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

© Springer Nature Switzerland AG 2020, corrected publication 2020
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

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

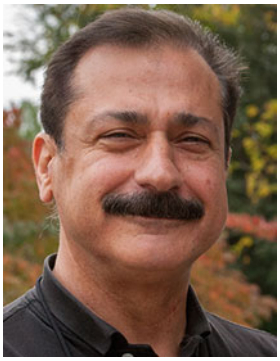
Acknowledgments

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

© Springer Nature Switzerland AG 2020

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

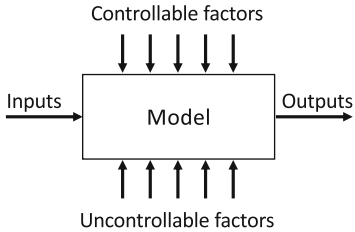


Fig. 1 Black box view on the input-output behavior of simulation models [24]

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*

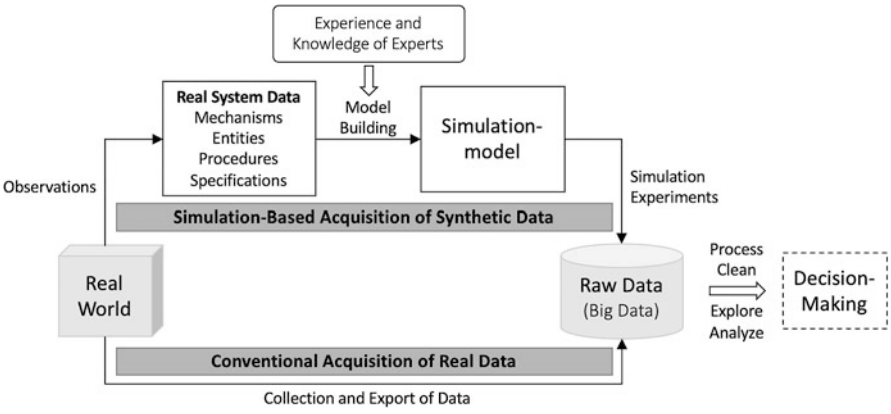


Fig. 2 Simulation-based and conventional data acquisition

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].

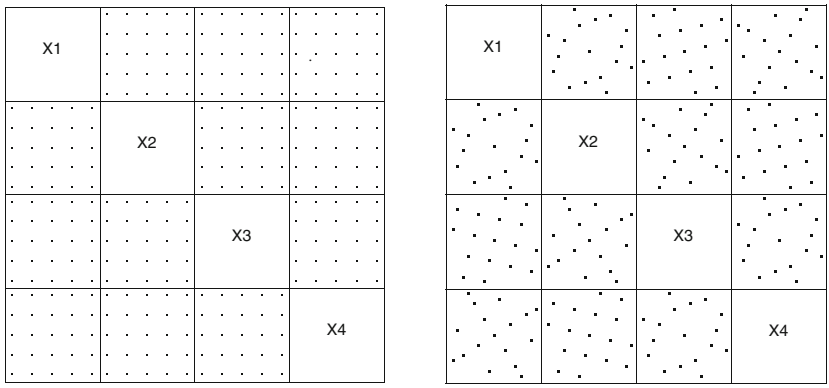
Factorial designs are usually defined via the number of factors (k) and the number of levels (m). Examples for common factorial designs are 2^k, 3^k, or the general m^k design. A 2^k factorial design is well-suited for the investigation of models with a smaller number of binary factors or factors with 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

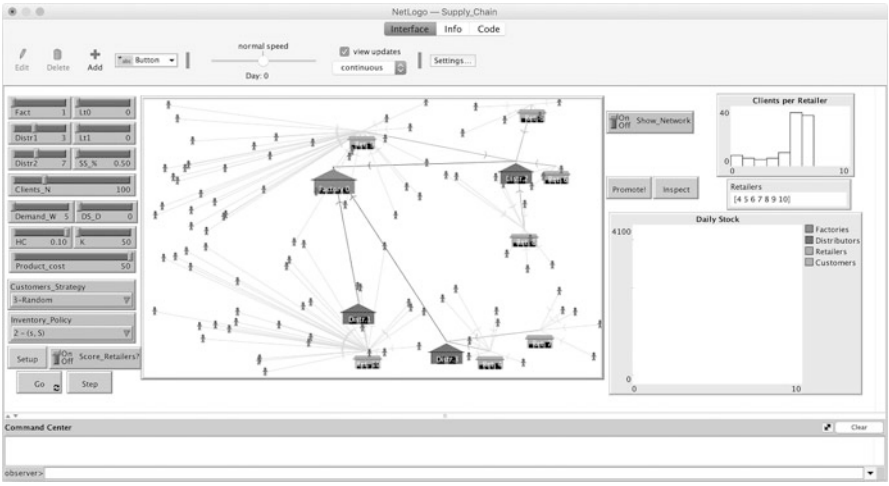


Fig. 5 Interface of the NetLogo simulation framework

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.

