DTU Library

Active learning metamodelling for survival rate analysis of simulated emergency medical systems

Antunes, Francisco; Amorim, Marco; Pereira, Francisco; Ribeiro, Bernardete

Published in:
Transportmetrica A: Transport Science

Link to article, DOI:
10.1080/23249935.2022.2046203

Publication date:
2024

Document Version
Peer reviewed version

FULL PAPER

Active Learning Metamodelling for Survival Rate Analysis of Simulated Emergency Medical Systems

Francisco Antunes[a b *], Marco Amorim[b], Francisco Pereira[c], and Bernardete Ribeiro[a]

[a]Technical University of Denmark, Kgs. Lyngby, Denmark;[b]Fraunhofer IAO, Fraunhofer Institute for Industrial Engineering IAO, Stuttgart, Germany;[c]University of Coimbra, Coimbra, Portugal.

ARTICLE HISTORY
Compiled October 26, 2021

ABSTRACT
Emergency Medical Services (EMS) constitute a crucial pillar of modern cities by providing urgent medical responses to citizens. As urbanised areas grow, adequate planning must be undertaken to ensure the health safeguard of people in need.

The study of EMS is often conducted via simulation, as the assessment of planning decisions is generally not feasible in the actual systems. However, simulation models can become computationally expensive to run. To address this issue, metamodels can be employed to approximate the input-output mappings underlying the former, being traditionally characterised by functional simplicity and computational speed.

In this work, a simulation metamodelling strategy supported on an active learning scheme is proposed to analyse the survival rate outcome of a simulated EMS. Through a series of training grids, the algorithm guides the exploration process towards simulation input regions whose output results match a specific survival rate of interest defined a priori. This provides a computationally efficient way of exploring the associated search space by channelling the computational effort to the most important input values combinations. Across two sets of experiments, we show that the presented approach can identify such relevant simulation input regions for different targeted survival rate thresholds while minimising the associated computational burden. Simultaneously, it is also shown that this approach is easily expandable to include multiple search values. The conclusions support the advantages of employing this kind of methodology in the EMS modelling field, where its application is still rather seldom and mostly unknown to the best of our knowledge.

KEYWORDS
Active Learning; Simulation Metamodelling; Gaussian Processes; Emergency Medical Service; Emergency Response

1. Introduction

In 2018, more than half of the world's population was estimated to be living in cities or urban areas. The next decade's projection is that this proportion reaches approximately 60% and that one-third of the people are likely to be settled in cities with five hundred thousand citizens or more (United Nations, 2018, 2020).

The exponential growth of the urbanised areas will generate increased pressure on the existing urban infrastructures and the need to plan for new ones (Rodrigue,

*CONTACT Francisco Antunes, Email: franant@dtu.dk

2020). The growing volumes of passengers will inevitably require more and improved transport systems services (Belokurov et al., 2020; Enoch et al., 2020; Ferreira et al., 2020; Zhang et al., 2020). Furthermore, due to the increasing international trade supported on an interconnected global economy, the impacts of this growth are not set to be locally restricted to the cities and their vicinities. As a consequence, the need for commercial trade plays a fundamental role in the increasing demand for freight transport (Archetti and Peirano, 2020; Dulebenets, 2020; Heinold, 2020; Pasha et al., 2020; Rosano et al., 2018). Despite the unprecedented social and economic shutdowns worldwide due to the COVID19 pandemic during 2020, the world economy is expected to expand around 5.6% by the end of 2021 (World Bank, 2021). In any case, if not taken into proportionate consideration, the aforementioned multifaceted growth can lead to undesirable and even irreversible transformations. Nevertheless, the potentially negative consequences of these changes, including environmental and health concerns (Nieuwenhuijsen, 2020), can be minimised and handled if proper technical and policy decisions are taken in the present, with a strategic planning view for the inevitable steady urban growth (Levy, 2016; Marengo, 2014; Martine et al., 2007; Weber and Puissant, 2003).

As the population density increases within urban settlements, so does too the demand for mobility. Transportation systems play a particularly vital role: they are the backbone that supports sophisticated social and economic advancements. A healthy economic activity inevitably requires the movement of people, goods, and services (Moore and Pulidindi, 2013). Thus, it is not a surprise that urbanisation generally follows the evolution of transportation (Rodrigue, 2020), and vice-versa (Hensher et al., 2004), within a positive feedback loop. Moreover, with this intense urbanisation and agglomeration of people, the concerns for health and safety conditions increase accordingly (Vlahov and Galea, 2002). To this end, Emergency Medical Services (EMS) constitute a crucial pillar of today's cities by providing urgent medical responses to their citizens, particularly those in serious life-threatening situations. In (Bélanger et al., 2019), a review of the last ten years of EMS research is provided, focusing on the current state of the art modelling tools and projecting future trends whereas Andersson et al. (2020) presents three concrete managerial cases of EMS in the scope of strategic decision support systems. On the other hand, Poulton et al. (2019) addresses the problem of optimising ambulance routing to decrease response times to life-threatening events under blue light conditions.

EMS performance is highly dependent on the characteristics and idiosyncrasies of the urban environment in which they are integrated and on the information available to the service operator. Especially in highly populated areas, such systems' emergency response is inevitably constrained by multidimensional urban dynamics. Heavy traffic and population movements have a particularly relevant and direct impact on the EMS's outcome (Amorim et al., 2019b). To address the potential hindrances in implementing and deploying emergency services, strategic, tactical, and operational planning must be undertaken to ensure the proper emergency response of people in need of medical assistance.

The study of EMSs is often conducted via simulation-based modelling, as the assessment of planning decisions and policy solutions is generally not feasible in the actual real-world systems. However, despite their obvious advantages, simulation models can become computationally expensive to run whenever a high degree of realism and detail is required (Law, 2015). To address this shortcoming, simulation metamodels (Friedman, 2012; Gramacy, 2020) can be used to approximate the simulation results. Moreover, as these models are still explicitly dependent, to a reasonable extent, on the data generated by the underlying simulators they are designed to mimic, their performance can be further improved with active learning (Settles, 2010; Wang and Zhai, 2016). As a learning paradigm within machine learning, active learning aims to attain high prediction performance with as few data as possible. This economical approach to model training is achieved by providing the subjacent algorithm to choose the points from which it learns actively.

In this work, by combining elements of both simulation metamodelling and active learning, an integrated methodology is proposed to study the survival rate outcome of an EMS simulator by exploring its input space more efficiently. Employing a Gaussian Process (GP) as a metamodel, the algorithm guides the exploration process in the direction of simulation input regions whose output results match a specific survival rate threshold defined in advance by the user. This is achieved by a series of training grids and sub-grids that get iteratively closer to the input regions of interest. The results show that this approach can identify such relevant simulation input regions while minimising the computational workload simultaneously, turning the exploration process more efficient. We also show that the approach is trivially adaptable to incorporate different output search values.

From the obtained results, we draw several practical implications for the EMS modelling field and possible directions for improvements in future works. The conclusions support the advantages of employing this kind of integrated active learning metamodelling strategies in the design, study and evaluation of simulated EMS. To the best of our knowledge, their application is still rather seldom and mostly unknown by the field researchers and practitioners.

The remainder of the paper is organised as follows. The next section summarily reviews the main ideas, concepts and tools supporting the presented methodology and several related applications. Section 3 present the proposed methodology, followed by Section 4 where we detail the experimental setting and subsequently explore the results. Finally, Sections 5 and 6 draw several immediate implications and future lines of improvement and applications within the research field of EMS.

2. Background

By definition, simulation metamodels (Friedman, 2012), also known as emulators (McClarren, 2018), surrogates (Gramacy, 2020), response surfaces (Box et al., 1987), or sometimes more broadly as auxiliary models, aim at approximating the input-output functional relationships inherently defined by simulation models. They are often employed to approximate and explore simulation results, thereby mimicking the behaviour of the simulator itself (Kleijnen, 1987, 1979), particularly within experimental settings characterised by computationally expensive simulation runs.

To adequately serve their purpose, simulation metamodels should be defined as fast, tractable and intelligible formulas whose both input and output domains generally match that of the original simulation model. Kleijnen and Sargent (2000) mentions four main goals for simulation metamodelling, namely, real-world problem understanding, output prediction, optimisation, and verification/validation. In this work, we mostly focus on the first two objectives. Consequently, it is assumed that the simulation model in use is perfectly calibrated and correctly describes the real-world situation under study.

