets. This helps us find the optimal tree depth which gives the best bias-variance trade off for each of our experiments. For this, we use the K-fold cross-validation algorithm, with five folds, based on accuracy, with a hyperparameter grid as the input for the search. This increases the performance of decision trees significantly.

2) Random Forests: Random Forests consist of a large number of decision trees that act as an ensemble. The fundamental concept behind a random forest is a simple one: a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models. Thus, it reduces the overfitting in decision trees and helps improve the accuracy. To improve the performance of random forests, we implement both Random Search Cross Validation and Grid Search with Cross Validation. Both these algorithms help us find the best hyperparameters for Random Forests. Of these, Random Search Cross Validation performs the best and is used to report the results.

In our experiments, we use both decision trees and random forests to improve our accuracy and precision while retaining the interpretability. Thus, the classification and regression models for each of 1, 3 and 5 seconds were trained using both a decision tree and a random forest.

IV. RISK EXPLANATION

In this section, we describe two approaches for explaining our risk prediction models (cf. Section III-F): (i) a tree-based ‘why’ and ‘what-if’ explanation which we have designed, and (ii) the Shapley Additive Explanation (SHAP) method [19] which we used to qualitatively evaluate our tree-based explanation method.

A. Why and What-If Explanation

Let f represent a learnt decision tree model (from our learnt models in Section III-F). Let T represents a tree for f, such that $T = \langle N, E\rangle$. N is a set of nodes and E is a set of edges connecting two nodes. We define a node $n \in N$ (or risk factor) in the tree as a tuple $n = (u, C)$ where: $u \in N$ is a unique numerical identifier for a node in T. C is a list of conditional statements. Each $c \in C$ is constructed by an inequality operator. The root node is the unique node with no parent, and a leaf is a node with no child. The level l_n of a node n is the number of edges from the root to that node.

To construct a *Why* explanation, we traverse T by starting from the root node (say $n_r \in N$) to a leaf node (say $n_l \in N$. We return the set of unique conditions C_w which satisfy the decision path of the input instance. *Why* explanation is then created using the information in C_w. Each $c \in C_w$ is then represented with linguistic terms that describe its meaning in the driving domain. The 'Why' explanation E_w is now a concatenation of the linguistic terms for all the $c \in C_w$.

'What-If' explanations are also referred to as counterfactual explanations. Counterfactual explanations are meant to contain information about the minimum change required in the input in order to obtain the closest alternative outcome (or a target output) from the model. To construct a 'What-If' explanation, we find the nearest sibling $n_l{}'$ to the current leaf node n_l which yields a different outcome/class to n_l. Where such leaf node $n_l{}'$ does not exist, we move a level up the tree and find the descendants of the sub-tree that lead to the desired $n_l{}'$ while avoiding leaf nodes that have been visited. Once $n_l{}'$ is discovered, the lowest common ancestor n_a of n_l and $n_l{}'$ is identified in the tree. n_a is the node from which the path p_w from the root to n_l and the path p_f from the root to $n_l{}'$ first differ. The condition at n_a is negated and added to the set of conditions (say C_f) resulting from $p_f \setminus p_w$ (where '\' represents set complement). Each $c \in C_f$ is then represented with linguistic terms that describe its meaning in the driving domain. The 'What-If' explanation E_f is now a concatenation of the linguistic terms for all the $c \in C_f$.

When the model in consideration (f) is an ensemble of trees (random forest), we perform a tree selection procedure which differ for both regression and classification tasks. An approach to obtain the final prediction (y) in an ensemble tree regressor model is by estimating the mean or the median of $Y : Y = \{y_i | 1 \leq i \leq n\}$ where n is the number of trees in the forest. In our implementation, we find the median of Y and use the tree whose output correspond to this median value to explain the model. Where n is *even*, we use the tree with the closet prediction value to the mean of the prediction $y_{\frac{n}{2}}$ and $y_{\frac{n+2}{2}}$

For an ensemble tree classifier, a sorted list of features is created based on the frequency of the occurrence of features across the forest. From the trees in the forest with same class prediction, we find the tree that has most of the recurring features. If there are more than 1 of such trees, we choose the tree with the highest prediction confidence.

