Downloaded from orbit.dtu.dk on: Mar 02, 2026



DTU Library

Active learning metamodeling for policy analysis: Application to an emergency medical service simulator

Antunes, Francisco; Amorim, M.; Pereira, Francisco Camara; Ribeiro, Bernardete

Published in:
Simulation Notes Europe

Link to article, DOI:
10.1016/j.simpat.2019.101947

Publication date:
2019

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Antunes, F., Amorim, M., Pereira, F. C., & Ribeiro, B. (2019). Active learning metamodeling for policy analysis: Application to an emergency medical service simulator. *Simulation Notes Europe*, *97*, 101947. https://doi.org/10.1016/j.simpat.2019.101947

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Active learning metamodeling for policy analysis: application to an Emergency Medical Service simulator

Francisco Antunes
University of Coimbra, Faculty of Sciences and Technology, Portugal
fnibau@uc.pt
Marco Amorim
University of Porto, Faculty of Engineering, Portugal
University of Stuttgart, Institute of Human Factors and Technology Management IAT, Germany
marco.amorim@iao.fraunhofer.de
Francisco Pereira
Technical University of Denmark, DTU Management, Denmark
camara@dtu.dk
Bernardete Ribeiro
University of Coimbra, Faculty of Sciences and Technology, Portugal
bribeiro@dei.uc.pt

Abstract

Simulation approaches constitute a well-established tool to model, understand and predict the behavior of transportation systems, and ultimately to assess the performance of transportation policies. Due to their multidimensionality nature, such systems are not often approachable through conventional analytic methods, making simulation modeling the only reliable tool of study. Nevertheless, simulation models can turn out to be computationally expensive when embedded with enough detail. An immediate answer to this shortcoming is the use of simulation metamodels that are designed to approximate the simulators' results.

In this work, the authors propose a metamodeling approach based on active learning that seeks to improve the exploration process of the simulation input space and the associated output behavior. A Gaussian Process (GP) is used as a metamodel to approximate the function inherently defined by the simulation model itself. The GPs are able to nicely handle the uncertainty associated with their predictions, which eventually can be improved with active learning through simulation runs. This provides a practical and efficient way to analyze the simulator's behavior and therefore to assess the performance of policies regarding the underlying real-world systems and services.

The authors illustrate the proposed methodology using an Emergency Medical Service (EMS) simulator. Two outputs are analyzed and compared, namely, the survival rate and response time averages. The medical emergency response time recommendation of eight minutes is explored as well its relation with the survival rate. The results show that this methodology is able to identify regions in the simulation input space that might affect the performance and success of medical policies with regards to emergency vehicle dispatching services.

Keywords: Active Learning, Simulation Metamodeling, Gaussian Processes, Kriging, Emergency Medical Service.

1. Introduction

Real-world transportation systems are characterized by their overwhelming complexity, not only due to the multitude of variables involved but also to their corresponding interrelationships. In order to model, understand, and then predict the behavior of such systems, and to assess their performances, simulation modeling is often the only reliable tool available. Specially due to their high dynamism and dimentionality, as well as their intrinsic stochastic nature, these systems can be hardly described or studied analytically [1].

Simulation models are virtual representations of the reality, often in a sufficiently simplified form, that are considered as experimental and virtual environments to test different system designs, and therefore to understand the impact of certain policies and interventions [2]. Due do their exploratory nature, simulation tools prove to be highly advantagous for policy analysis [3, 4]. However, simulation models can become computationally expensive to run, exhibiting prohibitive runtimes along with great workloads. To address this shortcoming, simulation metamodels are usually employed to approximate the simulation model itself and thus the function inherently defined by it.

Along with the consideration of simulation metamodels with the objective to decrease the burden of conducting expensive computer experiments, a machine learning paradigm, called active learning [5], can be also taken into account. Active learning is particularly useful in situations in which data is difficult to obtain, as it aims to enhance the models' predictive performance with fewer training data points. Therefore, both active learning and simulation metamodeling approaches can be jointly used, on the one hand, to minimize the need for simulation runs, and on the other hand to achieve and maintain a reasonably good approximation of the simulation model. The Gaussian Process (GP) framework [6], also known as Kriging, is a well-known modeling tool, widely applied in numerous research and application fields. Due to its Bayesian formalism and highly non-linear properties, it constitutes an excellent option for designing active learning strategies based on simulation metamodeling settings.

This work presents an active learning metamodeling methodology to address the problem of policy analysis within the context of computationally expensive simulation models. A GP is considered to approximate the simulator's behavior and then used to explore the simulation input space. The fitness of the GP is then iteratively improved with active learning via simulation runs, by decreasing the associated variance of the given predictions over the simulation input region in study. This provides an alternative way to perform policy analysis while avoiding, at the same time, a potential large number of simulation runs.

The presented methodology is tested using an Emergency Medical Service (EMS) simulator. Two simulation outputs are studied, namely, the average survival rate and the average response time. Then, motivated by [7], the medical emergency response time recommendation of 8 minutes (480s) is analyzed, as well as its relation with the survival rate. The results show that this methodology is able to identify regions within the simulation input space that directly influence the successful application of medical policies with regards to emergency vehicle dispatching services.

2. Literature Review

The development and application of simulation metamodels [8, 9, 10, 11] can be traced back to the 70's [12]. Their main purpose is to serve as parcimounious approximations for simulation models, so that expensive simulation runs can be avoided. Specific features such as mathematical simplicity, speed and interpretability are usually attributed to metamodels. Consequently, the use of metamodels within simulation analysis provides an additional level of understanding of the underlying system, as well as of the relationships between the system input and output variables, while maintaining a computationally simple and economic approach to the problem [13]. Simulation metamodels are often described by computationally fast and easy-to-implement functions that approximate the true but unknown function intrinsically defined by the simulation model itself. It is common that many of these inputs are shared with those of the simulation model, although it is not entirely necessary, as [14] points out.

The earliest applications of metamodeling involved simple queuing simulation systems [8] and used linear regression metamodels. At the time, as the computational resources were evidently low and scarce, when compared to the current days, the use of metamodels emerged as a practical tool to overcome the difficulties posed by even the simplest simulation systems. Although the technology evolved ever since, essentially providing more computational power, it also leveraged the demand and the opportunities for modeling

increasingly complex systems. Both computational power and system models complexities increased in the same direction and with a similar intensity, therefore explaining the need for the use of metamodels nowadays. This shows that the application of metamodeling techniques is not only exclusively related with the available computing power but also to the demand of highly detailed models, which are eventually conditioned by it. Notice that with more realistic models usually comes large or even prohibitive simulation running times, which cannot be used for real-time applications or practical simulation output behavior analysis. In these situations, the use of metamodels as an auxiliary tool is always preferred [14] and can eventually be used to explore the simulation behavior in a less expensive manner.

