A comprehensive study of agent-based airport terminal operations using surrogate modeling and simulation

De Bosscher, Benjamin C.D.; Mohammadi Ziabari, Seyed Sahand; Sharpanskykh, Alexei

DOI
10.1016/j.simpat.2023.102811

Publication date
2023

Document Version
Final published version

Published in
Simulation Modelling Practice and Theory

Simulation Modelling Practice and Theory 128 (2023) 102811

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

journal homepage: www.elsevier.com/locate/simpat

A comprehensive study of agent-based airport terminal operations using surrogate modeling and simulation

Benjamin C.D. De Bosscher[a], Seyed Sahand Mohammadi Ziabari[a,b,*], Alexei Sharpanskykh[a]

[a] Air Transport and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands
[b] Faculty of Science, Mathematics and Computer Science, University of Amsterdam, 1098 XH Amsterdam, The Netherlands

ARTICLE INFO

Keywords:
Agent-based modeling
Surrogate modeling
Interpretable machine learning
Airport terminal

ABSTRACT

Airport terminals are complex sociotechnical systems, in which humans interact with diverse technical systems. A natural way to represent them is through agent-based modeling. However, this method has two drawbacks: it entails a heavy computational burden and the emergent properties are often difficult to analyze. The purpose of our research is therefore to accurately abstract and explain the dynamics of airport terminal operations by means of computationally efficient and interpretable surrogate models, based on an existing detailed agent-based simulation model. We propose a methodology consisting of two stages. Stage I involves the development of faithful surrogates. A sample is collected according to an active learning strategy, upon which Gaussian process regression, higher-order polynomials, gradient boosting, and random forests are fitted. Stage II then applies state-of-the-art techniques from the emerging field of explainable artificial intelligence to interpret and understand these models. Both model-agnostic and model-specific methods are considered, and their results are synthesized in order to explain the emergent properties. We prove the efficacy of this approach by conducting two case studies on AATOM, an existing Agent-based Airport Terminal Operations Model. The first case study examines the total expenditure on discretionary activities, such as shopping and dining. A combination of poor staffing strategies and high occupancy rates on certain flights was found to disrupt the terminal journey of passengers on subsequent flights. As a result of these knock-on phenomena, less free time is left for discretionary activities, which has a negative effect on the total expenditure. The second case study examines the throughput of security checkpoints. While throughput increases with passenger numbers, a clear point was observed where the checkpoint reaches its maximum capacity. This leads to longer queues and therefore higher waiting times. It even goes so far as to put passengers at risk of missing their flight, especially with poor staffing strategies. Altogether, we clearly observed the preservation of emergent phenomena in surrogate models, and conclude that their combination with interpretable machine learning is an effective way to explain the dynamics of complex sociotechnical systems.

[*] Corresponding author at: Technical University of Delft, Delft, the Netherlands.
E-mail address: s.s.mohammadiziabari@uva.nl (S.S. Mohammadi Ziabari).

https://doi.org/10.1016/j.simpat.2023.102811
Received 5 April 2023; Received in revised form 24 June 2023; Accepted 19 July 2023
Available online 23 July 2023

1. Introduction

In recent decades, the aviation sector has benefited from stable growth in air traffic demand. While long term prospects have long been taken for granted, abrupt events such as a financial crisis or the outbreak of a disease have shown the vulnerability of this supposition [1]. Furthermore, a growing number of people are concerned about the environmental impact [2]. It proves that airlines should operate more agile and lean: they must react quickly to such events and adapt to the new status quo. Airports, in turn, are the infrastructural epicenter of the system, so their operations are directly affected by changes in passenger numbers. This demonstrates the need for reliable models of terminal operations. Such models would be useful to prevent chaotic events, like in the aftermath of the COVID-19 pandemic at European airports [e.g., 3,4].

The modeling of airport terminal operations has been previously explored by numerous scholars. Pao-Yen Wu and Mengersen [5] summarized these efforts in a meta-study, wherein was concluded that agent-based simulation models are most commonly used for operational planning and design purposes. Indeed, such models are preeminent for high levels of detail without compromising the complexity and emergent properties of sociotechnical systems like an airport terminal [6]. Notwithstanding, their computational requirements are often substantial, which might become a limiting factor as the scale of the simulation increases. To address this limitation, a worthy alternative is the consideration of surrogate modeling, also known as meta-modeling. A surrogate mimic's model responses through so-called black-box approximation functions [7]. Fundamentally, the principle is subject to a dichotomy between savings in computational requirements and fidelity to the original model [8]. It is presumed to be viable as long as the reduction in computation time justifies the associated lower level of accuracy [9].

However, there are additional arguments. Namely, another disadvantage of agent-based modeling is that the emergent properties are only revealed a posteriori [10]. One is thus required to run several simulations to obtain some tangible information. As a result, it is rather challenging to thoroughly analyze the dynamics of a system based solely on its agent-based model. Surrogates allow for a wider range of interpretation possibilities, making them useful in this case too [8]. Traditional examples of such practices are the analysis of regression coefficients or the evaluation of feature sensitivities [e.g., 11,12]. Yet, a promising direction is the rapidly emerging field of explainable artificial intelligence (XAI). Its aim is to interpret and explain the reasoning behind machine learning algorithms [13]. Hence, there is an overlap with the purposes of meta-modeling and so it makes sense to exploit the synergies between the two disciplines. Applying the methods of XAI to surrogates will identify which rules, variables and characteristics of airport terminal operations are most decisive, ultimately leading to a better understanding of emergence in the system.

To our knowledge, existing literature on the common ground between surrogate modeling and interpretable machine learning is rather scarce. We therefore contribute by demonstrating the further potential of metamodeling in this direction, and in particular its application to agent-based airport terminal operations models. The ability to detect and elucidate emergent properties is of paramount importance to realize these ambitions. As such, the objective of the research can be summarized as to accurately abstract and explain the dynamics of airport terminal operations by means of computationally efficient and interpretable surrogate models, based on an existing agent-based simulation model. Apart from the scientific contributions, the outcome is mainly relevant for airport and airline managers. It leads to detailed insights into terminal processes, and provides them with efficient and effective decision-making tools. This enables them to act agile and adapt quickly to changing conditions, thereby maximizing service against available resources.

We intend to achieve the objective by proposing the following two-stage methodology. The starting point is AATOM, which is the abbreviation for agent-based airport terminal operations model. It was recently designed and calibrated by Janssen et al. [14], and has been further developed ever since. AATOM is known for its high-fidelity to the actual terminal system, although it suffers from large computational requirements. Hence, the first stage of our methodology relates to the creation of surrogate models. This includes generating a data set, training black-box functions, and validation. The process is not necessarily linear, as data collection can be combined with training surrogates — commonly known as active learning or adaptive sampling [12]. Once they reach a satisfactory level of accuracy; the second stage then uses them for the interpretation of the agent-based model. Both traditional and more advanced techniques from the field of XAI are considered. Several approaches are thus explored, although it is not the intention to implement as many as possible. Instead, we select those that ultimately yield the best insights into AATOM. The end product is therefore a concise synthesis of a multitude of results from different interpretation methods. If done right, the emergent properties of the sociotechnical system should come to light, demonstrating the relevance of this methodology. Altogether, its novelty is not situated in the individual components, but rather in their consolidation where we take advantage of the synergies.

The paper is organized as follows. It starts with compiling the theoretical background in Section 2. This is done by reviewing three research dimensions, namely the modeling of airport terminal operations, surrogate modeling, and interpretable machine learning. After that, Section 3 further elaborates on the methodology based on its two stages. The AATOM simulator is described next in Section 4: the main principles behind the model are illustrated, along with the specific settings for the simulations. Examples include the terminal layout, airport strategies, airline time slots, and so on. Section 5 continues by presenting the results. First, the performance of the meta-models is examined, after which two examples show the strengths of synthesizing different interpretation methods. In particular, we analyze: 1) the discretionary spending behavior of passengers, and 2) how the throughput at security is affected by different scenarios in the airport terminal. Finally, the results and their implications are further discussed in Section 6 and a conclusion is drawn up in Section 7, along with recommendations for future work.

2. Theoretical background

Three research fields are related to our research. This section provides the necessary groundwork by reviewing the state-of-the-art of each of these fields. Respectively, that is the modeling of airport terminal operations, surrogate modeling, and interpretable machine learning. We also discuss existing examples of their specific combination.

2.1. Modeling airport terminal operations

Airport terminals are central to passenger handling. It is the place where departure, arrival, and transfer flows congregate, each of which has its own characteristics and goals [15]. In particular, we focus on the departure flow of passengers. Typical activities include the check-in, security check, border control for international destinations, and possibly some non-aeronautical activities such as shopping or dining [16]. Scholars have shown great interest in modeling these processes as they are subject to stochasticity and non-trivial complexity inherent in natural human behavior.

Tosic [17] is one of the earliest available review studies, yet it has not lost its relevance. The author identifies several ingredients of modeling airport terminal operations, the most pertinent of which are the following. First of all, the demand of air traffic is usually forecast with traditional statistics. This is important for planning purposes and thus forms the basis for rigorous decision-making. The more recent literature has further subdivided it into problems with strategic, tactical and operational horizons [e.g., 18,19]. Secondly, one can also consider specific physical locations; examples are single check-in counters or border control. They are often modeled using queuing theory, where performance is measured by quality of service. The results can then be directly benchmarked against the International Air Transport Association expectations [20]. The models are either stochastic or deterministic, the former being closer to reality at the cost of greater complexity. Thirdly, terminal operations may be viewed from the perspective of the process itself, like security screening at the checkpoint. That allows to optimize it as a whole rather than the components individually. Processes are generally modeled in two ways: analytically or through simulation. Analytical approaches are quick, exact and not overly complicated. However, this affects their fidelity to the real world [21]. A simulation-based approach is therefore preferable if the system entails a certain degree of complexity. Lastly, the entire terminal building can be taken into account at once. In the end, individual processes influence one another and as such contribute to the overall emergent properties. Most examples in the literature are simulation-based, which makes sense as analytical approaches often fail to capture much of the complexity associated with the dynamics of socio-technical systems. Depending on the level of detail, one can still distinguish between microscopic, mesoscopic and macroscopic models, although the former is rather the standard when considering operational flows [21].

Alternatively, the more recent meta-study of Pao-Yen Wu and Mengersen [5] differentiate existing airport terminal models according to their use case. They identified four purposes: capacity estimation, operational planning, security risk evaluation, and performance measurement. This becomes particularly interesting in combination with Tosic [17], as it reveals appropriate methods to realize our research ambitions. Notably, most models to represent the operations of an entire departure flow seem to be agent-based. That is a microscopic bottom-up approach capable of simulating the behavior of individual passengers, along with the interactions between them and the environment [5]. It became particularly relevant as computing power increased over the years, giving researchers the opportunity to create simulation models that are meticulously close to reality [22]. Hence, agent-based modeling is indeed very suitable if one requires detailed information about terminal processes, which is crucial for understanding emergent properties.

In line with the above observations and suggestions, Janssen et al. [14] have recently developed such an agent-based architecture and a simulator for airport terminal operations. We use this model to prove our methodology, so Section 4 further elaborates on its working principle and usage. Nevertheless, it is rather known for its heavy computational requirements, making surrogate modeling a viable alternative.

2.2. Surrogate modeling

Despite currently available technology, advances to understand complex systems in detail are often hampered by computational limitations. This has encouraged the development of surrogate modeling, which aims to fit black-box functions between the input and output of an expensive model in an attempt to accurately mimic its behavior [8,7]. The concept is not new and knows several applications in engineering: Forrester et al. [8] mentions structural analysis, computational fluid dynamics, geo-statistics, etc. More recently, it is also finding its way to emulate agent-based simulation models, as demonstrated in the meta-study by Pietzsch et al. [23]. The benefit is that, once a meta-model is trained, it can approximate the output at a fraction of the computational effort that would otherwise be required. Moreover, it facilitates the analysis of the underlying system. An example is Elshawi et al. [24], where they managed to gain detailed insights into relationships between several variables to assess people’s risk of hypertension based on local and global surrogates.

Meta-modeling starts with a design of experiments (DOE), the purpose of which is the creation of a training sample [8]. One typically strives to maximize the amount of statistical information in the data against the number of observations. After all, the original model of the complex system usually has large computational requirements. Several strategies exist for the DOE, although literature generally distinguishes between one-shot, space-filling, and adaptive approaches [25,26]. The latter in particular has been receiving a lot of academic attention lately, despite it being rather complicated. We describe the strategy as a process where an algorithm iteratively samples in regions from which a surrogate model benefits the most. Hence, it is also known as ’active learning’ because an emulator is trained simultaneously [25]. The notion is that with the same or even a smaller sample size, adaptive approaches should outperform the other two because data points are selected in such a way as to maximize the accuracy of the meta-models.

The second degree of freedom is the architectural choice of the surrogate. In the early days, this was limited to mostly linear models. They are however still widely used, mainly because of their simplicity [27]. Later on, Gaussian processes and radial basis functions made their entrance as they proved to be more accurate, albeit at higher computational costs [27,28]. Nowadays, research into the possibilities of machine learning is thriving. Its application to surrogate modeling has not been overlooked with support-vector machines, decision trees, ensembles, and neural networks as the most recent additions to the field [29,9].

2.3. Interpretable machine learning

The traditional approach to explain the behavior of a model has been to conduct a sensitivity analysis (SA) [30]. It allows to investigate the effect of input parameters on the output. While its usefulness remains undisputed, further research in this area has been flourishing of late. This is fueled by the European Union's adaption of the General Data Protection Regulation, which requires transparency and a profound understanding of the rationale behind machine learning implementations [31,32]. Seemingly simple questions, whether the outcome is trustworthy or not, or how the algorithms arrive at their solutions, are actually quite challenging to answer. The literature calls it explainable artificial intelligence (XAI), which even goes so far as questioning whether accuracy or explainability should be optimized [33].

The motivation for developing surrogates to understand a complex system is twofold. On the one hand, more extensive analyses can be performed as they are much faster than the original model. On the other hand, they also consist of internal functional relationships between input and output parameters, which may reveal emergent properties. Methods belonging to the former are commonly referred to as agnostic [34]. The SA is one of them, although state-of-the-art examples are dependency plots [35], examining the relevance of features [36], local interpretable model-agnostic explanations [37], and Shapley additive explanations [38]. Alternatively, methods belonging to the latter are associated with a particular architecture. That is, they are model-specific [34]. A well-known example is the interpretation of regression coefficients, for which widely accepted statistical significance tests are available to evaluate whether and which parameters truly influence a response [13]. Another possibility is to extract decision rules from tree-based ensembles, such as RuleFit [39]. This enables an IF-THEN type of reasoning and yields insights that are quite unique. Clearly, there are myriads of options which show that research into XAI is still emerging. For those new to the field, we refer to Barredo Arrieta et al. [13] for relevant definitions and taxonomies, while practical illustrations are found in Belle and Papantonis [40], Elshawi et al. [24], and Molnar [41].

2.4. Surrogate modeling for the interpretation of agent-based models

Research on the employment of meta-modeling for understanding agent-based models is rather scarce. An example is the work of Janssen et al. [11], in which linear regression was applied as one of the methods to examine causality between features and responses. In addition to causal graphs and Sobol sensitivity indices, regression weights proved useful in elucidating the strength and direction of relationships. Nevertheless, the authors emphasize that emergent properties are best observed when insights from different angles are combined. This also helps against spurious interpretations, as causalities may be inferred that do not exist in reality.

