Article

Scenario Optimisation and Sensitivity Analysis for Safe Automated Driving Using Gaussian Processes

Felix Batsch [1,2,*] , Alireza Daneshkhah [3,*] , Vasile Palade [3] and Madeline Cheah [2]

1   Institute for Future Transport and Cities, Coventry University, Coventry CV1 5FB, UK
2   HORIBA MIRA Ltd., Nuneaton CV10 0TU, UK; madeline.cheah@horiba-mira.com
3   Research Centre for Data Science, Coventry University, Coventry CV1 5FB, UK; ab5839@coventry.ac.uk
*   Correspondence: batschf@coventry.ac.uk (F.B.); ac5916@coventry.ac.uk (A.D.)

Abstract: Assuring the safety of automated vehicles is essential for their timely introduction and acceptance by policymakers and the public. To assess their safe design and robust decision making in response to all possible scenarios, new methods that use a scenario-based testing approach are needed, as testing on public roads in normal traffic would require driving millions of kilometres. We make use of the scenario-based testing approach and propose a method to model simulated scenarios using Gaussian Process based models to predict untested scenario outcomes. This enables us to efficiently determine the performance boundary, where the safe and unsafe scenarios can be evidently distinguished from each other. We present an iterative method that optimises the parameter space of a logical scenario towards the most critical scenarios on this performance boundary. Additionally, we conduct a novel probabilistic sensitivity analysis by efficiently computing several variance-based sensitivity indices using the Gaussian Process models and evaluate the relative importance of the scenario input parameters on the scenario outcome. We critically evaluate and investigate the usefulness of the proposed Gaussian Process based approach as a very efficient surrogate model, which can model the logical scenarios effectively in the presence of uncertainty. The proposed approach is applied on an exemplary logical scenario and shows viability in finding concrete critical scenarios. The reported results, derived from the proposed approach, could pave the way to more efficient testing of automated vehicles and instruct further physical tests on the determined critical scenarios.

1. Introduction

With the help of technological advances in vehicle technology, fatal accidents have been steadily declining over the past decades [1]. The rate at which fatal accidents are declining has slowed down, however, suggesting that further improvements can only be achieved by significantly reducing the proportion of accidents which are caused by human error. The latest developments in advanced driver assistance systems and proof-of-concept prototypes of automated vehicles (AVs) are very promising to overcome this challenge [2]. The introduction of higher levels of automation (SAE levels 3+ [3]) into wider public use is currently hindered by the satisfactory assurance of their safety in every possible situation. For instance, to formulate a statistical safety case which supports the superiority of AVs over human drivers, an estimated 440 million km of on-road driving with AVs is necessary [4].

Current research thus focuses on reducing the necessary testing with physical hardware by utilizing virtual testing in simulation. A common approach focuses on the circumstance that most day-to-day driving is uneventful and thus tries to prioritise dangerous and safety-critical scenarios which happen very rarely [5]. Most methods are therefore

aimed at finding these critical scenarios by pursuing a scenario-based testing approach, where AVs are evaluated with the help of vehicle simulation [6]. When implemented in simulation, scenario-based testing allows for an automatic generation of interesting test cases, which can then be further evaluated in tests involving physical hardware.

A terminology for describing scenarios in the different phases of development, test and validation of AVs is presented in Reference [7]. Scenarios can be abstracted into three levels, depending on where they are used in the development process. The highest abstraction is the functional scenario, which describes the involved entities and behaviours semantically, mostly in natural language. A logical scenario, as the middle level of abstraction, specifies the parameters which were described linguistically in the functional scenario. The logical scenario also specifies value ranges for each parameter in the scenarios state space, thus giving a formal description of the scenario. On the lowest level of abstraction, a concrete scenario specifies concrete values for each parameter from the respective value ranges. We make use of this terminology, which is visualised in Figure 1.
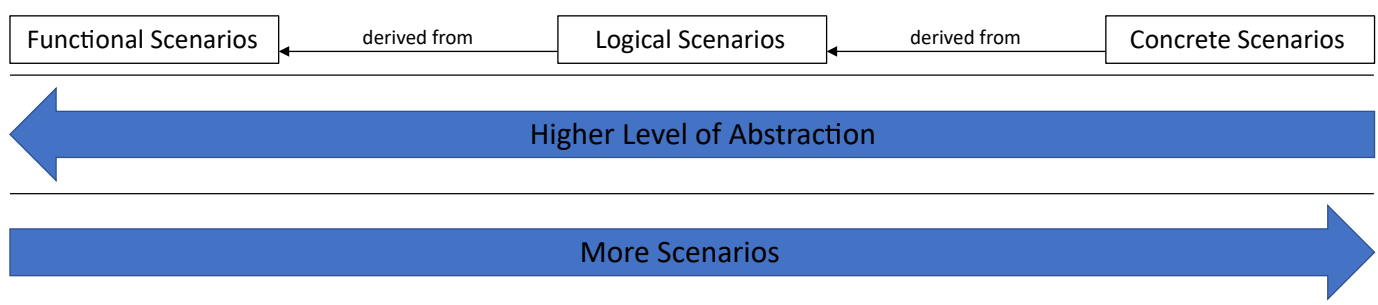

Figure 1. Terminology of the three scenario types (after Reference [7]).

A challenging problem in scenario-based testing is the combinatorial explosion of possible concrete scenarios. Defining a concrete scenario to include the interactions of the AV with its surroundings, especially with other traffic participants leads to an indefinite amount of concrete scenarios. There is a need for a novel method that efficiently finds the parameter combinations that lead to critical concrete scenarios. In our previous work [8], we have shown how stochastic machine learning can be used to estimate the performance boundary of an AV in a given logical scenario, by simulating a limited number of concrete scenarios. The performance boundary separates the scenario space into regions according to their criticality.

Contributions

We build on our previous work [8] to further reduce the number of necessary simulations for concrete scenarios. Starting from a small set of concrete scenarios, which have been executed in simulation to find their respective criticality measure, a novel algorithm is developed to propose and select candidates of concrete scenarios which are predicted to be most critical. This approach iteratively explores the parameter space of a predefined logical scenario and thus reduces the necessary simulation effort, while at the same time holistically covering the parameter space of the logical scenario. We apply this optimisation method on a complex interaction scenario between an AV and a pedestrian stepping onto the road.

Furthermore, we use the Gaussian Process (GP) models to compute several variance-based sensitivity analysis (SA) measures for the previously optimised scenario data set of a pedestrian stepping onto the road. The results derived from conducting the SA, including the main and interaction effects between the input parameters of the logical scenarios, enable us to effectively determine the input parameters with the highest impact on the scenario outcome. SA can thus be used to reduce the complexity of a logical scenario into a computationally tractable problem and to our knowledge, such a sensitivity based assessment of scenarios using GP emulators has so far not been conducted.

The remainder of this paper is organised as follows. In Section 2, we review relevant literature in the field of scenario-based testing and supervised machine learning with multivariate GPs. Section 3 formally introduces GP and what modifications are necessary for them to be used in a classification setting. Further, we introduce the sensitivity analysis used for assessing the variance-based SA indices, including main and interaction effects; and Sobol indices in Section 4. In Section 5, we present the optimisation method, including the iterative algorithm to find the most critical scenarios. Section 6 provides the results of applying the optimisation method to the exemplary scenario of a pedestrian stepping onto the road. It further presents the results of the sensitivity analysis on the optimised data set constructed with the optimisation method, and a second functional scenario of a traffic jam approach, taken from literature. Finally, we conclude with Section 7 and point to open areas for future work.

2. Related Work

2.1. Scenario-Based Validation of AVs

Scenario-based testing has emerged as an active field of research, pursuing the aim of validating the safety of AVs. It is inherently connected to the definition of scenarios in automated driving research ([7,9]), and examines the interactions and relations of the AV (also called ego vehicle) with one or several other actors in defined environmental conditions [10]. By testing the AV in concrete scenarios, the decision making of the AV can be validated based on the outcome of the concrete scenario. It is here assumed that the decision making of the AV changes, depending on the specified behaviour of other involved actors and the various environmental conditions, similar to human decision making while driving.

The execution of such scenario-based tests is predominantly done with vehicle simulation software, which combines the dynamics of the ego vehicle with a virtual representation of the world and other traffic participants, as well as sensor emulation. Most of the subsequently presented methods for scenario-based testing utilise simulation. Scenario-based testing of AVs is however not limited to simulation and can also be done with Hardware-in-the-Loop or field-operational tests on the road or proving ground [6]. Using simulation in executing the tests offers however benefits in terms of cost and execution time, as large numbers of concrete scenarios can easily be carried out.

A scenario-based validation process should cover all functional scenarios defined in the concept phase of the development process [7]. After these functional scenarios are defined, the respective logical scenarios can be constructed, by specifying the parameters and value ranges. To actually carry out a scenario, a concrete scenario needs to be drawn from the logical scenario space by setting the values of each parameter. The simplest approach is to sample values from specified probability distributions of each parameter, such as Monte Carlo (MC) sampling. The values of the concrete scenarios in Reference [11] are, for example, drawn from a uniform distribution. The MC simulation approach would however only yield accurate outcomes, if a high number of concrete scenarios were carried out, making it computationally expensive. But when a sufficient number of concrete scenario are carried out, this approach covers the whole parameter space of a logical scenario, and is thus often used as a baseline for comparison to other methods [12].

The MC sampling based approach can be improved, if the probability distributions used for sampling concrete values are changed from uniform distributions to distributions which are closer to the true distributions of the parameters characterising the functional scenarios. Zhang et al. [5] proposed a related approach to improve the computational efficiency of the MC method by starting with MC sampling, but benefiting from the important sampling concept to sample from a tractable distribution, which can be considered a close distribution to the true one.

The proposed method achieved much higher computational efficiency in the magnitude of hundreds to thousands of times greater compared to the original MC method. This results in more scenarios being evaluated that can be classed as rare events and might

be problematic for AVs [5]. Importance sampling methods are especially suited here and have been used to evaluate adaptive cruise control systems [13]. Further, they were found to improve the evaluation time of testing a car following scenario by a factor of up to 100,000 and a lane change scenario by up to 20,000 [14,15].

An improvement over MC simulation was also shown by using subset simulation on the test case of a lane change scenario [5]. It was shown to offer a similar improvement as importance sampling by focusing on a subset of scenarios that have a higher probability of failure.

Variance reduction techniques, such as importance sampling and subset simulation, rely heavily on the availability and validity of prior knowledge to shape the probability distribution used for sampling. Such data can be sourced from accident databases or large scale naturalistic driving trials [14], but this carries the risk of excluding scenarios that emerge from new automation technology. This new technology is often not included in historic accident databases and naturalistic driving trials and thus questions the validity of the used prior probability distributions.