B. Tree SHAP Explanation Method

For a learnt model f, the Kernel SHAP algorithm [19] explains a prediction with respect to a chosen reference or an expected value by estimating the SHAP values of each feature i from $1, ..., M$. The SHAP values are computed as:

- generate all subsets S of the set $F \setminus \{i\}$
- for each $S \subseteq F \setminus \{i\}$ find the contribution of feature i as $CT\{i|S\} = f(S \cup \{i\}) - f(S)$
- compute the SHAP value according to:

$$\phi_i := \frac{1}{M} \sum_{S \subseteq F \setminus \{i\}} \frac{1}{\binom{M-1}{|S|}} CT(i|S) \qquad (6)$$

Tree SHAP explanation algorithm optimises the procedure above to compute exact SHAP values for tree based models in reduced time complexity. These SHAP values show the contributions (both positive and negative) of the features in the model. We show the 10 most contributing features to the models' prediction (see Figure 6). This will provide a basis for us to compare our tree-based method with Tree SHAP. Our tree-based explanation method should generate explanations with reference to some of the features in Figure 6.

V. EXPERIMENTAL RESULTS

A. Lyft Dataset

In order to evaluate the efficacy of the proposed approaches, we utilise the Lyft Level 5 Prediction Dataset [4]. The dataset is primarily composed of a set of scenes collected across 1118 hours of autonomous driving activity from 20 different vehicles. These scenes are accompanied by a manually constructed semantic map detailing the road network, as well as a metadata file. The metadata describes part of the transformation from the semantic map frames based upon a geodetic datum, to the world frame used for the scene data. The scene data is structured as follows: the top level consists of a series of scenes, each scene is $\sim 25\,s$ and is composed of ~ 250 frames sampled at 1 *Hz*. Each frame contains a timestamp, translation and rotation values for the ego vehicle, and the relevant agent objects. Agent objects do not persist between frames, and instead use a tracking id to identify the same entity between frames. For our experiments, the first 300 scenes were used to generate the feature vectors, of which 20% were used for testing and the five-fold method was used for cross validation. See Figure 1.