Another example is De Leeuw et al. [42], who implemented random forests and artificial neural networks to derive feature importances. The results clearly show which input parameters their surrogates are most reliant on. These are usually solid indicators of the drivers behind a response, although further conclusions are difficult to draw. Hence, it is a valuable addition to the overall analysis, but solely the importances will not reveal any dynamics nor emergence in agent-based models.

Lastly, ten Broeke et al. [43] proposed to split the feature space first. They use a classification algorithm to distinguish different types of model behavior from one another. Only thereafter, support-vector regression is deployed to mimic responses in each of the distinct regions. This enables them to differentiate between parts of the domain, which in turn leads to a more in-depth analysis.

Despite the latest developments, it is clear that the full potential of surrogate modeling and XAI has not yet been exploited in the interpretation of agent-based models. We fill the gap by proposing a two-stage methodology in the next section. First, surrogate models are created, to which various techniques from the field of interpretable machine learning are then applied. This paves the way for detecting and visualizing emergent properties of complex systems behind agent-based models.

3. Methodology

As relevant dimensions of the research have been touched upon in the theoretical background, current section continues with the

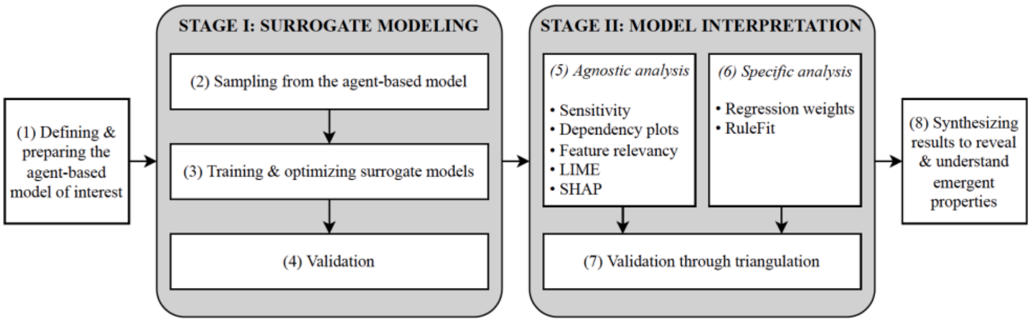

STAGE I: SURROGATE MODELING
STAGE II: MODEL INTERPRETATION
(1) Defining & preparing the agent-based model of interest
(2) Sampling from the agent-based model
(3) Training & optimizing surrogate models
(4) Validation
(5) Agnostic analysis
• Sensitivity
• Dependency plots
• Feature relevancy
• LIME
• SHAP
(6) Specific analysis
• Regression weights
• RuleFit
(7) Validation through triangulation
(8) Synthesizing results to reveal & understand emergent properties

Fig. 1. High-level overview of the two-stage methodology.

methodology. A high-level overview is depicted in Fig. 1. The first step is to define and prepare the agent-based model of interest. This model is typically highly detailed and close to reality, but computationally intensive. Consequently, there are two reasons that make surrogate modeling an attractive alternative. On the one hand, it gives access to much faster models. On the other hand, they enable us to better understand the underlying system — recall that agent-based models reveal the emergent properties only a posteriori, thereby requiring numerous simulations runs [10]. These two reasons are reflected in stage I and stage II of the methodology. The former consists of sampling, fitting surrogate models, and validation. The latter is concerned with agnostic and specific analyses, after which their outcome is validated through triangulation. Finally, the results of the second stage are synthesized in order to interpret and understand the complex dynamics and emergence of the focal agent-based model. The purpose is not to make the analysis as elaborate as possible, but rather to select the results that ultimately lead to the best insight.

3.1. Defining and preparing agent-based models

Before diving into the possibilities of surrogate modeling, one must first have access to an agent-based model of the system under investigation. Our general presumption is that the model has been verified and validated, although it is strongly recommended to conduct some additional tests to see whether this is actually the case. Namely, surrogates cannot be expected to yield excellent performance if their training data is of poor quality. Hailpern and Santhanam [44] provide an overview of common practices for software testing and verification.

Next to that, sociotechnical systems are characterized by high degrees of stochasticity; a natural consequence of the human behavior to which they are subject. This results in variable model responses. Even if the input settings are exactly the same, output parameters will differ in another simulation run. It is usually considered beneficial and makes agent-based modeling a powerful paradigm to analyze such systems. Notwithstanding, surrogate models are different as they are generally deterministic in nature. In other words, they associate one input vector with one specific output vector, making it difficult for them to incorporate stochasticity. There is thus a need to stabilize responses of agent-based models before their data can be used to train surrogates. In accordance with the law of large numbers, a straightforward approach to achieve this is by averaging the results over many simulation runs without changing the input parameters [45]. The coefficient of variation can then be evaluated to determine exactly how many simulations are needed — a common approach to describe the statistical dispersion of a particular outcome [46]. Once the responses are deemed sufficiently stable, one can proceeded to stage I, starting with sampling from the agent-based model.

3.2. Sampling from agent-based models

The surrogate modeling process commences with the creation of a training data set. This is often referred to as the design of experiments (DOE), which aims to extract as much statistical information as possible from the focal agent-based model [8]. While several sampling strategies exist, it follows from the theoretical background in Section 2 that adaptive designs are state-of-the-art. We therefore focus in particular on such approaches. Once an initial sample is available, the general procedure is to iteratively evaluate a meta-model and select a new point to sample until some stopping criterion is reached [25]. There are still a few degrees of freedom, so Algorithm 1 presents an overview of the proposed strategy, of which we now discuss the consecutive steps.

Since active learning requires the prediction uncertainty of a surrogate, one must first train the model to evaluate its output. This in turn can only be done if an initial sample is available. In the literature, scholars often use one of the following four sampling methods: Latin hypercube sampling, orthogonal arrays, Hammersley sequences, and uniform designs [27]. They all have pros and cons, though we find the Hammersley sequence best suited to serve especially the exploration purpose of an initial sample. Wong et al. [47] describe the technical details, but in essence it is based on a mathematical formulation to generate sequences with low discrepancy while maintaining a uniform distribution in high-dimensional spaces. That also makes the method space-filling and computationally efficient, albeit only quasirandom and rather poor at covering the boundaries of the domain [48,47]. Regarding the size of the initial sample, Loeppky et al. [49] recommend 10 times the number of input features if a Gaussian process regression model is used as surrogate [25]. This turned out to be the case in the while loop of the active learning algorithm.

The surveys of Liu et al. [25] and Fuhg et al. [26] are excellent starting points to design such a while loop. Fundamentally, it consists of two key ingredients: 1) a way to obtain the uncertainty of a meta-model, and 2) an acquisition function to translate the uncertainty into a specific data point that is most interesting to sample next. Liu et al. [25] distinguish four strategies for the first ingredient. One can use either the variance, disagreement between a committee of surrogates, cross-validation prediction error, or derivatives. The former is by far the most effective and efficient way to infer about surrogate model uncertainty over an entire domain — especially the

Algorithm 1
Proposed adaptive sampling strategy.

```
Use the Hammersley sequence to generate an initial set of features        ⇒ Size = 10 X dimensionality
Sample the corresponding response from the agent-based model
**While** the stopping criterion is not reached **do**                    ⇒ Based on meta-model accuracy
    Train a Gaussian process regression model
    Identify the next data point to sample based on the EIGF acquisition function
    Sample the responses corresponding to the selected data point from the agent-based model
    Add the data point to the training set
**end while**
```

combination with Gaussian process regression is powerful, as it naturally yields the prediction variance [25,50]. For the second ingredient, Fuhg et al. [26] rigorously reviewed 14 promising acquisition functions. Various drivers were evaluated and based on their assessment, we chose the expected improvement for global fit (EIGF), developed by Lam [51]. Namely, it yields a reliable and solid performance for meta-modeling purposes, without increasing the theoretical complexity or computational requirements too much. Mathematically, the EIGF acquisition functions is expressed as

$$\text{for} f \text{in} D \in \mathbb{R}^{\text{n}}, x_{\text{next}} = \underset{x \in D}{\arg\max} \left[\alpha \left(\widehat{\sigma}^2(x) \right) + (1-\alpha) \left(\widehat{f}(x) - f(x^*) \right)^2 \right] \tag{1}$$

where f denotes the target, $\widehat{f}$ the surrogate approximation, D the feature space, n its dimensionality, $\widehat{\sigma}^2$the variance of the surrogate, x the input vector, and x^* the nearest already sampled data point. We later added a balancing factor α to control the trade-off between exploration and exploitation during the sampling process. The closer to one, the more preference to exploration, and vice versa. This may change throughout the loop, e.g., to allow for a decaying strategy that gradually shifts the emphasis from exploration to exploitation.

The while loop of the active learning algorithm continues until a stopping criterion is reached. That is a predefined condition which determines when the sample ought to be sufficiently informative. Usually, it is based on computational or time constraints, though one can also look at the attained meta-model accuracy [25,26]. We opt for the latter because the computational budget for current research allows this. More specifically, the active learning algorithm continues until accuracy no longer improves by increasing the iteration.

In addition to a training set, it is also desirable to have independent validation and test sets for the hyperparameter optimization and model assessment, respectively. Common approaches include splitting the training set beforehand or pursuing a cross-validation strategy [52]. However, this is not straightforward when the sample is gathered adaptively, as the benefits of active learning could be jeopardized. We therefore prefer to create two additional data sets. They are both randomly sampled from the agent-based model, but only those input parameter combinations that were not selected during the adaptive sampling process. This continues until the size of each equals about 20% of the entire sample; a typical proportion for validation and test sets [52,53].

Finally, the implementation of the DOE must be verified. Explanatory testing is used to eliminate most of the bugs, although unit and integration tests are also necessary to ensure that the software behaves as expected at a deeper level [44]. After this, one can proceed with training and optimizing surrogate models.

3.3. *Training and optimizing surrogate models*

When a data sample is available, the next step of stage I is to train and optimize surrogates. In essence, we aim to find an appropriate function $\widehat{f}$ so that the prediction error e is as small as possible compared to the actual outcome of the focal agent-based model, denoted by f. This is mathematically described as

$$f(x) = \widehat{f}(x) + e \tag{2}$$

where x denotes an n-dimensional input vector [54]. Put differently, the purpose is to accurately mimic the response behavior of the original function f by fitting a machine learning algorithm on the sample [8].

A follow-up question is then what modeling method is most suitable to replicate the relationship between features and outcome. Various alternatives have been tried out over the years, a comparison of which is provided in Table 1. We summarized the principal advantages and disadvantages of common architectures. Expressions that linearly combine mathematical terms are referred to as linear models; specific examples are linear regression, polynomials, and multivariate adaptive regression splines [7,55,56]. They are widely popular, mainly because of their simplicity and interpretability [27]. Secondly, there are Gaussian processes and radial basis functions [57,7]. Both use Euclidean distances to known observations, although the former builds surrogates as a stochastic process and the latter as a weighted sum of basis functions [58,7]. They became prevalent surrogate models as scholars reported high

Table 1
Comparison between common architectures for surrogate modeling.

Architecture	Advantages	Disadvantages	Ref.
Linear models	Simple, interpretable, can be extended to a higher-order polynomial or spline	Bounded to their functional form, less suitable for complicated problems	[27,9,28]
Gaussian processes	Yields the uncertainty of the prediction, high accuracy, integrated hyperparameters tuning	Computationally intensive, ill-conditioned covariance matrix, complicated mathematics	[9,28,55]
Radial basis functions	Captures non-linear relationships without having a too complicated construction	Difficult to interpret, basis functions and hyperparameters need to be defined, ill-conditioning	[9,28,8]
Support-vector machines	Flexible, robust, low risk of over fitting, performs well with small samples	Difficult to interpret, complex hyperparameters, computationally intensive	[7,55,52]
Decision trees	Powerful, handle different data types without much preprocessing, performs especially well in boosted/bagged ensembles	Deteriorating interpretability when used in an ensemble, prone to oversitting, responses are not smooth	[29,53,52]
Neural networks	Powerful, completely customized, captures complex relationships and interactions in a high dimensional. environment	Prone to overfitting, computationally intensive, difficult to interpret, complex hyperparameters, require a lot of data	[53,52,55]

accuracies, especially for non-linear relationships [27,28]. Finally, more powerful machine learning algorithms are gradually gaining the upper hand, with support-vector machines, decision trees, and neural networks as the latest additions to the field [29,9]. While there is some academic debate about the accuracy of support-vector machines (e.g., a superior, comparable and inferior performance is mentioned by Wang and Shan [27], Bhosekar and Ierapetritou [7], Williams and Cremaschi [59], respectively), the dominance of tree ensembles and neural networks is not questioned [52]. The latter, however, suffers from severe constructional complexity, and also requires a large data sample and a substantial computational budget [53,52].

Now that the salient characteristics of common architectures are clear, we move on to selecting appropriate alternatives in order to achieve the research objective. First and foremost, recall from Section 3.2 that a Gaussian process regression model is central to the adaptive sampling strategy. It is therefore automatically part of our selection. Secondly, the highest accuracy is presumably attained by tree ensembles, making random forests and gradient boosting also promising candidates. Finally, we include polynomials because they are reasonably interpretable without losing the ability to capture curvilinear patterns [29]. The other methods are not taken into account; they are either too inaccurate, too complicated, or too difficult to interpret.

Next, the selected architectures should also be optimized, as the performance of machine learning algorithms can be severely affected by the choice of hyperparameters. Notwithstanding, finding the optimal combination is often a difficult and, above all, computationally demanding task [60]. Popular tuning methods are therefore a grid search, random search, Bayesian optimization, and evolutionary metaheuristics [61–63]. We prefer Bayesian optimization as it is an efficient algorithm with high chances of finding a combination close to the global optimum. Moreover, it does not require in-depth knowledge of promising candidates. Two drawbacks are its complexity and difficulty to multi-thread, although these are easily outweighed by the benefits [62,63]. For an introduction to Bayesian optimization, we refer the reader to Shahriari et al. [64], but in essence it works as follows. An auxiliary model is fitted onto a prediction error metric of the surrogate, then the parameter combination is selected that leads to the smallest error, after which this combination is evaluated on the actual surrogate. The auxiliary model is updated with the new information and the process continues sequentially until a predefined number of iterations is arrived at [61,62].

Lastly, the training and optimizing step must also be verified. This depends on exactly how it was implemented, but one typically employs existing packages, such as Python's scikit-learn [65] or scikit-optimize [66]. These packages are generally tested before being released, so there is no need to do this again. However, one should still perform integration tests to ensure that interfaces are properly embedded in the software [44].

3.4. Surrogate model validation

Validation is important as it gives an idea about the generalization power of meta-models. Namely, it shows to what extent they remain close to the agent-based model under investigation. The out-of-sample performance is usually measured by evaluating validation metrics on an independent test set [53].

Undoubtedly, one of the most popular metrics is the coefficient of determination (R^2). It indicates the proportion of variation in a response of the agent-based model that is explained by input parameters in the surrogate [67,46]. The R^2 is a very useful and informative metric, but it does not directly measure the prediction error. For that, one can resort to the root-mean-square error (RMSE), which yields the expected error of a surrogate [8]. Alternatively, there is also the mean absolute error (MAE). The RMSE and MAE are closely related, although an essential distinction is that the former squares the discrepancies between actual and predicted responses and takes the root of their average, instead of using the absolute value. This makes the RMSE naturally more sensitive to outliers [67]. Both the RMSE and MAE are expressed on the same scale as the evaluated output parameter. While this can be convenient, the performance of surrogate models for different responses cannot be compared when they are expressed differently. The mean absolute percentage error (MAPE) resolves the issue by calculating the prediction error relatively. One drawback, however, is that the indicator tends to exaggerate the error for responses close to zero [67].