Song et al. (2017) summarise the metamodeling methodology in three major steps, namely, 1) definition of the experimental design, 2) metamodel specification, and 3) metamodel learning/fitting. The first step consists of strategically sampling the input space to generate a data set for the metamodel training. This step is shared with the standard simulation procedures discussed in the previous section. Then, steps 2 and 3 are conducted sequentially. While the former involves selecting a family of functional forms for the metamodel, the latter regards the fitting procedure of the selected metamodel to the data set obtained from step 1. For a more in-depth description of the metamodeling approach and unfolding in ten detailed steps suggestions, please refer to (Kleijnen and Sargent, 2000).

As already mentioned, simulation metamodels are essentially functions that are employed in such a way as to mimic the underlying simulation input-output mappings, which are often intractable or unknown. This approximation is conducted by fitting these functions to pre-generated sets of simulation results. As expected, metamodels still rely on data produced by the underlying simulation models, which can be viewed as a modelling bottleneck, particularly in the context of expensive simulation models or when intensive and systematic input space exploration is required. Additionally, it is no surprise that simulation metamodels also suffer from the popular problem deemed as "curse of dimensionality", especially in the context of large-scale modelling settings, as pointed out by Wang and Shan (2006). Hence, depending on the problem's nature and for the sake of tractability and interpretability ease, many applications restrict the metamodel fitting procedure to a particular and bounded subset of a much larger simulation input space and may not even consider all the input variables at once. This bounded sub-set is called the domain of applicability, or experimental region, encompassing the definition of the input domain in which the metamodel should be a valid approximation for the underlying model and real-world problem under study (Kleijnen and Sargent, 2000).

Within the previously mentioned kind of experimental scenarios where labelled (simulation) data is hard to obtain, active learning (Li and Sethi, 2006; Settles, 2012; Wang and Zhai, 2016) emerges as an opportune learning paradigm and modelling approach aiming at increasing a model's predictive performance with as few training points as possible. This is essentially accomplished by designing algorithms with the ability to actively choose the training points from which they learn iteratively, being ultimately characterised by the sequential selection of the most informative points that simultaneously boost the model's training efficiency and predictive performance. Besides the learning model itself, the entire process is also guided by a label provided called the oracle, whose task is to provide on-demand labelled instances which are subsequently incorporated into the expanding training set. In the context of simulation metamodelling, the oracle is represented by the simulator corresponding to the computer implementation of the underlying simulation model. Finally, the iterative process cyclically repeats itself across the alternating phases of model fitting, label acquisition, and training set expansion until a certain stopping criterion is satisfied.

The ability to measure the informativeness of a given unlabelled data point or to quantify model uncertainty is a core concept subjacent to any active learning system. Essentially, the idea is that seeking the most informative points and exploring high uncertainty regions of the feature space reduces data redundancy and boosts model training efficiency, besides helping in attaining a good trade-off balance between model prediction performance and computational costs of label acquisition.

Bridging both approaches of simulation metamodelling and active learning is the Gaussian Process (GP) framework. Due to its Bayesian properties, it provides predictions in the form of Gaussian probability distributions, with specified mean and variance. As a proxy for either information or uncertainty, this predictive variance can be naturally adopted to develop active learning strategies to seek the most informative points and explore the uncertainties within the simulation input space. GPs have a long tradition of being employed as metamodels (Gramacy, 2020; Kleijnen, 2009), as well as in active learning settings (Ling et al., 2016; Yue et al., 2020). When applied to spatial data, the GP framework is called Kriging (Chilès and Desassis, 2018) within geostatistics, constituting a widely used interpolation technique in the field.

Within the field of simulation and machine learning applied to EMS planning, state of the art has yet to build grounding supports. While simulation-based optimisation approaches are quite popular among researchers and practitioners and the body of knowledge is rich, there is still a lot to grow on the usage of new technologies and methods, particularly those that might take advantage of big data and real-time information, as indicated by Bélanger et al. (2019).

Simulation modelling is a recurrent tool to design, plan and study EMS. For example, in (Begicheva, 2020) the authors use simulation to identify the most important variables affecting the usage of EMS. In (Hu et al., 2020), simulation is used to identify ambulance travel times under different traffic conditions. These works focus primarily on identifying the underlying characteristics or parameters that can serve as input for a planning or assessing model. On the other hand, Enayati et al. (2020) proposes a stochastic model to optimise the EMS service planning and assesses it using a simulation model to compare the proposed model against existing policies. Similarly, Yang et al. (2019) implemented a simulation model to evaluate the performance of the ambulance location problem.

It is therefore clear that simulation is a reliable approach as an assessment tool in EMS planning. However, it still lacks generalised solutions to offer decision-makers the tools to assess and comprehensively compare multiple scenarios. Here is where we believe that metamodelling techniques, combined with active learning that helps reduce the hardly avoidable computational costs, have the potential to answer those needs. The first insights into the potential of metamodelling on this topic can be seen in (Yousefi and Yousefi, 2019) and in recent earlier works (Amorim et al., 2019a; Antunes et al., 2019a,b). The present work aims at stimulating the much-required developments and further applications within the field.

3. Methodological Approach

The methodology explored in this paper follows an active learning strategy integrated within a metamodelling setting. In the following, we first briefly present the GP framework, as it constitutes the core modelling tool of the work developed herein, and then the active learning metamodelling approach itself with more detail.

3.1. Gaussian Processes

A GP is a collection f of random variables in which any finite set forms a multivariate Gaussian distribution (Rasmussen and Williams, 2006). Any GP is completely defined by two functions: the mean, $m_f(\mathbf{x})$, and the covariance (or kernel), $k_f(\mathbf{x}, \mathbf{x}')$, where $\mathbf{x}$ and $\mathbf{x}'$ are two different input observations, and is commonly denoted by $\mathcal{GP}(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}'))$. Within a standard regression problem $y = f(\mathbf{x}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the random noise term, a GP prior is placed over $f(\mathbf{x})$, i.e., $f(\mathbf{x}) \sim \mathcal{GP}(m_f(\mathbf{x}), k_f(\mathbf{x}, \mathbf{x}'))$.

It is usual for the mean and covariance functions to have a certain number of free parameters, also called hyperparameters of the GP, which can be estimated via marginal likelihood maximisation with respect to the training data. After this fitting procedure, the conditional distribution for an unobserved data point $\mathbf{x}_*$ is given by $f_*|X, \mathbf{y}, \mathbf{x}_* \sim \mathcal{N}(k_f(X, \mathbf{x}_*)^\top [K_y]^{-1}\mathbf{y}, k_f(\mathbf{x}_*, \mathbf{x}_*) - k_f(X, \mathbf{x}_*)^\top [K_y]^{-1} k_f(X, \mathbf{x}_*))$, where K_y is the covariance matrix of the noisy observations, X is the design matrix composed by the independent variables and $\mathbf{y}$ is the vector containing the values of the dependent variable. Each GP prediction comes in the form of a Gaussian distribution. Hence, instead of a single point-wise estimate, GPs provides an entire range of possible values, weighed by a density function. This intrinsic Bayesian property allows the GPs to encode the uncertainty associated with its estimates and the underlying process being studied. The predictive variance $k_f(\mathbf{x}_*, \mathbf{x}_*) - k_f(X, \mathbf{x}_*)^\top [K_y]^{-1} k_f(X, \mathbf{x}_*)$ can be viewed as a proxy for both data and prediction uncertainties. Higher variance translates into more dispersion, which in turn implies more variability around the mean. This property allows straightforward implementations of active learning schemes by setting, for example, the predictive variance as an information criterion for selecting new data points.

It is worth mentioning that, whereas the mean function can be set to zero in most applications, the covariance function, however, plays a crucial role in establishing the GP's modelling performance. The latter essentially encodes the notions of similarity, spatial nearness and joint variability among the observations. This is the reason why GPs are can be viewed as kernel-based machines. A common default and general-purpose choice for their kernel is the Squared Exponential, also known as Radial Basis Function, generically defined as $k_f(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{x}')^\top M (\mathbf{x} - \mathbf{x}')\right)$, where σ_f^2 is the variance of the underlying signal function f, $M = diag(\boldsymbol{\sigma})^{-2}$, and $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_D]^\top$ is a positive, real-valued vector containing the characteristic length-scales of each of the D dimensions of the input feature space. These individual scales allow the fitting procedure to select the most appropriate weights for each input dimension according to their magnitude and importance, thus explaining why it is also called the Squared Exponential with Automatic Relevance Determination (SE-ARD). Besides being infinitely differentiable, it can serve as a universal approximator for arbitrary functions (Micchelli et al., 2006), among several other sound properties (Rasmussen and Williams, 2006).

