Transactions on Computational Science and Computational Intelligence

Hamid R. Arabnia · Kevin Daimi
Robert Stahlbock · Cristina Soviany
Leonard Heilig · Kai Brüssau *Editors*

Principles of Data Science

Springer

Transactions on Computational Science and Computational Intelligence

Series Editor
Hamid Arabnia
Department of Computer Science
The University of Georgia
Athens, Georgia
USA

Computational Science (CS) and Computational Intelligence (CI) both share the same objective: finding solutions to difficult problems. However, the methods to the solutions are different. The main objective of this book series, "Transactions on Computational Science and Computational Intelligence", is to facilitate increased opportunities for cross-fertilization across CS and CI. This book series will publish monographs, professional books, contributed volumes, and textbooks in Computational Science and Computational Intelligence. Book proposals are solicited for consideration in all topics in CS and CI including, but not limited to, Pattern recognition applications; Machine vision; Brain-machine interface; Embodied robotics; Biometrics; Computational biology; Bioinformatics; Image and signal processing; Information mining and forecasting; Sensor networks; Information processing; Internet and multimedia; DNA computing; Machine learning applications; Multiagent systems applications; Telecommunications; Transportation systems; Intrusion detection and fault diagnosis; Game technologies; Material sciences; Space, weather, climate systems, and global changes; Computational ocean and earth sciences; Combustion system simulation; Computational chemistry and biochemistry; Computational physics; Medical applications; Transportation systems and simulations; Structural engineering; Computational electro-magnetic; Computer graphics and multimedia; Face recognition; Semiconductor technology, electronic circuits, and system design; Dynamic systems; Computational finance; Information mining and applications; Astrophysics; Biometric modeling; Geology and geophysics; Nuclear physics; Computational journalism; Geographical Information Systems (GIS) and remote sensing; Military and defense related applications; Ubiquitous computing; Virtual reality; Agent-based modeling; Computational psychometrics; Affective computing; Computational economics; Computational statistics; and Emerging applications. For further information, please contact Mary James, Senior Editor, Springer, mary.james@springer.com.

More information about this series at http://www.springer.com/series/11769

Hamid R. Arabnia • Kevin Daimi • Robert Stahlbock
Cristina Soviany • Leonard Heilig • Kai Brüssau
Editors

Principles of Data Science

Springer

Editors
Hamid R. Arabnia
University of Georgia
Athens, GA, USA

Robert Stahlbock
University of Hamburg
Hamburg, Hamburg, Germany

FOM University of Applied Sciences
Hamburg/Essen, Germany

Leonard Heilig
University of Hamburg
Hamburg, Hamburg, Germany

Kevin Daimi
University of Detroit Mercy
Detroit, MI, USA

Cristina Soviany
Features Analytics
Nivelles, Belgium

Kai Brüssau
University of Hamburg
Hamburg, Hamburg, Germany

ISSN 2569-7072 ISSN 2569-7080 (electronic)
Transactions on Computational Science and Computational Intelligence
ISBN 978-3-030-43980-4 ISBN 978-3-030-43981-1 (eBook)
https://doi.org/10.1007/978-3-030-43981-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

Data science combines statistical techniques, data analysis methods, and machine learning algorithms and techniques to analyze and understand tangible trends in data. Through data science, one can identify relevant issues, collect data from various data sources, integrate the data, conclude solutions, and communicate the results to improve and enhance organizations' decisions and deliver value to users and organizations. Data science draws techniques and methods from mathematics, statistics, information science, and computer science. The demand for data scientists is constantly increasing; scientists and practitioners are faced with numerous challenges caused by exponential expansion of digital data together with its diversity and complexity. The scale and growth of data considerably outpaces technological capacities of organizations needed to process and manage their data.

Thinking of massive omnipresent amounts of data as strategic assets and the aim to capitalize on these assets by means of analytic procedures is more relevant and topical than ever before. Although there are very helpful advances in hardware and software, there are still many challenges to be tackled in order to leverage the potentials of data analytics. Obviously, technological change is never ending and appears to be accelerating. Nowadays, the world seems to be especially focused on data science, and an ever-increasing impact on our society is expected. Many industries are working toward "Version 4.0," with digitization, digitalization, and even digital transformation of traditional processes resulting in improved workflows, new concepts, and new business plans. Their goal usually includes data analytics, automation, automatization, robotics, AI, and other related fields.

This book provides readers with a thorough understanding of various research areas within the field of data science. To this extent, readers who are into research will extract and conclude various future research ideas and topics that could result in potential publications or thesis. Furthermore, this book will contribute to data scientists preparation or enhancing the knowledge of current data scientists. It will introduce readers to various techniques for data acquisition, extraction, and cleaning, data summarizing and modeling, data analysis and communication techniques, data science tools, deep learning, and various data science applications in different domains.

Principles of Data Science introduces various techniques, methods, and algorithms adopted by data science experts in the field and provides detailed explanation of the data science perceptions that are properly reinforced by various practical examples. It acts as a road map of future trends suitable for innovative data science research and practice and presents a rich collection of manuscripts in highly regarded data science topics that have not been fully compiled before. It is edited by full professors with long experience in the field of data science and by data science experts in industry.

Athens, GA, USA — Hamid Arabnia
Detroit, MI, USA — Kevin Daimi
Hamburg, Germany — Robert Stahlbock
Nivelles, Belgium — Cristina Soviany
Hamburg, Germany — Leonard Heilig
Hamburg, Germany — Kai Brüssau

The original version of the book was revised: The affiliation of Robert Stahlbock has been updated. The correction to this book can be found at: https://doi.org/10.1007/978-3-030-43981-1_13

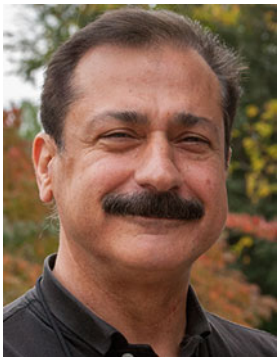
Acknowledgments

We would like to thank the faculty and researchers below for the generous time and effort they invested in reviewing the chapters of this book. We would also like to thank Mary James, Zoe Kennedy, Brian Halm, and Pearlypercy Joshuajayakumar at Springer for their kindness, courtesy and professionalism.