Table 2
Comparison between model-agnostic interpretation methods.

Method	Advantages	Disadvantages	Ref.
Sensitivity analysis	Simple and evident, gives solid indications, several implementations exist with different levels of sophistications (local and global)	Interactions are not always considered, can be computationally intensive, questions about the variance as a proxy for variability	[68, 69,30]
Dependency plots	Intuitive and easily understood, no ambiguity, heterogeneity can be detected, straightforward implementation	Features are analyzed separately when in fact they may be correlated, difficult to consider more than two dimensions at the same time	[41, 24,35]
Feature relevancy	Easily understood, incorporates feature interactions, provides one comprehensive overview	Perturbations can lead to unstable results, relevancy may be divided among correlated features	[41, 36]
LIME	Fully customized, easily understood, straightforward implementations, yields an actual model rather than just insights	Unstable results: there may be large local differences, no theoretical basis for why this should work, may degrees of freedom	[41, 70]
SHAP	Based on valid theoretical arguments, provide a comprehensive overview, local and global interpretations are consistent	Computationally intensive, approximations are necessary, rather complicated, not evident to interpret the result correctly	[41, 40,38]

3.5. Model-agnostic analysis

When the concept of meta-modeling was introduced, we argued that it can play an important role in understanding an underlying system. Surrogates are not only much faster, but they also have internal functional relationships between the input and output parameters, unlike agent-based models [8,10]. As a result, they enable: 1) more extensive analyses, and 2) explicit insight into their rationale. Interpretation methods that relate to the former are referred to as model-agnostic [34]. They are independent from machine learning algorithms and can therefore be applied to essentially any model. Only the effect of input parameters on the output is analyzed, thereby completely disregarding an algorithm's internal working principle [13]. This makes agnostic approaches flexible and gives them a large user base. Table 2 compares salient characteristics of the five most common methods [40,41,68]. We now briefly discuss them in more detail.

First and foremost, there is the sensitivity analysis (SA), the general purpose of which is to examine how model responses behave with respect to changes in their input parameters. Put differently, it attributes response variability to the respective features. Both local and global approaches exist, such as a one-at-a-time SA or variance-based Sobol sensitivity indices [68,69]. While a SA shows to what extent the output of a machine learning model is affected by changing features, it says nothing about the actual form of the relationship. Such information can be obtained by analyzing the dependency of an outcome on its input parameters [40]. Typical examples are partial dependence plots by Friedman [35] and the closely related individual conditional expectation curves [24,40]. These graphs visualize how responses behave as a function of a feature: for instance, whether the nature of a relation is (non-)monotonic, (non-) linear, and so on [41]. Thirdly, the outcome of a surrogate is expressed as a function of input variables, so one may wonder which ones are more determinative than others. In this sense, examining the relevancy of features is a useful approach to understanding them. Fisher et al. [36] proposed to calculate their importances based on perturbations. The more a meta-model relies on a certain parameter, the greater its importance, and vice versa [24,34]. Next, there is LIME, which is short for local interpretable model-agnostic explanations. It originated from the research by Ribeiro et al. [37] and is currently one of the more popular approaches in the field of XAI. The core idea is to explain the reasoning behind black-box algorithms by means of fitting an auxiliary model that is locally faithful [40]. Logically, this model must be transparent and easy to interpret, such as linear regression or decision trees [41,37]. The explanations at a particular point in the domain are then presumed to be in accordance with the local behavior of the machine learning algorithm. Finally, we discuss LIME which is one of the more recent additions to XAI research. While the aforementioned agnostic interpretation methods do provide useful insights into the relationships between features and responses, the rationale behind a surrogate model's outcome remains unclear [24].

3.6. Model-specific analysis

Unlike model-agnostic, model-specific approaches explain the behavior of machine learning algorithms based on their internal properties [24,34]. Functional relationships between input and output parameters are used to interpret how responses come about. Consequently, the application of these methods is strongly limited to algorithms that provide access to such information. This is an evident drawback, but it should be borne in mind that agnostic approaches do not explicitly reveal the rationale behind an outcome [24]. They do explain how responses are affected, albeit on the basis of intermediate steps where proxies are the rule rather than the exception. For direct interpretations, one should resort to model-specific approaches. Given our selected architectures in Section 3.3, there are two methods that are deemed feasible: the analysis of polynomial regression weights and RuleFit for the tree-based ensembles. We now briefly elaborate on them, alongside a summary of salient characteristics in Table 3.

Linear regression, in addition to simplicity and accessibility, is often praised for its transparency. Indeed, one can easily understand how results are obtained and the linear structure enables decomposition, enhancing the overall interpretability [13,41]. The analysis is based on its coefficients: these can be interpreted as the effect on the response of one unit change of the corresponding feature [41]. Regression weights are in that sense global measures of sensitivity. However, it is crucial to understand that the coefficients have dimensions, and therefore an order of magnitude. This can be solved by standardization, which eliminates the scaling issues [68]. Secondly, tree-based machine learning algorithms are among the popular alternatives, mainly because of their impressive performance [53]. Notwithstanding, this comes at the expense of interpretability. They are usually so complex that users can no longer understand their reasoning. Friedman and Popescu [39] therefore suggested RuleFit to analyze decision rules. In essence, rules are extracted from the ensemble and then a regularized linear regression model is fitted with both the decision rules and direct feature effects [41]. The parameter value of a decision rule equals one if the condition is met, and zero otherwise. The further interpretation is similar to ordinary linear models, namely regression coefficients are analyzed, revealing the impact and direction of the corresponding influences on a response [41].

Table 3
Comparison between model-specific interpretation methods.

Method	Advantages	Disadvantages	Ref.
Regression Weights	Well-accepted and mature, does not require many calculations, transparent, accessible	Mediocre performance of the underlying model, multicollinearity can lead to fallacious conclusions	[68, 41,13]
RuleFit	Combined strengths of linear regression and tree ensembles: powerful without compromising transparency, flexible, yields a model	Overlapping decision rules deteriorate the explanatory power, often results in mediocre performance as it remains a linear model	[41, 39]

Finally, both the agnostic and specific methods must again be verified. Since one typically uses existing packages that have already been tested, the same applies as for the training and optimization step in Section 3.3. There is thus no need to retest them, although the integration itself still needs to be verified [44].

3.7. Triangulating interpretation methods

The second stage of the methodology concludes with validation, which aims to enhance the credibility of the overall model interpretation. Its main question is whether the results are in line with expectations and if not, how anomalies can be explained. However, unlike the surrogate model validation in Section 3.4, quantitative performance indicators are not suitable for such an assessment. Numerous insights from distinct methods are involved, so triangulation would be a better approach. Scholars describe it as a means of validation where the combination of different data sources, investigators, theories, etc. leads to research findings that are more reliable and therefore more credible [71,72]. It is based on the presumption that conclusions become stronger when a similar outcome is obtained from different perspectives [71].

In our case, one should qualitatively evaluate consistencies and inconsistencies between the results of considered meta-model interpretation techniques. Some specific examples could be: 1) comparing global sensitivity indices, feature importances, and regression weights, 2) assessing whether local methods such as LIME or a one-at-a-time SA match global patterns in dependency plots.

3.8. Synthesizing results to understand emergent properties

The final step is to reveal and understand emergent properties of the focal agent-based model by synthesizing insights from stage II. In practice, this step is often combined with the previous one from Section 3.7, although there is a crucial difference. That is, the purpose of triangulation is to assess consistencies and inconsistencies between the interpretation results, while a synthesis is to explain emergence in the system under investigation. Consequently, the latter yields a concise summary of perspectives from relevant model-agnostic and model specific methods.

Nonetheless, different perspectives are usually necessary to detect such phenomena. This is one of the reasons why the AbACaD methodology proposed by Janssen et al. [11] was rather successful; the authors combined causal graphs with regression weights and sensitivity indices, as discussed in Section 2.4. Such an approach enables one to read between the lines and use synergies to deeply understand what is actually going on in the agent-based model. A specific example with our selected interpretation methods could be as follows. First, several points in the feature space are examined with a local approach, such as LIME or a one-at-a-time SA. This gives an initial impression of how a response is affected and indicates whether there are mutual differences within the domain.

4. Description of the agent-based model

To demonstrate the applicability of our two-stage methodology, we aim to detect and explain emergent phenomena in a complex sociotechnical system. It follows from the theoretical background in Section 2 that a passenger terminal is the epitome of such a system: cognitive, social, technical, and organizational factors play a major role. With this in mind, Janssen et al. [14] recently developed an agent-based airport terminal operations model (AATOM) — existing alternatives were not accurate enough, too generic, too difficult to use, or the source code was not openly available. AATOM is designed with an object-oriented philosophy, allowing users to easily model the associated passenger flows. This makes it a very versatile tool, as it can be completely adapted to any set of requirements. One of its main features is that it contains prebuilt components, such as check-in desks or a security checkpoint [14]. Consequently, the layout of an entire terminal can be built with just a few lines of code. Since its introduction, the model has shown its capabilities in various studies. Some recent examples are Janssen et al. [73] on the relationship between checkpoint security and efficiency, Janssen

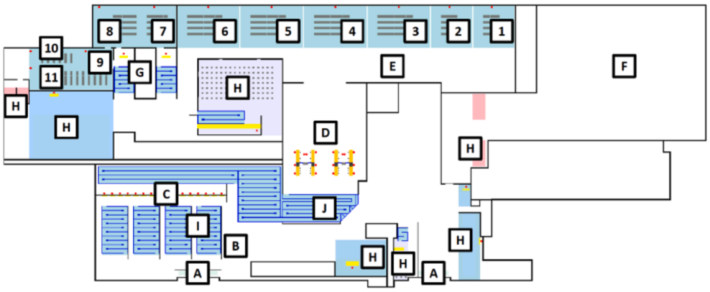


Fig. 2. Terminal layout of RTHA represented in the model. Passengers arrive through entrances (A) in the public area (B). Those who have not checked-in online can do so at the counters (C) via designated queues (I). Thereafter, all passengers continue via queues (J) to the security checkpoint (D) to access the restricted area. This area is split up into a departure hall (E) and an arrival hall (F). The arrival hall is not further developed as our research focuses on solely the outbound passenger flow. The departure hall has gates 1 to 6 for flights with destinations in the Schengen area and gates 7 to 11 for flights outside the Schengen area (the numbers on the map correspond to the gate numbers). To access the latter gates, passengers should go through border control to have their passports checked (G). Along the journey, passengers are free to make use of the facility areas for non-aeronautical activities (H) [14].

et al. [74] on the management of airport security risks, and Mekic et al. [6] with an analysis on non-aeronautical activities and their impact on terminal operations. We focus in particular on the latter, as this is currently the most advanced version to simulate the operations of an entire terminal.

An agent-based model is known to consist of an environment, agents, and interactions between them [75]. These three components of AATOM are now briefly discussed in respective order. First of all, the environment contains all elements of an airport terminal. That includes various functional areas with physical objects, but also more abstract items such as flights [76]. The former is depicted in Fig. 2, which resembles the terminal layout of Rotterdam The Hague Airport (RTHA). We specifically opted for RTHA due to the availability of data and associated insights from a previous study (see [77]). The layout was also available in the latest version by Mekic et al. [6], although some changes have been implemented. Regarding the flights, we consider a typical morning rush hour at RTHA. The schedule is summarized by Table 4. Secondly, cognitive agents are the key players in an environment.

Three types can be defined in AATOM: passengers, operators, and orchestrators [76]. The former is trivial, operators are generally the employees in the terminal (e.g., security officers, check-in staff, cashiers, etc.), and orchestrators help with coordination and monitoring (e.g., employees who open or close check-in counters based on an airport's strategy). Agents have certain goals on which they act and interact accordingly. Behind their reasoning is a three-layered hierarchical architecture, allowing AATOM to realistically model human behavior. The structure is visualized by Fig. 3. In essence, they operate as follows: observations are perceived and interpreted, allowing agents to reason so that their activities can be set. This eventually leads to the actuation of specific actions. The final component of AATOM is that agents can interact with the environment as well as with each other. The model reflects these two types of interaction in many different ways [76]. For example, check-in employees managing flights or security officers using sensors at the checkpoint are concrete cases of interaction between agents and the environment. On the other hand, border control agents checking passports or an X-ray operator instructing another security officer to further examine some suspicious baggage are examples of agent-to-agent interaction. For more detailed information, the reader is referred to Janssen et al. [76] and Janssen et al. [14] as the key principles behind the architecture of AATOM have now been touched upon.

Finally, relevant input and output parameters are discussed. Knowing that AATOM was created as a versatile tool, we emphasize the fact that essentially everything can be customized and adapted to the requirements of a user. Nevertheless, several calibrated presets are available to restrain complexity. Mekic et al. [6, pp. 20–21] made a comprehensive overview, though two examples are the distribution of the time required for checking-in at airport counters and the distribution of passengers arriving at the terminal. We use the defaults for most of these settings. The remaining features that are considered variables for our case study are listed in Table 5. The call-to-gate strategy is bounded between 15 and 60 min before departure, while more information about flights, check-in strategies and security checkpoint strategies are provided by Tables 4, 6-8, respectively.

5. Results

This section showcases how the methodology can be deployed in practice. We apply it to the AATOM implementation of Mekic et al. [6] and start with evaluating the surrogates' performance; a critical step to ensure they generalize sufficiently well before being used for further analysis. Subsequently, the departure flow in RTHA is examined on the basis of two specific case studies. First, the total expenditure of passengers on discretionary activities in the terminal is analysed, and then the saturation of throughput at security. Both cases are largely determined by emergence in airport terminals, so studying them will prove that our methodology is an effective way to explain such phenomena in complex sociotechnical systems — the objective of this research.

5.1. Surrogate model performance

Before assessing fidelity, one must first collect data and tune the surrogate model architectures. We visualize the distribution of selected data points, analyze summary statistics of the responses, and show how the stopping criterion of the active learning scheme is reached. It turns out that the training sample is sufficiently informative with 300 data points in total. This automatically leads to validation and test sets with both 100 additional observations, so that the proportion of each equals 20% of the entire sample.

Table 4
A typical flight schedule at RTHA in the fall of 2019. While technically based on assumptions, it resembles the morning rush hour at the airport, which is considered a busy period in the terminal. We take into account 7 flights from 3 airlines to both Schengen and Non-Schengen destinations. The check-in counters are in accordance with the terminal layout in Fig. 2 of the Scientific Paper; it goes from the first on the left to the 16th on the right. Furthermore, the number of passengers correspond to usual aircraft types at RTHA, with a flight being canceled if the occupancy is less than 50% of its total capacity. Finally, note that the simulation time in AATOM matches with the time slots. A simulation starts 3 h before the first flight and ends when the last flight departs.