A relatively unexplored field concerns the selection of the correct parameters to describe a logical scenario (and by extension, the concrete scenarios derived from it). During the process of defining logical scenarios from the functional scenarios, parameters that formally describe the functional scenario need to be chosen. This is frequently done based on expert knowledge on which parameters might have a higher impact on the scenario outcome.

Furthermore, determining the effect of the scenario parameters on the outcome is important for understanding the behaviour of AVs, and can be done via sensitivity analysis of scenarios. In the context of testing AVs, SA are commonly applied to find sensitivities of model parameters in AV sub-modules. It has been applied to assess which parameters of a radar simulation have the greatest impact on the performance of a spatial clustering algorithm which is applied to the simulated output of the radar models [16]. Vehicle dynamics models have also been analysed towards their sensitivities, as the quality of the used vehicle dynamics models has an effect on AV simulations [17]. In order to find the optimal calibration of the trajectory planning module of an AV for several different functional scenarios, sensitivity analyses can be used to reduce the complexity of a genetic algorithm based optimisation as shown in Reference [18]. The sensitivity analysis is herein used to restrict the search space of possible calibration parameters by excluding parameters with a sensitivity below a certain threshold from further simulation. The presented approach includes the simulation of the AV on a set of functional scenarios, it is however not specified how these scenarios are defined, nor does the sensitivity analysis and optimisation include a variation of these functional scenarios.

2.2. Supervised Machine Learning Algorithms

Supervised machine learning offers powerful tools for building input-output relationships of observed systems and have shown their efficacy in a range of problems, such as image classification and control [19,20].

The most prominent supervised machine learning methods are Artificial Neural Networks, which are very powerful for building classifications models of high dimensional data, and solving problems such as object classification in images. These advances were made possible by the wide availability of large amounts of annotated data, reducing overfitting [19]. This makes them however unreliable in the application at hand, where preexisting data is sparse or expensive to obtain due to computationally expensive simulations or costly tests involving physical hardware.

A supervised machine learning method that can build surrogate models based on small data sets is the Support Vector Machine (SVM) [21,22]. They were shown to be effective in scenario prediction of human driving as shown in Reference [23]. The disadvantage of SVMs compared to GPs is that they do not provide an inherent estimate of the confidence

of a prediction. Furthermore, GPs can offer better adjustment in nonlinear modelling by allowing custom kernel functions [24].

The k-Nearest Neighbour (k-NN) algorithm can also be used to model a relationship between input and observed output of an unknown function [25]. The method predicts new data points by relating known data in the vicinity of the predicted point. As the predicted point moves closer to the decision boundary between classes, the reliability of the method declines, however. This makes it less suited for the evaluation of automated vehicles, where it is critical to find corner case scenarios on the boundary between two performance modes.

3. Gaussian Processes

Gaussian Processes are a class of supervised machine learning algorithms, which describe the functional relation between input and output data as a multivariate Gaussian distribution. They are a powerful nonlinear regression and classification method [24]. An application of GPs is the modelling and prediction of trajectories, such as vehicle and pedestrian trajectories [26,27]. They have also been used to model the driving intention of human drivers on intersection scenarios [28,29].

Using GPs to build a surrogate model of a scenario-based testing framework for AVs has been relatively unexplored. A comparable study for autonomous, unmanned underwater vehicles was conducted by Reference [30], where the state space of the vehicle was adaptively searched using a Gaussian Process Regression (GPR) model. In this paper, we adopt the notion of the performance boundary, which separates the different modes of the system under test, and adapts the concept to automated ground vehicles.

GPR was used in an automotive context in Reference [31] to model the probability distribution of a logical scenario, which was then used to conduct importance sampling. The logical scenario considered was a lane change scenario, where they could show an improvement over crude MC sampling.

In our application, we use Gaussian Process Classification (GPC) and Regression (GPR) to optimise the parameter space of a given logical scenario. Once trained, these GP models allow for a quick prediction of untested concrete scenarios. We apply the optimisation method on a complex logical scenario, which is further described in Section 6.1. The scenario data is obtained by evaluating concrete scenarios in the vehicle simulation software CarMaker [32], and for the purpose of this study, the simulation is regarded as ground truth. For this to be valid in the overall scope of AV validation, the simulation must be validated by physical tests.

3.1. Formal Description of Gaussian Processes

We lean on the extensive work of Reference [24], who formally defined the GP model by a prior distribution

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')),\tag{1}$$

where $m(\mathbf{x})$ denotes the distribution mean and $k(\mathbf{x}, \mathbf{x}')$ the kernel function. We used the radial basis function (RBF), also known as squared exponential kernel in this paper [24].

Given a data set $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, \ldots, n\}$, consisting of n samples, wherein $\mathbf{x}_i$ denotes the vector of input data taken from the input space $\mathcal{X}$, and $y_i = f(\mathbf{x}_i)$ the corresponding output observations from the simulation. With the definition of a Gaussian Process from Equation (1), a joint prior distribution for observed outputs f and predicted outputs f_* is described as:

$$\begin{bmatrix} f \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K & K_*^T \\ K_* & K_{**} \end{bmatrix}\right),\tag{2}$$

with an assumed mean of zero and the covariance matrices

$$K = k(\boldsymbol{X}, \boldsymbol{X}) \tag{3}$$

$$K_*^T = k(\boldsymbol{X}, \boldsymbol{X}_*)$$

$$K_* = k(\boldsymbol{X}_*, \boldsymbol{X})$$

$$K_{**} = k(\boldsymbol{X}_*, \boldsymbol{X}_*)$$

for all observed and predicted data points. Here $\boldsymbol{X}$ denotes a $d \times n$ matrix of the training inputs $\{x_i\}_{i=1}^n$ (also known as the design matrix, c.f. Section 5.1), d stands for the dimension of input space $\mathcal{X}$, and $\boldsymbol{X}_*$ is the matrix of test inputs. The subscript $*$ differentiates the predicted data from the actually observed data. Without loss of generality, the mean function is usually assumed to be zero, which does not limit the mean of the posterior to zero [24].

The posterior distribution of $f(\cdot)$ given $\mathbf{y} = \{f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)\}$ is a Gaussian distribution given by

$$f(\mathbf{x})|\mathbf{y} \sim \mathcal{N}(m_*(\mathbf{x}), \sigma^2 k_*(\mathbf{x}, \mathbf{x}')) \tag{4}$$

where

$$m_*(\mathbf{x}) = t(\mathbf{x})^T K^{-1} \mathbf{y} \tag{5}$$

$$k_*(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{t}(\mathbf{x})^T K^{-1} \mathbf{t}(\mathbf{x}') \tag{6}$$

$$\mathbf{t}(\mathbf{x})^T = (k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n)), \tag{7}$$

σ^2 designates the variance of the input data, and $k(\mathbf{x}, \mathbf{x}')$ is the kernel function which both will be introduced in Section 3.2.

The posterior distribution of the predicted value is obtained by conditioning the joint prior distribution to

$$\boldsymbol{f}_* | \boldsymbol{f}, X, X_* \sim \mathcal{N}(K_* K^{-1} \boldsymbol{f}, K_{**} - K_* K^{-1} K_*^T). \tag{8}$$

The predicted value $\boldsymbol{f}_*$ is given by a Gaussian distribution with mean and covariance and can be evaluated over $\boldsymbol{X}_*$. The inherent property of the GPR of providing a predictive value along with its variance is later used in the optimisation method described in Section 5.

3.2. Learning the Hyperparameters

The RBF kernel function $k(\mathbf{x}, \mathbf{x}')$ is used in this paper:

$$k(\mathbf{x}, \mathbf{x}') = \exp\{-(\mathbf{x} - \mathbf{x}')^T \mathbf{B}(\mathbf{x} - \mathbf{x}')\}, \tag{9}$$

where $\mathbf{B} = \{(\sqrt{2}b_i)^{-2}\}_{i=1}^d$, a diagonal matrix of positive smoothness parameters, and d is the dimension of $\mathbf{x}$. These parameters are represented in a vector $\boldsymbol{\theta} = (b_1, \dots, b_d, \sigma^2)$, and are called the hyperparameters. The hyperparameters b_i's have the effect of re-scaling the distance between two inputs $\mathbf{x}$ and $\mathbf{x}'$. Thus, they determine how close two inputs $\mathbf{x}$ and $\mathbf{x}'$ need to be such that the correlation between $f(\mathbf{x})$ and $f(\mathbf{x}')$ takes a particular value.

A stationary and isotropic RBF was used in this GP, which is invariant to both translation and rotation of the input data set. The hyperparameters can be optimised in order to find the best fit of the GP, given the training data $\mathcal{D}$. The optimal hyperparameters are found by maximising the log marginal likelihood

$$\boldsymbol{\theta}_{max} = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta}), \tag{10}$$

where the log marginal likelihood is given by

$$\log p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{f}^T K^{-1} \boldsymbol{f} - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi. \tag{11}$$

3.3. Gaussian Process Classification

In a classification setting, our data set $\mathcal{D}_c = \{(x_i, y_i)\}$ consists of input features x_i and associated discrete class labels y_i. A GPC can be fit on this data set and give prediction in form of binary class probabilities y_*. To that end the output of a regression model is squashed through a logistic function (e.g., sigmoid function, $\sigma(\cdot)$). This transforms the output from a domain of $(-\infty, \infty)$ to $[0, 1]$.

Firstly, a latent variable f_* is predicted corresponding to the input x_*

$$p(f_*|X, y, x_*) = \int p(f_*|X, x_*, \mathbf{f}) p(\mathbf{f}|X, y) \, d\mathbf{f}. \tag{12}$$

and the class prediction can then be calculated using

$$p(y_*|X, y, x_*) = \int \sigma(f_*) p(f_*|X, y, x_*) \, df_*. \tag{13}$$

Equation (12) is a non-gaussian likelihood function, due to the discrete class labels in y. The integral can be computationally approximated using a Laplace approximation.

We use a GPC in the optimisation method in Section 5 to differentiate between scenarios with valid and invalid time-to-collision label.

4. Sensitivity Analysis

In this section, the global sensitivity analysis (SA) of the model output is briefly described. This allows for an evaluation of the relative importance of inputs when they are varied generously, that is, when their uncertainty is acknowledged over a wide range. The most common approach to global SA is the analysis of variance of the model response originally proposed by Reference [33]. This approach, known also as Variance-based, can capture the fraction of the model's response variance explained by a models input on its own or by a group of model inputs. In addition, this approach can also provide the total contribution to the output variance of a given input, that is its marginal contribution and its cooperative contribution. There are several methods, as discussed in References [33–35], to compute this SA approach. In this paper, the GP-based SA method [36,37], which is computationally more efficient, will be adopted and used to compute the sensitivity measures.