Kai Brüssau, University of Hamburg, Germany
Satyadhyan Chickerur, KLE Technological University, India
Andrei Chtcheprov, Fanny Mae, USA
Paulo Cortez, University of Minho, Portugal
Kevin Daimi, University of Detroit Mercy, USA
Rinkaj Goyal, Guru Gobind Singh Indraprastha University, India
Leonard Heilig, University of Hamburg, Germany
Seifedine Kadry, Beirut Arab University, Lebanon
Zeashan Khan, Bahria University, Pakistan
Nevine Makram Labib, Sadat Academy for Management Sciences, Egypt
Mahmoud Abou-Nasr, University of Michigan Dearborn, USA
Diego Pajuelo, University of Campinas, Brazil
Emil Ioan Slusanschi, Politehnica University, Romania
Sorin Soviany, National Communications Research Institute, Romania
Robert Stahlbock, University of Hamburg, Germany and FOM University of Applied Sciences, Hamburg/Essen, Germany
Vivian Sultan, California State University, Los Angeles, USA
Bayu Adhi Tama, Pohang University of Science and Technology, Republic of Korea
Gary Weiss, Fordham University, USA

Contents

About the Editors



Hamid R. Arabnia received his PhD in Computer Science from the University of Kent (England) in 1987. He is currently a professor (emeritus) of Computer Science at the University of Georgia (Georgia, USA), where he has been since October 1987. His research interests include parallel and distributed processing techniques and algorithms, supercomputing, data science (in the context of scalable HPC), imaging science, and other compute-intensive problems. His most recent activity include: studying ways to promote legislation that would prevent cyber-stalking, cyber-harassment, and cyber-bullying. As a victim of cyber-harassment and cyber-bullying, in 2017 and 2018, he won a lawsuit with damages awarded for a total of $3 million (including $650 K awarded for attorney's costs). Since this court case was one of the few cases of its kind in the United States, this ruling is considered to be important. Prof. Arabnia is editor in chief of *The Journal of Supercomputing* (Springer) and book series editor in chief of "Transactions on Computational Science and Computational Intelligence" (Springer). He is a senior adviser to a number of corporations and is a fellow and adviser of Center of Excellence in Terrorism, Resilience, Intelligence and Organized Crime Research (CENTRIC).

Kevin Daimi received his PhD from the University of Cranfield, England. He is currently professor emeritus at the University of Detroit Mercy. His research interests include data science, computer and network security, and computer science and software engineering education. Two of his publications received the Best Paper Award from two international conferences. He has been a member of the International Conference on Data Mining (DMIN) since 2004 and of the Program Committee for the 2019 International Conference on Data Science (ICDATA'19). He participated in a number of data science workshops in the United States and abroad. He is a senior member of the Association for Computing Machinery (ACM) and of the Institute of Electrical and Electronics Engineers (IEEE) and a fellow of the British Computer Society (BCS). He served as a program committee member for many international conferences and chaired some of them. In 2103, he received the Faculty Excellence Award from the University of Detroit Mercy.

Robert Stahlbock is a lecturer and researcher at the Institute of Information Systems, University of Hamburg. He is also lecturer at the FOM University of Applied Sciences since 2003. He received his Diploma in Business Administration and his PhD from the UHH. His research interests are focused on managerial decision support and issues related to maritime logistics and other industries as well as operations research, information systems, business intelligence, and data science. He is author of research studies published in international prestigious journals, conference proceedings, and book chapters. He serves as guest editor of data science-related books, as reviewer for international leading journals, and as member of conference program committees. He is general chair of the annual International Conference on Data Science since 2006. He also consults companies in various sectors and projects.

Cristina Soviany completed her MSc in Computer Science from Politehnica University of Bucharest, Romania, and her PhD in Applied Sciences from Delft University of Technology, the Netherlands. She is a technologist with strong academic, R&D, and more than 14 years of entrepreneurial experience. She has published in many scientific magazines and presented in several international conferences like Money 2020, MRC, MPE, RegTech Summit NY, B-Hive conf., and Vendorcom events. She is currently the co-founder and CEO of Features Analytics, a young AI technology company based in Belgium. She has been awarded the prize for leading the most innovative technology company in Europe in December 2011 and benefits from continuous financial support of Belgian Ministry of Economy and Scientific Research. Prior to starting Features Analytics, she has worked as a senior scientist for Philips Applied Technologies, Netherlands. She then joined the Advanced Medical Diagnostics (AMD), a start-up company based in Belgium, for 6 years. At AMD, she was in charge of leading the development of an innovative technology for cancer tissue characterization in 3D ultrasound data.

Leonard Heilig is a lecturer and researcher at the Institute of Information Systems, University of Hamburg. He completed his MSc in Information Systems and his PhD from the UHH. His current research interest is centered around cloud computing, operations research, and data science with applications in logistics and telecommunications. He spent some time at the University of St Andrews (Scotland, UK) and at the Cloud Computing and Distributed Systems (CLOUDS) Lab, University of Melbourne, Australia. He served as guest editor for several international journals and consults companies in various sectors and projects.

Kai Brüssau is a lecturer and researcher at the Institute of Information Systems, University of Hamburg. He received his Diploma in Business Mathematics and his PhD from the UHH. In his research as well as in his courses, he cooperates with bachelor and master students in several projects belonging to the fields of operations research, data science, and business analytics. Therefore, the application of optimization and data mining methods for solving practical problems is his main interest. In many industry projects, he works together with several companies, e.g., a telecommunication provider, port logistics enterprises, and manufacturers. He also focuses on developing new approaches and implementing them in different application systems.

Simulation-Based Data Acquisition

Fabian Lorig and Ingo J. Timm

1 Introduction

Most data science approaches rely on the existence of big data that is acquired and extracted from real-world systems for further processing. However, for some analyses or investigations, real data might not be available. Potential reasons are, for instance, accessibility to or existence of the system of interest such that data cannot be acquired. Other possible restrictions are economical or time limitations that do not allow for the efficient extraction of required data. In other disciplines, similar challenges are addressed using computer simulation. Here, artificial systems serve as a substitute for real-world systems, which enable a more efficient, viable, and unlimited generation of synthetic data instead.