Nr.	Slot (UTC)	Simulation time(s)	Airline	Destination	Gate	Check-in	Passengers
(1)	04:55	9000	A	Schengen	6	9–12	(75, 149)
(2)	04:55	9000	B	Schengen	1	1–4	(75, 149)
(3)	05:00	9300	B	Non- Schengen	7	13–16	(75, 149)
(4)	05:05	9600	C	Non- Schengen	9	13–16	(49, 98)
(5)	05:10	9900	B	Schengen	4	5–8	(75, 149)
(6)	05:30	11,100	B	Schengen	2	1–4	(75, 149)
(7)	05:45	12,000	B	Schengen	5	5–8	(75, 149)

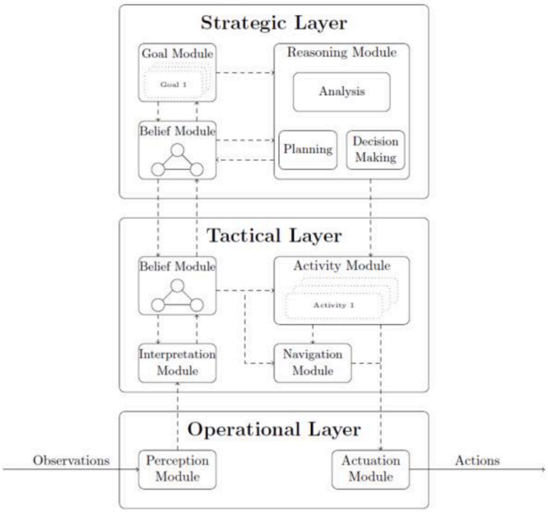


Fig. 3. Architectural layout of agents in AATOM [6]. In essence, they operate as follows: observations are perceived and interpreted, allowing agents to reason so that their activities can be set. This eventually leads to the actuation of specific actions.

Table 5
Considered input parameters of AATOM. A remark regarding the number of passengers is that Pax_t is defined for every available time slot t. In other words, if RTHA has 7 available time slots during the simulated time frame, the model requires 10 input parameters (i.e., 7 parameters to define the number of passengers and 3 strategy parameters).

Input parameter	Unit	Explanation
Pax_t	[#]	An integer indicating how many passengers are traveling on the flight on time slot *t*. It is strictly positive, bounded by the maximum capacity of an aircraft. If the occupancy rate is below 50%, it becomes 0 because the flight is canceled.
$\text{CTG}_{\text{strategy}}$	[s]	A positive real number that determines the time when passengers are called to their fate prior to the departure time. It represents the airport's call-to-gate (CTG) strategy [6].
$\text{CI}_{\text{strategy}}$	[-]	An integer that determines the number of open check-in counters over time. It represents the airport's check-in (CI) strategy. An orchestrator agent couples the number with a predefined strategy [6].
$\text{SC}_{\text{strategy}}$	[-]	An integer that determines the number of open lanes at the security checkpoint over time. It represents the airport's security check (SC) strategy. An orchestrator agent couples the number with a predefined strategy [6].

Table 6
Presumed check-in staffing strategies at RTHA. The table shows 9 possibilities of how many desks are available. This number can change over time, which is expressed in seconds prior to the departure of a flight. Note that passengers can only check-in between 2 h and 45 min before take-off. All flights of the schedule assume the same strategy over a certain simulated time frame.

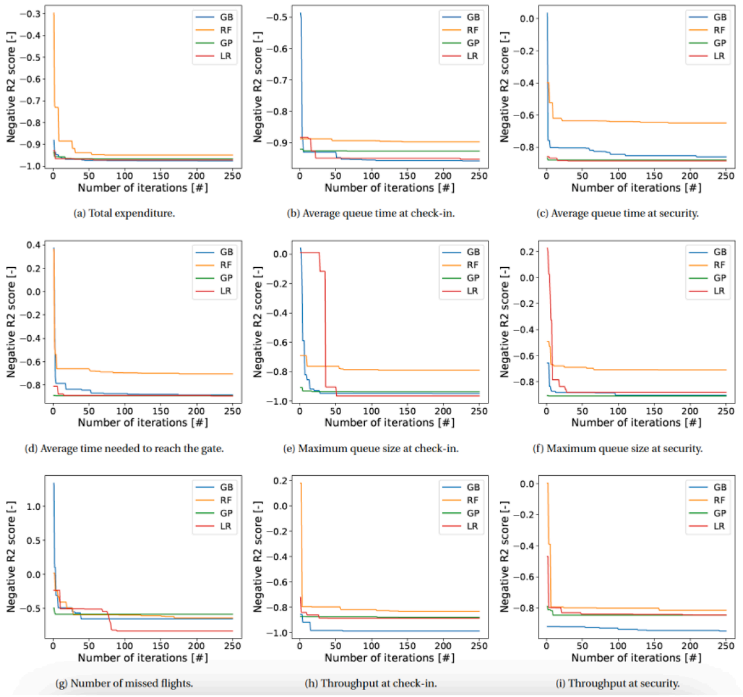
Available check-in counters as a function of time before departure of a flight							
Strategy	>7200	7200–6300	6300–5400	5400–4500	4500–3600	3600–2700	<2700
(1)	0	1	1	1	1	1	0
(2)	0	2	2	2	2	2	0
(3)	0	1	2	2	2	1	0
(4)	0	1	1	2	2	1	0
(5)	0	1	2	2	1	1	0
(6)	0	1	2	2	2	2	0
(7)	0	1	1	2	2	2	0
(8)	0	2	2	2	1	1	0
(9)	0	2	2	2	2	1	0

Secondly, hyperparameter tuning is done. Convergence plots showed that the algorithms can be deemed optimal after 50 initial trials and 200 subsequent Bayesian iterations. With that, the next step is to evaluate the surrogates' out-of-sample performance.

We perform validation in Table 9. Per output parameter of AATOM, the metrics are calculated for Gaussian process regression (GP), polynomial regression (LR), random forests (RF), and gradient boosting regression (GB) — the four selected meta-model architectures. The R2 and MAPE are dimensionless, but the RMSE and MAE are expressed in the same unit as the corresponding response, given in Table 8. Finally, the surrogate model that yields the best performance for each response is indicated by an asterisk.

The first thing that immediately stands out is the disappointing generalization power of random forests. Their performance is clearly inferior to the other architectures, often with quite a large discrepancy. The initial hypothesis was that overfitting posed the issue, although their accuracy no longer improves near the stopping criterion of the active learning algorithm, nor did regularization help. This is in stark contrast to LR, GP and GB, whose performance is actually rather impressive. While the validation metrics of these three architectures are generally comparable, regularized polynomials seem to mimic AATOM most accurately. Namely, they have been selected 7 out of 9 times, with only the throughput at check-in and security being better estimated by gradient boosting. This may be somewhat surprising, but a plausible explanation could be that the associated responses behave similarly according to the format of higher-order polynomials. Their combination then naturally produces superior results. For instance, the average queuing time at security is resembled with an expected error of about 1 min and the total expenditure with an error of about 40 euros. Only the number of missed flights appears to be more challenging: the coefficient of determination decreases to 0.80. Yet, even that is still acceptable, because it is the response most influenced by higher-order knock-on effects and less directly by the features themselves. Consequently, it becomes inherently more difficult to predict (see also the conclusions of De Leeuw et al. [42], which are consistent with our results). Furthermore, note the relative difference between the RMSE and MAE — the number of missed flights has the highest of all, indicating the presence of outliers. In spite of that, LR convincingly remains the best performing architecture for the response, while the tree-based ensembles are inadequate. Altogether, LR, GP and GB seem to mimic the output parameters of AATOM rather well, despite the fact that RTHA is a complex sociotechnical system. However, there is evidently "no free lunch", as multiple architectures must be considered and carefully optimized per individual parameter to achieve a high accuracy [78].

After understanding the hyperparameters defined for each model according to [59], the next step is to prepare the Bayesian optimization algorithm. The high-level principle has been explained in Theoretical background and Methodology Sections. Briefly, however, a meta-model is fitted onto a prediction error metric of the machine learning model as a function of its hyperparameters. Then, an acquisition function proposes new parameters to try, so that after some iterations the combination is found that yields the highest accuracy. We opt for the traditional choice, as the machine learning models in [59] do not have an excessive dimensionality, while the vast majority of hyperparameters are numerical. Secondly, one should decide on the acquisition function. Archetti and Candelieri [12] discuss three common possibilities, being the probability of improvement, the expected improvement, and the lower confidence bound. The first is one of the oldest and prefers exploitation over exploration. It does not really incorporate the extent of the improvement, which is why scholars came up with the second option—the most popular alternative up to date [49]. However, the expected improvement still favors exploitation. Furthermore, recall that the error metric to fit the meta-model on was previously set to the coefficient of determination, as argued in theoretical background Section. The prediction error is evaluated on the independent validation set, the sole purpose of which was to optimize the hyperparameters. Finally, there remains the number of iterations that must be performed. In principle, the sequential optimization algorithm continues until the coefficient of determination of the machine learning model does not improve further. It is thus rather difficult to decide on this in advance. Moreover, it can also differ per response and model. We therefore start with an estimate of 50 randomly selected parameter combinations for the initial set, after which 200 additional iterations are performed by the Bayesian optimizer. In the next paragraph will be determined whether or not that is sufficient.



(a) Total expenditure.

(b) Average queue time at check-in.

(c) Average queue time at security.

(d) Average time needed to reach the gate.

(e) Maximum queue size at check-in.

(f) Maximum queue size at security.

(g) Number of missed flights.

(h) Throughput at check-in.

(i) Throughput at security.

Fig. 4. Convergence plots of the hyperparameter tuning process towards the global optimum.

The optimization process can be easily monitored by creating convergence plots. They are presented in Fig. 4, for each of the nine responses. In essence, the graphs show how the performance of machine learning models improves over a number of iterations. Note that all four architectures are combined in one plot per response.

The coefficient of determination is on the vertical axis. It is the negative score, as the implementation in scikit-optimize requires a minimization problem [31]. Consequently, accuracy increases with a lower value for the negative R^2. Sometimes it happens that the indicator is positive in the beginning of the optimization. That is, the coefficient of determination is actually negative. This is correct and does not indicate an issue with the underlying computation, but it means that the machine learning model is severely underperforming [59]. On the horizontal axis, there are the number of iterations. The first 50 are the randomly selected hyperparameter values for the initial set, followed by 200 iterations of the Bayesian optimizer. That makes a total of 250. Each trial shows the score corresponding to the best parameter combination of the attempts so far. Thus, the convergence graphs either decrease or remain constant; they never rise. From the plots is clear that the biggest improvements are realized in the beginning. However, this is not surprising, since hyperparameters are arbitrarily chosen from the search space at that time. Hence, it is likely that one of these trials turns out to be rather decent, which explains why there can be a sudden gain in accuracy at some point before 50 iterations. The Bayesian optimization algorithm takes over after that, which results in many smaller decrements. Step by step, it searches for the global optimum. Whether it actually manages to do so is hard to say, although it is evident that the curves stagnate near the end. This suggests that most of the advancements in model performance are realized. Therefore, we conclude that 50 initial random selections and 200 subsequent Bayesian updates are sufficient. Any minor improvements after that would not outweigh the additional computational requirements. Finally, it is worth noting that random forests seem to be underperforming almost consistently: the discrepancy to the other models is oftentimes rather large. This may be somewhat surprising. GB, GP and LR convergence towards more or less comparable scores, except for the number of missed flights and the throughput at both check-in and security. Respectively, the higher-order polynomial and twice the gradient boosting algorithm turn out to be exceptionally better than the others in those cases. Nonetheless, keep in mind that the optimization results in Fig. 4 are evaluated on the validation and not on the test set. Hence, these insights can be misleading regarding the actual performance of the machine learning models.

After 250 iterations, the machine learning models are considered to be optimal. These are the thus the four models for each response that deliver the best possible performance. The next step is then to select only one of the four architectures per response. Of course, that is the model which can most accurately mimic the response in question. Notwithstanding, one might be interested in the ultimate hyperparameter combinations to which the Bayesian optimization algorithm has converged. The result is presented in Table 10 — only the combinations of the selected machine learning models are included for the sake of the report. A higher-order polynomial is chosen for the first seven responses. Most of them have a degree of three, except for the total expenditure and the average waiting time at security, for which four was found to be better. Also, they all seem to benefit from regularization. The vast majority of their regression coefficients are zero because of the selected values for α. The two responses with a fourth-order polynomial have only 148 and 307 non-zero terms against a total of 91,389 coefficients, while the others have respectively 136, 212, 112, 142, and 184 non-zero terms against a total of 9138. It is thus clear that the resulting polynomials are sparse: the final number of non-zero terms does not explode, despite having rather high degrees. In other words, the models reap the benefits from adding higher-order complexity without experiencing too many of the associated drawbacks. Unnecessary terms are marginalized by regularization, which in this case can actually be seen as a form of feature (and feature interaction) selection. Finding the right balance between the degree and α is not a straightforward task, but the Bayesian optimization algorithm seems to handle it well. In fact, this approach turns out to be quite powerful for meta-modeling AATOM's responses, as it was chosen seven times out of nine. For the remaining two responses, the throughput at check-in and security, gradient boosting was the preferred machine learning architecture. The squared loss function appears to be the best option for both cases, along with fairly low learning rates and a large number of trees. So, the influence of new trees in the sequence is limited, although many are added to compensate for that. Furthermore, the throughput at check-in requires at least 14.2% of the training sample before splitting a node, but otherwise there are not many requirements regarding a number of data points at internal nodes or leaves. They are all very close to or at the lower bound of their search space, which essentially means that the trees are not constrained by these hyperparameters. Hence, at first glance it seems that trees are allowed to be fully grown. This may be somewhat surprising, as it severely deteriorates the performance of gradient boosting regression in general [79].

Besides visualizing how the input parameters were chosen over their domain, it is also interesting to take a closer look at the corresponding responses. Doing so provides a first insight into the output of AATOM, which will be meta-modeled in the current research. The most relevant summary statistics of the training sample are tabulated in Table 11. From left to right, there is the mean response, its 95% confidence interval (abbreviated as CI), the standard deviation (abbreviated as SD), and the minimum, median and maximum value of the response. The averages are in line with the expectations. It is interesting to see their interconnections: the average time it takes to reach the gate from entering the terminal, for example, includes the waiting time at security. For those who have not checked in beforehand, the waiting time at check-in is also included. These results show that, on average, the majority of time in the terminal is spent at security. Furthermore, note that the throughput at security is higher than the one at the check-in counters. However, this is very logical. The latter only counts passengers who do so at the airport itself, while everyone has to go through the security checkpoint—there is no way around that. Subsequently, the numbers of passengers who completed checkin and security have the smallest confidence interval and the number of missed flights the largest, relative to the mean. The same can be said of the standard deviation, although the result for the latter response is rather extreme. Namely, the standard deviation of the number of missed flights is more than twice as high as its mean. This shows a high degree of dispersion in the response, which must be right-tailed since it cannot be negative for obvious reasons. Furthermore, the minimum value is zero for all output parameters. While this may seem strange at first, it actually makes sense because flights are canceled if their occupancy rate is less than 50%. Hence, there is a scenario where this applies to all flights, resulting in zero agents in the airport terminal. Such a situation is of course not very plausible in practice, although it is technically possible. In the end, airlines still have the flexibility — albeit limited — not to use their allocated time slots, which are fixed.

Flight cancellations therefore remain an option to ensure the versatility of the surrogate models. Next, the median is another interesting statistic, and especially its discrepancy with the mean. It shows to what extent responses are influenced by extrema. We only consider those whose median is outside the 95% confidence interval of the mean, which is true for all except throughput at security and total expenditure. Apart from the throughput at check-in, the remaining responses seem to have a lower median than the average. The number of missed flights is skewed rather extreme, as its median value equals the minimum. This means that in at least 50% of the simulated scenarios, everyone catches their flight. Finally, there are the maxima in the last column, which show no peculiarities. The only response with an exceptionally high maximum is again the number of missed flights. That nearly 190 passengers may not be at their gate on time during the simulated schedule is a lot. However, the reason behind it is evident as it must be caused by a very busy routine with extremely poor airport terminal strategies.