The GP framework is a well-known modeling approach widely used as a simulation metamodel [15, 16, 17, 18] and, when applied to regression problems, it is also called Kriging [19]. Initially, GP-based metamodels had only been used in deterministic simulations. Nevertheless, [20] started using GPs for random simulations, showing that they had great potential in this kind of applications too. Since then, several similar approaches arised [21, 22, 23]. [24] generally discusses the use of GPs as a reliable metamodel for the design and analysis of computer experiments. [15] constructs two GP representations and develops an experimental design to extend the metamodel framework to account for heteroscedasticity in the simulation output. In [25], the authors developed a method based on application-driven sequential designs using a GP metamodel. [26] shows that the GP, as a metamodeling technique, is able to accurately approximate the complex functions often associated with non-linear and non-monotonic probabilistic design space. In [27] the authors introduce several diagnosis metrics to validate the GP framework as a simulation metamodel.

In the context where simulation data can be difficult to obtain (e.g. computationally expensive simulation experiments) in a systematic way, active learning can prove to be a powerful learning paradigm to enhance the application of simulation metamodels. As a subfield of supervised machine learning, active learning is essentially an iterative sampling strategy that allows any algorithm designed upon it to actively select the data points from which it learns. Thus, instead of selecting a large number of random points, the algorithm searches for the most informative ones, in a sequential way, so that both the model training efficiency and its prediction performance are improved with as few training data points as possible [5].

According to [28], an arbitrary active learning strategy encompasses five essential elements enclosed in the following quintuple

$$(\mathcal{L}, \mathcal{U}, \mathcal{M}, \mathcal{O}, \mathcal{Q}).$$

First, $\mathcal{L}$ is the labeled training data set. Then, the set of the unlabeled data points is represented by $\mathcal{U}$. Generally, $\#\mathcal{U} >> \#\mathcal{L}$, i.e., the number of unlabeled data points is much higher than that of the labeled ones. The machine learning model is represented by $\mathcal{M}$. Depending on the nature of the problem being modeled, it can be a classification or a regression model, which in turn affects the nature of the labels in $\mathcal{L}$ of being discrete or continuous, respectively. $\mathcal{O}$ denotes the oracle whose role is to provide labeled instances from the underlying process in study. Finally, $\mathcal{Q}$ is the query function that encodes the strategies and criteria for finding and selecting the most informative instances of $\mathcal{U}$ to be added to $\mathcal{L}$. There have been many query strategies formulations which essentially translate into different perspectives to approach the problem in question. As mentioned by [5], depending on both the nature of the problem and the model being used, several query frameworks can be adopted, such as uncertainty sampling [29], query-by-committee [30], expected model change [31] and error reduction [32], variance reduction [33] or density-weighted methods [34].

Closely related to the concept of the query function is the definition of a stopping rule. Due to its iterative nature, techniques based on active learning must be stopped at a certain time, either manually by the user or automatically by a stopping criterion. In both cases, the stopping rule must take into account the trade-off between the overall prediction performance and generalization capacity of the machine learning model and the associated costs of acquiring new labeled data. Such costs could be, for example, the computational workload or running time associated to any simulation model.

Active learning has been used in a variety of research fields. However, this paper focus on those applications involving simulation metamodels, particularly the GPs, where the simulation model plays the role of oracle. As mentioned erlier, by providing a fully non-linear Bayesian approach, the GPs allow for an intuitive way to develop active learning algorithms due to their ability to explicitly model the uncertainty present in the data. As high predictive variance can be associated with high degree of uncertainty, both the posterior mean and variance, provided by the GPs for any given data point, can be directly used to explore the simulation input space and thereby guide the search for the most informative data points. Such active

learning schemes involving GPs are usually associated to exploration-exploitation problems, often studied in Bayesian Optimization contexts [35, 36].

According to [37], while a pure exploration approach aims to learn an unknown function of interest as accurately and fast as possible, in an exploration-exploitation setting the goal is to find the input that maximizes the output of an unknown function, in a equally fast manner. Under an active learning strategy, Bayesian optimization uses a reward function to more efficiently select the next unlabeled data point, according to an automatic trade-off between the domain regions where the objective function is very uncertain (exploration) and those of where the same function is expected to be attain high value (exploitation), as seen in [38].

Within the transportation literature, the application of simulation metamodels is still rare and relatively recent. The available research can be broadly categorized into traffic prediction and optimization of networks, as mentioned by [39]. Some of such works include, for example, metamodeling for mesoscopic simulation [40, 41] and for travel behavior and dynamic traffic optimization [42]. In [43] the authors proposed a restricted batch-mode active learning strategy to address the problem of efficiency of the metamodeling process using a simple traffic simulation and a Demand-Responsive Transportation (DRT) system simulator.

3. Methodological Approach

In this work, a straightforward pool-based active learning strategy is adopted. The experimental design is depicted in Figure 1. Here the unlabeled data set $\mathcal{U}$ is entirely available for querying and represents the simulation input region in which the simulator's behavior is analyzed. The pool of labeled instances $\mathcal{L}$ is comprised of simulation results, i.e., input-output $(n+1)$-dimensional tuples. The machine learning model $\mathcal{M}$ is a GP, whereas the query function $\mathcal{Q}$ is based on the analysis of the predictive variance provided by the latter at each point in $\mathcal{U}$.

The general idea within this experimental setting is to assume that the functional relationship between the simulation input vector $\mathbf{x}$ and the output y is described by a GP. After the GP is fitted to $\mathcal{L}$, the provided conditional distribution is used to predict the output values (or labels) for each unlabeled data comprised in $\mathcal{U}$. This makes it possible to bypass simulation runs and to approximate the simulator behavioral structure, making the exploration process more efficient. The predictive variance is used as a measure of fitness, which should decrease as the iterative process evolves. Eventually, the final trained GP model is used as a simulation metamodel to explore the behavior of the simulator and then to conduct policy analysis and assessment. With respect to the implementation, the Gaussian Processes for Machine Learning (GPML) tool is used. It constitutes a free, open source and well-stablished Matlab package provided and maintained by [6]. More details regarding the GP framework, as well as the adopted methodology are presented in the following.

3.1. Gaussian Processes

Despite being quite an old topic in the field of probability and statistics, the application of Gaussian Processes within machine learning tasks has emerged in the past decade. As described in [6], a Gaussian Process (GP) is a stochastic process from which each finite set of variables follows a multivariate Gaussian distribution. It is usually simply denoted as $\mathcal{GP}(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}'))$, where $m_f(\mathbf{x})$ and $k_f(\mathbf{x}, \mathbf{x}')$ are respectively a mean and a covariance function, with $\mathbf{x}$ and $\mathbf{x}'$ being two different input data points. Consequently, a GP is sufficiently characterized by these two functions. For sake of simplicity, the mean function can be set to zero or any other constant value. This does not correspond to a significant restriction, as the mean of the process is not forced to be zero. On the other hand, the covariance function is of utmost importance for the modeling process, as it crucially encodes the notion of similarity between two given data points