Coding of Bits for Entities by Means of Discrete Events (CBEDE): A Method of Compression and Transmission of Data

Reinaldo Padilha França, Yuzo Iano, Ana Carolina Borges Monteiro, and Rangel Arthur

1 Introduction

Conceptually, data are quantifiable contents that have no value and are considered a basic unit of value, since the information is the result of the processing of that data, the interpretation of the data, or, finally, its meaning. In today's world, the greater the amount of data, the greater the amount of information available and, consequently, the greater the knowledge acquired by mankind, generating people's need to interpret data and, from this, create predictions, validating the affirmative initial that the universe is made of data, after all [1].

In the past, most of the data was unprocessed; that is, it was not transformed into information, unlike the current era, which, with the ability to process in the cloud, companies are seeking to transform data into information to interpret and generate important insights for your business [2].

A simulation is a tool that allows in a practical way to build knowledge and a systemic view of a given problem, providing the necessary capabilities for the design of a solution, allowing also changes of several aspects in a wireless telecommunication system. The target in study of this chapter, evaluating it without the need for the construction of a physical experimental setup, which is totally

R. P. França (✉) · Y. Iano · A. C. B. Monteiro (✉)
School of Electrical and Computer Engineering (FEEC), Department of Communications (DECOM), University of Campinas (UNICAMP), Campinas, SP, Brazil
e-mail: padilha@decom.fee.unicamp.br; yuzo@decom.fee.unicamp.br; monteiro@decom.fee.unicamp.br

R. Arthur
Faculty of Technology (FT), Telecommunications Engineering, University of Campinas (UNICAMP), Limeira, SP, Brazil
e-mail: rangel@ft.unicamp.br

© Springer Nature Switzerland AG 2020

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_2

efficient. It is related to data science, because there are certain assumptions about the data that is desired to make that are represented in the system under study, having possible scenarios where some solutions to problems often do not have exact solutions. Thus, it is necessary to be able to generate all possible outcomes, which is not the case under study [3–5].

The simulation is a method used to study the performance of a wireless telecommunication system, reproducing as accurately as possible the characteristics of the original system. Thus, allowing to generate a variance in the results to be able to test for model stability, in the same way, stimulating the critical analysis of data, the formulation of questions and their respective discovery of answers of a given problem in study. It is possible to implement different types of system architecture to analyze different layers, such as physical, transport, transmission, and higher layer, improving and validating the system for different applications [4, 6].

Manipulating the model and analyzing the results it is possible to conclude that how several factors affect the system. These factors relate to some data science problems such as stochastic and random problems. It can be solved with simulation, for example, Monte Carlo simulation, used in the study of this chapter, as a mobile wireless channel, susceptible to several impediments like multipath, fading, shadowing, and noise, among other interferences, and so proposes new methodologies [7].

Multipath fading affects most forms of radio communications links in one way or another, where because they do not have a regular surface and a stable atmosphere, the electromagnetic beam changes in its characteristics. This occurs in any environment where there is multipath propagation and the paths change for some reason. These variations are statistical nature, is known as fading, being caused by absorption, obstacles, reflections, and atmospheric ducts altering the trajectory, amplitude, phase, and polarization. This phase may attenuate, reinforce, or even distort the signal, resulting in multiple versions of signals transmitted across different paths before they reach the receiver. Changing in this way, not only their relative strengths but also their phases, the lengths will change, also causing distortion to the radio signal; such attenuations can be represented with the statistical distribution of Rayleigh [8–11].

The simulation of discrete events is one of the most used techniques when it is a decision support technique. It is mainly used in the sense to denote the modeling that suggests representing the system as a sequence of operations performed on entities (transactions) of certain types such as data packets and bits, among others, searching for the optimal solution of a problem made by the analysis of a computational model which describes the behavior of the system under study, applied in several areas of knowledge and presenting very significant results. In this sense, the approach of simulation employed considers the principal entity of the modeled system, where are discrete items of interest in a discrete event simulation, its meaning depends on what is modeled and the type of system. This technique usually has its use oriented in the modeling of concepts having a high level of abstraction; that is, clients in a queue, emails on a server, flow of vehicles, and transmission of data packets in communications systems, among others [12–14].

In this study, a methodology for wireless systems implemented using an AWGN channel and advanced modulation format DBPSK in simulation environment is proposed. With the objective of improving the transmission capacity of data and information content through the channel and to compensate for additional complexity of modulation by multipath techniques. In this context a better performance related to memory consumption is expected, achieving improved decision-making based on data, comprising the competitive advantages oriented according to data science.

With the focus that a huge amount of data is being generated and collected every second around the world, and companies are increasingly focused on creating competitive advantages by exploiting this data, which should be able to evaluate such data every time faster and more efficient transmission and the response of your system.

A bit (data) treatment with discrete events was modeled in the step of bit-generation of a communication system, is the differential of this study, the use of discrete events applied in such low level of abstraction in the system. The results show better computational performance related to the memory utilization of the studied model.

The present chapter is organized as follows: Sect. 2 discusses data science and Sect. 3 discusses the traditional method of telecommunications and modeling of transmission channel AWGN. Section 4 presents and describes the proposal. Section 5 presents the results and, finally, in Sect. 6, the conclusions are presented, highlighting the potential of the research.