Even though the summary statistics in Table 11 show the most important properties of the responses at a glance, they are all subject to different scales. It is therefore rather difficult to compare them with one another. This can be resolved by standardization and plotting the results in box plots, as can be seen in Fig. 5. The insights from before are now also directly visible in these graphs. For example, the minimum and median of the number of missed flights do indeed coincide, resulting in a strongly right-tailed distribution. Furthermore, apart from the latter, the interquartile ranges of the remaining responses are more or less comparable. They are not extremely dispersed and while there may be degrees of skewness, the median is never really far from the average. However, it is rather interesting that most of the responses' skew is caused by the values outside the interquartile range. This is especially visible in the throughput at the check-in counters and at security, which show heavy left-tails. Other than that, the waiting time at security and the associated maximum number of passengers in the queue generally appear to be the least variable. All in all, there is certainly some heterogeneity between the responses, albeit not extreme. The distribution for the number of missed flights is skewed the most, so it will be interesting to see how this affects the meta-model performance.

Finally, in addition to statistics and standardized visualizations, it is also relevant to check whether the responses are correlated. This shows how they are influenced altogether and proves the presence of significant relationships, if any. A correlation matrix is presented in Table 12, along with their p-values. First and foremost, the vast majority of correlation coefficients are rather high, indicating strong positive relationships. These results are not surprising in principle, though some are very high. For example, the correlation be-tween the average time to reach the gate and the average queue time at security is almost perfect. The latter is a direct part of the former, so it does make sense, but also implies that waiting time at security is the main determinant of the total time spent in the airport terminal. Notwithstanding, one must beware of the fallacy of questionable causes [66]. The fact that there is correlation does not necessarily mean that one is caused by the other. An example is the relation between the throughput at security and the maximum number of passengers in the check-in queues. One can see that they are moderately correlated with a coefficient of 0.56. Practically speaking, however, the two responses do not have much in common. Instead, they are both simultaneously affected by the number of passengers in the terminal. The more agents in the system, the higher the two responses are likely to be, which explains the correlation. Hence, the coefficients ought to be interpreted as such: it shows how the input parameters influence AATOM's responses in relation to one another. Furthermore, note that none of the output parameters are negatively correlated with a significance level of at least 5%, and that the total expenditure has the least overall correlation. Other than very weak positive relationships with the throughput at check-in and security, the response has no significant coefficients, making it rather standalone. Lastly, a remarkable finding is the result between the number of missed flights and the security checkpoint's throughput. They are almost unrelated, while there is a weak correlation with the throughput at check-in. This may seem strange, but it actually makes a lot of sense. Think of it as follows. The busier it gets, the more agents that have to pass through check-in and security. The same goes for those who end up

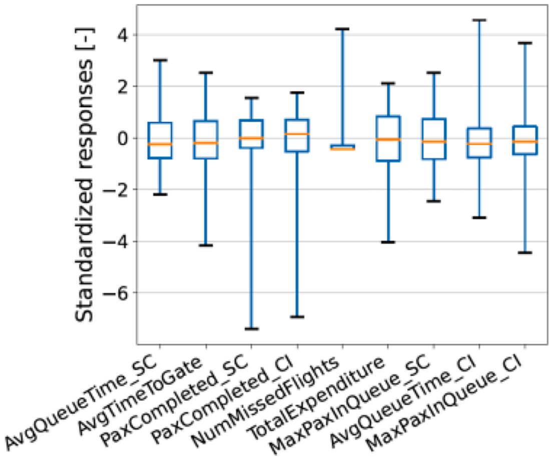


Fig. 5. Standardized box plots of the responses obtained from the training sample.

missing their flight, but one should realize that this is most likely caused by long waiting times at the checkpoint—note the fairly strong correlation between these two. Indeed, the main bottleneck of the system is at security, which becomes saturated as the crowd grows. At that point, people will start to miss their flight while the throughput remains constant. The checkpoint cannot handle more passengers than its maximum capacity allows. With this knowledge, in combination with the fact that the check-in counters can achieve a much higher throughput in the current terminal layout, one can explain the difference in correlation coefficients with the number of missed flights. These were the most relevant insights from the matrix. Although the results were not surprising per se, they do provide a thorough understanding into how the responses behave in relation to one another.

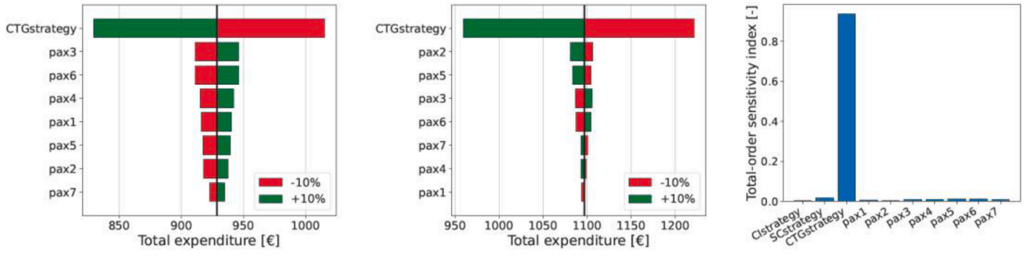
5.2. *Analysis of the total expenditure on discretionary activities*

The first case study examines the spending behavior of passengers on non-aeronautical activities. In fact, this was the main topic of the analysis by Mekic et al. [6], though we go more in-depth to demonstrate the strengths of synthesizing interpretation techniques applied to surrogate models. The total expenditure represents the amount of money all passengers together spent on activities such as shopping, dining, etc., during the simulated time frame. These events are of course not mandatory to catch a flight and hence not a priority, so passengers will only consider them if they have enough time. Readily, it shows that the expenditure is an ideal starting point to analyze emergence; the indicator is affected by various interdependent phenomena in the airport terminal.

We commence the analysis in Fig. 6 by exploring the sensitivity of features. First, two one-at-a-time assessments are performed in Fig. 6a and 6b, which depict tornado diagrams of local sensitivities. An uncrowded scenario is compared against a crowded one, both assuming poor airport staffing strategies (check-in and security check strategy 1) and an early call-to-gate (48 min before departure). The crowd is controlled by adopting a load factor of about 65% and 85% on all flights, respectively. Poor staffing strategies in combination with an early call-to-gate does not provide the ideal condition for discretionary activities — passengers have less spare time in the terminal. Nevertheless, the baseline values of the tornado diagrams suggest that busier scenarios lead to more spending. This makes sense, as larger crowds are naturally expected to have a higher expenditure. Both diagrams associate the greatest sensitivity to the call-to-gate strategy, though note that it has a negative direction. In other words, the sooner passengers are called to the gate, the less they spend along their journey and vice versa. While this is not surprising, a more striking difference is the influence of certain flights' load factor. They are all harmonious for the uncrowded terminal, but not when it gets busier. For flights 2 and 5 in particular, the total expenditure decreases as more passengers travel on those flights. This is rather counterintuitive, so we resort to other methods to explain the negative effect and why it depends on the scenario. We conclude the sensitivity analysis by plotting total-order Sobol indices in Fig. 3c. They attribute the variance of a response to the features in proportion to their contribution, so total expenditure appears to be most sensitive to the call-to-gate strategy. The global impression thus corresponds to the local impressions, although it is more pronounced.

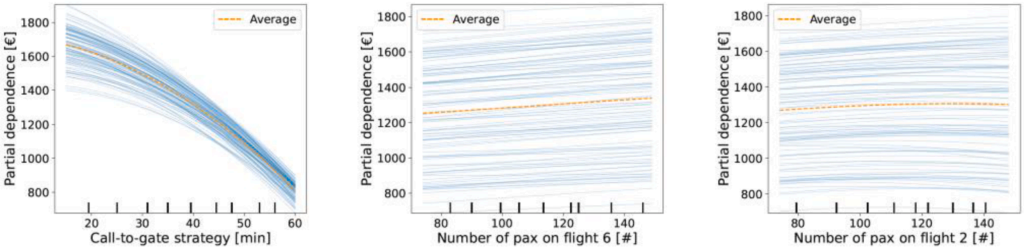
In Fig. 7, the analysis continues with partial dependence plots, which visualize the marginal effect of a feature on the total expenditure. On the left, Fig. 7a shows the effect of the call-to-gate strategy. As expected, the curve has a downward slope, indicating a lower spending for higher strategies. More interestingly, however, is that the gradient increases the sooner passengers are called to their gate. This suggests that the total expenditure is less sensitive to the feature in the lower part of its domain. To illustrate with a specific example, changing the call-to-gate strategy from 20 to 30 min will reduce the spending less than changing it from 50 to 60 min. Secondly, Fig. 7b and 7c depict the marginal effects of the number of passengers on flight 6 and 2, respectively. Both plots are consistent with the previous one-at-a-time sensitivity analysis. For flight 6, it holds that larger occupancy rates yield more expenditure in the terminal; the partial dependence rises almost linearly and there is a solid homogeneity among the individual conditional expectation curves. The same cannot be said of flight 2, where spending first rises, then levels off, and eventually falls back slightly. This indeed indicates that at some point, adding passengers may have a negative effect on the ability to engage in non-aeronautical activities. Note, however, the increased heterogeneity near the upper bound of the feature space. It does not seem to be case for all scenarios in the terminal. For example, if all the other flights have very few passengers, then it is probably the other way around. Nonetheless, some flights clearly hinder a number of passengers from enjoying leisure activities during their journey.

Certain flights, such as the second and sixth, are assigned to the same check-in counters, although that in itself is nothing unusual. Yet, it is remarkable that flight 2 and 5, which mostly affect expenditure in the negative direction according to Fig. 7b, are both first in line. This could indicate the presence of emergent knock-on phenomena where passengers of one flight hinder those of another, but



(a) Local sensitivity uncrowded terminal. (b) Local sensitivity crowded terminal. (c) Total-order global sensitivity.

Fig. 6. Sensitivity analysis of the total expenditure.



(a) Marginal effect of the call-to-gate.

(b) Marginal effect of flight 6.

(c) Marginal effect of flight 2.

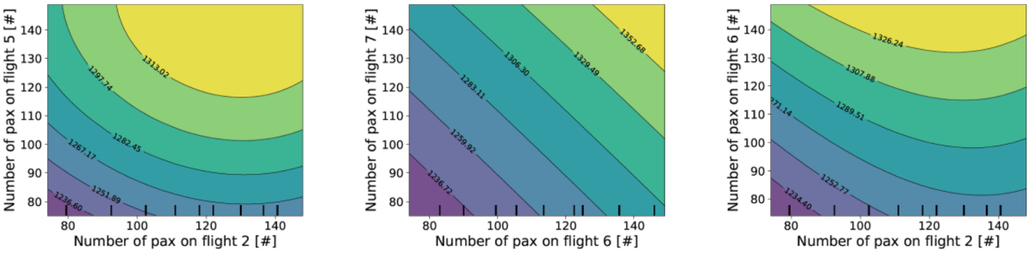
Fig. 7. One-dimensional partial dependence and individual conditional expectation plots of the total expenditure.

only if the terminal is sufficiently crowded. In turn, waiting times at check-in and security increase, ultimately leading to fewer opportunities for discretionary activities. That would be a logical explanation for the observed pattern in the partial dependence plot of flight 2, and also why it is not the case for others like flight 6. The third flight goes against this logic, although note the lower capacity on its successor — the knock-on effect is therefore presumably less powerful. To prove whether our reasoning is actually true, we further examine the partial dependence in Fig. 8, but now from a two-dimensional perspective.

5.3. *Analysis of the saturation at security*

The second case study originates from the observation that throughput at security and the number of missed flights are not correlated at all, while correlation between the latter and average waiting time at security is 0.80 ($p < 0.01$). The analysis commences again with a local method, but this time with insights from LIME instead of a one at-a-time sensitivity analysis. Fig. 9 explains how the throughput at security is influenced. We compare an uncrowded terminal against a crowded one; both scenarios presume good staffing strategies and an early call-togate, the values of which are shown in the graphs. A trivial conclusion is that high load factors lead to positive contributions to the number of passengers passing through security, and vice versa. Furthermore, note that constantly operating the checkpoint at full capacity — security check strategy 16 — positively impacts the total flow. That is logical, as it delivers the best possible service, thereby maximizing throughput. Notwithstanding, one should remain vigilant about the discrepancy between predictions of LIME and the surrogate. The error is around 30 passengers for both scenarios, which is rather large compared to the size of the feature impacts.

Interdependent relationships are best understood by plotting the partial dependence, so Fig. 10 visualizes three relevant cases. The marginal effect of flights 2 and 6 on the number of passengers who completed security is depicted in Fig. 10a. First of all, note that decision boundaries are not as smooth as in previous dependence plots. This makes sense because it is one of the two responses that is mimicked by gradient boosting rather than by a polynomial; the outcome of the former is inherently more erratic. Secondly, at a given occupancy rate on flight 6, the response levels off when the number of passengers on the second flight exceeds 120–130. That indeed seems to confirm our earlier hypothesis: at some point, the security checkpoint becomes saturated, hence adding passengers no longer increases throughput. The phenomenon is not visible in flight 6, which is logical as it is much later in the schedule — obstructing the checkpoint is therefore less likely. In fact, its load factor and the throughput practically have a positive linear relationship. These findings are also apparent in the one-dimensional counterparts, depicted by Figs. 11a and 11b. A natural follow-up question is then what happens to passengers who failed to complete the security check on time. Fig. 11b shows the consequence by plotting the marginal effect of flights 2 and 6 on the total number of missed flights. The graph confirms our expectations, namely from about 120 passengers on flight 2, more and more people do not reach their gate on time. Flight 6 also has a slight impact, albeit rather limited because contour lines in the critical region are almost vertical. Finally, Fig. 11c shows that the effect of flight 2 even accelerates as it approaches its maximum capacity, suggesting that there must be some degree of knock-on phenomena. The sixth flight does not appear to be affected, as otherwise it would have been visible in Fig. 11b through more horizontal contours in the upper part. Nonetheless, flight 2 clearly has a considerable impact on whether or not the checkpoint may become obstructed. Note, however, the heterogeneity

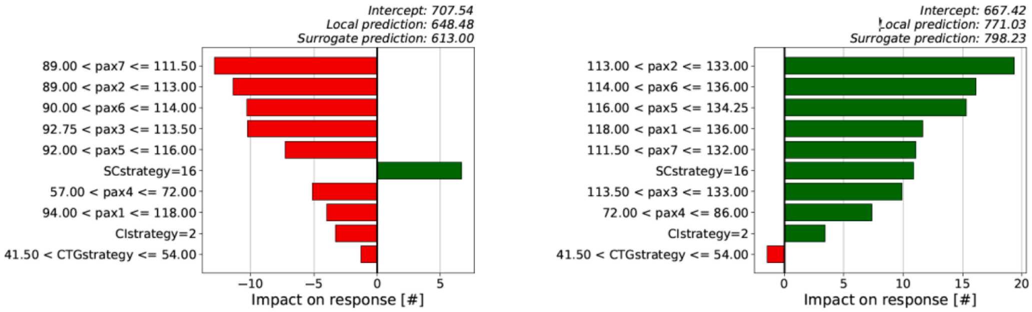


(a) Marginal effect of flight 2 and 5.

(b) Marginal effect of flight 6 and 7.

(c) Marginal effect of flight 2 and 6.