From a regression point-of-view, and as an intrinsic Bayesian approach, the GP modeling assumes a prior over functions, i.e., in the functional dependency established by $y = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$, it follows that $f(\mathbf{x}) \sim \mathcal{GP}(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}'))$. Most of the common mean and covariance functions typically have several free parameters, also called hyper-parameters of the GP, which can be optimized by marginal likelihood maximization subjected to the training data. The conditional distribution of a new unlabeled data point $\mathbf{x}_*$ is then given by $f_*|X, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(k_{f*}^\top [K_y]^{-1}\mathbf{y}, k_{f**} - k_{f*}^\top [K_y]^{-1} k_{f*})$, with $k_{f*} = k_f(X, \mathbf{x}_*)$, $k_{f**} = k_f(\mathbf{x}_*, \mathbf{x}_*)$, K_y the covariance matrix, X the design matrix and $\mathbf{y}$ the set of training target values. Note that the prediction provided by the GP is a fully-defined Gaussian distribution, rather than a single

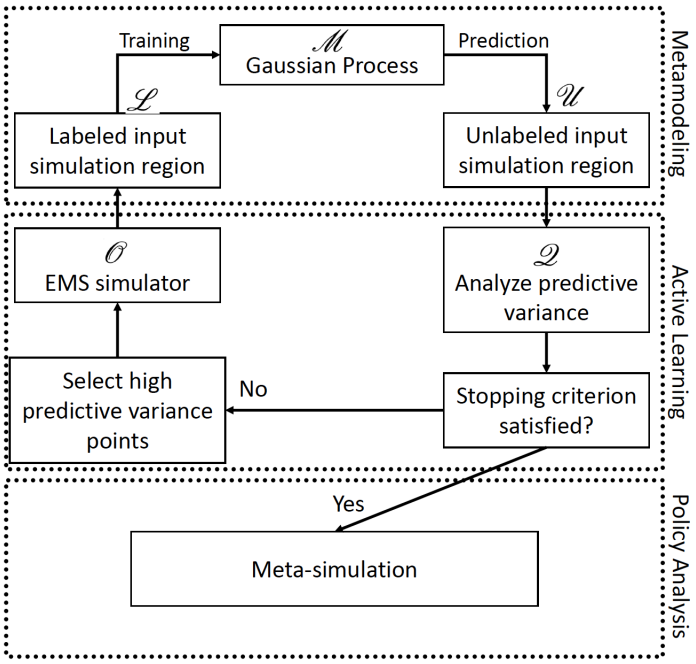


Figure 1: Active learning metamodeling experimental design. It is divided into three main blocks or steps. First, the simulation metamodeling comprises the approximation of the simulation model using a GP. Then, an active learning strategy is used to iteratively increase the fitting quality of the GP, in this case, by decreasing its total predictive variance across the unlabeled input simulation region. Finally, when the stopping criterion is verified, policy analysis by means of the provided meta-simulator is conducted.

point-wise estimate. This Bayesian property allows the GP to encapsulate the uncertainty not only of its owns predictions, but also that of the underlying signal being modeled. In turn, this uncertainty can be intuitively used to as an information criterion to design active learning strategies.

In this work, the Squared Exponential, or Radial Basis Function, given by

$$k_f(\mathbf{x}, \mathbf{x}') = \sigma^2 exp(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2l^2})$$

is used. Here, σ^2 and l^2 are respectively the signal variance and the characteristic length-scale. For the GP mean, a constant function dependent on the average of output values of the training data set was used, $m_f(x) = k$.

3.2. Metamodelling Strategy

This work follows a modeling approach similar to that presented in [43], and it is summarily depicted in Figure 2 in the form of pseudo-algorithm.

The algorithm starts with the initialization data set which is comprised of the first simulation input-output results. A GP is trained in this data set ($\mathcal{L}$), via likelihood maximization, and both the predictive mean and variance are obtained for each point in the input simulation region of interest ($\mathcal{U}$). Then, the algorithm selects the top five data points with the highest predictive variance values, top_5. This is meant to force an active selection of points, rather than selecting them at random, as the hypotheses is that the predicted values with higher predictive variance contain more information regarding the underlying process than those with lower variance. After this selection, the algorithm requests the simulator for the real output values (labels) corresponding to the unlabeled points in top_5. This is also called a batch-mode active learning. It is particularly useful as it allows for the query processes to be parallelized. In this case, the points are being selected in batches of five, which are, eventually, added to $\mathcal{L}$. The process repeats until the stopping criterion is satisfied.

Inputs: $\alpha \in (0,1), \mathcal{L}, \mathcal{U}$.

While $CTV \geq (1-\alpha)ITV$ **do**

1: Train the GP in $\mathcal{L}$ and predict the output values for each point in $\mathcal{U}$, $k_{f*}^{\top}[K_y]^{-1}\mathbf{y}$, and obtain their corresponding predictive variances, $k_{f**} - k_{f*}^{\top}[K_y]^{-1}k_{f*}$.
2: Determine the top 5 highest predictive variance points in $\mathcal{U}$, top_5
3: By simulation requests, obtain the true values for top_5, $\mathcal{L}_{+}^{top}$ as the new labeled set.
4: Expand the labeled pool: $\mathcal{L} = \mathcal{L} \cup \mathcal{L}_{+}^{top_5}$.
5: Update CTV.

end

Output: GP trained in $\mathcal{L}$ and associated predictions over $\mathcal{U}$.

Figure 2: Active learning metamodeling pseudo-algorithm.

The stopping criterion is defined by the relation between the Initial Total Variance (ITV) and the Current Total Variance (CTV). Whereas the former is computed in the first iteration, the latter represents the current amount of variance summed over $\mathcal{U}$ and it is updated in each of the following iterations. As α lies in $(0,1)$, the process stops in CTV reaches a fraction of ITV. In this work, $\alpha = 0.1$, which means that the stopping rule is satisfied when CTV is less than $(1-\alpha)\%$, or 90% of ITV. Note that in the first iteration, $CTV = ITV$ holds. However, as the process evolves, CTV has a tendency to decrease since the GP will inevitably acquire more training data points, and thus greater knowledge about the simulation function. It is also worth noticing that the number of iterations required to satisfy this criterion is directly dependent on the size of the initial training set as well as on the value of α itself. If the initial set is both sized-up and representative enough, then the GP may be able to generalize well for the unobserved points present in $\mathcal{U}$. On the other hand, if α is near 1, then the criterion will be easily satisfied in a few iterations. Therefore, the right trade-off between these two elements must be carefully taken into account.

3.3. EMS simulator

In Emergency Medical Service (EMS) planning, evaluating strategic and tactical decisions in the real-world systems, either via policies or operational solutions, is quite often physically unfeasible. Simulation and optimization approaches are thus preferable to mimic the systems' real conditions and to produce insightful assessments of decisions planned to be implemented. This is a synthetic process that is able to provide empirical evidences of how a certain choice or change might impact a certain targeted performance metric.

A simulation algorithm to numerically compute EMS solutions performance in terms of average response time and average survival is adapted from the work of [44], whose algorithm is resumed in Figures 3 and 4. This optimization model is used to assess the performance of station and vehicle locations in an urban area. It also provides a platform to measure the performance of dispatching policies and other tactical decisions. This work focuses on strategic and tactical decisions and their performance using two metrics: the classical average response time, and a survival metric that tries to capture victims' outcomes.