In many disciplines and domains, scientific advance increasingly relies on the application of simulation. It is used for the generation and validation as well as for the illustration and imparting of knowledge. To this end, simulation can be applied both as a scientific method in terms of simulation studies and as a practical tool for educational purposes [37]. Either way, individual models are required, which are configured and executed with respect to a specific purpose. In many fields of application, simulation models exist for different purposes and are often provided in domain-specific repositories, e.g., OpenABM [14] or CellML [19]. In addition, numerous simulation frameworks exist for different modeling paradigms

F. Lorig (✉)
Department of Computer Science and Media Technology, Internet of Things and People Research Center (IoTaP), Malmö University, Malmö, Sweden
e-mail: fabian.lorig@mau.se

I. J. Timm
Center for Informatics Research and Technology, Trier University, Trier, Germany
e-mail: itimm@uni-trier.de

H. R. Arabnia et al. (eds.), *Principles of Data Science*, Transactions on Computational Science and Computational Intelligence,
https://doi.org/10.1007/978-3-030-43981-1_1

that facilitate the creation of new models, e.g., [9]. Many of these frameworks do not require advanced technical or programming skills such that they can be utilized by both novice and professional users from different domains and with different backgrounds.

The application of simulation is particularly reasonable when empirical studies or observations are too costly, inconvenient, time-consuming, dangerous, or generally impossible. Instead of investigating a real-world system, *cause-effect relationships* of this system are modeled and simulated. This allows for observing the behavior of the model or individual mechanisms within the model under specific circumstances to confirm or refute assumptions or theories. For this purpose, the values of the model's exogenous variables (*inputs*) are systematically altered to observe the impact they have on the endogenous variables (*outputs*) that are used to measure the model's performance or behavior. By this means, large amounts of synthetic data can be acquired for the investigation of systems and phenomena using data science methods.

This chapter introduces simulation as a technique for the systematic acquisition of synthetic data in data science. Instead of generating a vast data basis by simulating all possible parametrizations of a model, this chapter presents techniques from the field of *data farming*, which enable the problem-related extraction of data in respect of a specific problem. By this means, simulation can help to address data science challenges that are especially associated with the volume of data. The resulting relationship between simulation and data science is bilateral: Simulation experiments enable the efficient acquisition of synthetic data for the use in data science, and data science provides approaches for deriving insights from simulation models.

To outline advantages and opportunities simulation offers for data science, this chapter is structured as follows: Sect. 2 introduces simulation as method for modeling, executing, and investigating artificial systems. In Sect. 3, the relationship between simulation and data science is outlined to illustrate how simulation models can be used for the acquisition of synthetic data as part of the data science process. Different approaches for the systematic design and execution of experiments are presented in Sect. 4, with focus on the comparison of different data farming approaches in respect of data science needs. In Sect. 5, two free-to-use simulation frameworks are introduced, which facilitate the conducting of simulation experiments. Finally, the opportunities simulation offers for data science are summarized.

2 What Is Computer Simulation?

The history of modern computer simulation starts in the 1940s, when the invention of the ENIAC general-purpose computer enabled scientists to automatically execute mathematical computations for solving numerical problems [39]. Nowadays, more than 70 years later, scientific progress often inherently relies on the use of simulation, and research without simulation became unimaginable. Axelrod even

introduced simulation as a third way of doing science besides deductive and inductive research, i.e., empirically and theory-driven approaches [2]. Based on this classification of scientific advance, some authors postulate the emergence of *data-intensive science* as a fourth paradigm of research, with focus on large data sets from different sources [11]. A special emphasis lies on the strong connection between computational sciences such as simulation and data science. Due to the large amount of data that can be generated by means of simulation, a demand for dedicated techniques arises to explore and extract relevant information.

Simulation is often utilized when the application other approaches is too costly, time-consuming, or cannot deal with the investigated system's complexity [3]. For instance, when analyzing crisis of the banking system, it might be necessary to investigate and understand the fractional-reserve banking mechanisms that allow banks to grant credits as well as its consequences to the banking system itself. In this example, experimentation with the real-world system is impossible as it might expose the financial market to unforeseeable threads. Likewise, the creation of a banking market under laboratory conditions that can be safely used for experimentation is not feasible due to financial and pragmatic reasons. The real-world system is also too complex to be analyzed by means of numerical approaches because of the large number of heterogeneous and independently acting market participants. Thus, Law [18] proposes simulation as technique of choice.

The conducting of a simulation usually consists of two distinct yet mutually dependent tasks: *model building* and *experimentation*. Hence, the corresponding discipline is also referred to as *Modeling & Simulation* (M&S). As this chapter addresses the practical application of simulation for means of data acquisition, the focus lies on the experimentation part of M&S, and it is assumed that a suitable simulation model already exists. Comprehensive introductions on the building of simulation models are, for instance, provided by Bonabeau [4], Carson and John [5], and Sokolowski and Banks [36].

With respect to the conducting of simulation experiments, a *black box* perspective on the model is often sufficient (cf. Fig. 1). Here, the inner states and mechanisms of the simulation model are not considered, and only the *input-output behavior* is investigated [41]. *Inputs* represent exogenous factors that affect the model's behavior such as uncontrollable environmental influences or control variables. *Outputs* of the model are those variables that can be used for observing and assessing the behavior or performance of the model. Simulation is often used to examine

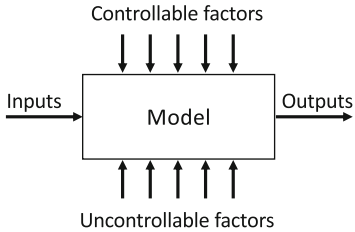


Fig. 1 Black box view on the input-output behavior of simulation models [24]

the relationship between inputs and outputs, i.e., which particular inputs influence specific outputs or how certain values of inputs minimize or maximize outputs.

To analyze the relationship between the model's inputs and outputs, experiments must be systematically executed to generate suitable data and to gradually exploit the model's *response surface* [20]. For this purpose, *data farming* techniques can be applied, which pursue an approach that takes place before *data mining* [13]. While data mining focuses on the discovery of patterns in data sets, data farming starts one step prior to this and targets the generation of relevant data on the model's behavior. Referring to agricultural farmers, relevant data for the analysis is deliberately "grown," and data samples can be drawn from different parts of the model's response surface to selectively assess the quality of data. Data miners, in contrast, can neither influence the quality of the data set nor generate more data. Still, both approaches depend on each other. On the one hand, data farming must provide suitable data sets that can be further processed by means of data mining and other data science techniques. On the other hand, data science provides approaches that allow for deriving information and knowledge from data that was generated by simulation models.