3.2. Active Learning Strategy

The strategy followed in this work combines crucial elements of both active learning and simulation metamodelling, similar to that of (Antunes et al., 2018) and (Antunes et al., 2019a). Figure 1 briefly depicts the proposed methodology, whose general objective is to approximate the behaviour of the simulation model under study around predefined output values of interest and corresponding triggering input region. This is achieved by using a GP that, while modelling the functional relationship between the simulation input and output spaces, ultimately guides the training process itself towards such simulation output value.

Three essential ingredients must be defined a priori to starting the algorithm. These are the predefined simulation output search value (sV) of interest, the initial training grid ($trGrid$), and the search grid ($sGrid$). The algorithm is designed to provide the user with a directional search towards the sets of input values or input regions that trigger, via simulation, output values that lie close to sV by successive predictions over $sGrid$. It is worthwhile to mention that, contrary to sV and $sGrid$, which are both fixed structures, *trGrid* evolves as part of the active learning scheme.

For the sake of simplicity, this work focuses on the two-dimensional case, although the proposed methodology also applies to high-dimensional spaces. The method is deemed as directional since the GP effectively directs the iterative metamodelling process towards the input regions that are more likely to assume output values closer to sV. This directionality is accomplished by considering a series of iterative training grids and corresponding sub-grids that approximate the simulation results in the vicinity of sV. These grids are sequentially added to the increasing training grid. The GP, used as a metamodel, guides the expansion process of *trGrid* towards identifying simulation input regions whose simulation results are close to the predefined output value of interest. This allows the user to concentrate the computational workload (associated with each simulation run) specifically in these input regions, turning the exploration process not only faster but more efficient.

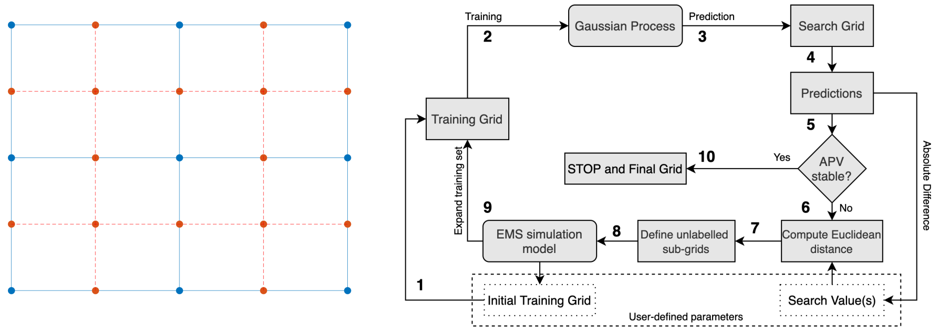
Figure 1(a) depicts an example of a training unit grid and its associated sub-grids. The unit grid is composed of four contiguous sub-grids, defined by a set of nine points arranged in a square shape. The same principles apply to any rectangular shape. Each vertex represents a Cartesian pair of values within the simulation input space whose corresponding simulation results (output values), *simR*, were previously computed. The training set denoted by $(trGrid, simR)$ is then a labelled data set. Furthermore, observe that the unit grid and sub-grid have several points in common, which could imply repeated simulation runs. Although this might prove useful for simulation models that exhibit a high degree of stochasticity, such scenarios were not particularly addressed in this work. As the iterative process advances, the sub-division of the successive grids continues recursively.

On the other hand, the search grid *sGrid*, besides being characterised by a static structure, has a different nature and purpose. It is only comprised of input data points for which the associated simulation results are not known. Thus, it comprises what is often called unlabelled simulation instances or observations, and it is exclusively used for prediction purposes. It should be highly dense so that the GP can predict over *sGrid* with increased great detail. Contrary to the training stage, prediction does not require much computational effort, and it is often a relatively fast procedure. This is valid not only for the GP framework but also for the vast majority of the machine learning tools.

In practice, *sGrid* is the simulation input region in which the user aims to explore the simulation output behaviour, in particular, to eventually locate the set of input values that trigger sV. Although not completely required, some prior knowledge regarding the simulation model is desirable when defining this grid. There is no point in defining a search grid in which the search value is never triggered. Hence, expert opinion and experience are of utmost importance in this stage so that *sGrid* encompasses some prior knowledge as to where sV is most likely to be triggered within the simulation input space.

Following Figure 1(b), the algorithm starts with the labelling process of the points contained in the initial training set. For each input point in this grid, a simulation experiment is run so that the corresponding simulation output result is obtained. The vector of all these output values is denoted by *simR* and the associated data set by $(trGrid, simR)$. Notice that in this first iteration, *trGrid* matches the training unit depicted in blue previously seen in Figure 1(a).

Afterwards, a GP is fitted to $(trGrid, simR)$, and predictions are made over the unlabelled set *sGrid*, effectively serving as a simulation metamodel. These predictions are estimates for the simulation output results associated with the points within the



(a) (b)

Figure 1.: (a) Grid-based training unit (blue) with corresponding sub-grids (red), and (b) proposed active learning methodology with 1-10 representing the iterative flows in chronological order.

Inputs: sV, $trGrid$, $sGrid$
While APV not stable *do*
Train a GP using $trGrid$ and obtain predictions over $sGrid$.
Select unlabelled sub-grids whose prediction points are closest to sV.
Run the simulator to obtain the output results for the selected sub-grids.
Expand $trGrid$ with the newly obtained simulation results.
Update APV.
End
Output: Trained GP, $trGrid$.

Figure 2.: Pseudocode for the presented active learning metamodelling strategy.

search grid. As mentioned in Section 3.1, the GP predicts in the form of a probability distribution, particularly providing the predictive variance associated with each prediction. This variance is then used to compute the Average Predictive Variance (APV) over $sGrid$, and its stability is accessed. For obvious reasons, this step is not applicable during the first iteration, as there are no previous iterations whose APV values can be compared to. Finally, sub-grids are defined, and some are selected to incorporate $trGrid$, naturally expanding it. This selection is conducted according to the predictions provided by the GP by computing the point-wise Euclidean distance between the GP surface and the nearest point comprising the horizontal plane defined by $z = sV$. Hence, only those sub-grids whose predicted values are closer to sV are added to the training grid.

Hereupon, the entire active learning process is iteratively repeated until APV shows no signs of significant variation between iterations. Observe that APV is viewed as a measure of the GP's overall prediction performance. As the training set is expanded by successive finer grids whose simulation results lie closer to sV, APV is expected to decrease from iteration to iteration. This decrease should reach a lower bound, representing the point from which the GP is not capable of further improving its predictions. At this point, the simple addition of new points to the training set is a waste of time and computational resources. In the end, this methodology is designed to provide a final and fine mesh grid that encloses the region contained in the simulation input space that specifically triggers the search value of interest. This approach is summarised in the form of a pseudo-algorithm in Figure 2.

4. Experimental Setting

In this section, an experimental illustration of the proposed approach is presented. To this end, a recently developed EMS simulator, designed for a real-world case study, was used, and two sets of experiments were conducted, including the validation of the obtained results.

4.1. EMS simulator

In this work, the EMS simulator developed by (Amorim et al., 2018) was used as a case study. This simulation model implements the allocation and dispatching of emergency vehicles according to the closest idle vehicle rule (Haghani and Yang, 2007; Jagtenberg et al., 2017; Yang et al., 2005). The underlying model relies on an agent-based simulation design where a city agent serves as the operator of the emergency medical service. After answering emergency calls, it allocates and dispatches idle vehicles to emergency locations. Each event is represented by an agent that encloses several characteristics that describe the type of medical emergency, location, and timestamp. Within the simulation, these events are set off according to a historical emergency call database and, as well as to the probability of location change, forwardly discussed in more detail. The vehicles are also formulated as agents that immediately react to the city agent requests.