B. Dynamic Risk Assessment Results

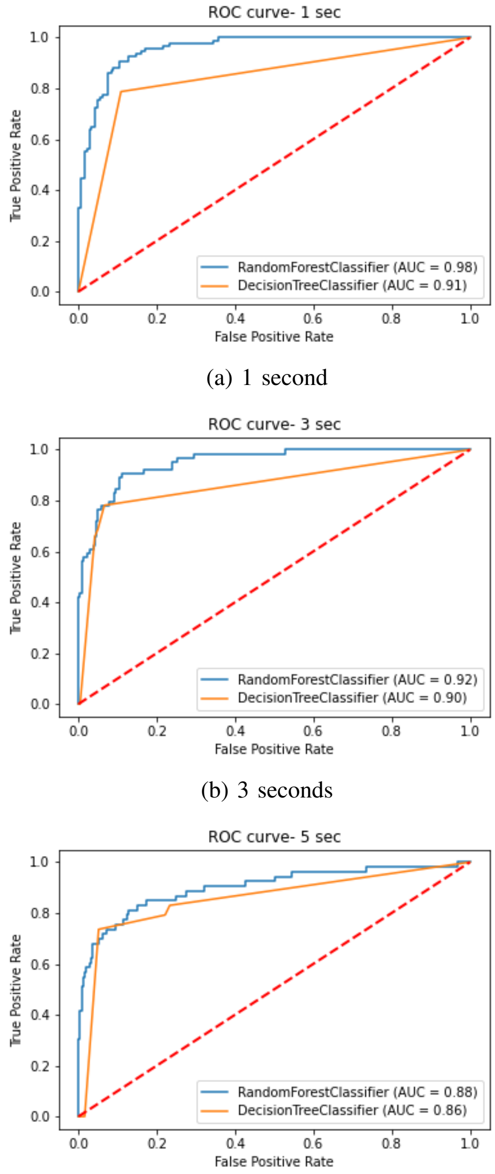
Table I compares different classification models where 'DT' stands for Decision Tree and 'RF' stands for Random Forest. The ROC curve at various instants are show in Fig. 4. Table II compares the results of different regression models. While the difference between the decision tree and random forest models are not very evident in the classification case, they are very distinct in the regression case. This is fairly intuitive as it is much more difficult for the algorithm to predict continuous values than it is to predict a binary value. There is a stark difference between the performance of decision trees in the two cases because they are sensitive to small perturbations in data. This makes it hard to apply it to the regression case where the data has very small changes. Moreover, since they are non-smooth, they are also prone to out-of-sample predictions. Another trend that is very evident is that the performance decreases as we increase the time horizon for our predictions. This is expected as it is difficult to make accurate predictions of a dynamic human-driven vehicle that can change its attributes in less than a second. For example, it is more likely that our assumption of constant acceleration or deceleration may hold true for $1\,s$ rather than for $5\,s$. Table III compares the regression model among different agent types. The 'count' column represents the number of instances of the particular class present in the dataset. The models perform the best on cars and the least on pedestrians. However, this is evidently not proportional to the percentage of data they constitute. This is because it is easier to estimate the velocity and acceleration of vehicles than pedestrians. Moreover, velocity and acceleration of pedestrians are not comparable to those of vehicles and this might induce errors as they'll give a large relative velocity and acceleration. The dimensions of pedestrians are also smaller compared to vehicles so the loom point method might not work accurately even though the Lyft dataset does contain the span of every single agent. The reason for this is that pedestrians are typically harder to observe accurately. This is partially because they are mainly found on pavements where they are more likely to be obscured and partially because their physical shape does not correspond well to a rectangular shaped bounding box i.e. the span. Hence, we will explore other agent-specific metrics in future work. Fig. 5 displays the normalised feature importance score of each feature based on how relevant that particular feature is in helping the model make a prediction. It is evident from the graph that the relative distance and the Planar Time-to-Collision are the two most important parameters for the Random Forest model to make predictions. This is in-line with our expectation from the risk metric.

C. Risk Explanation Results

Our tree-based explanation method can assist developers and engineers to identify the most influential risk factors from learnt risk assessment models. Moreover, through counterfactual inference, our techniques can provide explanations which describe how risk factors need to be 'changed' to decrease the overall risk in safety-critical driving scenarios.

TABLE I: Comparing Different Classification Models

Time	Model	RMS Error	AUC	F_1 Score
1 sec	DT	0.313112	0.91	0.873064
	RF	0.280056	0.98	0.895877
3 sec	DT	0.285831	0.90	0.851776
	RF	0.291492	0.92	0.844609
5 sec	DT	0.313112	0.86	0.784305
	RF	0.291492	0.88	0.816935



(a) 1 second

(b) 3 seconds

(c) 5 seconds

Fig. 4: ROC Curve for Different Temporal Predictions. We can see how performance decreases as we predict for longer time horizons.

We provide a qualitative assessment of our proposed tree-based natural language explanation generation technique. Tree SHAP is known to have higher level of faithfulness in contrast to LIME [19]. Natural language explanations which can be rendered as speech are useful in time critical domains where participants have limited chance to observe a chart or heatmap. Our tree-based method can generate counterfactual explanations to meet certain requirements. For example, the desired counterfactual class can be explicitly set for a classification task. In the case of regression, a counterfactual explanation which contains information on how to obtain a prediction within a certain value range can