In many scientific publications, mutual benefits are outlined that emerge from the combination of simulation and data science. Feldkamp et al., for instance, combine data farming and visual analytics to investigate the relationship between inputs and outputs of simulation models [8]. Following the *knowledge discovery in databases* process, the authors make use of a data farming approach to acquire data on the model's behavior which is then further analyzed via clustering and visual analytics to identify influential inputs of the model.

Conversely, simulation is also applied as technique in data science, e.g., as part of predictive analytics to validate the used models or to generate sample data of a system's behavior [27]. It is also utilized as independent application, e.g., in data analytics for addressing big data challenges [35]. To this end, Shao et al. demonstrate different applications of simulation in manufacturing and emphasize how data can be generated, which is required for the analysis of domain-specific data analytics applications [35]. Especially the combination of both disciplines allows the user to overcome existing shortcomings. Costs of data processing and acquisition can be reduced when applying data farming to artificial simulation models. Additionally, data points that are missing in the data set can easily be substituted by observations from the model. Finally, for the generation of simulation models, the need to understand all possible cause-effect relationships within the real-world system decreases as relevant mechanisms can be learned from data.

3 Computer Simulation for the Acquisition of Data

After introducing computer simulation as method for the generation of synthetic data and presenting approaches that combine simulation and data science, this section outlines the methodological relationship between simulation and data science.

It is illustrated how simulation can be integrated into the process of data acquisition as required in data science. Especially, the use of simulation as a technique for the generation and acquisition of synthetic raw data is addressed.

According to O'Neil and Schutt, the *data science process* starts from the real world [26]. Here, data exists in a variety of forms and contexts such that raw data might be directly acquired or observed from systems within this world. In consecutive steps, the goal of data science approaches is then to process and clean the raw data to prepare them for further analysis. Yet, the acquisition of raw data from the real world is not always feasible or possible. Among other things, this might be due to limited access to the system of interest, the required amount of time and money, or the existence of the system.

Here, benefits become apparent that simulation holds for data science: The use of M&S allows for the creation of an artificial system, which might be a suitable alternative to investigating a system in the real world. Compared to real-world systems, modeled systems can be executed and investigated with slowed or accelerated speed. They are not subject to access restrictions, and the initial state of the system can be restored at any time and at no expense. Moreover, artificial systems do not necessarily require the existence of the underlying real-world system, which additionally allows for the generation and investigation of fictive and theoretical systems or effects.

As intended by the design science process, the conventional data acquisition process relies on the collection and export of data from a real-world system, e.g., from the data warehouse of a company or other sources of big data. Gained raw real-world data is then processed, cleaned, explored, and quantitatively analyzed to derive qualitative insights that can be used as basis of a decision-making process.

In contrast to this, the simulation-based data acquisition approach extends and partially replaces this conventional approach of data acquisition (cf. Fig. 2). Instead of accessing big data in the real world, only a specific set of small data (*real*

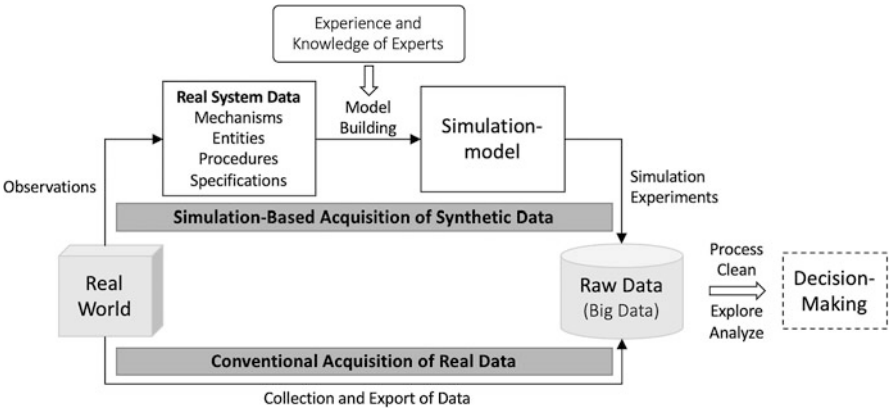


Fig. 2 Simulation-based and conventional data acquisition

system data) is acquired [22]. This includes information that is required for the model building process, e.g., information on the system's mechanisms, involved entities, process flow, and further specifications. Moreover, data that is required for the calibration and parametrization of the model is extracted. Besides real system data, experience and knowledge of domain experts are also required for the model building. After verifying and validating the developed model, simulation experiments are conducted and data farming approaches applied to systematically generate synthetic big data that is required for the application of further design science methods.

Considering trends that come along with the digitalization of our society, e.g., *Internet of Things* (IoT), the potentials of simulation-based data acquisition can be illustrated. This especially includes the use of artificial data for the evaluation of innovative technologies. For instance, Renoux and Klügl outline how agent-based simulation of inhabitants in smart homes can be used to gather realistic artificial sensor data on human behavior [29]. Such data can then be used to test augmented living algorithms or to identify patterns for learning rules on activities of inhabitants of smart homes. To this end, the authors also refer to *OpenSHS*, a simulator which can be used to extrapolate small data into big data with the aim of testing and evaluating IoT models using smart home data [1].

4 Design and Execution of Experiments

To enable and facilitate data farming on a simulation model, standardized *experimental designs* are used to derive experiment plans, which define all experiments that must be executed [32]. In this section, different experimental designs are presented that can be used for the systematic acquisition of data with respect to data science needs. This goes beyond the recommendation of big data only ("the more, the better") but also aims at the heterogeneity of the used or generated data, i.e., how adequately and evenly the data points cover all parts of the investigated response surface. To ensure the systematic investigation of the model's parameter space and the generation of heterogeneous data, simple *factorial designs* are introduced first. Especially for models with a great number of inputs, the suitability of basic factorial designs is often limited due to the combinatorial explosion of parametrizations that are suggested by the experiment plan. To overcome this limitation, this section also introduces more advanced *fractional factorial designs*. In contrast to basic factorial designs, fractional factorial designs investigate the model's parameter space more efficiently by reducing the number of simulated model parametrizations.