2 Data Science

The information age, transforming data into useful information, is a science of prime importance; in order to make more effective use of the databases of a company as the main subsidy guiding corporate decisions, data science was created. Resulting from the technological innovation that is constantly happening, generating advances that transform the world around us. With this data science is an interdisciplinary field of data research that solves real business problems, with the use of the scientific method and advanced techniques of data analysis, machine learning, and artificial intelligence. It is an essential area for positioning organizations at the heart of the new and so-called Industry 4.0, also having one of the important effects of digital transformation on the democratization of knowledge, which is now virtually free [15, 16].

However, exploiting the full potential of this digital transformation is only possible if we explore the data capacity generated by these innovations. In terms of "fields of knowledge," the area of data science is an intersection between computer science, engineering, mathematics, and statistics with business areas, involving knowledge of economics and administration, in general, and as a whole. In addition

to the abundance of data available, driving the revolution in the data industry is the technologies that change how we collect, store, analyze, and transform information [15–17].

It may be noted that throughout the history of mankind, the milestones of our civilization have been characterized by the progress in our ability to observe and collect data. In this way, data science is the extraction of knowledge for business decision-making through a wide range of data, be it in the form of big data or in a traditional database, whereby drawing insights from the data this results in the aid of decision-making in organizations [16, 17].

Throughout modern history, even small amounts of data have provided us with important information in the search for solutions to some of our greatest challenges. Currently, the field of data science has further increased its space by the possibility of generating value to organizations, delivering results quickly and objectively as a solution to complex problems. Mankind has always recorded its data, examples of which are the records of information in stone, papyrus, and printed books, and, later, computers have been one of the main motivators of human progress. What we now have in counterpart are the new online platforms like YouTube, Netflix, Snapchat, Facebook, and Instagram and many others that generate large data masses, being called data-driven companies; that is, data-oriented companies that use data science to make decisions [18].

Data science is an area of great interest that is promoting the democratization of mathematics and statistics in the world. Every day at least quintiles of bytes are thrown into the network, counting only the data generated by the companies and their consumers. Especially those that deal with customers, wherein in this scenario it is proven that decisions based on data science generate real growth in business, transforming this science into one of the main engines that are shaping organizations in the information age. Thus, being able to generate knowledge in different levels of internal and even external behavior of organizations. Still considering the gain in computational power, what is currently possible, and also in the capacity of data collection and storage [15–17].

Customization of service, knowledge of the target market, more assertive methods of analysis, increased return on investment, creation of focused digital strategies, agility in decision-making processes for a large number of employees, and improvement in service delivery are only some of the benefits of a data science process in a business. Since analyzing the current global scenarios one can see that the market is no longer what it used to be, since almost the whole industry is being affected by the great volume and omnipresence of the big data philosophy, and no company is immune. Where the lack of knowledge by the organization in the field of data science is very detrimental to the company, where those who have this knowledge gain competitive differential [15, 18].

3 Traditional Technologies for Telecommunications

Errors are inevitable in data transmission and wrong bits arise along with the transmission in telecommunication systems as a result of distinct, unpredictable, and random effects in time, for example, the noise caused by electromagnetic induction, sync failures between the transmitter and the receiver, defects of electronic components, fading, attenuation, and interference. This way to analyze the performance of a digital communication system, in general, is measured by the probability of bit detection errors (BER). As an important quality indicator for a telecommunication link, it is used to express the ratio of the number of received bits with reception error from the total number of bits sent in the transmission over a certain time interval [19].

In the same way, another widely used model with respect to its simplicity by mathematical modeling, and what applies to a large set of physical channels and mobile, is called additive white Gaussian noise (AWGN). Thus, the noise is additive, where the received signal is equal to the transmitted signal with the addition of noise, giving the most widely used equality in communication systems. The term white is related to all frequencies in the visible spectrum; referring to the Gaussian model it is used to describe the probability distribution of the noise samples. It is related to the time domain, where the samples can acquire both positive and negative values, and the values close to zero have a higher probability of occurrence while the values far away from zero are less [19, 20].

Still, in the context of wireless communications, the main source of thermal noise is the addition of random signals arising from the vibration of atoms in the receiver. However, it is susceptible to several interferences including multipath, fading, shadowing, and noise, among others, where such deficiencies cause an enormous degradation in the transmission and, finally, in the performance of the system [19–21].

Rayleigh fading is an ideal model for heavily built urban environments for radio signals with the propagation of signals, meaning that where there is no dominant propagation along a line of sight between the transmitter and the receiver [11].

This directly impacts on a very important factor being the advance in the capacity of processing in the cloud, through horizontal processing with clusters. Without such an increase in processing capacity data science would certainly not exist; this is because traditional vertical processing is expensive and inefficient for large amounts of data. Wherefrom the specialization of computational capacity provided by cloud computing vendors such as Amazon (AWS), Google (GCP), and Microsoft (Azure), among others, the possibility of on-demand hardware leasing and redistribution to achieve maximum efficiency it was possible, many projects became feasible with cloud computing [18].