The simulation model uses an agent-based model controlled by a city agent that takes the role of the emergency medical service. It allocates and dispatches vehicles using a closest dispatching rule [45, 46, 47]. A network agent simulates traffic stochastic conditions and EMS vehicle movements by using nodes and arcs as an abstraction of the reality, pre-computing travel times for different periods of the day and the week using Google's Directions API. Events are agents that abstract EMS calls and are triggered according to a historical database and a stochastic location change, denoted by w_1.

The model runs in short fixed time steps and everytime a step is made an update process is triggered, as depicted in Figure4. This trigger initiates the city agent tasks. Firstly, the city agent contacts the event agent to check the status of the current events. The event agent updates its list and reports back to the city. If a new event exists, it will request the city for assistance. If all events status are up-to-date then the time will advance. At the same time, the city communicates with the network to keep track of the vehicles' location and status. The network reports such status to the city and also informs the event agent when a victim has been assisted (storing the vehicle arrival time in the database for performance assessment) and when the victims arrive at the hospital. The assisting time is a stochastic variable that assumes a gamma

Algorithm 1 Simulation algorithm

Definitions:

N = set of nodes n
n = node, where s = node of type station and h = node of type hospital
V = set of vehicles v_s
v_s = vehicle in station s
S = set of stations s
H = set of hospitals h
E = set of events e^t_n
e^t_n = emergency event occurring at node n during t
M = set of matrices M^t
M^p = matrix of real travel times for period p
T = total simulation time
t = time
$step$ = temporal resolution
$f()$ = programming function

While $t < T$:

1. **Update city( )** "set t and activate e^t_n."
2. **Update network( )** "interact through every v_s to travel one *step* and transfers it to destination nodes"
3. **Update events( )** "activates e^t_n and the vehicle dispatching algorithm"
   -Network calculates time travel from all stations
   -Network returns the shortest one
   -Vehicle dispatching algorithm runs
4. **Update vehicles job( )** "updates v_s status"
   -If v_s arrived to e^t_n , activate assisting timer
   -If assisting timer ends, request network to be processed to h
   -If v_s arrived to h, transfers v_s to s.
5. **Update results( )** "calculates the EMS performance at the current *step*"
6. $t = t +$ *step*

Figure 3: Pseudo-algorithm for the studied EMS simulation model.

distribution with a mean of 10 minutes and a standard deviation of 5 minutes, according to the observations from [7]. In case a new event is activated during the events update, a request for assistance is sent to the city, which, after receiving the network report status, decides on the response strategy by selecting the proper vehicle to be allocated to the new event.

The stochastic location change captures a possible randomness in the demand by re-allocating a medical emergency event e_p at node p to node p', with a probability of w_1, according to $e_{p'} = p(g(E_s), e_p, w_1, w_2)$. Here, $g(E_s)$ is a function that chooses a location from a set of possible locations weighted by the observed demand at period s, and $p(e_1, e_2, w_1, w_2)$ is a function that picks either e_1 or e_2 with probability w_1 and w_2, respectively, where $w_1 + w_2 = 1$ must hold.

On the other hand, traffic stochasticity is controlled by parameter ε. It introduces an error term in any observed travel time $r_{s,l,p}$, which measures the time it takes to travel from point l to point p during s, according to $r'_{s,l,p} = r_{s,l,p} + r_{s,l,p} \times \mathcal{N}(\mu, \frac{\varepsilon^2}{4})$. When the mean is set to zero and the standard deviation to $\varepsilon/2$, a confidence of 95% exists that the travel time has at most an error of $\varepsilon \in [0, 1]$.

4. Results

In this section, the presented methodology is applied to the EMS simulator developed in [44]. As previously seen, this simulator is designed in terms of three kinds of input dimensions, namely, the location change probability, traffic error and vehicle station locations. In terms of value ranges, the first two lie in the interval $[0, 1]$, whereas the locations assume discrete positive values, $[0, 1, 2, 3, ...)$, representing the number of vehicles allocated to each position. Figure 5 shows the location of these stations, 90 in total, scattered

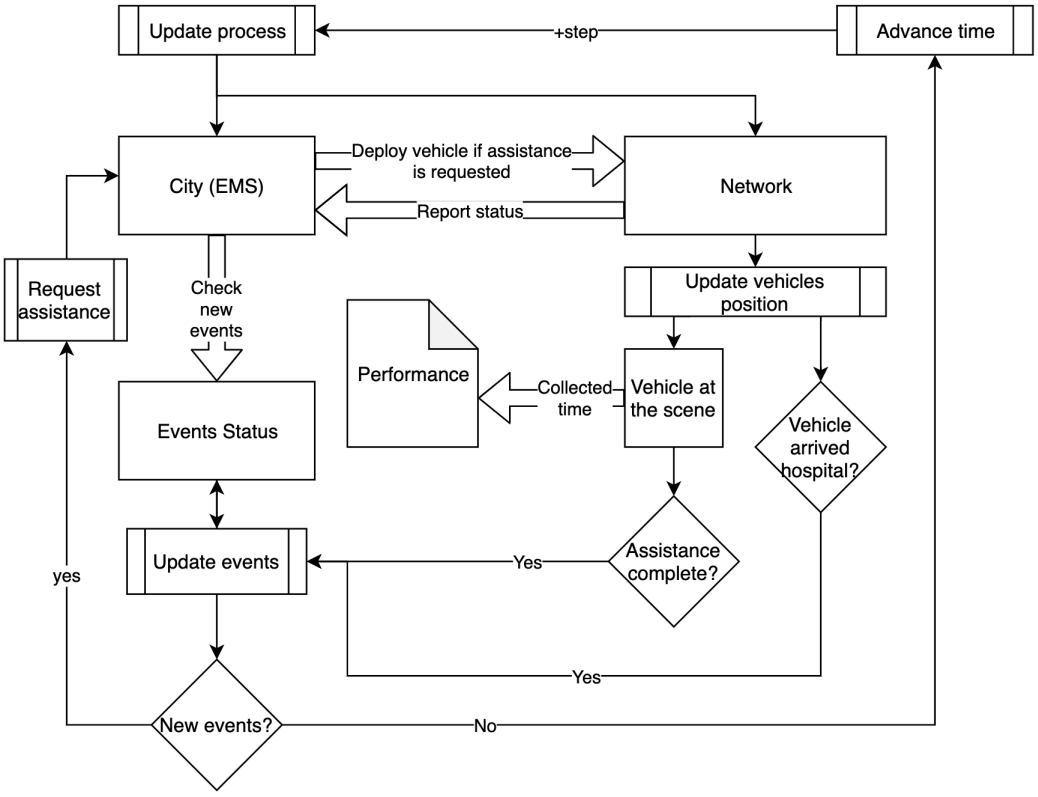


Figure 4: Flowchart describing the essential steps implemented in the studied EMS simulation model.