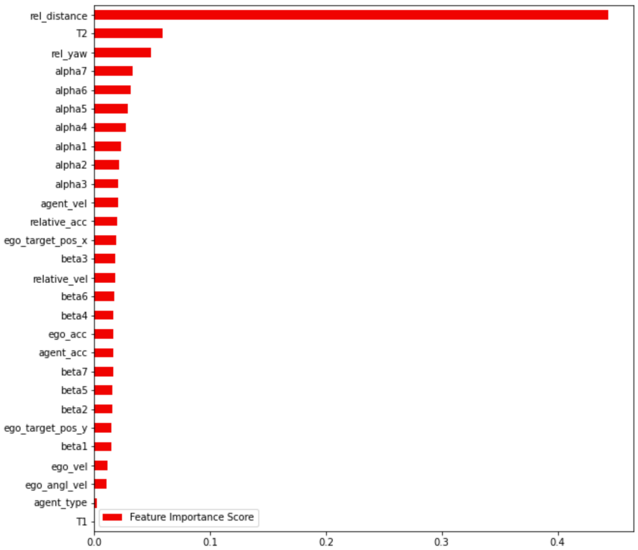
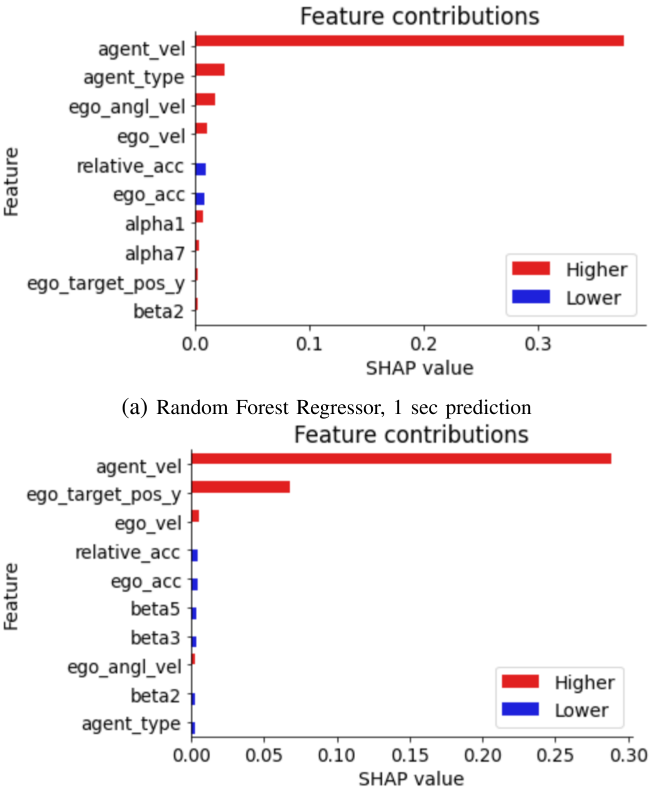


Fig. 5: Feature Importance Score assigns a value to each of the features from the feature vector based on how important that particular feature is in helping the model make a prediction. The average feature scores for the Random Forest classification model across the training set is shown.



(a) Random Forest Regressor, 1 sec prediction

(b) Random Forest Regressor, 5 secs prediction

Fig. 6: Explaining feature contributions for the example scene shown in Figure 1 (yellow agent). We show the 10 most contributing features to prediction based on SHAP values obtained from Tree SHAP algorithm for $1s$ and $5s$ predictions. Both predictions are for feature vector X passed to the RandomForest Regressor models.

TABLE II: Comparing Different Regression Models

Time	Model	RMS Error	EVS	R^2 Score
1 sec	DT	0.089790	0.513486	0.509954
	RF	0.036018	0.921205	0.921146
3 sec	DT	0.006891	0.449383	0.447278
	RF	0.051660	0.786132	0.785953
5 sec	DT	0.091964	0.369128	0.369034
	RF	0.058059	0.7485211	0.748506

TABLE III: Comparing Different Classes

Class	Count	RMS Error	EVS	R^2 Score
Car	519385	0.076093	0.754399	0.752412
Cycle	6688	0.053561	0.864631	0.735844
Pedestrian	43182	0.127931	0.695486	0.638030

be generated.

Explanation 1, 2, and 3 below are sample explanations generated using the tree-based method for the yellow agent in Figure 1. We generated natural language explanations ('why' and 'what-if') for the tree models' predictions. Our explanations made references to some highly contributing features (shown in Figure 6).