The use of DBPSK modulation (differential binary hase shift keying) has as one of its advantages eliminating phase ambiguity and the need for phase acquisition and tracking, which results in reduced energy cost advantages. Still considering the use of a noncoherent way of solving the need for coherent reference signal at the

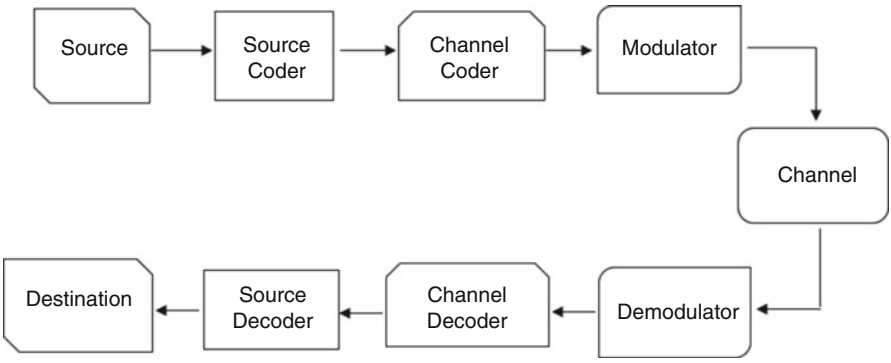


Fig. 1 Traditional model

receiver, wherein its constellation, the phase of the signal changes between two angles separated by 180°, one phase represents the binary 1 and the other phase for the binary 0. Since it does not require phase synchronization, it is very attractive and efficient for digital communications systems, widely used by wireless LANs compliant with the IEEE 802.11 standard [22, 23].

The research shown in this section presents an AWGN transmission channel with DBPSK modulation, where the Simulink environment of the MATLAB® software was used in its 64-bit version (8.3 – 2014a). In the respective modeling of Fig. 1, the signals corresponding to bits 0 and 1 were generated and then modulated, forwarding through the multipath Rayleigh fading channel with Jakes model with Doppler shift defined at 0.01 Hz, and also inserted to math function 1/u, relative to the channel fading effect.

The mathematical function is required to track the time variability channel where the receiver implementation generally incorporates an automatic gain control (AGC). After passing through the AWGN channel according to the parameters specified as the sample time of 1 sec, a power input signal of 1-watt, initial seed in the generator of 37 and in the channel of 67, Eb/No of 0–10 dB. This signal is then demodulated to perform the bit error rate (BER) of the channel and subsequent processing and generating of the signal BER graph.

The developed method directly involves the data science area since it studies the information (transmitted bits), its capture process, transformation (precoding the bits), generation (generation of the bits), and, later, data analysis (analyzes performed over the signal (information/modulated data) throughout the information process. Which method relates the major disciplinary areas of science such as computation, mathematics, and knowledge of the model in which it is under study, these disciplines being pillars for data science.

4 Proposed Method: CBEDE

The coding of bits for entities by means of discrete events (CBEDE) methodology consists of modeling relative to discrete events which is similar to the one shown in the previous section. The differential for the proposed method, it was added the discrete events process of precoding, where the treatment performed on the signal referring to bit 0 and 1.This signal then passes through a treatment by library of discrete events of the Simulink, and later passed by conversion to the specific format required for manipulation in the domain of discrete events, being both time-based signals and event-based signals were in the time domain.

Then, Entity Sink® shown in the following figure is relative to the end of the modeling of this signal as a discrete entity. This tool is responsible for marking the specific point in which Entity Sink will be located, where later the event-based signal conversion will be performed for a time-based signal. So, this time-based signal was converted to a specific type that followed the desired output data parameter, an integer, the bit. By means of the Real-World Value (RWV) function, where the current value of the input signal was preserved, then a rounding was performed with the floor function. This function is responsible for rounding the values to the nearest smallest integer. Zero-order hold (ZOH) is responsible for defining sampling in a practical sense and used for discrete samples at regular intervals, describing the effect of converting a signal to the time domain, causing its reconstruction, and maintaining each sample value for a specific time interval. The treatment logic on bits 1 and 0 according to the proposed method regarding the technique of discrete events is shown in Fig. 2.

It is the differential of this research and study is in the use of discrete events applied in a low level of abstraction possible in a communication system; that is, the bit generation. The model presented in Fig. 3 incorporates the traditional method modeled together with a proposal presented and explained previously. In the same way, that highlights the part modeled according to the proposal of discrete events, in blue.

It has also been applied to the proposed method in a wireless system, having a mobile wireless channel that is susceptible to several impediments including multipath, fading, shadowing, and noise, among other interferences [9]. Within this framework of application, it was modeled following the pattern of Figs. 2 and 3, where the signals relative to bits 0 and 1 are generated and then modulated in DBPSK. It was later passed through a multipath Rayleigh fading channel, both containing the Jakes model with Doppler shift defined at 0.01 Hz, as well as a block incorporated which has a math function 1/u.