Table 1: A data sample showing the dimensional structure with 92 features and two output variables: $x_1 - x_2$ lie in the interval $(0, 1)$, $x_3 - x_{92}$ assume discrete values in $\{0, 1, 2, ...\}$, and $y_1 - y_2$ are a real-valued variables.

x_1	x_2	x_3	x_4	...	x_{90}	x_{91}	x_{92}	y_1	y_2
0.0445	0.2921	0	0	...	0	0	0	0.3547	379.0279
0.4923	0.7756	0	0	...	2	1	0	0.1953	521.3361
0.0708	0.3740	1	0	...	0	0	0	0.3800	331.0639
0.2199	0.6818	2	1	...	0	1	0	0.2252	515.0807

in the city of Porto, Portugal. Note that not all the depicted labels follow a sequential order. The last four locations are respectively labeled with 96, 108, 109 and 112. The considered simulation outputs are the average survival rate and the average response time.

In Table 1 a sample of the used data is presented. The dimension, or feature, denoted by x_1 is the location change probability, whereas x_2 is the traffic error. The locations and the associated number of vehicles are coded in features x_3 to x_{92}. With respect to the simulation outputs, y_1 and y_2 are the survival rate and response time, respectively. Due to space limitations, only the results for x_1, x_2 and x_3 are analyzed in this paper.

Each output was modeled independently, i.e., a different GP was used to approximate the simulation results. Figures 6-9 present the obtained results. Similar conclusions can be drawn for both simulation output variables. Notice that the panels (a) and (b) of Figures 6 and 8 do not depict any kind of time series, but rather a straightforward way to simply represent the prediction originated from high-dimensional data and their corresponding confidence intervals. Therefore, particularly in these representations there is no notion of neighborhood between observations, nor the sequential order of each observation is of any relevance.

In both cases, the active learning metamodeling process started with the GP training stage using 100

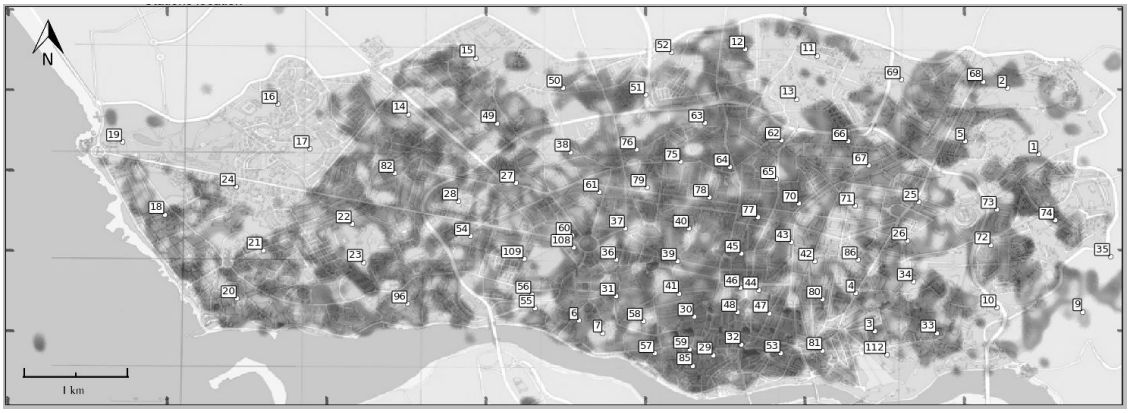


Figure 5: Locations of the 90 emergency vehicle stations in Porto, Portugal.

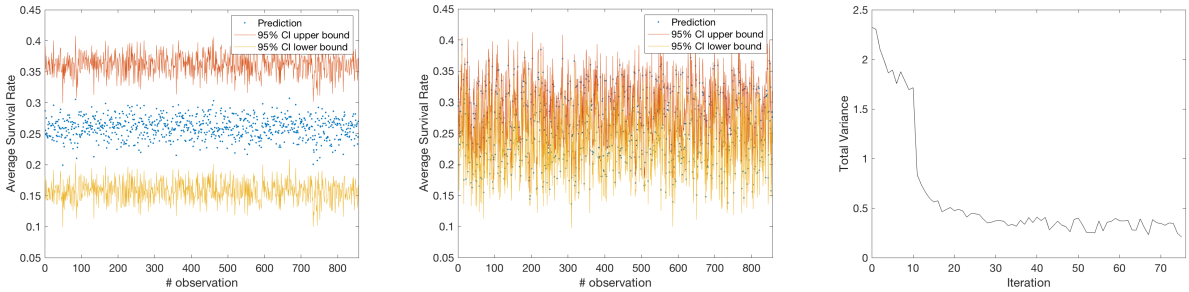
random simulation points. Then, the prediction was conducted over 856 points scattered along the simulation input space. Take, for instance, the case of y_2, the average response time. Figure 8(a) shows the first GP approximation along with a confidence interval of 95%. As expected for this first iteration, the uncertainty of the model, which is encoded into the variance of each prediction, is relatively high. It can be clearly observed that the GP is approximating the simulation output by values fairly around 450 (minutes). Moreover, these predictions are associated with large confident intervals, which is not desirable in real-world practice, especially for emergency-related operations decisions along which rapid responses ought to be undertaken. The width of these intervals can be viewed as a proxy for the uncertainty present not only within the data but also within the GP model itself. In the end, after 59 iterations, this variance decreased considerably, as depicted in Figures 8(b) and (c).

A similar behavior can be observed for y_1. However, contrary to the case of y_2, this time the active learning procedure required 16 extra iterations to achieve the same reduction of 90% of the total initial variance (see Figure 6(c)). It is worthwhile mentioning that, in both cases, the corresponding final GP approximations present a wider amplitude of predicted output values when compared to the results regarding the first iterations. In Figures 6(a) and 8(a), the GP is predicting inside narrow horizontal bands centered around 0.27 and 450, respectively, under a relatively large margin of uncertainty. Then, when the final iterations were reached, not only the associated predictive variance decreased, as mentioned before, but also the amplitude of the predicted values changed considerably. This translates into an improved prediction performance for both output variables, as the GP is now closer to the real underlying unknown function described by the simulation model.

In [7], an empirical study involving 3576 patients transported to a single Level I trauma center was conducted in order to assess the 8 minute guideline for ambulance response. The authors concluded that there was no significant difference in the survival rate, due to traumatic injuries, between the patients who were assisted within and above the established response time policy, respectively. In the same work, it is mentioned that the mortality odds ratio is of 0.81 for response times greater than 480s. Obviously, the odds for the patients' survival were of 0.19. Taking these two elements into account, a meta-simulation analysis was carried out in terms of the two outputs provided by the studied EMS simulator.

Figure 7(a) shows that the obtained survival rates averages were conditioned by both the traffic error and location change probability inputs. From traffic errors in the order of 60%, it is visible that emergency cases associated to lower rates of survival start to emerge. On the other hand, as the probability of location change increases, the rate of survival also decreases. Most of these low survival observations are concentrated in the upper right corner, slightly within $[0.5, 1.0] \times [0.6, 1.0]$. Similar conclusions can also be derived from the observation of Figures 7(b) and (c). Again, and this time with respect to the number of vehicles in location 1, which corresponds to variable x_3, most of the mortal occurrences are associated with higher values of location change probability and traffic error. This is particularly more evident for the latter simulation input.