In order to carry out the SA using the methods proposed in References [36,37], we need to examine how a function of interest, $f(\mathbf{x})$, depends on its input variables. For the case of this study, f will typically be the function that computes, for example, collision as a function of a vector of input parameters (speed of ego car, speed of pedestrian stepping onto street, distance between pedestrian and ego car at the time the pedestrian steps out, sensor range, and horizontal field-of-view of sensor) as described and illustrated in Section 6.1.

We first need to introduce some important notations. We denote a d-dimensional random vector as $\mathbf{X} = (X_1, \ldots, X_d)$, where X_i is the i_{th} element of $\mathbf{X}$, the sub-vector (X_i, X_j) is shown by $\mathbf{X}_{i,j}$. In general, if p is a set of indices, then $\mathbf{X}_p$ can be written for the sub-vector of $\mathbf{X}$ whose elements have those indices. $\mathbf{X}_{-i}$ is defined as the sub-vector of $\mathbf{X}$ containing all elements except X_i. Similarly, $\mathbf{x} = (x_1, \ldots, x_d)$ denotes the corresponding observed random vector $\mathbf{X}$. In this study, $\mathbf{X}$ is considered as an input vector consisting of all input parameters.

By considering the sensitivity of the input parameters in $\mathbf{X}$, we can determine which input parameters are most influential in inducing uncertainty in f. The main effects as the function over the input range, which are introduced in Appendix A, provide a visual tool to investigate how the logical scenarios respond to variations in each individual input parameter. In the next section, the variance-based SA indices, including first-order and total effects indices will be introduced.

4.1. Variance-Based Methods

The variance-based methods measure the sensitivity of the output $Y = f(\mathbf{X})$, the outcome of the scenario, to the changes in the model inputs in terms of a reduction in the variance of Y. A review of the variance-based approach can be found in Reference [34]. Two principal measures of the sensitivity of the model output, Y to an individual input

X_i are proposed. The first measure which is given in Equation (14), can be considered as the expected amount by which the uncertainty in Y will be reduced if we learn the true value of X_i. This can be also viewed as the variance of the main effect of X_i (as given in Equation (A2)), $V_i = var(z_i(X_i))$, or simply represented as,

$$V_i = var\{E(Y \mid X_i)\}. \tag{14}$$

The second measure, proposed by Reference [38], can be written as:

$$V_{T_i} = var(Y) - var\{E(Y \mid \mathbf{X}_{-i})\}, \tag{15}$$

which is the remaining uncertainty in Y that is unexplained after everything has been learnt except X_i.

These two measures (Equations (14) and (15)) can be converted into scale invariant measures by normalizing with $var(Y)$ as follows:

$$S_i = \frac{V_i}{var(Y)}, \quad S_{T_i} = \frac{V_{T_i}}{var(Y)} = 1 - S_{-i} \tag{16}$$

where S_i is the 1*st* order index for X_i, and S_{T_i} is the *total effect index* of X_i. The 1st order indices measure the portion of variability that is due to variation in the main effects for each input variable, while the total effect indices measure the portion of variability that is due to total variation in each input.

It should be noted that the variance measures are linked to the Sobol decomposition (as discussed in Appendix A), when the parameters are independent, and the total variance of $f(\cdot)$ can be represented as the sum of the variances for each term given in Equation (14), (see References [36,37,39] for further details about the Sobol decomposition and indices).

4.2. Emulators-Based Sensitivity Analysis

In principle, if the function of interest, $f(\mathbf{x})$ was adequately simple, the sensitivity measures, discussed in Section 4.1 and Appendix A, could be computed in an analytical manner. As the function of interest representing an AV system/scenario become more complex, which is the case for this paper, these SA measures cannot be analytically evaluated, and thus a computationally more efficient and robust model is required to compute these measures discussed in Section 4.1.

If $f(\mathbf{x})$ is computationally cheap enough and can be quickly evaluated for a large number of different inputs, standard MC methods would be sufficient to efficiently evaluate the SA measures described in Section 4.1. The MC-based computation techniques proposed in References [33,34,40] demand considerable (thousands to millions) function evaluations. Thus, these methods are impractical for a computationally expensive function, such as the vehicle simulation at hand. In order to tackle this computational complexity, the methodology proposed in References [36,37] was used and further developed based on the Bayesian paradigm. A Bayesian approach lets us estimate all the quantities that are required to examine the SA in modelling and predicting the outcomes of concrete scenarios (derived of the two addressed logical scenarios), amounting to a global sensitivity analysis.

Until we actually run a concrete scenario in simulation, the functional relationship $f(\cdot)$, between the scenario outcome and other input parameters are unknown for any particular input configuration $\mathbf{X}$. Within the Bayesian setting, it is therefore plausible to specify a prior distribution for the unknown input parameters, $\mathbf{X}$. Please note that placing prior distribution on the input parameters would provide a useful operational meaning to the SA measures of interest in this paper, and also provide a way to quantify the uncertainty on the input parameters as described in Reference [37].

The elicited prior can then be updated to the posterior distribution via the Bayesian paradigm in the light of the data, $\mathcal{D} = \{(\mathbf{x}_i, y_i) : y_i = f(\mathbf{x}_i), i = 1, \ldots, n\}$, generated via simulation from a set of known concrete scenarios, as described in Section 5.1. The resulting posterior distribution for $f(\cdot)$ can then be used to make formal Bayesian inferences about

the SA measures discussed in the previous section. Although there is still uncertainty about the exact value of the function $f(\cdot)$ at input values where it was not evaluated, we can further reduce uncertainty when the correlation of function values from one point to another is taken into account. This is done by taking the expected value of the obtained posterior distribution as a point estimate for $f(\cdot)$. Furthermore, this means that there are two different distributions being used in the SA computation: firstly, the distribution G representing the uncertainty in the input parameters **x**, which is then propagated to the output values through the function $f(\cdot)$; and secondly, the posterior distribution on $f(\cdot)$ which is very necessary for the efficient computation of the SA measures. The uncertainty on the model output (or the scenario output) can be reduced as much as required by evaluating the function $f(\cdot)$ through many simulation runs and thus increasing the training points. The next section describes the probabilistic approach for computing the SA measures using the Gaussian Process emulator.

Inference for Variance-Based Sensitivity Analysis Measures

This section will outline how the GP posterior distribution, which we derived in Section 3, can be used to estimate the sensitivity measures introduced in Section 4.1 and Appendix A. As stated in Reference [36], we can use the GP emulator developed to probabilistically estimate $f(\cdot)$, to make inference about the main and interaction effects of $f(\cdot)$, since they can be considered as linear combinations of $f(\cdot)$. Furthermore, since the posterior distribution of $f(\cdot)$ is a multivariate Gaussian distribution (c.f. Equation (4)), the resulting posterior distribution for the main and interaction effects is also a multivariate Gaussian distribution. Specifically, if the posterior mean for $f(\cdot)$ is given by Equation (5), then the posterior mean of the following quantity

$$E(f(\mathbf{X}) \mid \mathbf{x}_p) = \int_{\chi_{-p}} f(\mathbf{x}) dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p), \tag{17}$$

(recalling that χ_{-p} denote the input space associated with $\mathbf{x}_{-p}$, and $G_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p)$ is conditional distribution of $\mathbf{x}_{-p}$ given $\mathbf{x}_p$ under the input parameters distribution, G), can be written as

$$E_{post}\{E(f(\mathbf{X}) \mid \mathbf{x}_p)\} = T_p(\mathbf{x}_p)K^{-1}\mathbf{y}, \tag{18}$$

where

$$T_p(\mathbf{x}_p) = \int_{\chi_{-p}} t(\mathbf{x})^T dG_{-p|p}(\mathbf{x}_{-p} \mid \mathbf{x}_p). \tag{19}$$

The posterior mean of main effect or interaction can be similarly obtained as follows:

$$E_{post}\{z_i(X_i)\} = \{T_i(x_i) - T\}K^{-1}\mathbf{y} \tag{20}$$

Similarly, we can derive the standard deviations of the main effects and interactions. For further details see References [36,37].

After evaluating Equation (20) for all inputs X_i, we can visually assess the main effects by plotting their posterior means $E_{post}(z_i(X_i))$ against x_i, typically with bounds of plus and minus two posterior standard deviations. In order to see the influence of each variable, the input variables can be standardized and $E_{post}(z_i(X_i))$ drawn in a single plot for $i = 1, \ldots, d$. The results of the SA of the two logical scenarios of interests in this paper will be presented in Section 6.

Direct posterior inference for the variance-based measures introduced in Section 4.1, V_i and V_{T_i}, is more complex as these measures are quadratic functionals of $f(\cdot)$. See Reference [36] for detailed discussion on mathematical approached of dealing with quadratic functional forms of $f(\cdot)$.

5. Method for Scenario Optimisation and Data Acquisition through Simulation

In this section, we present the methodology to optimise a given logical scenario towards the most critical concrete scenarios. We discuss how the scenario data sets required to train the GP models and compute the SA measures, as described in Section 4, are generated using a vehicle simulation. The optimisation method is later applied to an exemplary functional scenario of a pedestrian stepping onto the road in front of the AV, described in Section 6.

The AV scenario, according to the scenario-based testing methodology described above, must be parameterised in terms of a set of input parameters. These scenario parameters define the functional scenario and by providing concrete values for each parameter, the concrete scenarios are constructed. The scenario parameters are selected to describe the geometric constitution of the functional scenario, the behaviour of the respective actors and the physical characteristics of the AV. Additionally, to supply a useful operational meaning to the sensitivity measures described in this paper, it is important that the input parameters considered for each scenario have a reasonable operational interpretation. Constricting the parameter space to reasonable ranges also facilitates the optimisation towards critical scenarios without excluding possible scenarios ex-ante. In order to ensure that the parameters are meaningful and to be able to model the uncertainty on these parameters, we assume these parameters are independently distributed according to some statistical distributions over realistically chosen ranges.