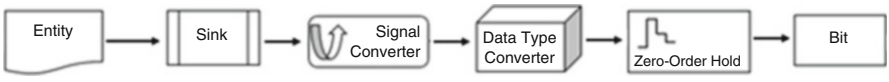


Fig. 2 Proposed bit precoding

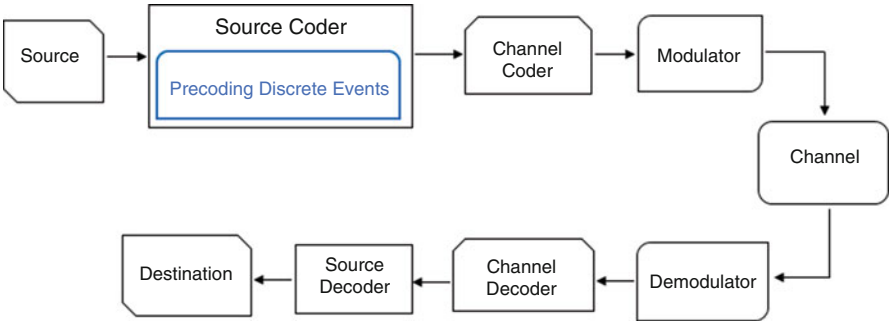


Fig. 3 The hybrid model without multipath fading

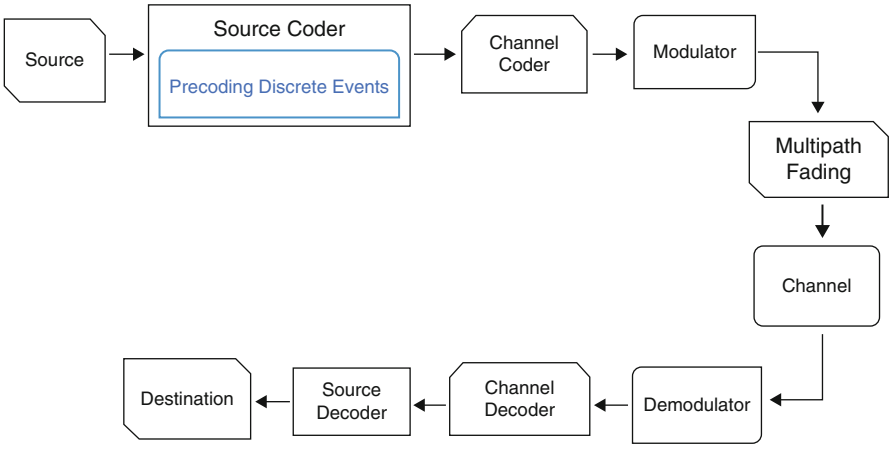


Fig. 4 The hybrid model with multipath fading

Such a function as previously said is required to track the time variability channel where the receiver implementation ordinarily incorporates an automatic gain control (AGC), with that the signal is followed to an AWGN channel. Thus, according to the parameters specified as sample time of 1 sec, power input signal of 1-watt, initial seed in the generator of 37 and in the channel of 67, Eb/No of 0–14 dB, and then the signal is demodulated to perform the bit error rate (BER) of the channel, as shown in Fig. 4.

5 Results and Discussion

Thus, in this section, the AWGN transmission channel with DBPSK modulation is presented, addressing the traditional methodology and the proposal. In this way, in Fig. 5, using 15,000 sec of simulation time, the flows of transmission of the DBPSK

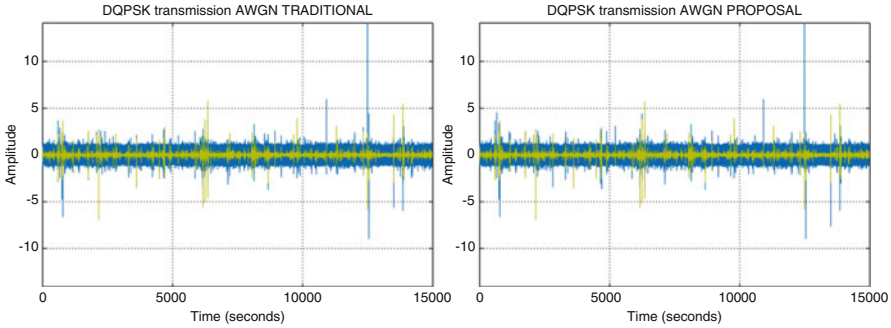


Fig. 5 Transmission flow DBPSK Rayleigh

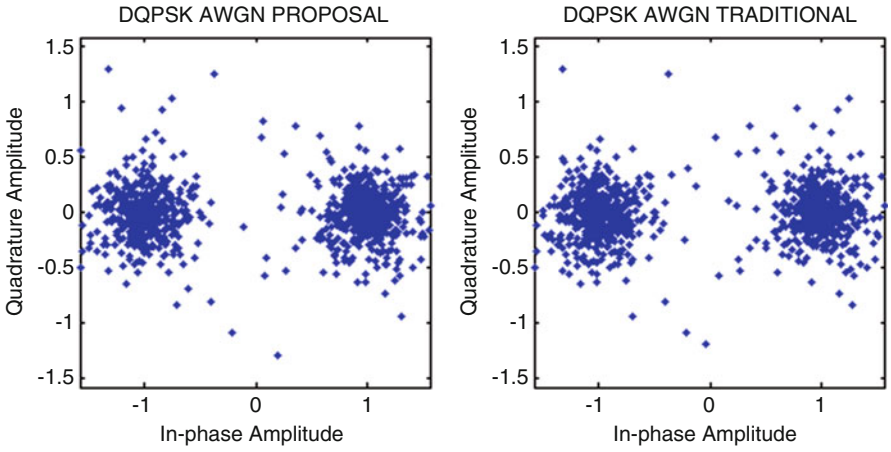


Fig. 6 Simulated DBPSK constellation Rayleigh

signal were placed side by side, in relation to CBEDE (right) and traditional method (left) for comparison, in context fading multipath with Rayleigh, noting that both methodologies also generated the same result.