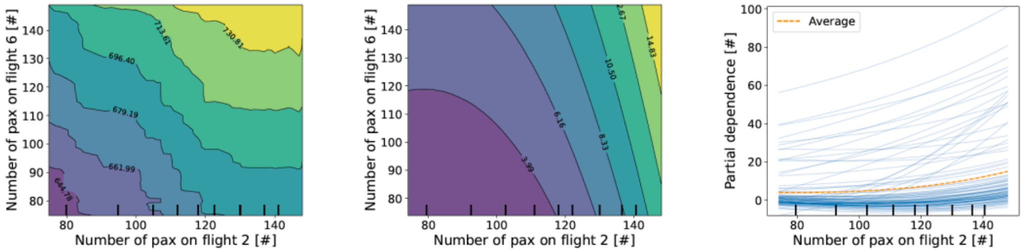
Fig. 8. Two-dimensional partial dependence plots of the total expenditure.



(a) Uncrowded terminal.

(b) Crowded terminal.

Fig. 9. Local interpretable model-agnostic explanations of the throughput at security. The interpretation is as follows. The vertical axis shows all input parameters and their assumed values, while the corresponding contributions to the response are plotted horizontally. These contributions can be considered as the impact on a prediction relative to the intercept of LIME's approximation. Bars appear red if the effect is negative and green otherwise. To connect the dots, LIME arrives at its local prediction by adding the individual contributions of all features to the intercept, which should then be close to the actual outcome of the investigated surrogate model.

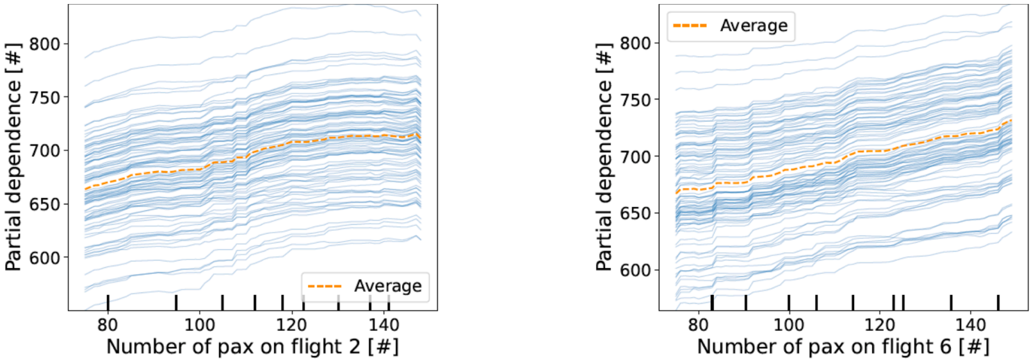


(a) Marginal effect of flight 2 and 6 on the throughput at security.

(b) Marginal effect of flight 2 and 6 on the number of missed flights.

(c) Marginal effect of flight 2 on the number of missed flights.

Fig. 10. Partial dependence plots of the throughput at security and number of missed flights.



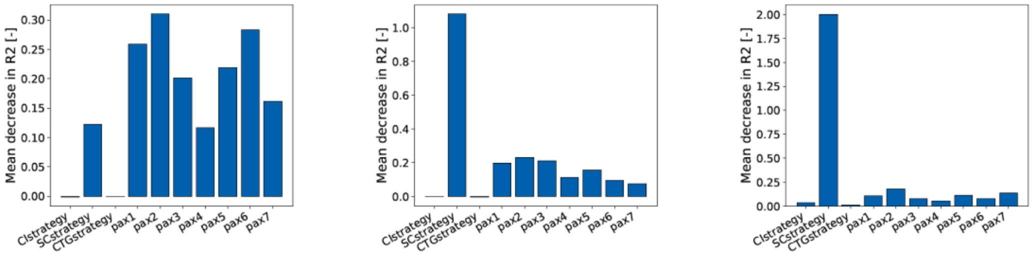
(a) Marginal effect of flight 2.

(b) Marginal effect of flight 6.

Fig. 11. One-dimensional partial dependence plot of the throughput at security.

among the individual conditional expectation curves in Fig. 11c. This proves that the conditions in the terminal remain important. For example, if it is not busy at all, then adding passengers to solely flight 2 will not necessarily increase the number of missed flights.

Next, we also analyze feature importances to see exactly which key drivers control the checkpoint's throughput, average waiting time, and the ensuing number of missed flights. The results are shown in Fig. 12, respectively. It is immediately noticeable that the graphs of the latter two are similar; both are driven primarily by the staffing strategy at security and to a lesser extent by occupancy rates. The opposite holds for the checkpoint's throughput, although the difference is not as pronounced. One should interpret these results as follows. Under normal circumstances, more passengers lead to more passage through security, which is therefore mainly determined by the load factor on flights. However, if the airport opts for a bad strategy, waiting times may increase considerably. The extent also depends on how busy it is, but personnel strategy is more decisive. That is logical, as it directly dictates the number of lanes



(a) Throughput at security.

(b) Average queuing time at security.

(c) Number of missed flights.

Fig. 12. Visualization of permutation feature importances.

to be opened. Ceteris paribus, fewer lanes will always lead to longer waiting times, but not necessarily to a lower throughput. This explains the difference between Figs. 12a and 12b. However, there is a risk that the waiting time, which we know is predominately driven by strategy, will continue to rise so passengers are no longer able to reach their gate on time. At that point, the number of missed flights will increase rapidly, especially when it is busy. Queuing time and the number of missed flights thus have the same key drivers. These findings are confirmed by plotting the marginal effect of the security check strategy on the three respective responses in Fig. 13. There is clearly a strong influence of certain strategies; especially those that close all but one lane after two hours appears to be really detrimental (see Table 7). While the relative effect on throughput may seem limited, note that Fig. 13a and 13c are actually complementary. Hence, passengers who failed to pass security on time, will eventually miss their flight. Based on these insights, we conclude that there are indeed emergent phenomena which can lead to saturation of the security checkpoint. If so, waiting times may increase substantially, resulting in a high number of missed flights. This particularly happens in combination with poor staffing strategies.

6. Discussion

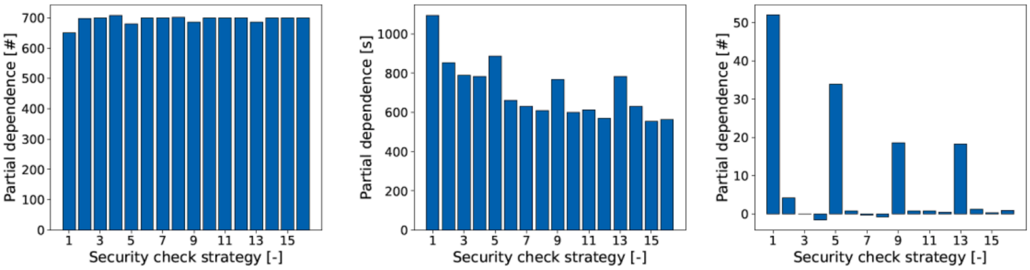
From the results, it is evident that the proposed methodology is rather effective in explaining the dynamics of complex socio-technical systems, based on an existing validated agent-based simulation model. The main presumption was that one already has access to such a faithful model, though we remind the reader not to take this for granted. In practice, it will not always be the case and alternatives such as obtaining real-word data are often infeasible. Moreover, one must also be aware of imperfections in the model. AATOM is known to be meticulously close to reality, but there are always certain assumptions. For example, Mekic et al. [6] mention passengers' perfect sense of time, the neglect of boarding delays, and so on. Surrogate models are directly subject to these limitations.

The surrogates were created in stage I of the methodology. While the resulting out-of-sample performance is rather impressive, it is in line with expectations. Namely, our validation metrics are comparable to those of De Leeuw et al. [42], who meta-modeled similar responses of AATOM.

It is crucial to highlight that our investigation specifically focused on the departure flow in the terminal of RTHA, which represents a significantly more complex system compared to the one examined by the authors mentioned in [42]. Thus, it is important to note that increased complexity does not necessarily lead to a decline in surrogate model performance. It is worth mentioning our utilization of an advanced sampling approach, as active learning has proven to be highly effective [25,26]. Interestingly, the authors [42] mentioned earlier achieved high accuracy using random forests, which contradicts our findings. In our study, however, we found that random forests exhibited notably weak meta-modeling performance. There are two potential explanations for this discrepancy: the heightened complexity of the system we analyzed or a variation in the sampling strategy, as our methodological steps closely resembled theirs. Additionally, it is crucial not to underestimate the predictive capability of regularized higher-order polynomials, as they demonstrated superiority in 7 out of 9 cases. This emphasizes the significance of considering the strength of such models in predictive analysis.

This may be somewhat surprising at first, but we emphasize that they are known to perform particularly well when responses behave like polynomials [9]. In addition, they provide analytical expressions of these responses as a function of the input parameters. Such transparency is considered a major strength, especially if the ultimate purpose is to gain insight into underlying relationships.

Yet, one cannot always rely on transparency to explain the rationale behind surrogates. Stage II of the methodology therefore introduced a more generic approach, combining both model-agnostic and model-specific interpretation methods. Emergent phenomena were clearly observed and the first case study is accordant with the conclusions of Mekic et al. [6]. For example, delaying the call-to-gate does indeed increase total expenditure, as does shortening waiting times at security. However, our methodology allowed us to go further in depth. We could even attribute certain phenomena down to load factors on specific flights. Stage II thus provides a unique insight into actual root causes, which is a considerable strength. For the synthesis, mainly model agnostic methods were used. They are more accessible and easier to apply, while the model-specific methods were prone to the following weaknesses: 1) RuleFit proved to be very unstable, and 2) it was rather difficult to materialize polynomial regression weights into concrete implications. Regarding the latter, regularization may have drastically reduced the number of non-zero coefficients, the resulting expressions still consist of more than 100 terms each. This remains challenging to interpret, so transparency does not always promote explainability. The downside thereof is that we have not explicitly revealed how exactly a surrogate arrives at its outcome. Agnostic methods do clarify the behavior of responses, but not the internal reasoning of a model. Next, a weakness of stage II is finding the right illustrations



(a) Throughput at security.

(b) Average queuing time at security.

(c) Number of missed flights.

Fig. 13. Marginal effect of the security check strategy.

Table 7
Presumed security checkpoint staffing strategies at RTHA. The table shows 16 possibilities of how many lanes are available. This number can change over time, which is expressed in seconds over the simulation time. There is always at least 1 lane staffed, although between 30 min and 2 h the minimum increases to first 2 and then 4 lanes. The reason for this is to ensure a minimum available throughout, because otherwise the terminal may become so crowded that it no longer resembles reality.

Available lanes at security as function of the overall simulation time							
Strategy	<1800	1800–3600	3600–5400	5400–7200	7200–9000	9000–10,800	>10,800
(1)	1	2	4	4	1	1	1
(2)	1	2	4	4	3	2	1
(3)	1	2	4	4	4	3	2
(4)	1	2	4	4	4	4	4
(5)	1	3	4	4	1	1	1
(6)	1	3	4	4	3	2	1
(7)	1	3	4	4	4	3	2
(8)	1	3	4	4	4	4	4
(9)	2	4	4	4	1	1	1
(10)	2	4	4	4	3	2	1
(11)	2	4	4	4	4	3	2
(12)	2	4	4	4	4	4	4
(13)	4	4	4	4	1	1	1
(14)	4	4	4	4	3	2	1
(15)	4	4	4	4	4	3	2
(16)	4	4	4	4	4	4	4

Table 8
Relevant key performance indicators of AATOM.

Output parameter	Unit	Explanation
AvgQueueTime$_{CI}$	[s]	Indicates the average time that passengers wait in a queue until they can be served at an available check-in (CI) counter.
AvgQueueTime$_{SC}$	[s]	Indicates the average time that passengers wait in a queue until they can be served at a security checkpoint (SC) lane.
MaxPaxInQueue$_{CI}$	[#]	Indicates the maximum queue size at check-in during the simulated time frame.
MaxPaxInQueue$_{SC}$	[#]	Indicates the maximum queue size at security during the simulated time frame.
AvgTimeToGate	[s]	Indicates the average time it takes passengers to get to their gate. It is counted from the moment they arrive at the airport.
PaxCompleted$_{CI}$	[#]	Indicates the total number of passengers that have completed the check-in (CI) activity at the airport counters (i.e., throughput at check-in).
PaxCompleted$_{SC}$	[#]	Indicates the total number of passengers that have completed the security (SC) activity at the checkpoint (i.e., throughput at security).
NumMissedFlights	[#]	Indicates the total number of passengers who could not reach their gate at the time of departure.
TotalExpenditure	[€]	Indicates the total amount of money that all passengers together have spent during their non-aeronautical activities [6].

to include in the analysis. By considering several approaches, numerous results are obtained, from which a careful selection must be made. We found a one-at-a-time sensitivity analysis, LIME, and correlation coefficients particularly useful to pinpoint interesting directions. From there, partial dependence plots turned out to be the preferred alternative when visualizing relevant dynamics, along with support from Sobol indices and feature importances to identify key drivers. One final remark is that LIME raises questions about its reliability, despite providing comprehensible insights.

To generalize above findings, the scope of the two-stage methodology is not limited to solely agent-based models, nor to airport terminal operations. Firstly, it can be applied to essentially any type of model, although preferably one that entails a heavy computational burden. Otherwise, there are probably more efficient approaches, without the need to create surrogates as an intermediate step. That being said, it is not obligatory to use the methodology in its entirety. The two stages are in fact modular and may be deployed

Table 9
Validation of the surrogate model performance.

PaxCompleted$_{SC}$					AvgTimeToGate				PaxCompleted$_{CI}$			
Metric	GP	LR	RF	GB*	GP	LR*	RF	GB	GP	LR	RF	GB*
R^2	0.90	0.90	0.79	0.93	0.91	0.92	0.55	0.89	0.93	0.94	0.86	0.98
RMSE	17.52	17.62	25.75	14.94	64.32	61.34	143.80	72.14	7.55	7.40	11.17	4.14
MAE	12.59	12.86	20.29	11.51	48.11	46.19	113.45	58.27	5.02	5.66	8.11	3.30
MAPE	0.02	0.02	0.03	0.02	0.04	0.04	0.09	0.05	0.01	0.02	0.02	0.01
AvgQueueTime$_{SC}$					NumMissedFlights				TotalExpenditure			
GP		LR*	RF	GB	GP	LR*	RF	GB	GP	LR*	RF	GB
R^2	0.90	0.92	0.57	0.86	0.70	0.80	0.53	0.43	0.97	0.98	0.94	0.97
RMSE	68.63	63.64	142.90	80.68	9.28	7.51	11.58	12.85	52..05	42.44	69.23	52.25
MAE	52.50	49.54	118.74	64.33	7.09	4.42	6.94	6.60	40.34	33.76	55.61	42.00
MAPE	0.08	0.08	0.18	0.10	N/A†	N/A†	N/A†	N/A†	0.03	0.03	0.04	003
AvgQueueTime$_{CI}$					MaxPaxInQueue$_{SC}$				MaxPaxInQueue$_{CI}$			
GP		LR*	RF	GB	GP	LR*	RF	GB	GP	LR*	RF	GB
R^2	091	0.95	0.87	0.95	0.91	0.92	0.65	0.91	0.90	0.95	0.78	0.92
RMSE	19.46	14.13	23.73	15.13	9.15	8.69	17.99	9.06	0.58	0.43	0.86	0.39
MAE	13.79	9.78	16.16	10.25	6.60	6.69	14.59	7.24	0.43	0.29	0.66	0.39
MAPE	0.05	0.04	0.06	0.04	0.07	0.06	0.14	0.07	0.04	0.02	0.05	0.03

[*] Best performing surrogate model architecture for the associated response.
[†] Mathematically undefined because of division by zero.

Table 10
Selected surrogate model hyperparameter values per response.