Based on a given logical scenario description, we present a method to optimise its parameters towards concrete scenarios with the targeted level of criticality. An overview of the method is shown in Figure 2. The method extends the notion of modelling a logical scenario with a GP model as shown in Reference [8], by implementing an iterative update step, where new information is collected by executing selected concrete scenarios in simulation. The method starts with an initial set of concrete scenarios, which are evaluated using a vehicle simulation and scored with a criticality metric, as described in Section 5.2. This data set is then used to train the GP models, which in turn are used in an optimisation algorithm to create candidate scenarios predicted to be most informative in finding critical scenarios. This algorithm is described in Section 5.3. The method and the resulting, optimised data set of critical scenarios are described in Section 6 on an exemplary scenario of a pedestrian stepping onto the road.
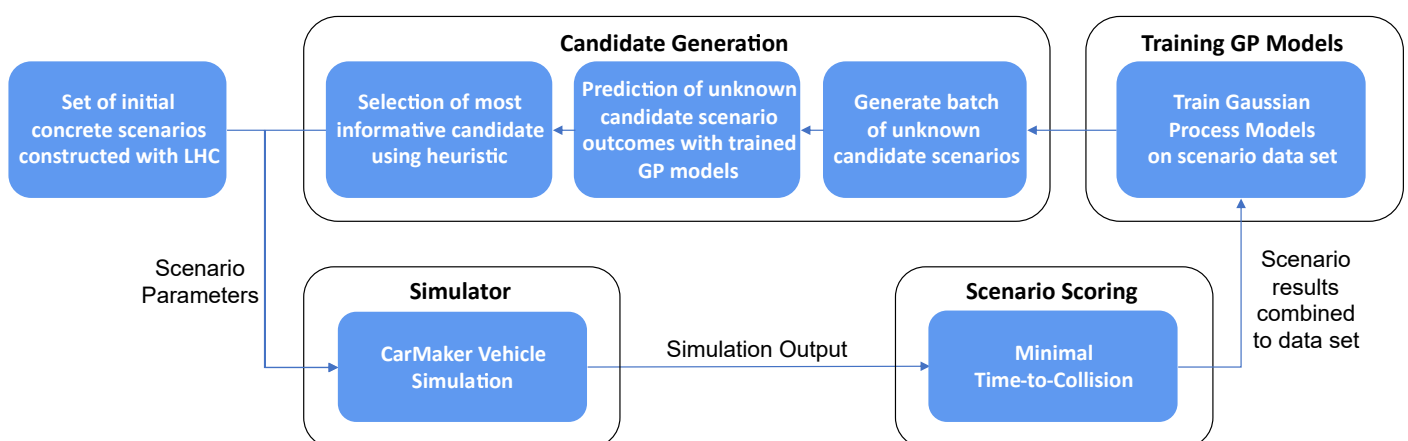

Figure 2. Overview of the automated vehicle (AV) Scenario Optimisation Method.

5.1. Initial Scenario Data Set and Data Generation via Simulation

In order to represent a logical scenario with a GP model, an initial data set of concrete scenarios is necessary to accurately train the GP model. In this section, we briefly describe the sampling method used to construct an initial data set that is optimally suited to train the GP models.

We use a Latin Hypercube (LHC) design to create an initial data set of predefined size [41]. In LHC, the scenario space is divided into equal parts, so-called Latin Hypercubes, where the sampling positions are randomly chosen from the input probability distribution. This ensures that the entire scenario space is covered as evenly as possible, without leaving large areas uncovered or areas where many samples are densely packed.

The LHC design has been found to be superior to a random sampling strategy (MC sampling) for modelling logical scenarios [8]. In a numerical experimental study conducted by Reference [42], it was illustrated that the sampling error for MC is 20 times larger than for the LHC based sampling method. However, the published theoretical results show that the sampling error of the MC goes down as $\mathcal{O}(n^{-1/2})$, whereas the sampling error for LHS is $\mathcal{O}(n^{-1})$, quadratically faster, for almost all distributions and statistics in common use. In other words, if you need n samples for a desired model accuracy using LHC, you will need n^2 samples for the same accuracy using MC sampling method [37].

While it is possible to increase the size of an MC sample data set by generating more data from the uniform distributions of input parameters, the same is not possible for data sets designed with the LHC method. The sample, based on the LHC sample method, will be optimally generated according to the requested sample size and the range of input parameters. As the computational complexity of training a GP model is $\mathcal{O}(n^3)$, a relatively small initial set of 100 concrete scenarios is chosen. A GP model trained on this data set is sufficient for use in the search algorithm described in Section 5.3.

Once a data set of concrete scenarios is available, the concrete scenarios are passed to the simulator to be executed, as shown in Figure 2. Here, the functional scenario is modelled and evaluated in the vehicle simulation software CarMaker [32]. The ego vehicle, along with the actors and geometric layout of the scenario are therein modelled in a way, that the scenario parameters can be set to the concrete values provided by the initial data set and the optimisation algorithm. Section 6.1 describes the modelling of the exemplary scenario in detail.

The drawback with this approach is, that with a change in the logical scenario, the modelling in the simulation needs to be adapted and the optimisation conducted again. The advantage of using simulation, however, is the ability to swiftly generate the required simulations of the concrete scenarios from the given logical scenario. It also provides all the necessary measurements to assign a score to each concrete scenario, indicating the outcome, and thus enabling the construction of a data set, which can be used to train the GP surrogate models.

5.2. Scenario Scoring

In order to be able to model the concrete scenario data using a Gaussian Process, a label or score needs to be assigned to each concrete scenario. Therefore, we assess the input parameters on their impact on the outcome of the scenario, by evaluating the data provided by the simulator. The score of the scenario needs to reflect the criticality, in order to focus the testing effort towards more critical scenarios, close to the performance boundary. The intention behind this is to omit concrete scenarios that certainly do or do not end in a collision. These are concrete scenarios far from the performance boundary and therefore less interesting for the validation of AVs. As scoring metric, the minimal time-to-collision (TTC) was therefore chosen as criticality measure. The TTC has previously been described as an effective measure for rating the severity of scenarios involving longitudinal conflicts in collision avoidance systems [43] and can thus be used to differentiate between critical and non-critical scenarios for AVs in the considered scenarios. The TTC can be calculated by

$$TTC = \frac{d_{rel}}{v_{rel}}, \tag{21}$$

wherein d_{rel} and v_{rel} are the relative distance and relative velocity between the ego vehicle and in the exemplary scenario, the pedestrian that is stepping onto the road. In order to obtain a single score for the concrete scenario, the TTC is calculated at every time step of

the simulation. The minimum TTC value over the entire duration of the concrete scenario is then taken as its criticality measure. It is to be noted that d_{rel} and v_{rel} are not equivalent to the scenario parameters d and S_1/S_2, as described in Section 6.1. The quantities, d_{rel} and v_{rel} change over the course of the scenario, depending on the actions of the ego vehicle and the pedestrian. The scenario parameters influence d_{rel} and v_{rel} and thus the TTC, but the exact relationship is to be analysed hereby.

Van der Horst and Hogema [43] specified the threshold for a critical scenario to be 1.5 s, which we will use as an optimisation criterion in the following section. A TTC threshold of 1.5 s will not target scenarios which result in a collision. While collision scenarios are certainly of value in an AV validation, the goal in this work is to find critical corner case scenarios, which are scenarios that almost resulted in a collision.

5.3. Methodology for Scenario Optimisation

After showing the efficacy of modelling logical scenarios using GP models in Reference [8], a logical next step is to optimise the validation process towards only simulating concrete scenarios that are most critical. The parameter space of the logical scenario described in Section 6.1 includes parameter combinations which could lead to less interesting concrete scenarios. These include, for example, the concrete scenarios, where the pedestrian crosses the road long before the ego vehicle reaches the intersection point or on the contrary, where the pedestrian crosses the road after the ego vehicle has passed. These concrete scenarios are not particularly interesting for a validation of the AV, as no direct interaction or conflict arises. In the following, we present a method to optimise the parameter space of the logical scenario to find the critical concrete scenarios that were determined by evaluating the TTC measure as outlined in Section 5.2.

We apply a two-stage GP approach for a given logical scenario, consisting of GPC and GPR models as defined in Section 3. Firstly, an initial data set $\mathcal{D}_I = \{(x^I)\}$ of concrete scenarios is created from the definition of the selected logical scenario. Since it is desired to evenly cover the entire parameter space of the logical scenario, the LHC approach as described in Section 5.2 was chosen as the optimised design to generate $\mathcal{D}_I$. The corresponding outputs (y^I) to the concrete scenarios are then determined by computing the criticality measure of each concrete scenario as described in Section 5.2.

The initial data set augmented by the computed outputs, $\mathcal{D}_I = \{(x^I, y^I)\}$, is converted into a classification data set $\mathcal{D}_C = \{(x^c, y^c)\}$ where the superscript c stands for classification, and a data set $\mathcal{D}_R = \{(x^r, y^r)\}$, which is used for regression modelling, thus the superscript r. For $\mathcal{D}_C$, the output or label (y^c) of each concrete scenario instance is binary, describing whether the concrete scenario has a valid TTC ($y^c = 1$) or not ($y^c = 0$). This is done in order to filter out scenarios where the pedestrian crosses the road far ahead or behind the ego vehicle, as previously described. Such concrete scenarios would not have a valid TTC, as either Equation (21) cannot be reasonably evaluated, or the TTC is obstructively large and thus not of interest. Furthermore, $\mathcal{D}_R$ is then the subset of the concrete scenarios in $\mathcal{D}_C$ with label $y^c = 1$, but the output (label) of the $\mathcal{D}_R$ data set, y^r, is the actual minimum TTC score as calculated by Equation (21).

These two data sets are then used in the two-staged GP approach, as described in Algorithm 1. Further inputs to Algorithm 1 are the total number of newly generated, critical scenarios N, the number of candidates used in each iteration n_{cand}, and the target TTC measure ttc_{target}.

Algorithm 1: Scenario Optimisation

Input: $\mathcal{D}_C = \{(x^c, y^c)\}$, $\mathcal{D}_R = \{(x^r, y^r)\}$, N, n_{cand}, ttc_{target}
Result: $\mathcal{D}_{crit}$

for *iter* $= 0$ **to** N **do**
$\quad$ $x^* \leftarrow \text{rand}(n_{cand})$ $\qquad\qquad$ /* create n_{cand} candidate scenarios */
$\quad$ $\mathcal{GPC} \xleftarrow{train} \mathcal{D}_C$ $\qquad\qquad\qquad\qquad\qquad\qquad$ /* GPC model */
$\quad$ $\mathcal{GPR} \xleftarrow{train} \mathcal{D}_R$ $\qquad\qquad\qquad\qquad\qquad\qquad$ /* GPR model */
$\quad$ $\hat{y}_c^* \leftarrow \mathcal{GPC}.\text{predict}(x^*)$
$\quad$ $x_{ttc}^* \leftarrow x^*(\hat{y}_c^* = 1)$ $\qquad$ /* get candidates predicted to have a TTC */
$\quad$ $\hat{y}_{ttc}^*, \hat{\sigma}_{ttc}^* \leftarrow \mathcal{GPR}.\text{predict}(x_{tcc}^*)$ $\qquad$ /* get predicted TTC of candidates
$\quad$ from GPR */
$\quad$ $x_{opt}^* \leftarrow \text{H}(\hat{y}_{ttc}^*, \hat{\sigma}_{ttc}^*, ttc_{target})$ $\qquad$ /* apply selection heuristic to GPR
$\quad$ predictions */
$\quad$ **while** $y_{new} = \varnothing$ **do**
$\quad\quad$ /* while no TTC in scenario found $\qquad\qquad\qquad\qquad\qquad$ */
$\quad\quad$ $x_{opt}^* \leftarrow x_{opt}^*.\text{pop}()$ $\qquad$ /* select scenario candidate that is most
$\quad\quad$ favourable */
$\quad\quad$ $y_{new} \leftarrow \text{sim}(x_{opt}^*)$ $\qquad\qquad\qquad$ /* get actual TTC from sim */
$\quad$ **end**
$\quad$ $\{\mathcal{D}_C, \mathcal{D}_R, \mathcal{D}_{crit}\}.\text{append}(x_{opt}^*, y_{new})$ $\qquad$ /* add simulated scenario and
$\quad$ corresponding TTC to respective data sets */
end

In order to find critical scenarios, firstly a large pool of n_{cand} concrete scenarios is generated at random and the GPC and GPR models are trained on the respective data sets. The GPC model is then used to predict which of the randomly generated candidate scenarios x^* have a valid TTC, meaning the corresponding predicted output label is $\hat{y}_c^* = 1$, based on the knowledge from previously executed scenarios in $\mathcal{D}_C$. Those candidate scenarios of the original candidate pool, which are predicted to have a valid TTC, x_{tcc}^*, are then used with the GPR model to predict the TTC measures of each scenario candidate in x_{tcc}^*. A heuristic as described in Reference [44] is used to rank the candidate scenarios according to their significance.