Regarding the traffic error, and taking into account the earlier presented details of the used EMS simulator, the obtained results meet the initial expectations. This simulation input represents the error associated



(a) (b) (c)

Figure 6: Results for the average survival rate output: (a) depicts the initial GP predictions over 856 simulation points, along with the corresponding 95% bounds, whereas (b) the final GP approximations. 100 random input points were used as the initial training set. Panel (c) shows the required number of iterations to satisfy the stopping criterion with $\alpha = 0.1$. In each iteration, the top five predictive variance points were selected to be added to the training set.

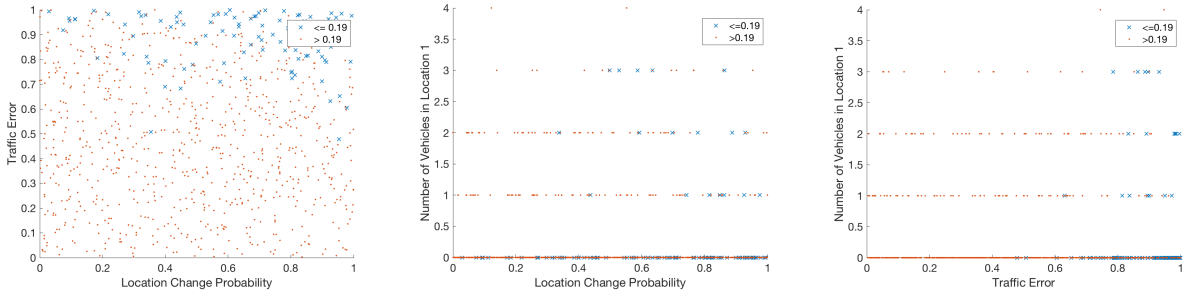
with the traffic prediction in comparison with the real traffic, immediately prior to the vehicles dispatching. Therefore, it makes sense that higher traffic errors may lead to inadequate operational decisions, resulting in higher response times and consequently in more fatal occurrences. With respect to the location change probability the results also confirm the intuition. Increasing this probability will lead to higher variability in the location of the life-threatening events or emergency calls, in comparison with historical data.

For the average response time, similar results were attained. In Figure 9(a), it can be concluded that the observations associated with response times greater than or equal to 480s are concentrated in $[0.0, 1.0] \times [0.6, 1.0]$. Contrary to what one would expect, not so many of these observations had led to survival rates below 0.19. This matches with the conclusions made by [7], i.e., that average response times greater then 8 minutes do not necessary lead to high mortality rates. Nevertheless, high response times combined with high probability of call location change, seem to lead to higher values of this output performance measure. In the end, it is true that when the traffic error increases, the delays in the vehicles arrivals increase accordingly.

It is important to take into account the kind of variation encoded into these two simulations inputs. Note that the traffic error does not have a direct implication on the vehicles' travel times. Instead, what is being measured by the EMS simulator is to which extent traffic errors lead to bad choices of dispatching the correct vehicles, both in terms of number and station locations. Given a certain emergency call, and ideally assuming no traffic congestion, the obvious vehicle to dispatch would be the closest one. However, when traffic congestion exists and, besides that, the real traffic information is only available with a certain degree of error, the dispatching solution is not so obvious. Moreover, the distance is no longer measured in space units, but rather in time ones, due to existence of traffic congestion, which in turns makes the problem more challenging. Take, for instance, the following example where station 1 is 10 minutes from the emergency event location (E), whereas station 2 is 20 minutes away from the same event. Given an error of $\pm 20\%$, this means that the emergency service operator may assume that stations 1 and 2 are, respectively, 12 and 16 minutes from E. In practice, specially in this kind of life-threatening situations, this represents a great error with potentially dramatic consequences. For this case, the decision result will be the same, i.e., a vehicle will be dispatched from location 1, which turns out to be the best decision. However, consider an alternate configuration, where both station 1 and 2 are, respectively, 10 and 11 minutes from E. Here, the same traffic error could lead the operator to take a poor decision based on the assumption that station 1 is distanced in 12 minutes and station 2 in 9 minutes, which, in reality, is not true. In such situation, this poor decision would be, for example, to send a vehicle from location 2, which is, in fact, 1 minute further than the other. Therefore, the same error can lead to different decisions and different outcomes.

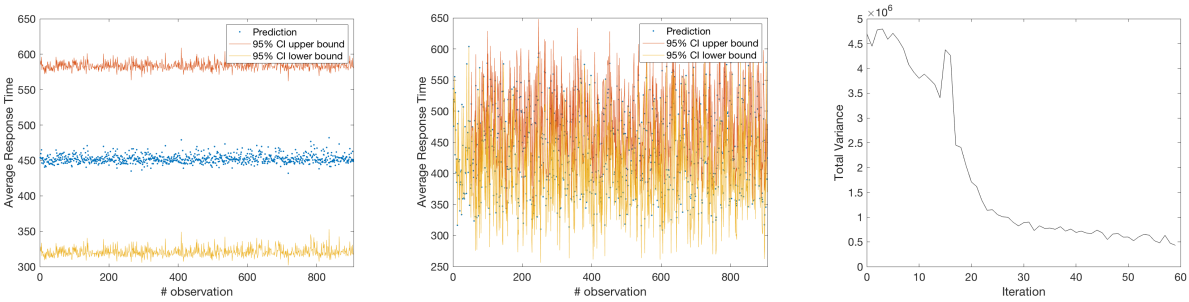
On the other hand, the probability change location is specially important to induce a stochastic behavior to the emergency calls' locations, so that the conclusions are not drawn exclusively from historical data. The simulation model must take into account the inherent dynamics of emergency events and should be able to respond accordingly.

Finally, in order to validate the generalization capacity of the obtained GP approximations for each simulation output, 30 random runs using 200 test points were conducted in a 10-fold cross-validation scheme.



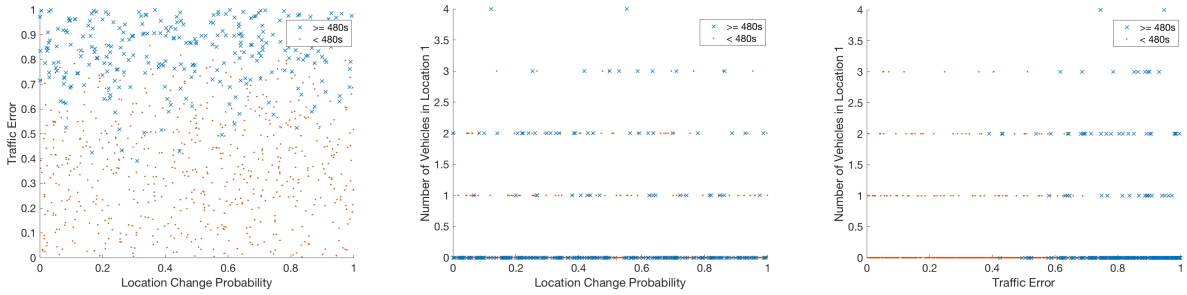
(a) (b) (c)

Figure 7: Comparison of the obtained predicted values for different input dimension with an average survival rate threshold of 0.19: (a) Location Change Probability versus Traffic Error, (b) Location Change Probability versus Location 1 and (c) Traffic Error versus Location 1.