In *experimental design* terminology, exogenous inputs of a model are referred to as *factors* that can be used to control the model during the experimentation. Each factor is defined by a set of admissible qualitative or quantitative values (*levels*), which it can take. In a manufacturing model, examples of potential factor levels might be simple logical values, e.g., factor *AutomatedAllocation* that can either be *true* or *false*; a set of discrete levels, e.g., factor *QueuingDiscipline* which can

take levels *FIFO*, *LIFO*, and *SPT*; or a range of numerical values, e.g., factor *NumberOfMachines* for which all whole numbers between 1 and 15 are admissible [33].

Factorial designs "cross" the levels of the factors to investigate all possible factor-level combinations [24]. In other words, if a model consist of two factors A and B with a respectively b levels, the *Cartesian product* $A \times B$ is applied which results in a set of $a * b$ possible parametrizations of the model. Compared to the conventional *one-factor-at-a-time* method, where only one factor is changed and tested in each experiment, factorial designs allow for the investigation of interactions between factors as multiple factors are tested at the same time [15].

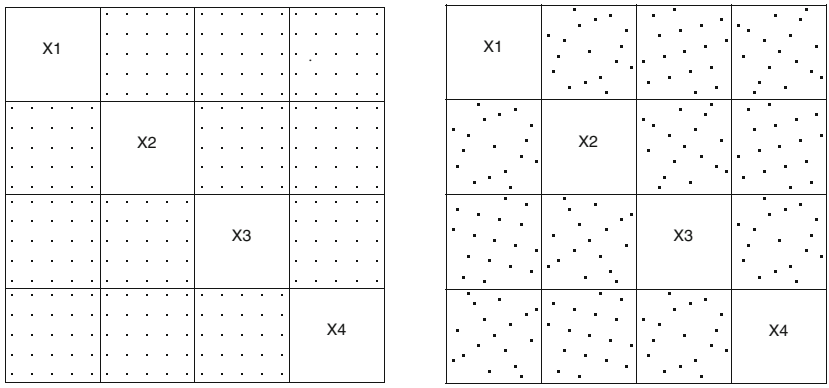
Factorial designs are usually defined via the number of factors (k) and the number of levels (m). Examples for common factorial designs are 2^k, 3^k, or the general m^k design. A 2^k factorial design is well-suited for the investigation of models with a smaller number of binary factors or factors with a limited number of levels. However, as both the number of factors and levels per factor increase, the number of resulting parametrizations also increases exponentially. This results in a combinatorial explosion of data points that are suggested by the experiment plan. For instance, the 2^k factorial design of a model with 10 factors and only 5 levels per factor consists of almost 10 million individual parametrizations (cf. Fig. 3). It also must be considered that many simulation models consist of stochastic components that, for example, represent real-world variations of processing times. The simulation of each parametrization must then be replicated multiple times to estimate the underlying probability distribution. Thus, Sanchez and Wan [33] suggest not to apply m^k designs in case the number of factors or levels exceeds ten.

Data generated by applying m^k designs allows for the identification of interactions between two and more factors. By confounding these interactions, the efficiency of m^k designs can be increased as only a *fraction* of the intended parametrizations needs to be executed. Resulting m^{k-p} fractional factorial designs generate an experiment plan which consists of a subset of parametrizations from the respective m^k design. Hence, the larger p is chosen, the less data but also information is generated [18].

Other examples of fractional designs that are well-suited for greater numbers of factors and levels are *Nearly Orthogonal Latin Hypercubes* (NOLH) and approaches

Fig. 3 Data requirements for factorial designs [33]

No. of factors	10^k factorial	5^k factorial	2^k factorial
1	10	5	2
2	$10^2 = 100$	$5^2 = 25$	$2^2 = 4$
3	$10^3 = 1{,}000$	$5^3 = 125$	$2^3 = 8$
5	100,000	3,125	32
10	10 billion	9,765,625	1,024
20	*don't even think of it!*	95 trillion	1,048,576
40		9100 trillion trillion	1 trillion



(a) 5^4 Factorial Design (b) NOLH Design

Fig. 4 Scatterplot matrices for (**a**) 5^4 factorial design and (**b**) NOLH design with 4 factors in 17 runs [33]

that combine different designs, e.g., *FFCSB-X* [34]. According to Sanchez, NOLH have good space-filling properties for $k \leq 29$, meaning that all parts of the simulation model's parameter space have the same probability of being investigated and require a considerably lower amount of data points. She also illustrates that while a 5^{10} design consists of almost 10 million data points, 33 parametrizations are sufficient for a NOLH design of 10 factors. Furthermore, reducing the number of required experiments allows for the execution of a sufficient number of replication per parametrization to investigate the distribution of the results as well as for the execution and combination of multiple NOLH designs.

Figure 4 illustrates the space-filling properties of NOLH by comparing the resulting coverage of the parameter space to a m^k factorial design. For each possible combination of two inputs x_1 to x_4, all investigated factor-level combinations are visualized. In the scatterplot matrix of the 5^4 design, the grid-like shape of the investigated tuples can be observed. Accordingly, the parts of the parameter space that fall between the grid cells are never analyzed. Thus, the scatterplot matrix of a specific m^k design will always be the same for a specific model. In contrast to this, the matrix of a NOLH design consists of a random permutation of all possible tuples in accordance with certain restrictions that ensure the coverage of the parameter space. Hence, the tuples that are suggested by the NOLH design are distributed randomly such that any possible factor-level combination might be suggested by the design.

An example of a sophisticated design that combines different more basic designs is FFCSB-X. Here, *CSB-X* is applied after using fractional factorial design to estimate the direction of the factors' effects. It pursues a *divide-and-conquer* approach to determine those factors that have the greatest effect on the model's behavior. According to Sanchez and Wan, FFCSB-X is more efficient than CSB-X and can be applied for models with more than 1,000 factors and with a large

number of discrete or even continuous factor levels [33]. Yet, the application of this approach is challenging as it requires advanced simulation knowledge and is not pre-implemented in standard simulation frameworks. Other more basic designs are often available in ready-to-use packages, e.g., via the R Archive Network (CRAN) or MATLAB.