The heuristic trades of the proximity of the candidate scenario to the criticality threshold ttc_{target} with the predictive variance supplied by the GPR model, which indicates the uncertainty of the prediction. Thus concrete scenarios which are close to the criticality threshold and unknown to the predictive GPR model are scored highest. A straddle heuristic as given in Reference [44] was found to yield the best selection results:

$$H(s) = 1.96 * \hat{\sigma}_{ttc}^* - |\hat{y}_{ttc}^* - ttc_{target}|. \tag{22}$$

In Equation (22), $\hat{\sigma}_{ttc}^*$ and $\hat{y}_{ttc}^*$ designate the predictive variance and TTC of a given candidate scenario, provided by the GPR model, and $s = (\hat{y}_{ttc}^*, \hat{\sigma}^*, ttc_{target})$.

The most favourable scenario candidate x_{opt}^* is then selected according to the highest score calculated by Equation (22). This concrete scenario is then actually evaluated in the Carmaker simulation model. If no valid TTC is found, the next best scenario candidate is chosen according to $H(s)$.

Finally the data sets $\mathcal{D}_C$ and $\mathcal{D}_R$ are updated with the new information ($\{(x_{opt}^c, y_{new}^c)\}$ and $\{(x_{opt}^r, y_{new}^r)\}$) from the simulation, and the found critical concrete scenario ($\{x_{opt}^*, y_{new}\}$) is added to a data set of critical scenarios $\mathcal{D}_{crit}$. Algorithm 1 is executed until the predefined number of N critical scenarios are found.

Since the data sets $\mathcal{D}_C$ and $\mathcal{D}_R$ used in the GPC and GPR models of the logical scenario are updated after each iteration, Algorithm 1 iteratively improves the knowledge of the performance boundary of critical scenarios around ttc_{target}.

6. Results

In the following the results of the proposed optimisation method are presented. Firstly, the previously mentioned exemplary scenario of a pedestrian stepping onto the road in front of the AV is described in Section 6.1. The results of the optimisation method applied to this exemplary functional scenario are then presented in Section 6.2, including the optimised data set of critical scenarios.

Furthermore, we conduct a probabilistic sensitivity analysis as introduced in Section 4, on the optimised scenario data set generated with the method from Section 5. Additionally, we conduct the same sensitivity analysis on a previously generated scenario data set taken from Reference [8]. This data set is less complex and can thus be analysed visually to validate the sensitivity analysis. The results from the sensitivity analyses are presented in Section 6.3.

6.1. Description of the Pedestrian Step-Out Scenario

We show the efficacy of the optimisation method on the functional scenario of a pedestrian stepping onto the road in front of the ego vehicle. In this functional scenario, the ego vehicle is driving along a straight road, when suddenly a pedestrian crosses in front of the vehicle from the near side of the road. Furthermore, the pedestrian is occluded behind a parked car, and only becomes visible once they step onto the driving lane, as illustrated in Figure 3. The ego vehicle is similarly fitted with a non-pivoting radar sensor, and equipped with longitudinal control and emergency braking systems. The radar sensor was set to recognise obstructions such as pedestrians in the driving lane of the ego vehicle and pass these on to the vehicle control.
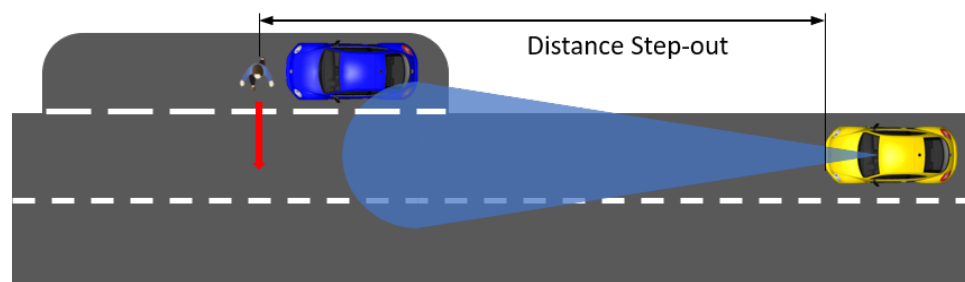

Figure 3. Pedestrian step-out scenario with the yellow ego vehicle, the pedestrian and a static blue vehicle, which acts as an obstruction so that the pedestrian is only seen once they step onto the driving lane.

The scenario can be described in its logical form using the following 5 input parameters:

1. Speed of ego vehicle (S_1);
2. Speed of pedestrian stepping onto street (S_2);
3. Distance between pedestrian and ego vehicle at the time the pedestrian steps out (d);
4. Sensor Range (blue cone) (SR);
5. Horizontal field-of-view of sensor (opening angle of blue cone) (H).

In addition to the initial speed of the ego vehicle, the speed of the pedestrian, which in this case moves in a straight line orthogonal to the ego vehicle, is parametrised. The radar sensor is parametrised using two input parameters: its horizontal field-of-view and its range. Furthermore, we add a parametrisation of the distance between the ego vehicle and the pedestrian, controlling when the pedestrian starts their movement to step onto the road. This spans up a five-dimensional logical scenario space, from where possible concrete values are sampled. The ranges of these five input parameters are listed in Table 1.

Table 1. Input parameters for the pedestrian step-out scenario and their ranges.

Simulation Parameters	Lower Limit	Upper Limit
S_1 [km/h]	10	50
S_2 [km/h]	2	15
d [m]	5	40
SR [m]	10	100
H [deg]	45	150

6.2. Optimised Critical Scenarios for the Pedestrian Step-Out Scenario

The optimisation as described in Section 5.3 was run for $N = 100$ iterations. This yielded 100 critical concrete scenarios in $\mathcal{D}_{crit}$, which are close to the critical TTC of $ttc_{target} = 1.5$ seconds. The initial data $\mathcal{D}_I$, constructed via LHC sampling to optimally fill the parameter space, had a size of 100 concrete scenarios and was executed in simulation. From $\mathcal{D}_I$, the classification data set $\mathcal{D}_C$ and the regression data set $\mathcal{D}_R$ were constructed and then fed into Algorithm 1. In every iteration of Algorithm 1, a new candidate scenario pool of size $n_{cand} = 1000$ was used.

To account for variations in the sampling, the optimisation was repeated three times with different initialisation seeds. The optimised concrete scenarios of one of the experiments are visualised in Figure 4. The concrete scenarios are coloured according to their actual minimum TTC measure, as evaluated in the simulation. In the three repetitions, between 71%–75% of the concrete scenarios in $\mathcal{D}_{crit}$ are within 0.5 s of the target TTC of 1.5 s, while only 9%–12% had a TTC higher than 2 s. Between 15%–18% of the concrete scenarios that were found had a TTC between 0 and 1 s.

Furthermore, we evaluated the Mean Absolute Error (MAE) and Root-Mean-Squared Error (RMSE) of the optimised scenario with respect to the target TTC. The resulting set of MAE = {0.086, 0.146, 0.114}, measured in seconds, shows very low deviation from the TTC target of 1.5 s. The RMSE was calculated to {1.516, 1.426, 1.297} seconds. The RMSE is more sensitive to outliers than the MAE. This can be observed here, as there are a few larger outliers in the data set of optimised scenarios. The largest outlier scenario had a minimum TTC of 11 s, as can be seen from Figure 4. The inclusion of these scenarios might be due to a slightly inaccurate prediction in the GP models, or they were included by the heuristic on purpose to explore the scenario space. This is the case, if one of the scenario candidates has a high predictive variance. It is then included by the heuristic according to Equation (22) to explore the scenario space and improve the GP models.
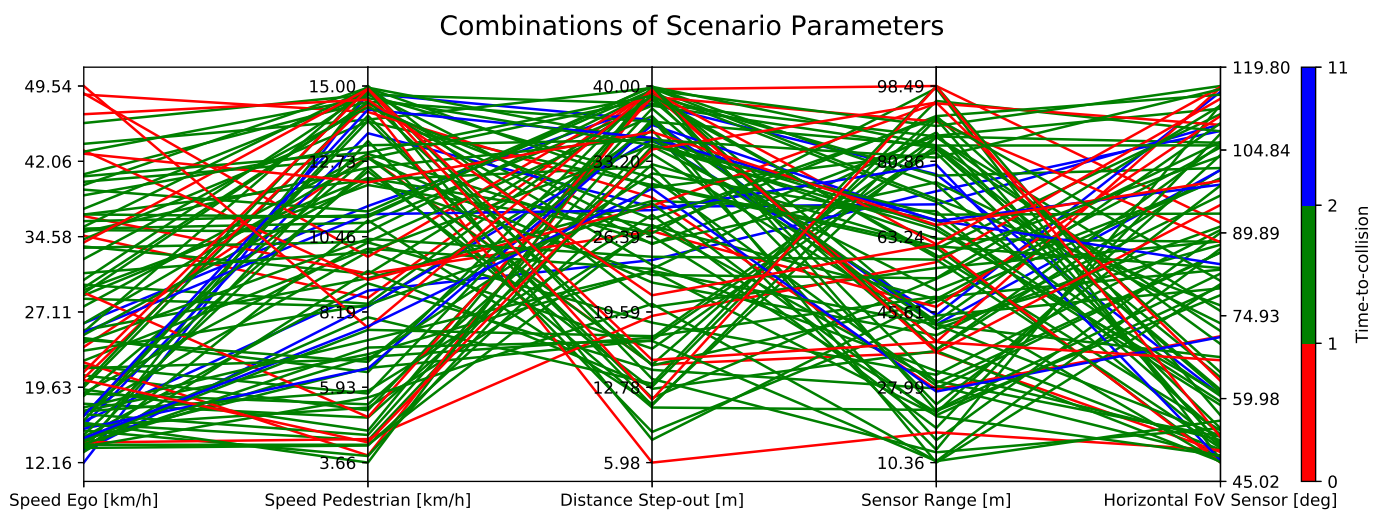

Figure 4. Combinations of scenario parameters of critical scenarios for the pedestrian step-out scenario.