Also used was the constellation diagram, useful for viewing the constellation of the modulated digital signal, comparing the performance of one system with another. In Fig. 6 the results of the constellations with 13 dB are shown, according to the CBEDE (left) and traditional method (right), in context fading multipath with Rayleigh.

The results were obtained through simulations with the models presented previously, on a physical machine with hardware configuration, which is an Intel Core i3 processor and 4GB RAM. In this way the `sldiagnostics` function was used, which calculates the sum of all the memory consumption processes used in the model in the simulation. Through `ProcessMemUsage` parameter, counting the amount of memory utilized in each phase of the simulation, displaying the total

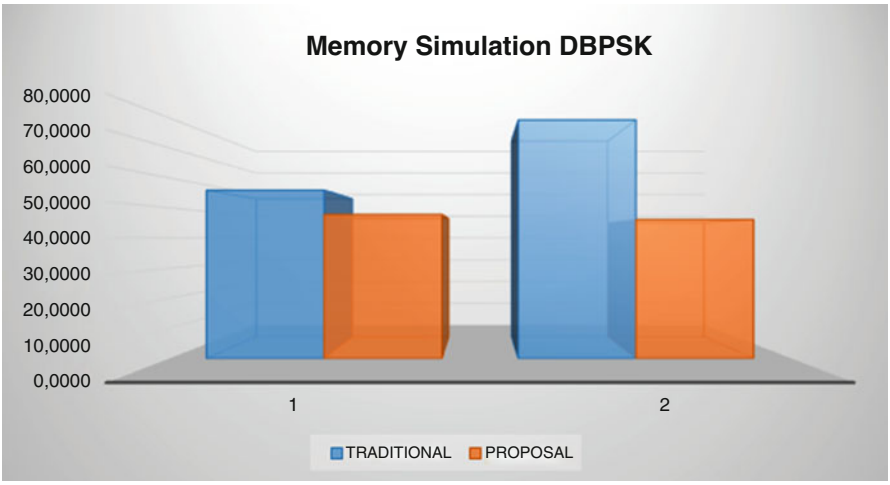


Fig. 7 Memory consumption DBPSK Rayleigh

Table 1 Computational improvement

Memory consumption		
Simulations	CBEDE	Traditional
1	47,7773	55,8633
2	46,0898	79,2422

amount in MB, taking that into account the first simulation of both models. This occurs because it is in the first that the variables are allocated and the memory reserved for the execution of the model, presented in Fig. 7.

In this way, it is understood that in the transmission of a channel containing the CBEDE and in another the traditional method, the same information content (quantity of bits), without any type of loss (signal and constellation) and having the same signal quality (BER). The results related to the memory consumption of the proposal are relative to the compression of this information transmitted and respectively received, as shown in Table 1 and related to Fig. 8.

In order to analyze the relationship between the simulation methodology (proposed x traditional) and the impact that is applied and directed on the physical layer of the channel, BER were made in MATLAB. In Fig. 8, the performance of the models according to the study is shown, along with the transmission with noise ranging from 0 to 10 dB.

As can be seen, CBEDE brings a new approach to signal transmission that facilitates the role of using data science. It allows the extraction of relevant information from data sets that do not seem to have a strategic value and transmitting and receiving them in a more efficient way. In this case, the transmission is performed in the discrete domain with the implementation of discrete entities in the bit generation process. What results in the possibility to increase the capacity of information transmission for communication systems facilitating the use and