Response (selected model)	Hyperparameter	Optimal value
Total expenditure (LR)	Degree	4
	α	0.494
Average queue time at check-in (LR)	Degree	3
	α	0.096
Average queue time at security (LR)	Degree	4
	α	0.217
Average time needed to reach the gate (LR)	Degree	3
	α	0.224
Maximum queue size at check-in (LR)	Degree	3
	α	0.005
Maximum queue size at security (LR)	Degree	3
	α	0.079
Number of missed flights (LR)	Degree	3
	α	0.013
Throughput at check-in (GB)	Loss function	Squared
	Learning rate	0.045
	Number of trees	990
	Required data points at a split	0.142
	Required data points at a leaf	0.006
	Considered features for a split	0.691
	Subsample size	0.257
Throughput at security (GB)	Loss function	Squared
	Learning rate	0.017
	Number of trees	1000
	Required data points at a split	0.001
	Required data points at a leaf	0.001
	Considered features for a split	0.969
	Subsample size	0.250

separately. For example, if accurate machine learning models are already available, one can go directly to the second stage. Conversely, if there is a mere desire to speed up the calculation process, then the first stage will suffice. The combination of both stages is thus only recommended if the ultimate purpose is to enhance the understanding of a particular model, for which conventional approaches fall short due to computational limitations. Secondly, we have applied the methodology to airport terminal operations, but its applicability certainly goes further. The only requirements are that responses should be numerical in nature and there must be a meaningful relationship between input and output parameters, direct or indirect. The former because promising interpretation methods for classification were not incorporated, such as in the case study of Belle and Papantonis [40]. The latter because otherwise it is rather difficult to visualize interdependencies and explain the overall dynamics of a system. A final consideration is whether it is

Table 11
Summary statistics of the responses calculated from the training sample.

Response	Unit	Mean	95% CI	SD	Minimum	Median	Maximum
AvgQueueTime_SC	[s]	808.61	[766.92, 850.29]	366.89	0.00	718.34	1915.81
AvgTimeToGate	[s]	1379.96	[1342.30, 1417.63]	331.50	0.00	1317.14	2216.35
PaxCompleted_SC	[#]	682.54	[672.06, 639.01]	92.22	0.00	681.90	825.65
PaxCompleted_CI	[#]	356.72	[350.89, 362.56]	51.36	0.00	364.62	447.20
NumMissedFlights	[#]	17.65	[13.07, 22.23]	40.29	0.00	0.00	187.47
TotalExpenditure	[€]	1211.85	[1177.81, 1245.88]	299.55	0.00	1191.48	1841.82
MaxPaxInQueue_SC	[#]	125.41	[119.60, 131.22]	51.16	0.00	118.44	254.44
AvgQueueTime_CI	[s]	275.53	[265.42, 285.63]	88.94	0.00	254.55	681.44
MaxPaxInQueue_CI	[s]	11.94	[11.64, 12.25]	2.69	0.00	11.55	21.81

Table 12
Correlation matrix of the responses obtained from the training sample.

Response	1	2	3	4	5	6	7	8
1.AvgQueueTime_SC								
2. AvgTimeToGate	0.98**							
3.PaxCompleted_SC	0.47**	0.53**						
4.PaxCompleted_CI	0.75**	0.78**	0.92**					
5.NumMissedFlights	0.80**	0.74**	0.01	0.41**				
6.TotalExpenditure	−0.11	0.02	0.18**	013**	−0.10			
7.MaxPaxQueue_SC	0.92**	0.91**	0.70**	0.86**	0.57**	−0.07		
8.AvgQueueTime_CI	0.36**	0.45**	0.17**	0.31**	0.41**	−0.08	0.31**	
9.MaxPaxInQueue_CI	0.62**	0.70**	0.56**	0.70**	0.47	−0.04	0.66**	0.86**

*Note: p <0.05.
** p <0.01.

desirable to have solely one surrogate model for a response across the entire feature space. For example, we found the number of missed flights more challenging to mimic. The predominant influence of higher-order interaction effects definitely plays a role, but the response is also zero for the vast majority of scenarios in the terminal. This makes sense, because missing a flight due to overcrowding is actually quite rare. It seems that surrogates cannot handle such patterns as well as agent-based models, so a better alternative may be to first distinguish between nominal and non-nominal scenarios, and then consider the respective regions of the feature space individually. A proof of concept is presented by ten Broeke et al. [43], where classification was applied first, followed by regression. To conclude, our proposed methodology is fairly generic, although there are boundaries to its applicability within which the best results are achieved.

Finally, we address two important limitations. On the one hand, a critical assumption is that responses are deemed deterministic — their stochasticity was averaged out by the law of large numbers. This is of course not true in reality and can even affect the emergence and knock-on effects in the system. On the other hand, the Hammersley sequence was selected as the initial sampling method. Based on our experience, we do not recommend this for future work. The reason is that in some exceptional cases, its beneficial properties only appear when the sample is large enough with respect to the number of features. While related issues were largely resolved by the subsequent active learning algorithm, another method, such as Latin hypercube sampling, could have prevented them in the first place [80,79]. Having discussed the implications of our methodology, the research is finalized with a conclusion and further recommendations.

7. Conclusions and future work

The motivation for our research originates from the observation that existing airport terminal operations models: 1) suffer from heavy computational requirements, and 2) reveal their emergent properties only a posteriori. These are typical challenges of agent-based modeling, the principle according to which they are usually built. Therefore, we introduced a two-stage methodology to analyze such systems in a more efficient way. The first stage involves the development of faithful surrogate models, whereafter the second stage applies techniques from the emerging field of explainable artificial intelligence to these abstractions. The novelty of our methodology lies thus in the amalgamation, rather than in the respective research fields themselves. Indeed, we have explored their common ground to take advantage of synergies. A successful application reveals the properties of the focal system, which in the case of a sociotechnical system mainly concerns emergent phenomena.

Proof of the methodology’s efficacy is provided by conducting two case studies on AATOM; a validated agent-based airport terminal operations model. On the one hand, we looked at the total expenditure on noncompulsory activities, like shopping and dining. It was found that the journey of some passengers may be disturbed in such a way there is an effect on their spending behavior. Knock-on phenomena were observed, with travelers from earlier flights holding up those from later flights at check-in and security. Consequently, less free time is left to engage in discretionary activities. It happens especially when the terminal is busy in combination with poor airport staffing strategies. This is a clear example of emergence, the root causes of which could even be associated to specific strategies and the occupancy on certain flights. On the other hand, we also examined the throughput at security. More passengers mean more passage, but there is an evident point where the checkpoint reaches its maximum capacity. As a result, throughput remains constant, while the queue and therefore the waiting time quickly increase. This even goes so far as to put passengers at risk of missing their flight. Again, unequivocal evidence of emergent properties, which are thus clearly preserved in surrogates. The key drivers of the phenomenon could also be traced back, along with the critical settings; it only occurs under certain conditions. Altogether, the case studies demonstrated that the proposed methodology is indeed able to accurately abstract and explain the dynamics of airport terminal operations through surrogate modeling an existing simulation model. This confirms the research objective and emphasizes the strengths of combining meta-modeling with interpretable machine learning. Unique and detailed insights were attained into properties and relationships that would otherwise have been very difficult to reveal due to computational limitations.

We conclude the research by recommending three promising directions for future work, based on our experience. First of all, it is believed that responses can be mimicked more accurately if an advance distinction is made between certain scenarios in the airport terminal. For example, one could distinguish between nominal and non-nominal situations, as mentioned in the discussion. Their associated dynamics are rather diverse, which raises the question of the best approach to meta-model disparate and rare events — a particularly relevant question for scholars interested in safety and security related issues. Secondly, we solely looked at averages or total values of responses, but it is evident that their behavior may change over the simulated time frame. Such changes can be quite considerable; think of the waiting time at security, which is strongly influenced by the flight schedule. Hence, including time as an extra dimension could reveal properties that were previously invisible. Doing so will certainly add complexity, yet it is crucial to enhance the understanding of a sociotechnical system even further. Our final recommendation relates to the interpretation of machine learning models. While the discipline is still emerging, most effort is clearly being put into model-agnostic methods. That makes sense because they have a broader applicability, though a model's explicit internal reasoning can only be elucidated through model-specific approaches. We therefore advocate further development of the latter.

Data availability

The data that has been used is confidential.

Acknowledgments

This research was funded by the 'European Regional Development Fund (ERDF) via the Kansen voor West II program' under the project 'Airport Technology Lab', grant number 'KVW-00235′.

References

[1] R. Alizadeh, J.K. Allen, F. Mistree, Managing computational complexity using surrogate models: a critical review, Res. Eng. Des. 31 (3) (2020) 275–298, https://doi.org/10.1007/s00163-020-00336-7. ISSN 0934-9839, https://link.springer.com/10.1007/s00163-020-00336-7.

[2] A. Graham, Managing Airports: An International Perspective, 4th edition, Routledge, New York, 2013. ISBN 978-0-415-52941-9.

[3] T. Patel and W. Wilkes. Strikes and labor shortages leave European airports in chaos. Bloomberg, June 2022. URL https://www.bloomberg.com/news/articles/2022-06-09/the-travel-boom-has-caught-airlines-still-in-bust-mode-off-guard.

[4] B. Timmins, K. Austin, Heathrow flight cancellations cause queues and 'chaos, BBC News (2022). https://www.bbc.com/news/business-61857008.

[5] P. Pao-YenWu, K. Mengersen, A review of models and model usage scenarios for an airport complex system, Transp. Res. A 47 (2013) 124–140, https://doi.org/10.1016/j.tra.2012.10.015. ISSN 0965-8564, https://linkinghub.elsevier.com/retrieve/pii/S0965856412001541.

[6] A. Mekic, S.S. Mohammadi Ziabari, A. Sharpanskykh, Systemic agent-based modeling and analysis of passenger discretionary activities in airport terminals, Aerospace 8 (6) (2021) 162, https://doi.org/10.3390/aerospace8060162. https://www.mdpi.com/2226-4310/8/6/162.

[7] A. Bhosekar, M. Ierapetritou, Advances in surrogate-based modeling, feasibility analysis, and optimization: a review, Comput. Chem. Eng. 108 (2017) 250–267, https://doi.org/10.1016/j.compchemeng.2017.09.017. ISSN 0098- 1354, https://linkinghub.elsevier.com/retrieve/pii/S0098135417303228.

[8] A.I.J. Forrester, A. S.bester, A.J. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, 1st edition, Wiley, 2008. ISBN 978-0-470-06068-1, https://onlinelibrary.wiley.com/doi/book/10.1002/9780470770801.

[9] S. Razavi, B.A. Tolson, D.H. Burn, Review of surrogate modeling in water resources, Water Resour. Res. 48 (7) (2012), https://doi.org/10.1029/2011WR011527. ISSN 1944-7973, https://onlinelibrary.wiley.com/doi/abs/10.1029/2011WR011527.

[10] F. Lamperti, A. Roventini, A. Sani, Agent-based model calibration using machine learning surrogates, J. Econ. Dyn. Control 90 (2018) 366–389, https://doi.org/10.1016/j.jedc.2018.03.011. ISSN 0165-1889, https://linkinghub.elsevier.com/retrieve/pii/S0165188918301088.

[11] S. Janssen, A. Sharpanskykh, R. Curran, K. Langendoen, Using causal discovery to analyze emergence in agent-based models, Simul. Modell. Pract. Theory 96 (2019), 101940, https://doi.org/10.1016/j.simpat.2019.101940. ISSN 1569190X, https://linkinghub.elsevier.com/retrieve/pii/S1569190X19300735.

[12] K. Cheng, Z. Lu, C. Ling, S. Zhou, Surrogate-assisted global sensitivity analysis: an overview, Struct. Multidiscip. Optim. 61 (3) (2020) 1187–1213, https://doi.org/10.1007/s00158-019-02413-5. ISSN 1615-147X, http://link.springer.com/10.1007/s00158-019-02413-5.

[13] A. Barredo Arrieta, N. D.az-Rodr.guez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable Artificial Intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115, https://doi.org/10.1016/j.inffus.2019.12.012. ISSN 1566-2535, https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[14] S. Janssen, A. Sharpanskykh, R. Curran, K. Langendoen, AATOM: an agent-based airport terminal operations model simulator, in: Proceedings of the 2019 Summer Simulation Conference (SummerSim '19), Assoc Computing Machinery, Berlin, 2019, p. 12.

[15] D. Curcio, F. Longo, G. Mirabelli, E. Pappoff, Passengers flow analysis and security issues in airport terminals using modeling & simulation. ECMS 2007, ECMS, 2007, pp. 374–379, https://doi.org/10.7148/2007-0374. ISBN 978-0- 9553018-2-7, https://www.scs-europe.net/dlib/2007/2007-0374.htm.

[16] S. Kalakou, F. Moura, Analyzing passenger behavior in airport terminals based on activity preferences, J. Air Transp. Manag. 96 (2021), https://doi.org/10.1016/j.jairtraman.2021. 102110. ISSN 0969-6997, https://linkinghub.elsevier.com/retrieve/pii/S0969699721000934.

[17] V. Tosic, A review of airport passenger terminal operations analysis and modelling, Transp. Res. A 26 (1) (1992) 3–26, https://doi.org/10.1016/0965-8564(92)90041-5. ISSN 0965-8564, https://linkinghub.elsevier.com/retrieve/pii/0965856492900415.

[18] K.C. James, M. Bhasi, Development of model categories for performance improvement studies related to airport terminal operations, J. Simul. 4 (2) (2010) 98–108, https://doi.org/10.1057/jos.2009.27. ISSN 1747-7778, https://www.tandfonline.com/doi/full/10.1057/jos.2009.27.

[19] L. Magalh.es, V. Reis, R. Mac.rio, A new methodological framework for evaluating flexible options at airport passenger terminals, Case Stud. Transp. Policy 8 (1) (2020) 76–84, https://doi.org/10.1016/j.cstp.2018.03.003. ISSN 2213-624X, https://linkinghub.elsevier.com/retrieve/pii/S2213624X18300749.

[20] IATA. Airport Development Reference Manual. Montreal, 9th edition, 2004. ISBN 978-92-9195-086-7.

[21] I.E. Manataki, K.G. Zografos, Development and demonstration of a modeling framework for airport terminal planning and performance evaluation, Transp. Res. Rec. 2106 (1) (2009) 66–75, https://doi.org/10.3141/2106-08. ISSN 0361-1981, https://journals.sagepub.com/doi/10.3141/2106-08.

[22] C. Macal, M. North, Tutorial on agent-based modeling and simulation, in: Proceedings of the Winter Simulation Conference, 2005, 2005, pp. 2–15, https://doi.org/10.1109/WSC.2005.1574234. ISSN: 1558-4305.

[23] B. Pietzsch, S. Fiedler, K.G. Mertens, M. Richter, C. Scherer, K. Widyastuti, M.C. Wimmler, L. Zakharova, U. Berger, Metamodels for evaluating, calibrating and applying agent-based models: a review, J. Artif. Soc. Soc. Simul. 23 (2) (2020) 9, https://doi.org/10.18564/jasss.4274. ISSN 1460-7425, https://jasss.soc.surrey.ac.uk/23/2/9.html.

[24] R. Elshawi, M.H. Al-Mallah, S. Sakr, On the interpretability of machine learning-based model for predicting hypertension, BMC Med. Inform. Decis. Mak. 19 (1) (2019) 146, https://doi.org/10.1186/s12911-019-0874-0. ISSN 1472-6947, https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-019-0874-0.