Explanation 1: RandomForest Regressor, $1s$
Why: "The predicted risk for the provided agent's attributes is 0.4922 because important features such as 'beta6' has a value between 0.0 $rad\,s^{-1}$ and 16.0179 $rad\,s^{-1}$, 'agent_vel' was below 5.2209 ms^{-1}, 'ego_vel' was below 0.0001 ms^{-1}."
What-If (counterfactual inference): "To get the risk prediction below 0.3, the following conditions should be true: 'alpha6' should be greater than 0.0 $rad\,s^{-1}$, 'agent_vel' should be above 6.794 ms^{-1}."

Explanation 1 was generated for a $1s$ random forest regressor prediction for a particular feature vector (say X). The counterfactual explanation is also generated for risk value lower than 0.3. When 'agent_vel' was set to 7 ms^{-1}, a risk value of 0.2614 was obtained. Increasing the 'agent_vel' makes the agent move farther ahead of the ego vehicle, thereby reducing collision risk.

Explanation 2: RandomForest Regressor, $5s$
Why: "The predicted risk for the provided agent's attributes is 0.3853 because important feature such as 'beta2' was above -1.105e-05 $rad\,s^{-1}$, 'agent_vel' was below 5.1108 ms^{-1}, 'ego_target_pos_y' was below 0.6182 m."
What-if (counterfactual inference): "To get the risk prediction below 0.3, the following conditions should be true: 'ego_target_pos_y' should be greater than 0.6182 m."

Explanation 2 was generated for a $5s$ random forest regressor prediction for feature vector X. The counterfactual explanation was generated for risk value lower than 0.3. When 'ego_target_pos_y' was set to 3, a risk value of 0.2754 was obtained. When the ego vehicle's target y is increased, the ego vehicle's destination is further south (where $(0,0)$ is the topmost left corner) which makes its trajectory farther apart from the agent when the agent is heading East.

> **Explanation 3:** RandomForest Classifier, $1s$
> *Why:* "The provided agent was classified as 'high risk' because important feature such as 'alpha1' was below $1.6972\,rad\,s^{-1}$, 'alpha5' has a value between -180.2083$\,rad\,s^{-1}$ and $0.0\,rad\,s^{-1}$, 'alpha7' was above -0.00046231$\,rad\,s^{-1}$, 'beta1' was above -2.9e-07$\,rad\,s^{-1}$, 'agent_vel' was below $23.5176\,ms^{-1}$, 'rel_yaw' was above -0.4258$\,rad$."
> *What-if (counterfactual inference):* "The closest class to the prediction is 'low risk'. To classify this sample as low risk the following conditions should hold: 'agent_vel' should be greater than $23.5176\,ms^{-1}$."

Explanation 3 was generated for a $1s$ random forest classifier's prediction for feature vector X. An explanation on how to obtain a counterfactual output (low risk) was also provided.

VI. CONCLUSION

With growing interest in the autonomous driving market, it is essential to account for its safety. In this paper, we utilised sensor data from autonomous vehicles to provide a comprehensive risk assessment that accounts for all kinds of conditions that may result in a collision. We provide an approach to infer risk from machine learning models (decision trees and random forests) trained on relevant features extracted from the Lyft dataset with predictions made at different future time horizons. We developed a tree-based explanation technique to explain the models' risk predictions. This work can be directly incorporated into safety systems in autonomous vehicles. It also serves as the basis for path prediction algorithms for high risk road users which can then assist the ego vehicle to perform maneuvers accordingly. The models obtained are high performing (with R^2 scores of 92.11% for the 1 second regression case) and are transferable and explainable on other datasets (e.g. KITTI dataset [20], Waymo dataset [21]). The explanation technique will enable developers in this field to identify the important attributes and how they can select these attributes to yield lower risks. Moreover, the explanations provide assistance for model debugging especially where safety is critical. While this paper has been written in the context of autonomous driving, the methods described could be extended to the risk assessment of any kind of autonomous robot by the dynamic entities in its environment.

ACKNOWLEDGMENTS

This work was supported by the Assuring Autonomy International Programme (Demonstrator project: Sense-Assess-eXplain (SAX)), a partnership between Lloyd's Register Foundation and the University of York. It was also supported by RoboTIPS, grant reference: EP/S005099/1.