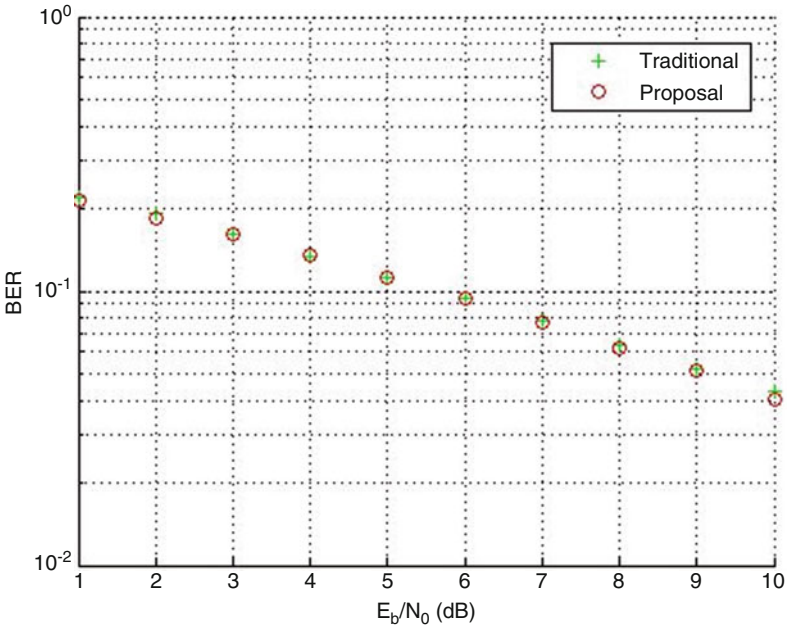


Fig. 8 BER between the models

application of data science. A simple example like the data generated by the marketing and sales sectors of an organization can be analyzed more quickly and efficiently, making a more fluid transmission with the proposal of this study. The behavior of the clients with respect to the actions carried out by the company and applying levels of satisfaction and evaluating them, as well as generate insights on the current demands of the target audience. It allows the verification of the dissatisfaction or not of some clients and seeks to generate solutions of care for these cases in isolation.

Given the above, the technique of discrete events can be applied in the treatment of bits in its generation stage, responsible for their conversion into discrete entities, the result of the use and application in lower level of application, acting on the physical layer. This already is an important parameter to meet the needs of an increasingly technological world and with a globalized market that is increasingly changing with the emergence of new technologies and business opportunities every day. Thus, the CBEDE has a positive impact on data science, where it is possible to perform advanced analyses of data in a prescriptive and predictive way, both of which are future scenarios. The first is dealing with the consequence of actions and the second with predictions; however, in order to obtain such benefits, it is necessary to invest in data science solutions and thereby create a data culture.

Considering that speed is a key issue when choosing methodology, whether it is used by a user, companies, or universities, the methodology presented can be

seen as a great ally to another advantage long sought by the companies that invest in data science being the possibility to maintain, in real time, a complete view of all information about the company. Such data systematization and information provision, which is relevant, ends up generating a fundamental knowledge for an increasingly assertive decision-making, allowing managers to achieve a number of other benefits.

With the result reaching up to 71.93% in the improvement of memory of the CBEDE (coding of bits for entities by means of discrete events) methodology employed in this process, since the amount of data is interest of data science and the speed of the transmission is primordial, the proposal of this chapter becomes quite attractive in the data science context.

Still considering that when it comes to data transmission is the memory consumption of the device used by the organization is of extreme importance. This occurs because currently ordinary users tend to generate data every second, worldwide, so the slowness is often related to the speed of communication and the technology implemented in the system structure used by these users. It can often generate device crashes, and inconvenience sometimes due to loss of data, even more, if these are in real time, and can be avoided with the application of the proposal, which provides an improvement in the transmission of data.

6 Conclusions

In ever more modern times, and technology increasingly present in our lives, such as the Internet of things (IoT) and big data, an astronomical amount of data is being generated every moment around the world. Being Data science is nothing more than the practice of extracting knowledge focused on decision-making through this large database, regardless of whether it is in big data or in a traditional database. For the present challenge is no longer in the simple storage of these data, but in the intelligent interpretation of this infinite collection of data, which are converted into precious information.

The objective of this research was the use of discrete event technique applied in the lowest level of abstraction possible in a telecommunication system, the bit generation presented in this chapter. It is an approach that works as a precoding process differentiated, in bits before the modulation process.

Companies are getting more and more embedded into the data ecosystem, where the world itself produces enormous amounts of information that has a direct impact on how organizations begin to determine their strategies and conduct their business, and so information compression is the by-product of the proposal, since the CBEDE acts on the bits, having a substantial impact on the compression methods performed in higher layers of a communication system. Where all types of information are hidden in the middle of various generated data (bits) by sales, marketing by a company or even hidden in the web itself, where the discovery of these answers is what constitutes, in reality, a competitive differential in the world currently.