Most real-world EMS are still developed and managed by rudimentary and outdated vehicle dispatching and reallocation rules. Essential factors such as traffic conditions, roadblocks, vehicles' exact positions, and emergency demand predictions are not entirely considered during EMS planning. These and other time-dependent elements do have a fundamental impact on the emergency system's response (Schmid, 2012). However, since the beginning of the 21st century, the proliferation of ubiquitous intelligent technologies (Kindberg and Fox, 2002) and the development of the Internet of Things (IoT) (Li et al., 2015), has culminated in unprecedented data growth. With this information technology revolution, EMS staff, practitioners, and researchers, in addition to the traditional historical data, are now able to access, collect and process large amounts of heterogeneous real-time information, such as traffic congestion, satellite-based navigation, and emergency alerts.

4.1.1. ***Inputs***

The mentioned EMS simulation model features three kinds of inputs, namely, the location change probability, traffic error, and station locations. Whereas the former two assume real values in the interval $[0, 1]$, the latter assumes discrete positive values, $[0, 1, 2, ...)$, additionally breaking down into 90 independent, although related, inputs dimensions. These inputs correspond to a 90-node emergency response network for possible station locations. For this work, 15 of these 90 possible locations were assigned to hold emergency stations. Figure 3 depicts the spatial distribution of the selected locations within the city of Porto. Here, darker areas correspond to higher values of emergency demand. This historical data consists of 35k emergency call records collected from the Portuguese Institute of Medical Emergencies (INEM) (Gomes et al., 2004) between 2012 and 2013. Table 1 summarises the locations and the number of vehicles per station.

On the other hand, the location change probability and the traffic error inputs are designed to reflect the operating conditions of the EMS somehow, as well as to induce

Table 1.: Stations used and number of emergency vehicles per station.

location	14	17	21	23	29	30	35	7	46	47	48	63	67	86	108
vehicles	2	2	3	2	5	4	2	2	3	4	4	1	4	1	3

a degree of variability often present in real-world systems. The former is particularly useful to emulate the expected stochasticity of the emergency events' spatial distribution. High values of probability change mean that the emergency locations present in the historical data are more likely to change. This introduces unforeseen dynamics in the spatio-temporal distribution of the emergency events, forcing the simulated EMS to respond accordingly.

With no surprise, the traffic error input relates to the road network conditions, although it does not directly affect the vehicle's travel times. Instead, it represents how traffic errors lead to bad operational decisions when dispatching vehicles. Consider the scenario where there is no traffic congestion, roadblocks, or other factors that can increase travel times. For a given emergency event, the optimal dispatching decision would be to assign the closest idle vehicle. However, the best dispatching decision is not trivial when heavy traffic occurs (e.g., peak hours). Moreover, errors might occur even with real traffic information, such as communication delays or data transmission problems.

Considering traffic errors instead of the traffic information allows measuring distances by travel times rather than in length units. In practice, the time it takes for a vehicle V to arrive at an emergency location E is far more crucial than the distance between its dispatching station location and E. Especially under severe traffic conditions, smaller distances do not imply reduced travel times. Therefore, traffic errors are considered in terms of time units, reflecting the intrinsic dynamics of traffic congestion and other road network conditions, ultimately leading to more robust dispatching decisions.

4.1.2. ***Outputs***

To evaluate the performance of the underlying EMS, two output metrics are considered in this simulation model, particularly the response time and survival rate. The vehicle response time measures how much time it takes for the medical staff to assist the victim. As a single-point quality measure for the entire dispatching operation, the sum of all emergency events or its corresponding average can be considered.

As for the survival rate, it represents the victims' likelihood of survival, which will depend simultaneously on the response times and severity of the emergencies. Again, the sum or average over all the medical occurrences can be considered to assess the overall performance of the EMS. In this work, only the survival output is taken into consideration for modelling purposes.

4.2. ***Results***

From a regression point-of-view, a GP is used to model the underlying functional relationship between the inputs, namely, location change probability and traffic error, and the output as the survival rate. Two sets of experiments were considered. Whereas the first considers only one search value, the second illustrates the methodology for two search values. In both experiments, we considered the mean the GP as the average of

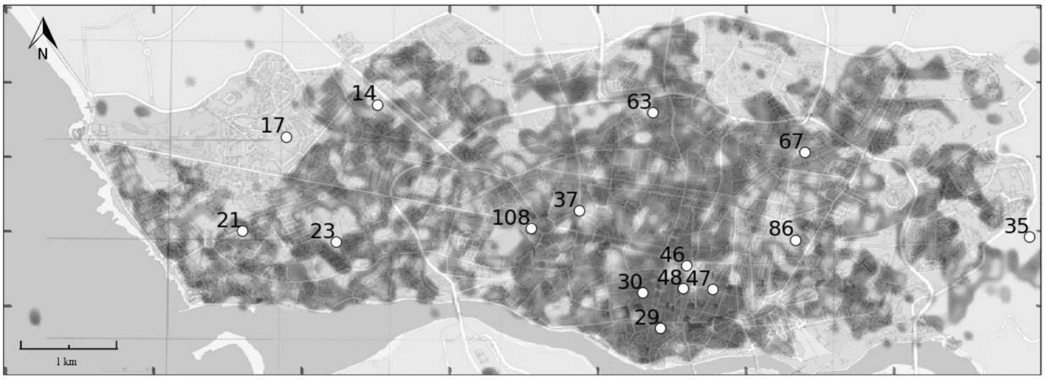


Figure 3.: Locations of the 15 emergency vehicle stations in Porto, Portugal. Darker regions represent higher emergency risk.

the simulation output value within the training set, whereas the covariance function was set to the widely known SE-ARD kernel already mentioned in Section 3.1. We also used the GP implementation available from Rasmussen and Williams (2006). The experiments were conducted in Matlab R2019b on a 2.7 GHz Quad-core Intel i7 processor with 16GB of RAM.

Our methodology aims to efficiently search for regions in the input simulation space that trigger a particular output value, or values, of interest. In the particular case of the mentioned EMS simulator, the proposed approach tries to identify sets of values defined within the space defined by the ranges of the probability change and traffic error inputs, $[0,1] \times [0,1]$, that trigger, through simulation, a user-specified output value lying in survival rate's range, $[0,1]$. An active learning and metamodelling approach is adopted so that exhausting and systematic computer experiments are avoided, thereby turning the exploration process of the EMS simulator's input space not only faster but more efficient.

As highlighted in Section 3.2, the proposed algorithm requires three components to be defined in advance, namely, sV, $trGrid$ and $sGrid$. The latter two are shared across the two sets of experiments. The initial training grid is set as the nine-point training unit depicted in blue in Figure 1(a), whereas the search grid is defined as a mesh grid of 10,000 unlabelled points uniformly distributed in $[0,1]^2$. The edges of both squared grids match those of the input domain being explored. The search values varied according to the experiment and were arbitrarily set for the sake of demonstration.

4.2.1. Experiment 1

In this experiment, the active learning strategy was applied to identify the input region most associated with survival rate values close to 50%. Thus, 0.50 was set as the search value of interest, i.e., $sV = 0.50$. The results are summarily depicted in Figure 4 and 5.

As the initial training grid is comprised of a set of labelled data points, that is, input-output tuples resulting from a series of computer experiments, a GP is fitted to these simulation results. This first approximation, represented by a three-dimensional surface, can be observed in Figure 4(a), and it corresponds to the GP predictions made over $sGrid$. The same panel does show not only the surface contours but also the input points constituting $trGrid$. Hereafter, using the GP predictions, the algorithm searches for the best-suited sub-grids (recall the red grids in panel (a) from Figure 1) to advance with the learning process. The best candidates are associated with those GP predictions whose values are closer to 0.50. This selection requires the computation of the Euclidean distance between the GP approximation and the plane $z = 0.5$. Observing Figure 4(b), the sub-grid delimited by $[0.0, 0.5] \times [0.0, 0.5]$ was selected as the best candidate to be added to *trGrid* for the following iteration. In other words, this means that it is most likely to contain simulation input values that trigger the search value 0.50.

The predictive variances related to each GP prediction are then averaged out over *sGrid*, and thus *APV* is computed. Finally, the selected sub-grid is labelled by the simulator. This is attained by running the simulation model with the input values associated with the selected sub-grid points. Naturally, each two-dimensional data point corresponds to a single simulation run.