6.3. Probabilistic Sensitivity Analysis of Scenario Data Sets

It is further necessary to analyse the sensitivity of the scenario parameters, in order to determine the efficacy of this method. We conducted a novel sensitivity analysis based on the GP models used to model the pedestrian step-out scenario, the results of which are presented in Section 6.3.1. The SA enables us to determine the influence of the scenario parameters on the outcome and could be used to reduce the sampling density of less influential parameters, and thus overall computational effort.

Furthermore, we conduct a sensitivity analysis on a second scenario data set taken from Reference [8]. This data set modelled a less complex traffic jam approach scenario, which could be visually analysed. The second scenario and corresponding results of the probabilistic sensitivity analysis are described in Sections 6.3.2 and 6.3.3.

6.3.1. Sensitivity Analysis of the Pedestrian Step-Out Scenario

As described above, the probabilistic SA measures are very important to evaluate the influence of the input parameters on the scenario outcome. Additionally, the SA would aid in simplifying the final model formulating the AV scenario. Reducing the dimensionality of the parameter space would be very practical in better visualising the performance boundary of the scenarios with a dimensionality larger than three. In this section, the probabilistic SA measures for the pedestrian step-out scenarios will be efficiently computed and illustrated using the GP-based method, as described in Section 4.2.

For the pedestrian step-out scenario, it is not feasible to examine the main effects from a plot of all the executed concrete scenarios, as the dimensionality of parameters is larger than 3. Thus a probabilistic sensitivity analysis is here more appropriate than for a simple scenario as presented in Reference [8]. The main effects are shown in Figure 5 and discernibly linear, with parameter S_1 having an opposite influence on the scenario outcome than S_2 and d. The almost horizontal lines of parameters SR and H and low values of their first-order sensitivity indices (c.f. Table 2) imply a negligible influence on the scenario outcome. This information can, for example, be used to change the sampling behaviour for these two parameters to be more conservative, as their effect on the scenario outcome is small. More dense sampling is however advised on S_2 and d and especially S_1, as it has the highest impact on the variability of the output.

By comparing the total effect sensitivity indices with the first-order sensitivity indices in Table 2, it is evident that interaction effects between input parameters are largely negligible. Table 2 presents the SA indices to evaluate the sensitivity of the scenario outcome with respect to changes in the input parameters presented in Table 1.

It is evident that 'speed of ego car (S_1)' and 'speed of pedestrian stepping onto street (S_2)' are the most influencing factors affecting the outcome of the pedestrian step-out scenario according to their variance contributions, 28% and 17.5%, respectively. It should be noted that SA indices reported in Table 2 are all computed based on 100 data points selected using the LHC sampling method. We can also say that 54.5% of total variance can be explained by the first-order interaction between (S_1, S_2), which is very significant. In other words, we can say that the total variance of the scenario outcome can be explained based on S_1, S_2 and their first-order interaction. The estimated posterior mean and variance of the fitted GP to this data set were reported to be 0.83 and 0.028, respectively.

Figure 5 and Table 2 illustrate the first order (S_i) and total effect (S_{T_i}) variance-based sensitivity indices for the input parameters. Using these sensitivity measures, we can similarly conclude that the outcome of the pedestrian step-out scenario is more sensitive to S_1, d (distance between pedestrian and ego vehicle), and S_2, consecutively. It seems the conclusions taken from these SA indices are slightly different from those taken from the main and the first-order interaction effects. We can argue that the interaction between the speeds of ego vehicle and pedestrian could be somehow represented in terms of the distance between pedestrian and ego car. Furthermore, these variance-based sensitivity indices suggest that both the sensor range and horizontal field-of-view of the sensor do

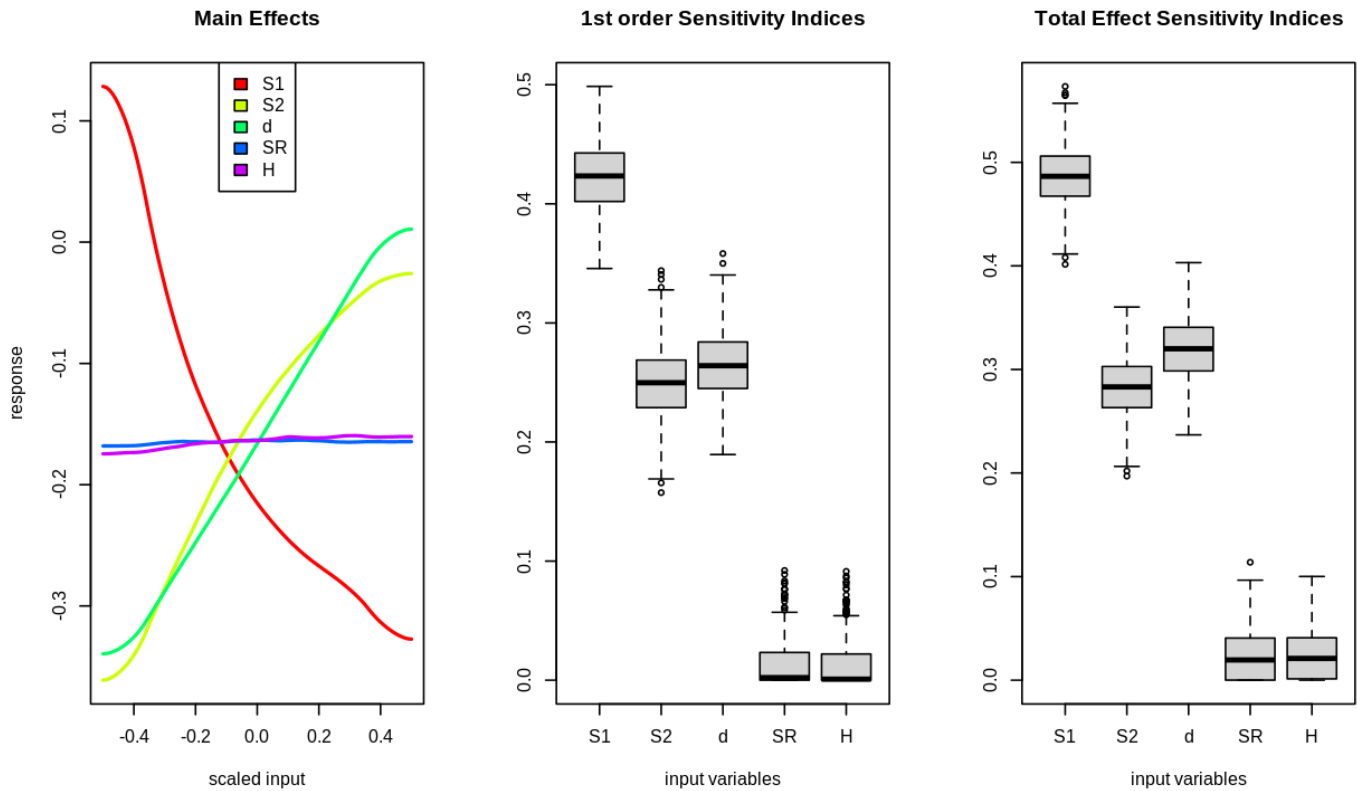not have a manoeuvresignificant impact on the outcome of the scenario (i.e., its minimum time-to-collision).

Figure 5. The estimated main effects, 1st and 2nd order sensitivity indices of the parameters in the pedestrian step-out scenario as discussed in Table 1.

Table 2. The emulator-based sensitivity analysis (SA) of the pedestrian step-out scenario (c.f. Section 6.1).

Parameters	Variance (%)	S_i	S_{T_i}
Speed of ego vehicle (S_1)	27.98	0.4221	0.4843
Speed of Pedestrian stepping onto street (S_2)	17.47	0.2468	0.2852
Distance between pedestrian and ego vehicle (d)	0	0.265	0.3215
Sensor Range (SR)	0	0.0124	0.0196
Horizontal field-of-view of sensor (H)	0	0.0128	0.0199
(S_1, S_2)	54.53		
Total Variance (%)	99.99		
Estimated mean output	$0.83 \pm 9.0905 \times 10^{-6}$		
Estimated variance output	0.028		

6.3.2. Scenario Description of the Traffic Jam Scenario

Furthermore, we took a scenario data set from Reference [8], and used the same SA methods on the data. Since the scenario considered in Reference [8] is simpler and only involves three scenario parameters, a direct comparison of the SA with the visual assessment of the scenario data set is possible and can thus validate the SA method.

The functional scenario considers an AV, equipped with a forward-facing radar sensor to detect objects, which approaches a traffic jam in which vehicles are moving considerably slower than the approaching ego vehicle. Furthermore, the road layout is a left-turning curve with a fixed radius of 50 m, where the vehicles under consideration travel on the outer

lane. The traffic jam is represented by the last vehicle driving in it to reduce simulation complexity and the ego vehicle has a clear line of sight on the traffic jam. Figure 6 illustrates the functional setup of the scenario. It is ensured that the vehicle always arrives at the traffic jam while turning on the curved road. An overtaking manoeuvre of the AV in this scenario is not desired, as the opposite lane might not be free.
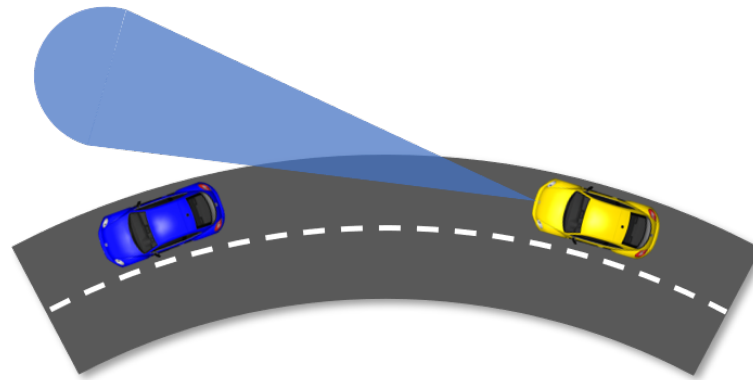

Figure 6. Traffic jam approach scenario with the blue target vehicle and the yellow ego vehicle equipped with a radar sensor.