Regardless of the used design, it must be ensured that a sufficient number of replications is executed for stochastic models to precisely measure the performance of the model [30]. For each execution of the model, the made observation of the model's output can be considered as a sample drawn from an unknown probability population. Thus, a sufficient number of experiments must be executed such that the sample mean ($\tilde{x}$) can be used to adequately estimate the population mean (μ). For this purpose, Hoad et al. suggest the use of confidence intervals [12].

Summarizing, factorial designs are well-suited to gather data on smaller models and to gain insights on interactions between the model's factors. When the application of factorial designs is limited, fractional factorial designs provide more efficient experiment plans that can handle a greater number of factors and levels. Yet, the reduced amount of data might also result in a reduced amount of information that can be gained from the simulation data. A detailed overview and comparison of different experimental designs are provided by Sanchez and Wan [33], Kleijnen et al. [16], and Montgomery [24]. However, with respect to the combination of simulation-based data acquisition and data science approaches, the execution of a great number of experiments as well as the quantity and controllability of generated data set might no longer be a showstopper. This is because data science provides more sophisticated and dedicated analytical approaches.

5 Simulation Frameworks and Toolkits

To apply experimental plans that were generated using factorial designs, the simulation model under investigation must be executed. Usually, simulation models are developed using commercial or free-to-use simulation frameworks rather than proprietary software developments. This facilitates the model building process, as commonly used modeling constructs or domain-specific formalisms are provided by these frameworks and can be applied out of the box. Moreover, a runtime environment is provided that enables the user to easily execute the model with a specific parametrization or to automatically observe the model's behavior under different parametrizations. To this end, scaling and parallelization of experiments as well as logging and first visualizations of output data are further functionalities that are often provided by such frameworks.

This section introduces *NetLogo* and *Repast Simphony* as related modeling environments that are applicable for novice as well as for professional simulation users. Both frameworks are especially well-suited for building and executing *agent-based models* in which actions of and interactions between individual entities (*software agents*) are investigated, e.g., in economic markets [10] or social networks [31]. In

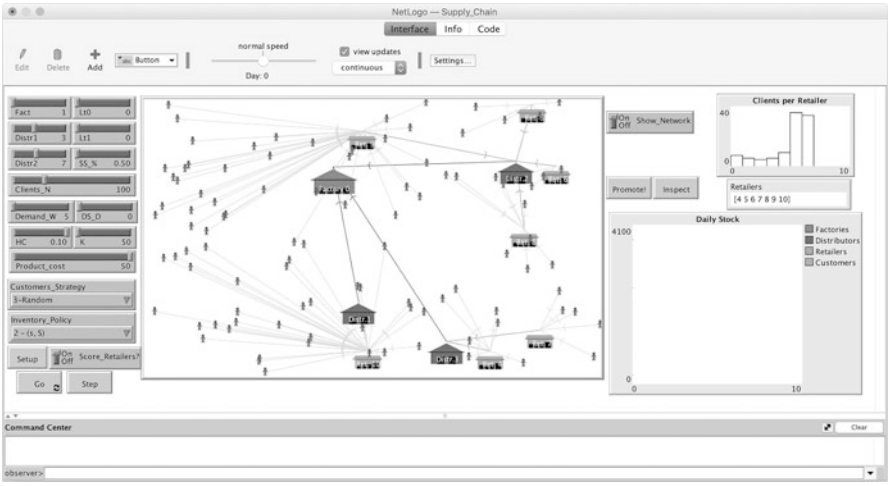


Fig. 5 Interface of the NetLogo simulation framework

contrast to other modeling paradigms, e.g., *system dynamics* or *microsimulation*, the autonomous and proactive decision behavior of each individual is in focus, which allows for the investigation of a system's behavior on a microlevel [7]. With respect to the simulation of agent-based models, the focus of this section also lies on the execution assistance *BehaviorSpace* that is provided by NetLogo. It facilitates the systematic execution of simulation experiments to examine how different factor levels influence the agents' behavior [38].

Of both frameworks, NetLogo is the one that is more suitable for novice users. It is lightweight, makes use of the functional and procedural *Logo* programming language, provides a user interface for the development and execution of the model, and facilitates the export and visualization of observed data (cf. Fig. 5). Moreover, NetLogo's model library consists of a great number of sample models that can be downloaded and modified as required.

Beside its ease of use for model building, NetLogo also provides assistance functionalities that facilitate the design and conducting of experiments. In *BehaviorSpace*, individual experiments can be configured as combination of different factor levels that shall be investigated. Referring to the m^k design, one or many distinct levels or a range of levels can be selected for each factor such that BehaviorSpace deduces and executes all possible factor-level combinations. Moreover, the number of replications can be determined that must be executed for each distinct parametrization of the model. After designing an experiment, all runs can be automatically executed, and the respective results are logged into a CSV file. NetLogo enables the use of multiple CPU cores to distribute and parallelize the execution of the runs.

Repast Simphony is more comprehensive compared to NetLogo and provides a greater range of functionalities, which allows for the building and executing of

more sophisticated simulation models. To import NetLogo models into Repast, the *ReLogo* language can be used, which is a NetLogo-inspired domain-specific language for the development of agent-based models in Repast [28]. Besides Repast Simphony, that is mostly Java-based, a C++-based version (*Repast HPC*) exists, which is intended for the use on clusters and supercomputers. In this regard, a comprehensive and practical introduction to agent-based modeling is provided by Wilensky and Rand [40] who maintain and teach NetLogo.

NetLogo and Repast Simphony are only two examples of a large number of simulation frameworks. In their literature review, Kravari and Bassiliades provide an overview on different platforms that can be used for agent-based modeling [17]. Even though not all agent platforms are also simulation frameworks, most of them include functionalities that facilitate the execution and simulation of multi-agent systems. Other prominent examples of agent simulation frameworks are *AnyLogic*, *MASON*, and *Swarm*.

In addition to agent-based simulation, there are also other established simulation paradigms. To efficiently model progress of time and to skip periods of time in which no relevant actions take place during simulation, the *discrete event* paradigm only calculates the next model state when specific predefined events take place. This is in contrast to continuous simulations, where time continuously progresses even if no actions take place. Franceschini et al. provide an overview of different frameworks that are suited for the simulation of discrete event systems [9].