The algorithm proceeds in an iterative manner, alternating between the GP training/fitting stage, simulation runs, and the expansion of the training data. Figures 4(d)-(f) show the results from the last three iterations, as well as the final configuration stages of *trGrid*. The algorithm was able to detect that the series of sub-grid defined within $[0.0, 1.0] \times [0.625, 0.750]$ specifically trigger simulation output values close to 0.50. This input region approximately matches the intersection area between the plane z and the GP surface. On the other hand, Figures 5(a)-(h) show the sequence of the distances between the search value and surface provided by the GP. Darker colour tones imply smaller distances. Here we can observe that the search procedure does not evolve linearly, not at least in the first iterations. As more points are added to the training grid, the GP reacts with different prediction behaviours until it eventually stabilises. The obtained simulation input region of interest is visible in Figure 5(h), here depicted as a dark narrow horizontal band.

The GP's fitting performance within the input region identified by the algorithm is supported by the finer grid of training points enclosing it. Note that the sub-grids gradually added to the training grid get narrower as the process advances. Thus, *sGrid* expands towards the input regions of interest defined by sequentially and increasingly finer grids. This is not only expected but also the ultimate goal of the proposed methodology. The algorithm stops when the addition of a new training point does not improve the fitting of the GP, i.e. when *APV* starts not to vary significantly from iteration to iteration. Figure 5(i) shows the evolution of *APV*. The algorithm required 17 iterations to stop.

It is worthwhile to mention that, in this work, the main objective of the GP is not to serve as the perfect simulation metamodel. Although it should be, of course, a reasonably good approximation for the underlying simulation model, most importantly, it should guide the active learning algorithm towards the identification of the most informative input points with regards to the simulation output search value. These points, arranged in grids, lead the algorithm in the direction of relevant input regions while minimising the computational workload. This ultimately turns the exploration process of such regions faster, as well as more efficient.

4.2.2. Experiment 2

In this experiment, the algorithm was extended to encompass two search values, namely, $sV_1 = 0.40$ and $sV_2 = 0.55$. Again, these values are arbitrarily set for the sake of illustration. Figures 6 and 7 summarize the results. The main goal of this experiment was to show that the proposed algorithm can be easily adapted to a multi-

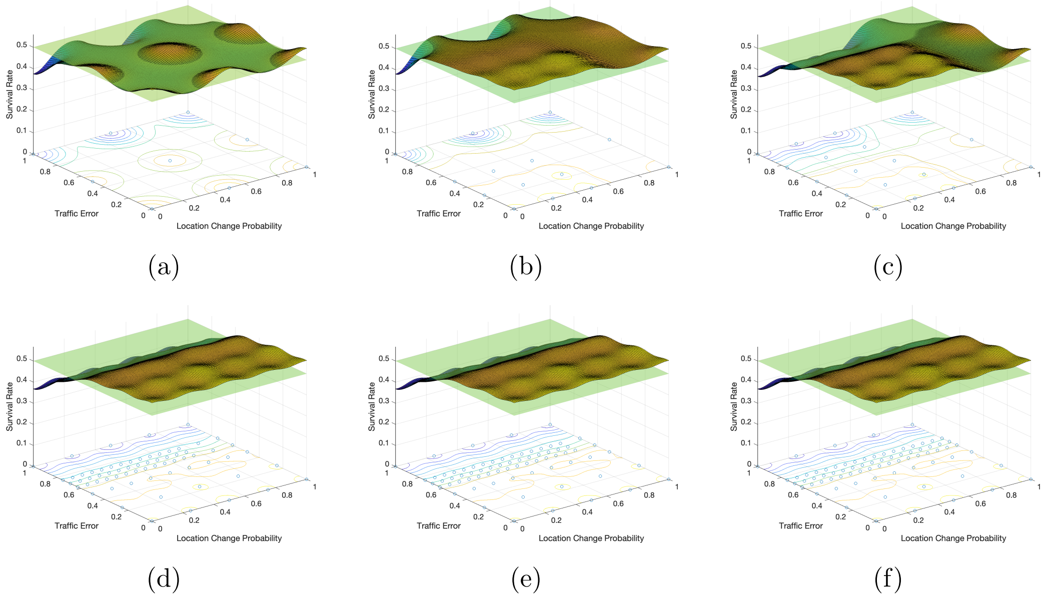


Figure 4.: Iterative GP surface approximations and associated countour plots with sequential training grids and $sV = 0.50$. Panels (a)-(c) and (d)-(f) correspond to iterations 1-3 and 15-17, respectively. The flat horizontal surface is located a $z = 0.50$ (search value).

directional nature. As expected, the problem gets more challenging as the number of search values, input domain range, and dimensionality increase.

Following the same strategy, the algorithm starts from the simple nine-point grid (see Figure 6(a)), ending up with a mesh grid of training points, as seen in Figure 6(f). Within this final grid, two series of finer sub-grids clearly stand out from the rest, in particular the grids defined by $[0.0, 1.0] \times [0.875, 1.0]$ and $[0.0, 1.0] \times [0.375, 0.50]$, corresponding to the input regions that trigger sV_1 and sV_2, respectively.

In Figure 7, panels (a)-(h) show several non-consecutive steps in the evolution of the distance (or absolute difference) between the GP surface and each of the search values. Here, notice that, contrary to the input region associated with sV_1, which is highlighted by the dark tone horizontal strip seen in panel (g), the same does not hold for sV_2 (see panel (h)). In terms of distances, we observe that instead of a single narrow region, a wider band is also visible, roughly defined within $[0.0, 1.0] \times [0.0, 0.5]$. Although the algorithm was able to identify the input region that effectively triggers the output value of 0.55, it does take any particular attention to the surroundings of that value. Consider the scenario where the GP partially exhibits a near-flat surface parallel to the search value plane ($z_2 = 0.55$ in this case) and that the distance between the two is sufficiently low. Here, the algorithm will ignore most of the associated GP predictions, to the detriment of those near the intersection zone between both surfaces, despite the closeness to sV_2. This is what has occurred in this experiment, and it is observable, for instance, in Figure 6(f). On the other hand, the same did not happen to sV_1 since the GP surface does not present a flat shape parallel to z_1 in the vicinity of this search value, as depicted in the same figure. An improvement to address this kind of situation should be considered in the future. Figure 7(i) shows that this time, the algorithm took 29 iterations to stop, as well as the expected decrease of APV.