In order to carry out scenario-based testing, a logical scenario representation is defined from this functional description. The scenario was parametrized with the following three parameters:

1. Speed of the ego vehicle (S_1)
2. Speed of the traffic jam (S_2)
3. Aperture angle of the radar sensor (AA)

The logical scenario description furthermore requires the input parameters to be restricted on realistic ranges, to loosely constrict the scenario space. These ranges can be found in Table 3, representing the logical description of the scenario.

Table 3. Input parameters for the traffic jam approach scenario and their ranges.

Simulation Parameters	Lower Limit	Upper Limit
S_1 [km/h]	40	70
S_2 [km/h]	5	20
AA [deg]	10	25

The concrete scenarios for the traffic jam scenario were scored on a binary scale, depending on whether the ego vehicle was able to prevent a collision (0) or not (1). This poses a binary classification problem, which was modelled using a GPC as described in Section 3.3. The exact description of the scenario data set and its modelling can be found in Reference [8].

6.3.3. Sensitivity Analysis of the Traffic Jam Scenario

The results of the SA for the traffic jam scenario are visualised in Figure 7, where the main effect plot reflects what is already known from studying the visualisation of the three scenario parameters as discussed and illustrated in Reference [8]. The influence of S_1 on the scenario outcome is contrary to the influence of S_2 and AA, with all of them having linear effects over the entire input domain. In concrete terms, this means, the logical scenario results in a collision for higher values of S_1 and lower values of S_2 and AA.
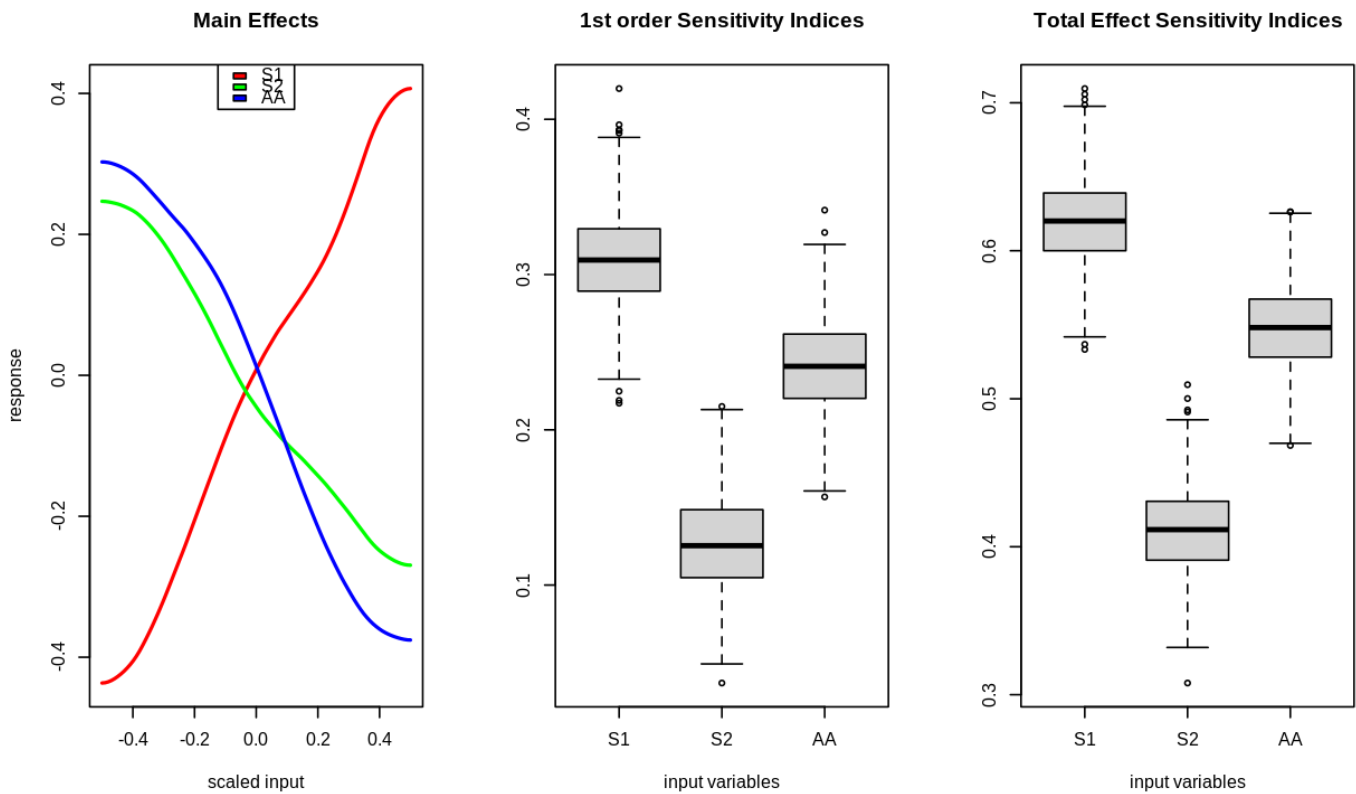
Figure 7. The estimated main effects, 1st and 2nd order sensitivity indices of the parameters in the traffic jam scenario as discussed in Table 3.

The box plot of the first-order sensitivity indices in the middle of Figure 7 shows that parameter S_1 has the highest effect on the scenario outcome, followed by AA and S_2. It is to be noted that the uncertainty in the box plots includes the uncertainty about the input-output relationship as well as the uncertainty of the integration estimates for the emulator.

For uncorrelated input parameters and a deterministic response, the first-order sensitivity index would be equal to the total effect sensitivity index, $S_i = S_{T_i}$. A comparison of the first-order sensitivity indices with the respective total effect sensitivity indices therefore presents a measure of the proportion of variability due to interaction between input parameters. From Figure 7 and the values listed in Table 4, it is discernible that the total effect of all three parameters is significantly higher than the first-order sensitivity indices. This indicates interaction effects between the input parameters causing a portion of the variability in the scenario outcome.

Table 4. The emulator-based SA of the traffic jam scenario (c.f. Section 6.3.2).

Parameters	Variance (%)	S_i	S_{T_i}
Speed of ego vehicle (S_1)	28.88	0.3122	0.6160
Speed of traffic jam (S_2)	16.64	0.1289	0.4088
Sensor Aperture Angle (AA)	31.18	0.2427	0.5438
(S_1, S_2)	3.72		
(S_1, AA)	6.21		
(S_2, AA)	3.87		
Total Variance (%)	90.49		
Estimated mean output	$0.537633 \pm 8.1986 \times 10^{-5}$		
Estimated variance output	0.136		

Table 4 reports the probabilistic SA measures for the scenario outcome (collision or no-collision) with respect to the changes of the input parameters discussed in Table 3. From the second column of Table 4, we can say that 31.2% of total variance (also known as the main effect and explained in Section 4) can be explained by AA (sensor aperture angle) based on 100 data points selected using the LHC sampling method, while main effects of 'Speed of ego vehicle' (S_1) and 'Speed of traffic jam' (S_2) are describing 28.9% and 16.6% of total variance, respectively. We can then conclude that the outcome of the scenario is more sensitive to variation of AA than S_2. Note that the main effect proportions do not sum to 100% of the variance. The remaining variance after the main effects are deducted from the total variance is thus explained by the first-order interactions (13.8%) and the third-order interaction (9.51%), as explained in Appendix A.

The data set used to train the GP emulator required for classification and computing the SA measure was almost balanced in the sense that 56% of concrete scenario outcomes resulted in a collision. The estimated posterior mean and variance of the fitted GPC to this data were reported to be 0.537 and 0.136, respectively.

Using the first order (S_i) and total effect (S_{T_i}) variance-based sensitivity indices from Figure 7 and the Table 4, we can conclude that the outcome of the traffic jam scenario is more sensitive to S_1, AA and S_2, consecutively.

7. Conclusions and Future Work

In this article, we showed how Gaussian Process models can be used to test AVs in a scenario-based setting. We used a vehicle simulation of scenarios to compile concrete scenario data and train probabilistic models to predict the scenario outcomes. GPs were formally introduced and it was shown that GP models can predict the outcomes of a logical scenario including several actors and their interactions.

We presented a method to optimise the parameter space of a logical scenario towards critical scenarios, measured by their minimal time-to-collision. This active learning method, as presented in Algorithm 1, was then evaluated on an exemplary functional scenario of a pedestrian stepping onto the road in front of the ego vehicle. The scenario was parametrised with 5 scenario parameters and the optimisation method sampled between 71–75% of new concrete scenarios within 0.5 s of the targeted, critical time-to-collision of 1.5 s. The Mean Absolute Error was between 0.086 and 0.146, while the Root-Mean-Squared Error was between 1.297 and 1.516, measured in seconds and for differently seeded initialisations of the method. The resulting data set of critical scenarios consisted of the targeted 100 concrete scenarios.

Furthermore, we formally introduced and conducted a probabilistic sensitivity analysis on the optimised data set and found that the configurations of ego speeds, actor speeds and the geometric layout of the scenario had the strongest effect on the scenario outcome. Sensor parameter variations were found to have a lower impact. This information can be used to further restrict the parameter space of the logical scenarios and denser sampling of the most important parameters.

The same probabilistic sensitivity analysis was also conducted on a simpler scenario data set of a functional scenario describing a traffic jam approach, taken from literature. On this functional scenario, a visual analysis of the parameter influences was possible and could therefore be linked to the results of the probabilistic sensitivity analysis and validate the sensitivity results. For the functional scenario of the pedestrian stepping onto the road, it is not possible to visually analyse the scenario parameter influences, due to the higher number of parameters.

A limitation of the optimisation method is the fitting of GP models in every iteration step, which is computationally expensive. This could be improved if the GP model fitting is adapted to only incorporate the newly found concrete scenario, instead of refitting the GP models in every iteration. Further research is also necessary into the efficacy of different selection heuristics, which trade off exploration and exploitation of the logical scenario parameter space. This is something we plan for future work, along with extending the GP

modelling to be able to classify multiple possible outcomes of the scenario, for example in several criticality classes and with different criticality metrics.