[25] H. Liu, Y.S. Ong, J. Cai, A survey of adaptive sampling for global metamodeling in support of simulation-based complex engineering design, Struct. Multidiscip. Optim. 57 (1) (2018) 393–416, https://doi.org/10.1007/s00158-017-1739-8. ISSN 1615-147X, https://link.springer.com/10.

[26] J.N. Fuhg, A. Fau, U. Nackenhorst, State-of-the-art and comparative review of adaptive sampling methods for kriging, Arch. Comput. Meth. Eng. 28 (4) (2021) 2689–2747, https://doi.org/10.1007/s11831-020-09474-6. ISSN 1134-3060, https://link.springer.com/10.1007/s11831-020-09474-6.

[27] G.G. Wang, S. Shan, Review of metamodeling techniques in support of engineering design optimization, J. Mech. Des. 129 (4) (2007) 370–380, https://doi.org/10.1115/1.2429697. ISSN 1050-0472, https://asmedigitalcollection.asme.org/mechanicaldesign/article/129/4/370/466824/Review-of-Metamodeling-Techniques-in-Support-of.

[28] P. Westermann, R. Evins, Surrogate modelling for sustainable building design a review, Energy Build. 198 (2019) 170–186, https://doi.org/10.1016/j.enbuild.2019.05.057. ISSN 0378-7788, https://www.sciencedirect.com/science/article/pii/S0378778819302877.

[29] R. De Neufville, A. Odoni, P. Belobaba, T. Reynolds, Airport Systems: Planning, Design, and Management, 2nd edition, McGraw-Hill, New York, 2013. ISBN 978-0-07-177058-3.

[30] T. Van Steenkiste, J. van der Herten, I. Couckuyt, T. Dhaene, Data-efficient sensitivity analysis with surrogate modeling. Uncertainty Modeling for Engineering Applications, PoliTO Springer SeriesSpringer International Publishing, Cham, 2019, pp. 55–69, https://doi.org/10.1007/978-3-030-04870-9_4. ISBN 978-3-030-04870-9.

[31] D.P. Kuttichira, S. Gupta, C. Li, S. Rana, S. Venkatesh, Explaining black-box models using interpretable surrogates, in: PRICAI 2019: Trends in Artificial Intelligence, 11670, Springer International Publishing, Cham, 2019, pp. 3–15, https://doi.org/10.1007/978-3-030-29908-8_1. ISBN 978-3-030-29907-1, https://link.springer.com/10.1007/978-3-030-29908-8_1.

[32] C. N.brega, L. Marinho, Towards explaining recommendations through local surrogate models, in: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, ACM, Limassol Cyprus, 2019, pp. 1671–1678, https://doi.org/10.1145/3297280.3297443. ISBN 978-1-4503-5933-7, https://dl.acm.org/doi/10.1145/3297280.3297443.

[33] R. Roscher, B. Bohn, M.F. Duarte, J. Garcke, Explainable machine learning for scientific insights and discoveries, IEEE Access 8 (2020) 42200–42216, https://doi.org/10.1109/ACCESS.2020.2976199. ISSN 2169-3536.

[34] M. Naser, An engineer's guide to eXplainable Artificial Intelligence and Interpretable Machine Learning: navigating causality, forced goodness, and the false perception of inference, Autom. Constr. 129 (2021), https://doi.org/10.1016/j.autcon.2021.103821. ISSN 0926-5805, https://linkinghub.elsevier.com/retrieve/pii/S0926580521002727.

[35] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. 29 (5) (2001) 1189–1232. ISSN 0090-5364, http://www.jstor.org/stable/2699986.

[36] A. Fisher, C. Rudin, F. Dominici, All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously, J. Mach. Learn. Res. 20 (177) (2019) 1–81. ISSN 1533-7928, http://jmlr.org/papers/v20/18-760.html.

[37] M.T. Ribeiro, S. Singh, C. Guestrin, Why should I trust you?": explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 1135–1144, https://doi.org/10.1145/2939672.2939778. ISBN 978-1-4503-4232-2, http://doi.org/10.1145/2939672.2939778.

[38] S.M. Lundberg, S.I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 30, Curran Associates, Inc., Long Beach, CA, USA, 2017, in: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

[39] J.H. Friedman, B.E. Popescu, Predictive learning via rule ensembles, Ann. Appl. Stat. 2 (3) (2008), https://doi.org/10.1214/07-AOAS148. ISSN 1932-6157, https://projecteuclid.org/journals/annals-of-applied-statistics/volume-2/issue-3/Predictive-learning-via-rule-ensembles/10.1214/07-AOAS148.full.

[40] V. Belle, I. Papantonis, Principles and practice of explainable machine learning, Front. Big Data 4 (2021) 39, https://doi.org/10.3389/fdata.2021.688969. ISSN 2624-909X, https://www.frontiersin.org/article/10.3389/fdata.2021.688969.

[41] C. Molnar, Interpreappend Machine Learning, 2nd edition, Lulu, 2019. ISBN 978-0-244-76852-2, https://christophm.github.io/interpretable-ml-book/.

[42] B. De Leeuw, S.S. Mohammadi Ziabari, and A. Sharpanskykh. Surrogate modeling of agent-based airport terminal operations. In Multi-Agent-Based Simulation XXIII, Auckland, New Zealand, Mar. 2022. URL https://mabsworkshop.github.io/articles/MABS_2022_paper_9.pdf.

[43] G. ten Broeke, G. van Voorn, A. Ligtenberg, J. Molenaar, The use of surrogate models to analyse agent-based models, J. Artif. Soc. Soc. Simul. 24 (2) (2021) 3, https://doi.org/10.18564/jasss.4530. ISSN 1460-7425, https://jasss.soc.surrey.ac.uk/24/2/3.html.

[44] B. Hailpern, P. Santhanam, Software debugging, testing, and verification, IBM Syst. J. 41 (1) (2002) 4–12, https://doi.org/10.1147/sj.411.0004. ISSN 0018-8670, https://ieeexplore.ieee.org/document/5386906.

[45] F. Dekking, C. Kraaikamp, H. Lopuha, L. Meester, A Modern Introduction to Probability and Statistics: Understanding Why and How. Springer Texts in Statistics, Springer, London, 2005. ISBN 978-1-85233-896-1.

[46] D.S. Fay, A Biologist's Guide to Statistical Thinking and Analysis, WormBook, 2013, pp. 1–54, https://doi.org/10.1895/wormbook.1.159.1. ISSN 15518507, https://www.wormbook.org/chapters/www_statisticalanalysis/statisticalanalysis.html.

[47] T.T. Wong, W.S. Luk, P.A. Heng, Sampling with Hammersley and Halton points, J. Graph. Tools 2 (2) (1997) 9–24, https://doi.org/10.1080/10867651.1997.10487471. ISSN 1086-7651, https://www.tandfonline.com/doi/abs/10.1080/10867651.1997.10487471.

[48] T.W. Simpson, D.K.J. Lin, W. Chen, Sampling strategies for computer experiments: design and analysis, Int. J. Reliab. Appl. 2 (3) (2001) 209–240. https://www.personal.psu.edu/users/j/x/jxz203/lin/Lin_pub/2001_IJRA.pdf.

[49] J.L. Loeppky, J. Sacks, W. J.Welch, Choosing the sample size of a computer experiment: a practical guide, Technometrics 51 (4) (2009) 366–376, https://doi.org/10.1198/TECH.2009.08040. ISSN 0040-1706, https://www.tandfonline.com/doi/abs/10.1198/TECH.2009.08040.

[50] F. Archetti, A. Candelieri, Bayesian Optimization and Data Science, Springer International Publishing, Cham, 2019, https://doi.org/10.1007/978-3-030-24494-1. Springer Briefs in Optimization ISBN 978-3-030-24494-1, https://link.springer.com/10.1007/978-3-030-24494-1.

[51] C.Q. Lam, Sequential Adaptive Designs In Computer Experiments For Response Surface Model Fit, Ohio State University, 2008. PhD thesis URL, https://rave.ohiolink.edu/etdc/view?acc_num=osu1211911211.

[52] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning: Data Mining, Inference, And Prediction. Springer Series in Statistics, 2nd edition, Springer, New York, NY, 2009. ISBN 978-0-387-84857-0.

[53] A. G.ron, Hands-On Machine Learning With Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, Volume 2th, O'Reilly Media, Sebastopol, CA, 2019. ISBN 978-1-4920-3264-9.

[54] L. Jia, R. Alizadeh, J. Hao, G. Wang, J.K. Allen, F. Mistree, A rule-based method for automated surrogate model selection, Adv. Eng. Inf. 45 (2020), 101123, https://doi.org/10.1016/j.aei.2020.101123. ISSN 1474-0346, https://linkinghub.elsevier.com/retrieve/pii/S1474034620300926.

[55] R. Yondo, E. Andr.s, E. Valero, A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses, Prog. Aerosp. Sci. 96 (2018) 23–61, https://doi.org/10.1016/j.paerosci.2017.11.003. ISSN 0376-0421, https://www.sciencedirect.com/science/article/pii/S0376042117300611.

[56] J.H. Friedman, Multivariate adaptive regression splines, Ann. Stat. 19 (1) (1991) 1–67, https://doi.org/10.1214/aos/1176347963. ISSN 0090-53642168-8966, http://projecteuclid.org/journals/annals-of-statistics/volume-19/issue-1/Multivariate-Adaptive-Regression-Splines/10.1214/aos/1176347963.full.

[57] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes For Machine Learning. Adaptive computation and Machine Learning, MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9.

[58] E. Schulz, M. Speekenbrink, A. Krause, A tutorial on Gaussian process regression: modelling, exploring, and exploiting functions, J. Math. Psychol. 85 (2018) 1–16, https://doi.org/10.1016/j.jmp.2018.03.001. ISSN 0022-2496, https://linkinghub.elsevier.com/retrieve/pii/S0022249617302158.

[59] B. Williams, S. Cremaschi, Selection of surrogate modeling techniques for surface approximation and surrogate-based optimization, Chem. Eng. Res. Des. 170 (2021) 76–89, https://doi.org/10.1016/j.cherd.2021.03.028. ISSN 0263-8762, https://www.sciencedirect.com/science/article/pii/S0263876221001465.

[60] F. Hutter, J. Lücke, L. Schmidt-Thieme, Beyond manual tuning of hyperparameters, Künstl. Intell. 29 (4) (2015) 329–337, https://doi.org/10.1007/s13218-015-0381-0. ISSN 0933-1875, https://link.springer.com/10.1007/s13218-015-0381-0.

[61] R. Andonie, Hyperparameter optimization in learning systems, J. Membr. Comput. 1 (4) (2019) 279–291, https://doi.org/10.1007/s41965-019-00023-0. https://link.springer.com/10.1007/s41965-019-00023-0.

[62] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: theory and practice, Neurocomputing 415 (2020) 295–316, https://doi.org/10.1016/j.neucom.2020.07.061. ISSN 0925-2312, https://linkinghub.elsevier.com/retrieve/pii/S0925231220311693.

[63] T. Yu and H. Zhu. Hyper-Parameter Optimization: a Review of Algorithms and Applications. arXiv:2003.05689 [cs, stat], Mar. 2020. URL https://arxiv.org/abs/2003.05689.

[64] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Taking the human out of the loop: a review of bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175, https://doi.org/10.1109/JPROC.2015.2494218. ISSN 1558-2256.

[65] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830. https://jmlr.csail.mit.edu/papers/volume12/pedregosa11a/pedregosa11a.pdf.

[66] T. Head, M. Kumar, H. Nahrstaedt, G. Louppe, and I. Shcherbatyi. Scikit-Optimize: sequential model based optimization in Python, Oct. 2021. URL https://zenodo.org/record/5565057.

[67] D. Chicco, M.J. Warrens, G. Jurman, The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation, PeerJ Comput. Sci. 7 (2021), https://doi.org/10.7717/peerj-cs.623. ISSN 2376-5992, https://peerj.com/articles/cs-623.

[68] E. Borgonovo, E. Plischke, Sensitivity analysis: a review of recent advances, Eur. J. Oper. Res. 248 (3) (2016) 869–887, https://doi.org/10.1016/j.ejor.2015.06.032. ISSN 0377-2217, https://linkinghub.elsevier.com/retrieve/pii/S0377221715005469.

[69] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: The Primer, John Wiley, Chichester, England, 2008. ISBN 978-0-470-05997-5.

[70] M.T. Ribeiro, S. Singh, and C. Guestrin. Model-agnostic interpretability of machine learning. arXiv:1606.05386 [cs, stat], June 2016. URL https://arxiv.org/abs/1606.05386.

[71] H. Noble, R. Heale, Triangulation in research, with examples, Evid. Based Nurs. 22 (3) (2019) 67–68, https://doi.org/10.1136/ebnurs-2019-103145. ISSN 1367-65391468-9618, https://ebn.bmj.com/lookup/doi/10.1136/ebnurs-2019-103145.

[72] V.A. Thurmond, The point of triangulation, J. Nurs. Scholarsh. 33 (3) (2001) 253–258, https://doi.org/10.1111/j.1547-5069.2001.00253.x. ISSN 1527-65461547-5069, https://onlinelibrary.wiley.com/doi/10.1111/j.1547-5069.2001.00253.x.

[73] S. Janssen, A. Sharpanskykh, R. Curran, Agent-based modelling and analysis of security and efficiency in airport terminals, Transp. Res. C 100 (2019) 142–160, https://doi.org/10.1016/j.trc.2019.01.012. ISSN 0968-090X, https://linkinghub.elsevier.com/retrieve/pii/S0968090X1830809X.

[74] S. Janssen, A. Sharpanskykh, R. Curran, AbSRiM: an agent-based security risk management approach for airport operations, Risk Anal. 39 (7) (2019) 1582–1596, https://doi.org/10.1111/risa.13278. ISSN 1539-6924, https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.13278.

[75] U. Wilensky, W. Rand, An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo, MIT Press, Cambridge, MA, USA, 2015. ISBN 978-0-262-73189-8.

[76] S. Janssen, A.N. Blok, A. Knol, AATOM - An Agent-Based Airport Terminal Operations Model, Delft University of Technology, Apr. 2018. https://research.tudelft.nl/en/publications/aatom-an-agent-based-airport-terminal-operations-model.

[77] S. Janssen, R. van der Sommen, A. Dilweg, A. Sharpanskykh, Data-driven analysis of airport security checkpoint operations, Aerospace 7 (6) (2020) 69, https://doi.org/10.3390/aerospace7060069. ISSN 2226-4310, https://www.mdpi.com/2226-4310/7/6/69.

[78] D.H. Wolpert, What is important about the no free lunch theorems? in: P.M. Pardalos, V. Rasskazova, M.N. Vrahatis (Eds.), Black Box Optimization, Machine Learning, and No-Free Lunch Theorems, Springer Optimization and Its Applications Springer International Publishing, Cham, 2021, pp. 373–388, https://doi.org/10.1007/978-3-030-66515-9_13. ISBN 978-3-030-66515-9.

[79] L. Kocis, W.J. Whiten, Computational investigations of low-discrepancy sequences, ACM Trans. Math. Softw. 23 (2) (1997) 266–294, https://doi.org/10.1145/264029.264064. ISSN 0098-3500, https://dl.acm.org/doi/10.1145/264029.264064.

[80] F.A. Viana. Things you wanted to know about the Latin hypercube design and were afraid to ask. In 10th World Congress on Structural and Multidisciplinary Optimization, page 9, Orlando, USA, May 2013. URL https://mae.ufl.edu/mdo/Papers/5176.pdf.