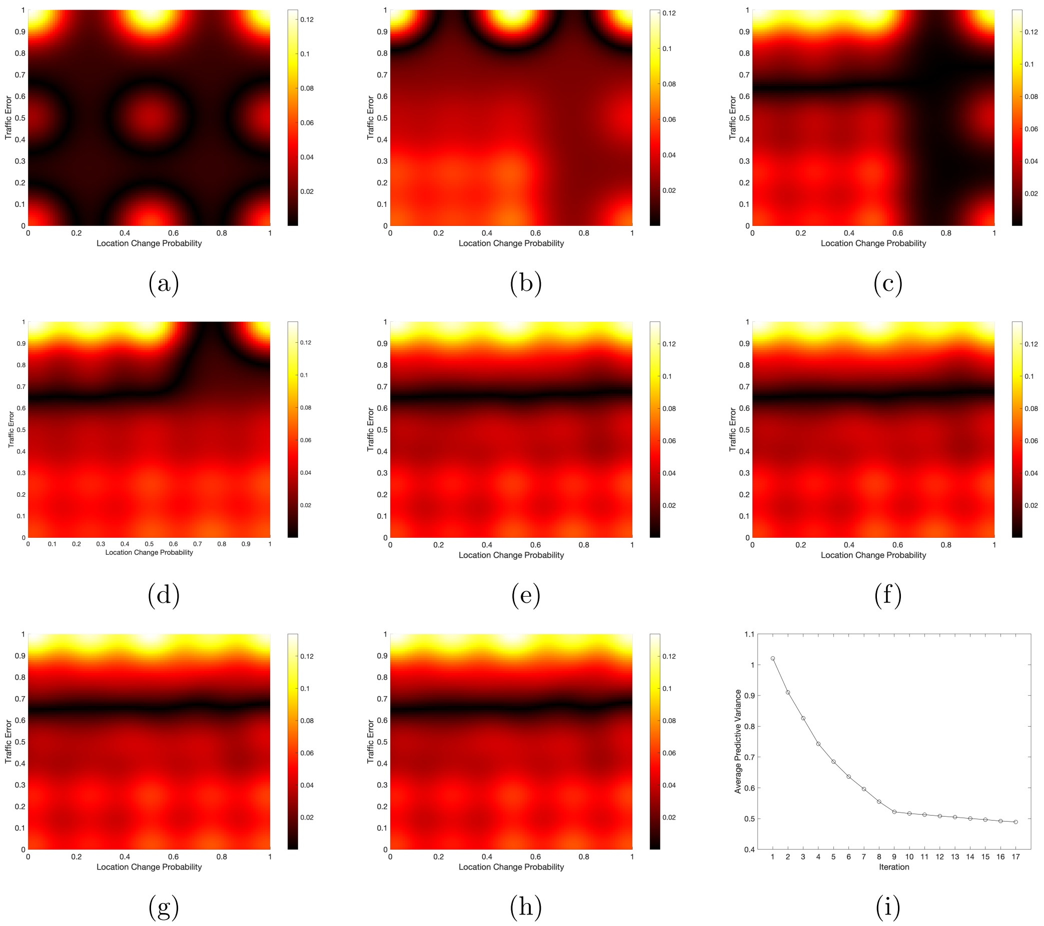
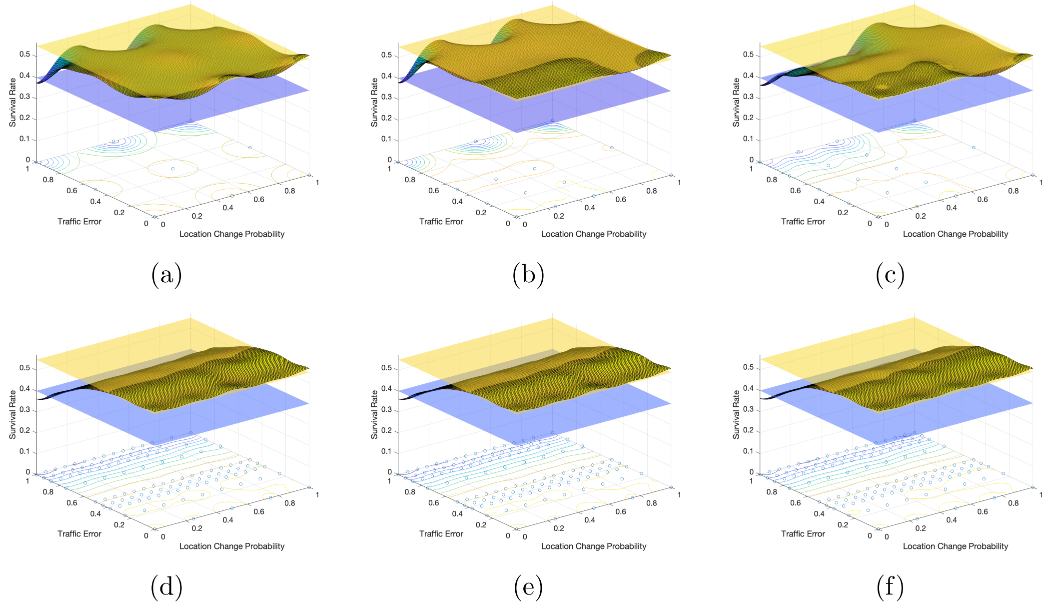


Figure 5.: Absolute difference between the obtained GP surface aproximations and $sV = 0.50$. Panels (a)-(d) and (e)-(h) correspond to iterations 1-4 and 14-17, respectively. Panel (i) depicts the evolution of the Average Predicted Variance.



(a) (b) (c)

(d) (e) (f)

Figure 6.: Iterative GP surface approximations and associated countour plots with sequential training grids with $sV_1 = 0.40$ and $sV_2 = 0.55$. Panels (a)-(c) and (d)-(f) correspond to iterations 1-3 and 27-29, respectively. The flat horizontal surfaces are located a $z_1 = 0.40$ and $z_2 = 0.55$, representing the two search values.

4.3. Validation

In this section, we present the results of a series of computer runs in order to assess the general predictive performance of the GP-based metamodel within the obtained final grids. We summarise these results in Table 2.

Several error-based metrics were used, namely Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Root Relative Squared Error (RRSE). The reported average values were obtained from 30 random computer runs and 10 test points. Each metric was computed by comparing the predictions provided by the GP from the last iterations against the corresponding simulation outputs. Within each experiment, we compared the GP's predictive performance inside and outside the obtained training grids. Here, the grids used for validation correspond to the finer grids that ultimately lead to the identification of the input region of interest, particularly $[0.0, 1.0] \times [0.625, 0.750]$, and $[0.0, 1.0] \times [0.375, 0.50] \cup [0.0, 1.0] \times [0.875, 1.0]$, respectively for experiments 1 and 2.

We can see in Table 2 that the GP consistently performs better inside each obtained grid as opposed to the outside. As more points are acquired during the active learning procedure, particularly towards the unknown input regions of interest, the density of the training grid increases locally. Thus, it is expected that the GP yields increased predictive performance within the finer obtained grids. This behaviour is particularly highlighted within the RMSE and MAE columns.

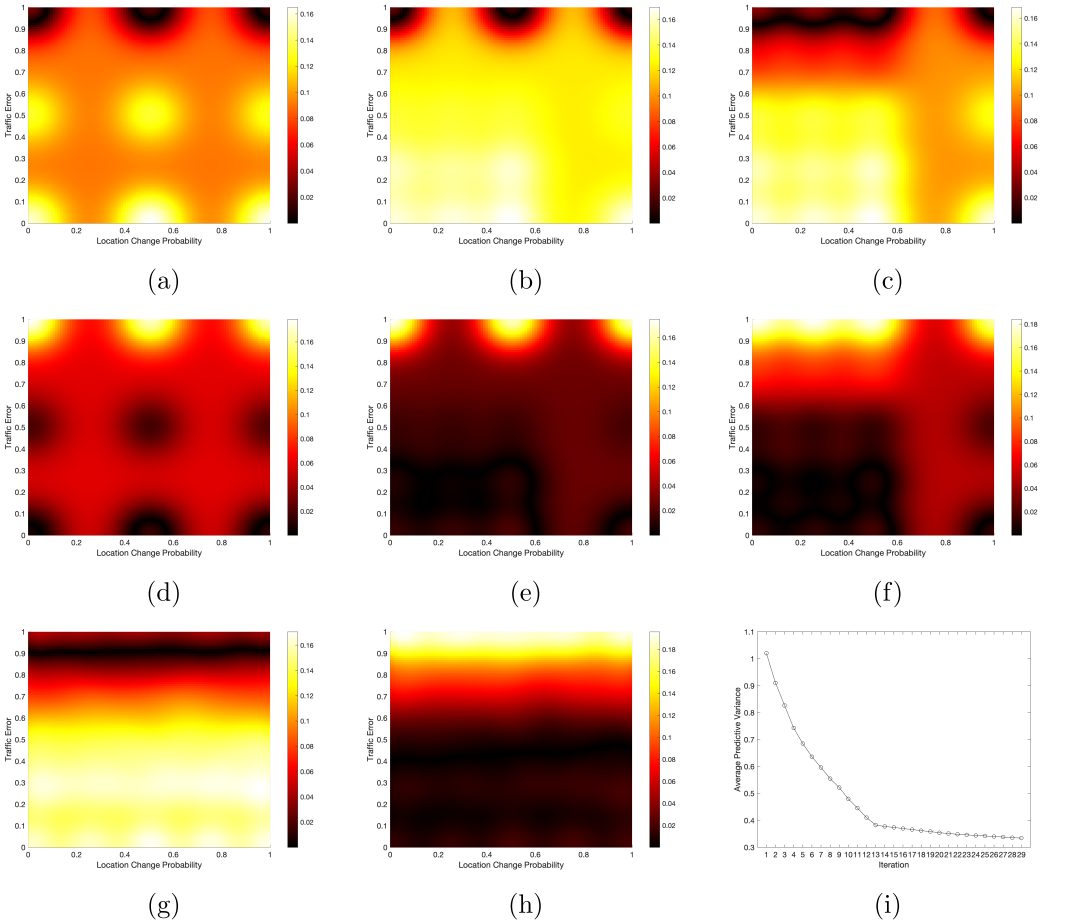


Figure 7.: Absolute difference between the obtained GP surface aproximations and the two search values. Iterations 1-3 are depicted in panels (a)-(c) and (d)-(f), respectively for $sV_1 = 0.40$ and $sV_2 = 0.55$. Panels (g) and (h) represent the two last iterations for each corresponding search value. Panel (i) depicts the evolution of the Average Predicted Variance.