(a) (b) (c)

Figure 8: Results for the average response time output: (a) depicts the initial GP predictions over 856 simulation points, along with the corresponding 95% bounds, whereas (b) the final GP approximations. 100 random input points were used as the initial training set. Panel (c) shows the required number of iterations to satisfy the stopping criterion with $\alpha = 0.1$. In each iteration, the top five predictive variance points were selected to be added to the training set.



(a) (b) (c)

Figure 9: Comparison of the obtained predicted values for different input dimension with an average response time threshold of 480 seconds (8 minutes): (a) Location Change Probability versus Traffic Error, (b) Location Change Probability versus Location 1 and (c) Traffic Error versus Location 1.

Table 2: Average results, and associated standard deviation values, obtained from 30 random computer runs using 200 test points and a 10-fold cross-validation scheme, for the two studied simulation outputs, average survival rate (y_1) and average response time (y_2).

Output	RMSE	MAE	RAE	RRSE	Corr.
y_1	0.0298 ± 0.0020	0.0237 ± 0.0017	0.4970 ± 0.0436	0.5287 ± 0.0397	0.8497 ± 0.0231
y_2	40.4023 ± 2.5809	32.1174 ± 1.9981	0.5335 ± 0.0353	0.5681 ± 0.0326	0.8238 ± 0.0215

These points were randomly sampled from the simulation input space. Note that, for each output variable, only the GP obtained in the last iteration was used for validation purposes. To evaluate the predicted performance of the these GPs, five well-known metrics were used, namely, the Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Root Squared Error (RRSE) and the Pearson's correlation coefficient (Corr.). The results are summarized in Table 2. These metrics were computed by comparing the predicted values given by the GP against the real known output values obtained from simulation sampling. The GP yields slightly better performance for the survival rate output (see RAE, RRSE and Corr. columns), despite requiring more iterations to satisfy the active learning stopping criterion (see Figures 6(c) and 9(c)). The standard deviation values provide a systematic assessment of the variability around the averaged results across the multiple cross-validation random runs. It can be concluded that, in both cases, the GP presents a good generalization and prediction performances.

5. Conclusion & Future Work

This paper presented a methodology based on active learning metamodeling to address the problem of exploring the behavior of simulators developed in the context of transportation simulation and policy analysis. Particularly, an Emergency Medical Service (EMS) was used for illustration and two important output thresholds were considered: 0.19 for the average survival rate and 8 minutes (480s) for the average response time.

The presented work shows that the proposed methodology is able to help in the identification of important regions of the simulation input space that have a direct impact in the performance or implementation of certain policies, while avoiding several simulation runs at the same time. Moreover, if the simulation input space proves to be sufficiently high-dimensional or if each simulation run shows prohibitive computational workload and runtimes, an exhaustive exploration process is virtually impossible. Therefore, the joint use of active learning strategies and simulation metamodels designed to identify policy-relevant regions within the such space, has great potential in practice, especially for decision making processes.

In the future, several research directions can be undertaken in order to extend and improve the presented work. Firstly, the methodology should improve in terms of graphical representation, although working with high-dimensional data constitutes a great challenge in that aspect. The graphical results (policy-relevant simulation input regions) should be provided in such a way that they make it easier for policies to be analyzed. Moreover, the presented methodology can also be improved through clustering techniques applied to the results so that such important input regions are more easily identifiable.

This work did not took into account any specific design for computer experiments, such as the well-known the Latin hypercube scheme. These designs provide sampling strategies that lead to a good and systematic understanding of the underlying properties of simulation models. This eventually improves the statistical significance as well as the prediction performance of simulation metamodels. Instead, pure random sampling was used. There are plans to apply such sampling schemes in the near future and use the current work as baseline benchmark.

In order to further validate its potential, this work should be replicated not only using other kinds of transport simulation models, but also considering more complex policy analysis, possibly taking into account combination of policies. Additionally, it would also be interesting to provide a certain degree of interactivity with the user. For instance, instead of letting the active learning independently decide which data points to request the simulator to label, the user could be able to actively engage in this process too. This could represent an interesting feature for decision making practitioners, as it would directly involve them and their expertise in the modeling process itself.

Further numerical experiments will be conducted in the future, especially concerning the tuning of the involved parameters, namely, the GP hyper-parameters, and the methodology-related parameters, i.e., α, size of the initial training set and number of active learning data points. This should improve both the fitting and generalization capacity of the obtained GP metamodels. Furthermore, sensitivity analysis over these parameters will also be conducted.

Due to its close relationship with simulation metamodeling, Bayesian optimization techniques will receive particular attention in future works. If the goal is to seek for those set of points that maximize (or minimize) the expensive and black-box function inherently defined by the simulation model, then such derivative-free optimization approaches should be considered.

Lastly, but not least, another important issue is the possible correlation between the simulation outputs, which was not considered in this work, as each output was modeled in a independent way. Multiple output regression, such as the multi-output Gaussian Process framework, is able to improve its prediction performance by considering relationships across the outputs. The integration of such models in the proposed methodology would also be an interesting line for future research.

Acknowledgments

The authors acknowledge the support of FCT (Portuguese national funding agency for science, research and technology) under the grants PD/BD/128047/2016 and PD/BD/52355/2013 during the development of this work. The research leading to these results has additionally received funding from the People Programme (Marie Curie Actions) of the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Individual Fellowship H2020-MSCA-IF-2016, ID number 745673.

The anonymous reviewers are also greatly acknowledged for their useful comments and suggestions which helped to improve this paper.

References

[1] A. M. Law, Simulation Modeling and Analysis, 5th Edition, McGraw-Hill Higher Education, 2015 (2015).

[2] M. C. Marengo, Urban simulation models: Contributions as analysis-methodology in a project of urban renewal, Current Urban Studies 2 (03) (2014) 298 (2014).

[3] S. C. Bankes, Exploratory modeling and the use of simulation for policy analysis, Tech. rep., RAND CORP SANTA MONICA CA (1992).

[4] S. Bankes, Exploratory modeling for policy analysis, Operations research 41 (3) (1993) 435–449 (1993).

[5] B. Settles, Active learning literature survey, Computer Sciences Technical Report 1648, University of Wisconsin–Madison (2010).

[6] C. E. Rasmussen, C. Williams, Gaussian processes for machine learning (Adaptive computation and machine learning), The MIT Press, 2005 (2005).

[7] P. T. Pons, V. J. Markovchick, Eight minutes or less: does the ambulance response time guideline impact trauma patient outcome?, The Journal of Emergency Medicine 23 (1) (2002) 43 – 48 (2002).

[8] J. P. Kleijnen, A comment on blanning's âĂIJmetamodel for sensitivity analysis: the regression metamodel in simulationâĂİ, Interfaces 5 (3) (1975) 21–23 (1975).