It is noticeable that many of the introduced frameworks make use of the Java programming language. Yet, there are also extensions of existing systems or additional packages that enable simulation by means of other languages, e.g., Python, which might be more familiar to data scientists. Examples include DES [23], ManPy [6], or Repast Py [25]. Especially with regard to data science, Python is a programming language that is frequently used for extracting, cleaning, and analyzing data sets, e.g., *Pandas* for exploratory data analysis or *scikit-learn* for machine learning. This facilitates first steps in the application of simulation for data scientists, as they can make use of a programming language they most likely are familiar with.

6 Investigation of a Credit Market

To emphasize how simulation can be applied with respect to data science requirements, this section introduces a simple NetLogo simulation model. Besides the formulation of potential analysis goals (*hypotheses* [21]), this section elaborates on the implementation of the model as well as the possibilities the model provides for data farming. The outlined model consists of a banking market and is taken from Hamill and Gilbert's introduction to "Agent-Based Modelling in Economics" [10].

Especially in economics, the use of simulation is promising to investigate cause-effect relationships between different entities in the system, to analyze the mechanisms behind certain phenomena, or to examine the effect new rules or norms

have on the system's behavior and resilience. In their book, Hamill and Gilbert introduce the banking market as an omnipresent system with sophisticated dynamics and partially unnoticed mechanisms. The authors especially stress on the credit system of *fractional reserve banking*, where banks can multiply money by granting credits, which are then used to pay money to third parties, which again potentially deposit the money in the bank. The deposited money can then be used to grant further credits that are smaller than the original credit. This mechanism results in a recursive credit system where the bank gains money from earlier credits, which are deposited by third-party retailers.

According to the authors, when analyzing such processes, most approaches leave out partial or monthly repayments of granted loans. This is undesirable as from the bank's perspective as the repaid money can be used to grant further credits which have a large effect on the bank's potential loan volume. When thoroughly implementing such mechanisms, respective models can be used to investigate the stability of the banking market. Potential triggers of crises can be analyzed, i.e., solvency crisis and liquidity crisis, and strategies for preventing or handling crises can be evaluated, e.g., regulatory frameworks and standards such as the global and voluntarily regulatory frameworks Basel I–III. To this end, understanding the relationship between credit institutions, regulators, and households seems most relevant such that potential questions that can be answered by means of simulation might be: *How does the borrower's budget affect the stability of fractional reserve banking?*

The presented model consists of one bank with an initial deposit of 1 million GBP and 10,000 households with an average monthly budget of 1,000 GBP. There are two kinds of loans, i.e., 25-year mortgages and 3-year consumer loans. According to the limitations of the regulators, the bank decides how much money to provide as credits to the households. The households then use the borrowed money to buy from other households who decide to deposit the money at the bank. This money is then again available to the bank and can be granted as further credits. Moreover, the borrowers of the loans repay on a monthly basis, and the bank also uses this money to grant further credits.

The described model can be used to conduct simulation experiments with different parametrizations of the banking system. Potential configurations that might be analyzed include different reserve ratios of the bank, the ratio of households that are borrowers and savers, or the amount of money the households spend for repayments. To investigate the resilience of the banking system under different circumstances, it can be simulated how the bank reacts to the absence of repayments in terms of profit and vulnerability.

To analyze different configurations of the model, the use of data farming approaches and experimental designs is reasonable. This allows for the systematic investigation of the model's parameter space and the identification of interactions between the model's factors as well as the overall impact of each factor. However, from a simulation perspective, the conducting of experiments is not sufficient for the identification of circumstances that lead to resilient or fragile banking markets.

Likewise, the analysis of real bank data is not satisfying as information from multiple bank crashes is required to derive patterns. Here, the potentials that emerge from combining simulation and data science can be highlighted. Based on a smaller amount of real system data, simulation allows for the generation of a large amount of artificial banking data for different policies of the bank, external regulations, and kinds of borrower. This set of big data can then be processed and explored to derive potential insights regarding the crisis resistance of the banking system. Without the use of simulation, data science methods would rely on real-world big data, which might not be accessible or exist at all, and result in limited possibilities to identify and analyze causal relationships in banking markets.

7 Conclusions

Limited access to real-world data imposes challenges on the acquisition of data and thus also on the application of data science techniques. To overcome a lack of real data, this chapter introduced the simulation-based acquisition of synthetic data. By modeling and executing an artificial system, limitations of big data acquisition are overcome and even fictive systems can become subject to data science approaches. To enable the efficient acquisition of synthetic data from simulation models, this chapter suggested data farming as a technique for the systematic extraction of data from the model's parameter space. Through this, the availability of real-world big data is no longer mandatory for the application of data science techniques, and the observation of smaller and specific real system data is sufficient. Finally, this chapter outlined the methodological relationship between simulation and data science and illustrated how data science can benefit from the utilization of simulation.

References

1. Alshammari, N., Alshammari, T., Sedky, M., Champion, J., & Bauer, C. (2017). Openshs: Open smart home simulator. *Sensors, 17*(5):1003
2. Axelrod, R. (1997). Advancing the art of simulation in the social sciences. In *Simulating social phenomena* (pp. 21–40). Berlin: Springer.
3. Banks, J., & Gibson, R. (1997). Don't simulate when…10 rules for determining when simulation is not appropriate. *IIE Solutions, 29*(9), 30–33.
4. Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences, 99*(suppl 3), 7280–7287.
5. Carson II, J. S. (2005). Introduction to modeling and simulation. In *Proceedings of the 37th Winter Simulation Conference* (pp. 16–23). Winter Simulation Conference.
6. Dagkakis, G., Papagiannopoulos, I., & Heavey, C. (2016). Manpy: An open-source software tool for building discrete event simulation models of manufacturing systems. *Software: Practice and Experience, 46*(7), 955–981.
7. Davidsson, P. (2000). Multi agent based simulation: beyond social simulation. In *International Workshop on Multi-agent Systems and Agent-Based Simulation* (pp. 97–107). Springer.