Table 2.: Averaged performance results from 30 random runs for the two experiments.

Experiment	Grid	Search Value sV	RMSE	MAE	RAE	RRSE
1	inside	0.50	0.0049	0.0039	0.3658	0.3917
	outside		0.0428	0.0360	0.6441	0.6677
2	inside	0.55	0.0033	0.0026	0.5255	0.5603
		0.40	0.0054	0.0045	0.3690	0.3805
	outside	0.55 & 0.40	0.0302	0.0231	0.5730	0.6451

5. Practical Implications

The proposed method is used, as stated, to explore the input space that leads to predetermined outputs. This process opens doors to a "reverse engineering" methodology, benefiting from existing transport system models, leading to a novel traffic and mobility planning approach. As of now, city planners, transport providers, and researchers build transport system models (or use existing ones) when faced with the necessity of improving a specific performance indicator of their system. With these models, they are able to test and assess possible measures, changes, or additions that may or may not lead to an improvement of the specified indicator. This process requires the definition of potential scenarios, and their respective analyses within the defined models, through an interactive process of trial-and-error in which the user continuously fine-tunes them or eventually proposes new ones until they reach a satisfactory output performance. Such scenario design and planning processes, which must be guided by specialised domain knowledge, traditionally require systematic simulation experimentation and exploration.

With the presented approach and the results obtained herein, there is the possibility of reversing the analysis by first defining the target output of interest or even a collection of possible output values and only then exploring the corresponding input domain that triggers them through simulation. This methodology would allow for an efficient and expeditious analysis of transport system requirements or changes, which would improve the decision mechanism and would offer a better overview of the possible solutions. Hence, the proposed approach provides EMS managers and planners a method to identify input domain regions that most influences a certain target performance of interest. In this case, the input domain is the location of vehicles and the uncertainty of traffic and demand. This allows them to quickly identify where to reinforce resources and where, for example, traffic uncertainty will have a higher impact. In practice, the approach can provide practitioners with a solution matrix with the station configurations that, for a certain condition, would improve the target metric such as the survival rate discussed in this paper.

This work was highly motivated by recent early work. In the same context of EMS and emergency response systems, Amorim et al. (2019a) has proposed an integrated strategic and tactical planning decision methodology to address the problem of optimal ambulance location, particularly relying on a GP-based metamodel to mimic the real-world system state as a complement to a location optimisation model. Using such a metamodel, a local search for feasible and reasonable solutions was employed with successful results. In most modelling situations, evaluating a single solution might require several minutes or even hours. Therefore, a metamodel-based simulation was preferred since it allows a rather fast evaluation of alternative solutions with a minor and controlled accuracy loss. In this particular setting, the metamodel took the role of the simulator, effectively proposing a heuristic approach based on simulation metamodelling to help solve the underlying hard optimisation problem.

On the other hand, it is worthwhile mentioning that a pre-trained GP was used during the entire optimisation process. Although this heuristic proved to attain good results, we believe that it can be further improved if active learning is added to the equation. Instead of using a pre-trained GP, a more dynamic learning approach can be easily adopted by integrating the three main entities, namely, the simulation model, the optimisation model, and the active learning metamodelling strategy as the link between them. From the optimisation problem perspective, both the simulation model and metamodel are treated as black-boxes that provide alternative empirical evidence with respect to the real-world system since experimenting in the latter can prove to be prohibitively or impractical. Therefore, active learning can be employed to effectively balance the challenging trade-off between computational parsimony and accuracy loss. More training points can be added by request as the optimisation process evolves to steer the learning towards important input regions and, consequently, to increase the prediction performance of the simulation metamodel.

To the best of our knowledge, the recent works of Amorim et al. (2019a) and Antunes et al. (2019a) constitute one of the first applications of simulation active learning metamodelling to an EMS simulator within a simulation/optimisation setting. As of now, most emergency response studies continue to rely on systematic and exhausting simulation experiments on a trial-and-error basis. This often poses a considerable hindrance to a parsimonious modelling and decision-making process. Hence, we are certain that the conjoint use of active learning and simulation metamodeling represent important complementary tools for researchers and practitioners in the field.

The obvious burden of this methodology falls on the data acquisition to train the metamodel, inevitably requiring the use of the simulation model. This drawback is also the main motivation for our directional approach, which steers the metamodel training phases to the most important simulation input regions from a modelling point of view. The proposed approach will always be dependent on the simulation model itself. The idea is that we extract as few training points as possible, especially avoiding redundancy, in the most informative way possible. Nevertheless, despite the complexity of the underlying problem and simulation model, we can eventually bypass the simulation burden of an in-depth exploration of scenarios. Additionally, we can even target specific simulation outcomes to understand the input regions directly responsible for influencing them.

6. Conclusions & Future Work

This paper presents a methodology combining active learning and simulation metamodelling, which identifies relevant regions within the simulation input space that specifically triggers an output value previously set by the user. The survival rate outcome of a state-of-the-art EMS simulation model, modelled after the city of Porto, Portugal, was used as a case study.

The results show that, by employing a GP as a metamodel, the presented approach can effectively guide the exploration process of the input space of a simulation model towards those input regions whose output values are close to predefined search values of interest while minimising the computational workload at the same time. This is achieved by allocating the simulation runs to cover specific subregions of the input search space, thus making the exploration process more efficient and faster. It is also shown that it is trivially generalisable to multiple output search values.

The presented work can be improved in several directions. Several plans for future work are as follows:

- The possible correlation between simulation outputs can be useful to improve the metamodelling performance. This would also allow the exploration of multiple distinct output metrics simultaneously. Hence, a multi-output regression setting, namely the multi-output GP framework, will be considered in upcoming developments.
- Natively invertible models, such as Normalizing Flows, should also be explored

as a natural means of a tentative inversion of the relationship between simulation input and output spaces.
- Adopting popular design for computer experiments, such as the Latin-hypercube, and combining them with the proposed sub-grid sequence, should further improve the statistical significance of the proposed approach.
- The grid-based search should be revisited. The generalisation for more appropriate geometric arrangements, depending on the properties of the simulation input space under analysis and the particular goals of the metamodelling approach, should be an object of further study.
- Compare the current approach with similar alternative methods, including different active learning schemes and stopping criteria.
- Improve the presented method by encompassing the development of heuristics (e.g., local search) within simulation-optimisation approaches, which are commonly used tools in the research field of EMS planning and related decision support systems.
- Explore new EMS simulation models with increased input/output dimensionality for further testing and to induce future developments.

As a final note, and following up on the last point, recall that the current simulation model focuses on exploring vehicle location and its impact on patient survival. In it, the uncertainty of the demand and traffic conditions is included as key parameters that directly influence the survival outcome performance of the EMS response. With this in mind, further improvements can be made in order to make the simulation model more realistic, namely:

- Inclusion of other operational inputs parameters such as vehicle types, crew composition and capacity would allow the proposed model to give further insights on the tactical planning;
- Dynamic vehicle location is also worth exploring and could be implemented in further research by expanding the input vector into a sequence of vectors that would describe the location of vehicles along a predefined period.
- Several other uncertainty parameters can be implemented to tackle issues such as the busyness of hospitals, patients' processing times, or even assistance time at the emergency location.
- Experiment with new survival functions. The usage of these functions for measuring the performance of EMS response is still scarce since more research on the topic regarding different emergency types is required.

In sum, it is our understanding that only a handful of works involving active learning and simulation metamodels techniques are currently available in the EMS literature. To this end, we believe that this work constitutes an important contribution to the related body of knowledge. We believe that the results presented herein will initiate fruitful discussions among researchers and practitioners and ultimately foster more developments and multidisciplinary applications within the EMS modelling field.

Acknowledgments

The authors acknowledge the support of FCT - Portuguese national funding agency for science, research, and technology - under grant No. PD/BD/128047/2016. We extend our thanks to the anonymous reviewers for their insightful comments and suggestions.

References

Amorim, M., Antunes, F., Ferreira, S., and Couto, A. (2019a). An integrated approach for strategic and tactical decisions for the emergency medical service: Exploring optimization and metamodel-based simulation for vehicle location. *Computers & Industrial Engineering*, 137:106057.