Author Contributions: Conceptualization, F.B. and A.D.; methodology, F.B.; software, F.B.; validation, F.B.; formal analysis, F.B. and A.D; data curation, F.B.; writing—original draft preparation, F.B. and A.D.; writing—review and editing, F.B. and A.D.; visualization, F.B.; supervision, A.D., V.P. and M.C.; project administration, A.D., V.P. and M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Function Decomposition for Main Effects and Interactions

Sobel [33] proves that any quadratically integrable function $f(\cdot)$ can be decomposed in terms of its main effects and interactions as follows:

$$y = f(\mathbf{x}) = z_0 + \Sigma_{i=1}^d z_i(x_i) + \Sigma_{i<j} z_{i,j}(\mathbf{x}_{i,j}) + \ldots + z_{1,2,\ldots,d}(\mathbf{x}) \tag{A1}$$

where the relationship between y and $\mathbf{x}$ is formulated via $y = f(\mathbf{x})$, $f(\cdot)$ is a function of uncertain quantities $\mathbf{x}$, and its expected value is denoted by $z_0 = E[f(\mathbf{X})]$. The function $z_i(x_i)$ appeared in Equation (A1) is the main effect of the ith variable, x_i, and defined as:

$$z_i(x_i) = E[f(\mathbf{X}) \mid x_i] - E[f(\mathbf{X})] \tag{A2}$$

The main effect, $z_i(x_i)$ is the function of x_i only that best approximates $f(\cdot)$ in the sense of minimizing the variance (calculated over the other variables) [36,37].

The function $z_{i,j}(\mathbf{x}_{i,j})$ given in Equation (A1), and defined in the Equation (A3), describes the first-order *interaction* between x_i and x_j, and similarly $z_{i,j,k}(\mathbf{x}_{i,j,k})$ is the second-order interaction, and so on.

$$z_{i,j}(\mathbf{x}_{i,j}) = E[f(\mathbf{X}) \mid \mathbf{x}_{i,j}] - z_i(x_i) - z_j(x_j) - E[f(\mathbf{X})] \tag{A3}$$

The main effect has a straightforward interpretation for the context of present study; it is the expected change to the scenario outcome that would be obtained if we were to know that parameter i has value x_i, taking into account the residual uncertainty in the other parameters. Additionally the decomposition of variance formula described in Equation (A4) implies that the variance of the main effect for parameter i is the expected amount by which the variance of f would be reduced if the value of parameter i was known.

$$var(Y) = E(var(Y|X) + var(E(Y|X)) \tag{A4}$$

From Equation (A1), it is visible that the functions in the Sobol decomposition are pairwise orthogonal [45]. The used prior distribution of the input parameters (denoted by G), affects the definition of the main and interaction effects terms given in Equation (A1) (more details are discussed in Section 4.2).

References

1. ADAC. *Zahlen, Fakten, Wissen. Aktuelles aus dem Verkehr*; Technical Report; ADAC e.V.: München, Germany, 2016.
2. Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making Bertha Drive—An Autonomous Journey on a Historic Route. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 8–20. [CrossRef]
3. SAE. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; Technical Report; SAE: Warrendale, PA, USA, 2018.
4. Kalra, N.; Paddock, S.M. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transp. Res. Part A Policy Pract.* **2016**, *94*, 182–193. [CrossRef]

5.   Zhang, S.; Peng, H.; Zhao, D.; Tseng, H.E. Accelerated Evaluation of Autonomous Vehicles in the Lane Change Scenario Based on Subset Simulation Technique. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 3935–3940. [CrossRef]

6.   Batsch, F.; Kanarachos, S.; Cheah, M.; Ponticelli, R.; Blundell, M. A taxonomy of validation strategies to ensure the safe operation of highly automated vehicles. *J. Intell. Transp. Syst.* **2020**, 1–20. [CrossRef]

7.   Menzel, T.; Bagschik, G.; Maurer, M. Scenarios for Development, Test and Validation of Automated Vehicles. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 1821–1827. [CrossRef]

8.   Batsch, F.; Daneshkhah, A.; Cheah, M.; Kanarachos, S.; Baxendale, A. Performance Boundary Identification for the Evaluation of Automated Vehicles using Gaussian Process Classification. In Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019; pp. 419–424. [CrossRef]

9.   Ulbrich, S.; Menzel, T.; Reschka, A.; Schuldt, F.; Maurer, M. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 982–988. [CrossRef]

10. Zlocki, A.; Eckstein, L.; Fahrenkrog, F. Evaluation and Sign-Off Methodology for Automated Vehicle Systems Based on Relevant Driving Situations. *Transp. Res. Rec. J. Transp. Res. Board* **2015**, *2489*, 123–129. [CrossRef]

11. Khastgir, S.; Dhadyalla, G.; Birrell, S.; Redmond, S.; Addinall, R.; Jennings, P. Test Scenario Generation for Driving Simulators Using Constrained Randomization Technique. *SAE Tech. Pap.* **2017**, 1–7. [CrossRef]

12. Althoff, M.; Mergel, A. Comparison of Markov Chain Abstraction and Monte Carlo Simulation for the Safety Assessment of Autonomous Cars. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 1237–1247. [CrossRef]

13. Gietelink, O.; De Schutter, B.; Verhaegen, M. Probabilistic validation of advanced driver assistance systems. *IFAC Proc. Vol.* **2005**, *38*, 97–102. [CrossRef]

14. Zhao, D.; Huang, X.; Peng, H.; Lam, H.; LeBlanc, D.J. Accelerated Evaluation of Automated Vehicles in Car-Following Maneuvers. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 733–744. [CrossRef]

15. Zhao, D.; Peng, H. From the Lab to the Street: Solving the Challenge of Accelerating Automated Vehicle Testing. *arXiv* **2017**, arXiv:1707.04792

16. Ngo, A.; Bauer, M.P.; Resch, M. A Sensitivity Analysis Approach for Evaluating a Radar Simulation for Virtual Testing of Autonomous Driving Functions. In Proceedings of the 2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), Singapore, 17–19 July 2020; pp. 122–128.

17. Nolte, M.; Schubert, R.; Reisch, C.; Maurer, M. Sensitivity Analysis for Vehicle Dynamics Models—An Approach to Model Quality Assessment for Automated Vehicles. *arXiv* **2020**, arXiv:2005.03872.

18. Fraikin, N.; Funk, K.; Frey, M.; Gauterin, F. Efficient Simulation Based Calibration of Automated Driving Functions Based on Sensitivity Based Optimization. *IEEE Open J. Intell. Transp. Syst.* **2020**, *1*, 63–79. [CrossRef]

19. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]

20. Al-Qizwini, M.; Barjasteh, I.; Al-Qassab, H.; Radha, H. Deep learning algorithm for autonomous driving using GoogLeNet. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 89–96. [CrossRef]

21. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]

22. Esmaeilbeigi, M.; Daneshkhah, A.; Chatrabgoun, O. Some Computational Considerations for Kernel-Based Support Vector Machine. In *Digital Twin Technologies and Smart Cities*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 177–189.

23. Remmen, F.; Cara, I.; de Gelder, E.; Willemsen, D. Cut-in Scenario Prediction for Automated Vehicles. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018; pp. 1–7. [CrossRef]

24. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; Adaptive computation and machine learning; MIT Press: Cambridge, MA, USA, 2006; p. 266.

25. Altman, N.S. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am. Stat.* **1992**, *46*, 175–185. [CrossRef]

26. Kim, K.; Lee, D.; Essa, I. Gaussian process regression flow for analysis of motion trajectories. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1164–1171. [CrossRef]

27. Ellis, D.; Sommerlade, E.; Reid, I. Modelling pedestrian trajectory patterns with Gaussian processes. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 1229–1234. [CrossRef]

28. Tran, Q.; Firl, J. Modelling of traffic situations at urban intersections with probabilistic non-parametric regression. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, QLD, Australia, 23–26 June 2013; pp. 334–339. [CrossRef]

29. Armand, A.; Filliat, D.; Ibanez-Guzman, J. Modelling stop intersection approaches using Gaussian processes. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; pp. 1650–1655. [CrossRef]

30. Mullins, G.E.; Stankiewicz, P.G.; Hawthorne, R.C.; Gupta, S.K. Adaptive generation of challenging scenarios for testing and evaluation of autonomous vehicles. *J. Syst. Softw.* **2018**, *137*, 197–215. [CrossRef]

31. Huang, Z.; Lam, H.; Zhao, D. Towards affordable on-track testing for autonomous vehicle—A Kriging-based statistical approach. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6. [CrossRef]

32. IPG Carmaker. 2019. Available online: https://ipg-automotive.com/ (accessed on 14 January 2021).

33. Sobol, I.M. Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.* **1993**, *1*, 407–414.

34. Saltelli, A.; Tarantola, S.; Chan, K.S. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* **1999**, *41*, 39–56. [CrossRef]

35. Sudret, B. Global sensitivity analysis using polynomial chaos expansions. *Reliab. Eng. Syst. Saf.* **2008**, *93*, 964–979. [CrossRef]

36. Oakley, J.E.; O'Hagan, A. Probabilistic sensitivity analysis of complex models: A Bayesian approach. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2004**, *66*, 751–769. [CrossRef]

37. Daneshkhah, A.; Bedford, T. Probabilistic sensitivity analysis of system availability using Gaussian processes. *Reliab. Eng. Syst. Saf.* **2013**, *112*, 82–93. [CrossRef]

38. Homma, T.; Saltelli, A. Importance measures in global sensitivity analysis of nonlinear models. *Reliab. Eng. Syst. Saf.* **1996**, *52*, 1–17. [CrossRef]

39. Daneshkhah, A.; Hosseinian-Far, A.; Chatrabgoun, O. Sustainable maintenance strategy under uncertainty in the lifetime distribution of deteriorating assets. In *Strategic Engineering for Cloud Computing and Big Data Analytics*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 29–50.

40. Bedford, T.; Wilson, K.J.; Daneshkhah, A. Assessing parameter uncertainty on coupled models using minimum information methods. *Reliab. Eng. Syst. Saf.* **2014**, *125*, 3–12. [CrossRef]

41. McKay, M.D.; Beckman, R.J.; Conover, W.J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics* **1979**, *21*, 239–245. [CrossRef]

42. Keramat, M.; Kielbasa, R. Latin hypercube sampling Monte Carlo estimation of average quality index for integrated circuits. *Analog Integr. Circuits Signal Process.* **1997**, *14*, 131–142. [CrossRef]

43. van der Horst, R.; Hogema, J. Time-to-Collision and Collision Avoidance Systems. In Proceedings of the 6th ICTCT Workshop, ICTCT: Salzburg, Austria, 1993; pp. 1–12.

44. Bryan, B.; Nichol, R.C.; Genovese, C.R.; Schneider, J.; Miller, C.J.; Wasserman, L. Active Learning For Identifying Function Threshold Boundaries. In *Advances in Neural Information Processing Systems 18*; Weiss, Y., Schölkopf, B., Platt, J.C., Eds.; MIT Press: Cambridge, MA, USA, 2006; pp. 163–170.

45. Daneshkhah, A.; Bedford, T. Sensitivity analysis of a reliability system using gaussian processes. In *Advances in Mathematical Modeling for Reliability*; IOS Press: Amsterdam, The Netherlands, 2008; p. 46.