8. Feldkamp, N., Bergmann, S., & Strassburger, S. (2015). Visual analytics of manufacturing simulation data. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 779–790). IEEE Press.
9. Franceschini, R., Bisgambiglia, P.-A., Touraille, L., Bisgambiglia, P., & Hill, D. (2014). A survey of modelling and simulation software frameworks using discrete event system specification. In *OASIcs-OpenAccess Series in Informatics* (Vol. 43). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
10. Hamill, L., & Gilbert, N. (2015). *Agent-based modelling in economics*. Chichester: John Wiley & Sons.
11. Hey, T., Tansley, S., Tolle, K. M., et al. (2009). *The fourth paradigm: Data-intensive scientific discovery* (Vol. 1). Redmond: Microsoft Research.
12. Hoad, K., Robinson, S., & Davies, R. (2010). Automated selection of the number of replications for a discrete-event simulation. *Journal of the Operational Research Society, 61*(11), 1632–1644.
13. Horne, G. E., & Meyer, T. E. (2004). Data farming: Discovering surprise. In *Proceedings of the 36th Winter Simulation Conference* (pp. 807–813). Winter Simulation Conference.
14. Janssen, M. A., Na'ia Alessa, L., Barton, M., Bergin, S., & Lee, A. (2008). Towards a community framework for agent-based modelling. *Journal of Artificial Societies and Social Simulation, 11*(2), 6.
15. Kleijnen, J. P. C. (2015). Design and analysis of simulation experiments. In *International Workshop on Simulation* (pp. 3–22). Springer.
16. Kleijnen, J. P. C., Sanchez, S. M., Lucas, T. W., & Cioppa, T. M. (2005). State-of-the-art review: a user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing, 17*(3), 263–289.
17. Kravari, K., & Bassiliades, N. (2015). A survey of agent platforms. *Journal of Artificial Societies and Social Simulation, 18*(1), 11.
18. Law, A. M. (2013). *Simulation modeling and analysis* (McGraw-Hill series in industrial engineering and management science, 5th ed.). Dubuque: McGraw-Hill Education.
19. Lloyd, C. M., Lawson, J. R., Hunter, P. J., & Nielsen, P. F. (2008). The cellML model repository. *Bioinformatics, 24*(18), 2122–2123.
20. Lorig, F. (2019). *Hypothesis-driven simulation studies – Assistance for the systematic design and conducting of computer simulation experiments*. Wiesbaden: Springer.
21. Lorig, F., Lebherz, D. S., Berndt, J. O., & Timm, I. J. (2017). Hypothesis-driven experiment design in computer simulation studies. In *Simulation Conference (WSC), 2017 Winter* (pp. 1360–1371). IEEE.
22. Maria, A. (1997). Introduction to modeling and simulation. In *Proceedings of the 29th Winter Simulation Conference* (pp. 7–13). IEEE Computer Society.
23. Matloff, N. (2008). Introduction to discrete-event simulation and the simpy language. Dept of Computer Science, University of California at Davis, Davis. Retrieved on 2 Aug 2009.
24. Montgomery, D. C. (2017). *Design and analysis of experiments*. Hoboken: John Wiley & Sons.
25. North, M. J., Collier, N. T., & Vos, J. R. (2006). Experiences creating three implementations of the repast agent modeling toolkit. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 16*(1), 1–25.
26. O'Neil, C., & Schutt, R. (2013). *Doing data science: Straight talk from the frontline*. Beijing: O'Reilly Media, Inc.
27. Ouyang, H., & Nelson, B. L. (2017). Simulation-based predictive analytics for dynamic queueing systems. In *Simulation Conference (WSC), 2017 Winter* (pp. 1716–1727). IEEE.
28. Ozik, J., Collier, N. T., Murphy, J. T., & North, M. J. (2013). The ReLogo agent-based modeling language. In *Simulation Conference (WSC), 2013 Winter* (pp. 1560–1568). IEEE.
29. Renoux, J., & Klügl, F. (2017). Simulating daily activities in a smart home for data generation. In *Proceedings of the 2017 Winter Simulation Conference*. IEEE.
30. Robinson, S. (2004). *Simulation: The practice of model development and use*. Chichester: Wiley.

31. Rodermund, S. C., Lorig, F., Berndt, J. O., & Timm, I. J. (2017). An agent architecture for simulating communication dynamics in social media. In J. O. Berndt, P. Petta, & R. Unland (Eds.), *Multiagent system technologies* (pp. 19–37). Cham: Springer International Publishing.
32. Sanchez, S. M. (2014). Simulation experiments: Better data, not just big data. In *Proceedings of the 2014 Winter Simulation Conference* (pp. 805–816). IEEE Press.
33. Sanchez, S. M., & Wan, H. (2012). Work smarter, not harder: A tutorial on designing and conducting simulation experiments. In *Proceedings of the Winter Simulation Conference* (p. 170). Proceedings of the 2012 Winter Simulation Conference.
34. Sanchez, S. M., Wan, H., & Lucas, T. W. (2009). Two-phase screening procedure for simulation experiments. *ACM Transactions on Modeling and Computer Simulation (TOMACS), 19*(2), 7.
35. Shao, G., Shin, S.-J., & Jain, S. (2014). Data analytics using simulation for smart manufacturing. In *Proceedings of the 2014 Winter Simulation Conference* (pp. 2192–2203). IEEE Press.
36. Sokolowski, J. A., & Banks, C. M. (2011). *Principles of modeling and simulation: A multidisciplinary approach*. New York: John Wiley & Sons.
37. Timm, I. J., & Lorig, F. (2015). A survey on methodological aspects of computer simulation as research technique. In *Proceedings of the 2015 Winter Simulation Conference* (pp. 2704–2715). IEEE Press.
38. Tisue, S., & Wilensky, U. (2004). Netlogo: A simple environment for modeling complexity. In *International Conference on Complex Systems*, Boston (Vol. 21, pp. 16–21).
39. Ulam, S. M. (1990). *Analogies between analogies: The mathematical reports of SM Ulam and his Los Alamos collaborators* (Vol. 10). Berkeley: University of California Press.
40. Wilensky, U., & Rand, W. (2015). *An introduction to agent-based modeling: Modeling natural, social, and engineered complex systems with NetLogo*. Cambridge, MA: MIT Press.
41. Zeigler, B. P., Kim, T. G., & Praehofer, H. (2000). *Theory of modeling and simulation*. Amsterdam: Academic Press.