Amorim, M., Ferreira, S., and Couto, A. (2018). Emergency medical service response: analyzing vehicle dispatching rules. *Transportation Research Record*, 2672(32):10–21.

Amorim, M., Ferreira, S., and Couto, A. (2019b). Corrigendum to how do traffic and demand daily changes define urban emergency medical service (uems) strategic decisions?: A multi-period survival approach [jth 12 (2019) 60-74]. *Journal of Transport & Health*, page 100570.

Andersson, H., Granberg, T. A., Christiansen, M., Aartun, E. S., and Leknes, H. (2020). Using optimization to provide decision support for strategic emergency medical service planning – three case studies. *International Journal of Medical Informatics*, 133:103975.

Antunes, F., Amorim, M., Pereira, F., and Ribeiro, B. (2019a). Active learning metamodeling for policy analysis: application to an emergency medical service simulator. *Simulation Modelling Practice and Theory*, page 101947.

Antunes, F., Pereira, F., and Ribeiro, B. (2019b). Directional grid-based search for simulation metamodeling using active learning. In *International Conference on Intelligent Transport Systems*, pages 32–46. Springer.

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

Archetti, C. and Peirano, L. (2020). Air intermodal freight transportation: The freight forwarder service problem. *Omega*, 94:102040.

Begicheva, S. V. (2020). Use of simulation modeling to select input variables in predictive models for the demand for emergency medical services in russia. In *2020 IEEE 22nd Conference on Business Informatics (CBI)*, volume 2, pages 101–105. IEEE.

Belokurov, V., Spodarev, R., and Belokurov, S. (2020). Determining passenger traffic as important factor in urban public transport system. *Transportation Research Procedia*, 50:52–58.

Box, G. E., Draper, N. R., et al. (1987). *Empirical model-building and response surfaces*, volume 424. Wiley New York.

Bélanger, V., Ruiz, A., and Soriano, P. (2019). Recent optimization models and trends in location, relocation, and dispatching of emergency medical vehicles. *European Journal of Operational Research*, 272(1):1 – 23.

Chilès, J.-P. and Desassis, N. (2018). *Fifty Years of Kriging, in Handbook of Mathematical Geosciences: Fifty Years of IAMG*, pages 589–612. Springer International Publishing.

Dulebenets, M. A. (2020). An adaptive island evolutionary algorithm for the berth scheduling problem. *Memetic Computing*, 12(1):51–72.

Enayati, S., Özaltın, O. Y., and Mayorga, M. E. (2020). Designing ambulance service districts under uncertainty. In *Optimal Districting and Territory Design*, pages 153–170. Springer.

Enoch, M. P., Cross, R., Potter, N., Davidson, C., Taylor, S., Brown, R., Huang, H., Parsons, J., Tucker, S., Wynne, E., et al. (2020). Future local passenger transport system scenarios and implications for policy and practice. *Transport policy*, 90:52–67.

Ferreira, M. C., Dias, T. G., and e Cunha, J. F. (2020). An in-depth study of mobile ticketing services in urban passenger transport: state of the art and future perspectives. *Smart Systems Design, Applications, and Challenges*, pages 145–165.

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

Gomes, E., Araújo, R., Soares-Oliveira, M., and Pereira, N. (2004). International ems systems: Portugal. *Resuscitation*, 62(3):257–260.

Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences.* CRC Press.

Haghani, A. and Yang, S. (2007). Real-time emergency response fleet deployment: Concepts, systems, simulation & case studies. In *Dynamic fleet management*, pages 133–162. Springer.

Heinold, A. (2020). Comparing emission estimation models for rail freight transportation. *Transportation Research Part D: Transport and Environment*, 86:102468.

Hensher, D. A., Button, K. J., Haynes, K. E., and Stopher, P. R. (2004). *Handbook of transport geography and spatial systems*. Emerald Group Publishing Limited.

Hu, W., Tan, J., Li, M., Wang, J., and Wang, F. (2020). Impact of traffic on the spatiotemporal variations of spatial accessibility of emergency medical services in inner-city shanghai. *Environment and Planning B: Urban Analytics and City Science*, 47(5):841–854.

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

Kindberg, T. and Fox, A. (2002). System software for ubiquitous computing. *IEEE pervasive computing*, 1(1):70–81.

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

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

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

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

Law, A. M. (2015). *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 5th edition.

Levy, J. M. (2016). *Contemporary urban planning*. Taylor & Francis.

Li, M. and Sethi, I. K. (2006). Confidence-based active learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1251–1261.

Li, S., Da Xu, L., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259.

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

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

Martine, G., Marshall, A., et al. (2007). State of world population 2007: unleashing the potential of urban growth. In *State of world population 2007: unleashing the potential of urban growth*. UNFPA.

McClarren, R. G. (2018). Gaussian process emulators and surrogate models. In *Uncertainty Quantification and Predictive Computational Science*, pages 257–274. Springer.

Micchelli, C. A., Xu, Y., and Zhang, H. (2006). Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667.

Moore, T. and Pulidindi, J. (2013). Understanding urban transportation systems: An action guide for city leaders. *National League of Cities. Available online at: www.nlc.org. Accessed on 26/10/21*, 31.

Nieuwenhuijsen, M. J. (2020). Urban and transport planning pathways to carbon neutral, liveable and healthy cities; a review of the current evidence. *Environment International*, 140:105661.

Pasha, J., Dulebenets, M. A., Kavoosi, M., Abioye, O. F., Theophilus, O., Wang, H., Kampmann, R., and Guo, W. (2020). Holistic tactical-level planning in liner shipping: an exact optimization approach. *Journal of Shipping and Trade*, 5(1):1–35.

Poulton, M., Noulas, A., Weston, D., and Roussos, G. (2019). Modeling metropolitan-area ambulance mobility under blue light conditions. *IEEE Access*, 7:1390–1403.

Rasmussen, C. E. and Williams, C. (2006). *Gaussian processes for machine learning (Adaptive computation and machine learning)*. The MIT Press.

Rodrigue, J.-P. (2020). *The geography of transport systems*. Routledge.

Rosano, M., Demartini, C. G., Lamberti, F., and Perboli, G. (2018). A mobile platform for collaborative urban freight transportation. *Transportation research procedia*, 30:14–22.

Schmid, V. (2012). Solving the dynamic ambulance relocation and dispatching problem using approximate dynamic programming. *European journal of operational research*, 219(3):611–621.

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

Settles, B. (2012). Active learning: Synthesis lectures on artificial intelligence and machine learning. *Long Island, NY: Morgan & Clay Pool*.

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

United Nations (2018). *The World's Cities in 2018, Data Booklet, ST/ESA/SER.A/417*. Department of Economic and Social Affairs, Population Division.

United Nations (2020). *World Cities Report 2020: The Value of Sustainable Urbanization*. United Nations Human Settlements Programme (UN-Habitat).

Vlahov, D. and Galea, S. (2002). Urbanization, urbanicity, and health. *Journal of Urban Health*, 79(1):S1–S12.

Wang, G. G. and Shan, S. (2006). Review of metamodeling techniques in support of engineering design optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 4255, pages 415–426.

Wang, X. and Zhai, J. (2016). *Learning With Uncertainty*. CRC Press.

Weber, C. and Puissant, A. (2003). Urbanization pressure and modeling of urban growth: example of the tunis metropolitan area. *Remote Sensing of Environment*, 86(3):341 – 352. Urban Remote Sensing.

World Bank (2021). Global economic prospects, june 2021.

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

Yang, W., Su, Q., Huang, S. H., Wang, Q., Zhu, Y., and Zhou, M. (2019). Simulation modeling and optimization for ambulance allocation considering spatiotemporal stochastic demand. *Journal of Management Science and Engineering*, 4(4):252–265.

Yousefi, M. and Yousefi, M. (2019). Human resource allocation in an emergency department: A metamodel-based simulation optimization. *Kybernetes*.

Yue, X., Wen, Y., Hunt, J. H., and Shi, J. (2020). Active learning for gaussian process considering uncertainties with application to shape control of composite fuselage. *IEEE Transactions on Automation Science and Engineering*, 18(1):36–46.

Zhang, X., Liu, H., Xu, M., Mao, C., Shi, J., Meng, G., and Wu, J. (2020). Evaluation of passenger satisfaction of urban multi-mode public transport. *PloS one*, 15(10):e0241004.