[9] J. P. Kleijnen, Regression metamodels for generalizing simulation results, IEEE transactions on systems, man, and cybernetics 9 (1979) 93–96 (1979).

[10] J. Kleijnen, Model behaviour: Regression metamodel summarization, Encyclopedia of Systems and Control 5 (1987) 3024–3030 (1987).

[11] L. W. Friedman, The simulation metamodel, Springer Science & Business Media, 2012 (2012).

[12] R. R. Barton, Simulation metamodels, in: Simulation Conference Proceedings, 1998. Winter, Vol. 1, IEEE, 1998, pp. 167–174 (1998).

[13] L. W. Friedman, I. Pressman, The metamodel in simulation analysis: Can it be trusted?, Journal of the Operational Research Society 39 (10) (1988) 939–948 (1988).

[14] J. P. Kleijnen, R. G. Sargent, A methodology for fitting and validating metamodels in simulation, European Journal of Operational Research 120 (1) (2000) 14–29 (2000).

[15] A. Boukouvalas, Emulation of random output simulators, Ph.D. thesis, Aston University (2010).

[16] T. Chen, K. Hadinoto, W. Yan, Y. Ma, Efficient meta-modelling of complex process simulations with time–space-dependent outputs, Computers & chemical engineering 35 (3) (2011) 502–509 (2011).

[17] S. Conti, A. O'Hagan, Bayesian emulation of complex multi-output and dynamic computer models, Journal of statistical planning and inference 140 (3) (2010) 640–651 (2010).

[18] J. P. Kleijnen, Kriging metamodeling in simulation: A review, European journal of operational research 192 (3) (2009) 707–716 (2009).

[19] J.-P. Chilès, N. Desassis, Fifty Years of Kriging, in Handbook of Mathematical Geosciences: Fifty Years of IAMG, Springer International Publishing, 2018, pp. 589–612 (2018).

[20] W. C. Van Beers, J. P. C. Kleijnen, Kriging for interpolation in random simulation, Journal of the Operational Research Society 54 (3) (2003) 255–262 (2003).

[21] J. P. Kleijnen, W. C. Van Beers, Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments, European Journal of Operational Research 165 (3) (2005) 826–834 (2005).

[22] A. Boukouvalas, D. Cornford, A. Singer, Managing uncertainty in complex stochastic models: Design and emulation of a rabies model, in: 6th St. Petersburg Workshop on Simulation, 2009, pp. 839–841 (2009).

[23] B. Ankenman, B. L. Nelson, J. Staum, Stochastic kriging for simulation metamodeling, Operations research 58 (2) (2010) 371–382 (2010).

[24] B. Jones, R. T. Johnson, Design and analysis for the gaussian process model, Quality and Reliability Engineering International 25 (5) (2009) 515–524 (2009).

[25] J. P. Kleijnen, W. C. Van Beers, Application-driven sequential designs for simulation experiments: Kriging metamodelling, Journal of the Operational Research Society 55 (8) (2004) 876–883 (2004).

[26] L. Wang, D. Beeson, S. Akkaram, G. Wiggs, Gaussian process meta-models for efficient probabilistic design in complex engineering design spaces, ASME Paper No. DETC2005-85406 (2005).

[27] L. S. Bastos, A. OâĂŹHagan, Diagnostics for gaussian process emulators, Technometrics 51 (4) (2009) 425–438 (2009).

[28] X. Wang, J. Zhai, Learning from Uncertainty, CRC Press, 2016 (2016).

[29] D. D. Lewis, W. A. Gale, A sequential algorithm for training text classifiers, in: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, Springer-Verlag New York, Inc., 1994, pp. 3–12 (1994).

[30] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: Proceedings of the fifth annual workshop on Computational learning theory, ACM, 1992, pp. 287–294 (1992).

[31] B. Settles, M. Craven, S. Ray, Multiple-instance active learning, in: Advances in neural information processing systems, 2008, pp. 1289–1296 (2008).

[32] N. Roy, A. McCallum, Toward optimal active learning through monte carlo estimation of error reduction, ICML, Williamstown (2001) 441–448 (2001).

[33] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural computation 4 (1) (1992) 1–58 (1992).

[34] B. Settles, M. Craven, An analysis of active learning strategies for sequence labeling tasks, in: Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, 2008, pp. 1070–1079 (2008).

[35] C. K. Ling, K. H. Low, P. Jaillet, Gaussian process planning with lipschitz continuous reward functions: Towards unifying bayesian optimization, active learning, and beyond., in: AAAI, 2016, pp. 1860–1866 (2016).

[36] D. R. Jones, M. Schonlau, W. J. Welch, Efficient global optimization of expensive black-box functions, Journal of Global optimization 13 (4) (1998) 455–492 (1998).

[37] E. Schulz, M. Speekenbrink, A. Krause, A tutorial on gaussian process regression with a focus on exploration-exploitation scenarios, bioRxiv (2017) 095190 (2017).

[38] E. Brochu, V. M. Cora, N. De Freitas, A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, arXiv preprint arXiv:1012.2599 (2010).

[39] W. Song, K. Han, Y. Wang, T. Friesz, E. del Castillo, Statistical metamodeling of dynamic network loading, Transportation Research Procedia 23 (2017) 263–282 (2017).

[40] B. Ciuffo, J. Casas, M. Montanino, J. Perarnau, V. Punzo, Gaussian process metamodels for sensitivity analysis of traffic simulation models: Case study of aimsun mesoscopic model, Transportation Research Record: Journal of the Transportation Research Board 2390 (2390) (2013) 87–98 (2013).

[41] X. Chen, Z. Zhu, X. He, L. Zhang, Surrogate-based optimization for solving a mixed integer network design problem, Transportation Research Record: Journal of the Transportation Research Board 2497 (2497) (2015) 124–134 (2015).

[42] L. Zhang, X. He, C. Xiong, Z. Zhu, et al., Bayesian stochastic kriging metamodel for active traffic management of corridors, in: IIE Annual Conference. Proceedings, Institute of Industrial and Systems Engineers (IISE), 2014, p. 1790 (2014).

[43] F. Antunes, B. Ribeiro, F. Pereira, R. Gomes, Efficient transport simulation with restricted batch-mode active learning, Transactions on Intelligent Transportation Systems PP (2018) 1–10 (2018).

[44] M. Amorim, S. Ferreira, A. Couto, Emergency medical service response: Analyzing vehicle dispatching rules, Transportation Research Record: Journal of the Transportation Research Board (2018) 0361198118781645 (2018).

[45] A. Haghani, S. Yang, Real-time emergency response fleet deployment: Concepts, systems, simulation & case studies, in: Dynamic fleet management, Springer, 2007, pp. 133–162 (2007).

[46] C. Jagtenberg, P. van den Berg, R. van der Mei, Benchmarking online dispatch algorithms for emergency medical services, European Journal of Operational Research 258 (2) (2017) 715 – 725 (2017).

[47] S. Yang, M. Hamedi, A. Haghani, Online dispatching and routing model for emergency vehicles with area coverage constraints, Transportation Research Record 1923 (1) (2005) 1–8 (2005).