REFERENCES

[1] Sébastien Demmel, Dominique Gruyer, Jean-Marie Burkhardt, Sébastien Glaser, Grégoire Larue, Olivier Orfila, and Andry Rakotonirainy. Global risk assessment in an autonomous driving context: Impact on both the car and the driver. *IFAC-PapersOnLine*, 51(34):390–395, 2019. 2nd IFAC Conference on Cyber-Physical and Human Systems CPHS 2018.

[2] Ben Gilbert. Self-driving cars still won't prevent the most common car accidents, according to a new study, 2020.

[3] Kaan Ozbay, Hong Yang, Bekir Bartin, and Sandeep Mudigonda. Derivation and validation of new simulation-based surrogate safety measure. *Transportation Research Record*, 2083(1):105–113, 2008.

[4] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. *arXiv preprint arXiv:2006.14480*, 2020.

[5] S.M. Sohel Mahmud, Luis Ferreira, Md. Shamsul Hoque, and Ahmad Tavassoli. Application of proximal surrogate indicators for safety evaluation: A review of recent developments and research needs. *IATSS Research*, 41(4):153–163, 2017.

[6] James R. Ward, Gabriel Agamennoni, Stewart Worrall, Asher Bender, and Eduardo Nebot. Extending time to collision for probabilistic reasoning in general traffic scenarios. *Transportation Research Part C: Emerging Technologies*, 51:66–82, 2015.

[7] Yang Li, Yang Zheng, Bernhard Morys, Shuyue Pan, Jianqiang Wang, and Keqiang Li. Threat assessment techniques in intelligent vehicles: A comparative survey. *IEEE Intelligent Transportation Systems Magazine*, PP:1–1, 2020.

[8] Stephanie Lefevre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *Robomech Journal*, 1, 2014.

[9] Andrzej Wardziński. Dynamic risk assessment in autonomous vehicles motion planning. pages 1 – 4, 2008.

[10] Tesfaye Hailemariam Yimer, Chao Wen, Xiaozhuo Yu, and Chaozhe Jiang. A study of the minimum safe distance between human driven and driverless cars using safe distance model, 2020.

[11] António Vasconcelos, Álvaro Seco, Ana Silva, and Luis Neto. Validation of the surrogate safety assessment model for assessment of intersection safety. *Transportation Research Record Journal of the Transportation Research Board*, 2432:1–9, 2014.

[12] Daniel Omeiza, Helena Webb, Marina Jirotka, and Lars Kunze. Explanations in autonomous driving: A survey. *arXiv preprint arXiv:2103.05154*, 2021.

[13] Ceena Modarres, Mark Ibrahim, Melissa Louie, and John Paisley. Towards explainable deep learning for credit lending: A case study. *arXiv preprint arXiv:1811.06471*, 2018.

[14] Nicasia Beebe-Wang, Alex Okeson, Tim Althoff, and Su-In Lee. Efficient and explainable risk assessments for imminent dementia in an aging cohort study. *IEEE Journal of Biomedical and Health Informatics*, 2021.

[15] Shih-Yuan Yu, Arnav V Malawade, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad A Al Faruque. Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions. *arXiv preprint arXiv:2009.06435*, 2020.

[16] M. Gadd, D. de Martini, L. Marchegiani, P. Newman, and L. Kunze. Sense–Assess–eXplain (SAX): Building trust in autonomous vehicles in challenging real-world driving scenarios. In *2020 IEEE Intelligent Vehicles Symposium (IV), Workshop on Ensuring and Validating Safety for Automated Vehicles (EVSAV)*, pages 150–155, 2020.

[17] Andrea Palazzi, Davide Abati, Simone Calderara, Francesco Solera, and Rita Cucchiara. Predicting the driver's focus of attention: the dr(eye)ve project, 2018.

[18] Managing a slow reaction time while driving, Jul 2019.

[19] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*, 2017.

[20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[21] Waymo open dataset: An autonomous driving dataset, 2019.