References

1. Hashem, I. A. T., et al. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems, 47*, 98–115.
2. Rittinghouse, J. W., & Ransome, J. F. (2016). *Cloud computing: Implementation, management, and security*. Boca Raton: CRC Press.
3. Yeoh, G. H., & Tu, J. (2019). *Computational techniques for multiphase flows*. Kidlington: Butterworth-Heinemann.
4. Digital Modulation in Communications Systems. (2001). *An introduction*. Agilent Technologies.
5. Padilha, R., Martins, B. I., & Moschim, E. (2016, December). *Discrete event simulation and dynamical systems: A study of art*. BTSym'16, Campinas, SP – Brasil.
6. Vasileska, D., Goodnick, S. M., & Klimeck, G. (2016). *Computational electronics: Semiclassical and quantum device modeling and simulation*. Boca Raton: CRC Press.
7. Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the Monte Carlo method* (Vol. 10). New York: Wiley.
8. Sasaki, N. K., & Moschim, E. (2007). *Simulação de Sistemas de Comunicação Óptica Baseada em Simulação a Eventos Discretos*. Campinas: Universidade Estadual de Campinas.
9. Pissinelli, J. G., Risso, L. A., Picanco, S. R. A., Ignacio, A. S. P., & Silva, L. A. (2015). *Modelo De Simulação De Eventos Discretos Para Análise De Fluxo De Veículos*. Fortaleza: ENEGEP.
10. Rangel, J. J. A., Costa, J. V. S., Laurindo, Q. M. G., Peixoto, T. A., & Matias, I. O. (2016). Análise do fluxo de operações em um servidor de e-mail através de simulação a eventos discretos com o software livre Ururau. *Produto & Produção, 17*(1), 1–12.
11. Shankar, P. M. (2017). *Fading and shadowing in wireless systems*. Heidelberg: Springer.
12. Padilha, R. (2018). *Proposta de Um Método Complementar de Compressão de Dados Por Meio da Metodologia de Eventos Discretos Aplicada Em Um Baixo Nível de Abstração*. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas. Campinas, SP – Brasil.
13. Bouanan, Y., Zacharewicz, G., Ribault, J., & Vallespir, B. (2018). Discrete event system specification-based framework for modeling and simulation of propagation phenomena in social networks: Application to the information spreading in a multi-layer social network. *Simulation, 95*(5), 411–427.
14. Artuso, M., & Christiansen, H. L. (2014). Discrete-event simulation of coordinated multipoint joint transmission in LTE-advanced with constrained backhaul. In *Proceedings of IEEE Eleventh International Symposium on Wireless Communication Systems*, pp. 106–110.
15. Pries, K. H., & Dunnigan, R. (2015). *Big data analytics: A practical guide for managers*. New York: Auerbach Publications.
16. Chen, H. M. (2017). Information visualization. *Library Technology Reports, 53*(3), 0024–2586.
17. Grant, R. M. (2016). *Contemporary strategy analysis: Text and cases edition*. Chichester: Wiley.
18. Klous, S., & Wielaard, N. (2016). *We are big data: The future of the information society*. London: Springer.
19. Barnela, M., & Kumar, D. S. (2014). Digital modulation schemes employed in wireless communication: A literature review. *International Journal of Wired and Wireless Communications, 2*(2), 15–21.
20. Rama Krishna, A., Chakravarthy, A. S. N., & Sastry, A. S. C. S. (2016). Variable modulation schemes for AWGN Channel based device to device communication. *Indian Journal of Science and Technology, 9*(20). https://doi.org/10.17485/ijst/2016/v9i20/89973.

21. Tozer, E. P. (2012). *Broadcast engineer's reference book* (1st ed.). Waltham: Focal Press.
22. Couch, L. W., II. (2013). *Digital and analog communication systems* (8th ed.). Upper Saddle River: Prentice-Hall.
23. Morreale, P. A., & Terplan, K. (2018). *CRC handbook of modern telecommunications*. London: CRC Press.

Big Biomedical Data Engineering

Ripon Patgiri and Sabuzima Nayak

1 Introduction

The Big Data is exploding in every data-intensive field around the world since its inception, especially, in the IT industries. It is able to show its existence and dominate the market within a few years. The Big Data is composed of a high volume of a variety of data. These data are increasing at a massive rate. The highly increasing massive dataset causes a perplex dilemma on how to store, process, and manage since the conventional system does not work in large-scale data. Therefore, there are numerous ways to deal with the dilemma of Big Biomedical Data Engineering (BBDE) using Big Data, for instance, NoSQL. The Big Data seems more theoretical, but Big Data engineers have difficulty in maintaining the mammoth sizes of data. Besides, scientists deal with tremendous biomedical data on a daily basis. Thanks to the Big Data paradigm that makes our life easy in biomedical data engineering (BDE). Moreover, an organization can enhance their performance by employing Big Data technology. Therefore, the boundary of Big Data extends to every field nowadays, for example, science, engineering, economy, politics, and business.

The biomedical research is known as a useful research for the well-being of human being. Engaging Big Data in biomedical can generate numerous possibilities. It is a common practice nowadays. Every year enormous amount of health data is created [54]. Interestingly, every individual generates 0.4 TB data, 6 TB of genomic information, and 1100 other health data by 2020 [22]. The clinical data also will be doubled every 73 days from 2020 onward [22]. The collective healthcare data are tremendous in size and increasing at a very high pace on a daily basis. The

R. Patgiri (✉) · S. Nayak
National Institute of Technology Silchar, Silchar, Assam, India
e-mail: ripon@cse.nits.ac.in

© Springer Nature Switzerland AG 2020

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_3

huge data visualization enhances the abilities to find trends, identify outliers, and perform quality checks [21]. The visualization process stitches together the variety of data types for a common goal. There is a research opportunity in the technological singularity [20] that can exist in Big Biomedical Data Engineering (BBDE) which is to be exposed. However, this singularity can only be discovered after excessive experimentation of the technology or after deployment of the technology in the field of healthcare.

The Internet of Things (IoT) is a disruptive technology predicted by [17]. Things are becoming smart and connected to the Internet. The IoT is a booming research area and billions of IoT devices are already available in the market. The IoT is an integral part of smart hospital nowadays [5]. IoT devices are being deployed in hospitals to upgrade conventional hospitals to smart hospitals. Therefore, the IoT devices have become the key sources of generating massive amount of data in healthcare system.

Ta et al. [70] classify the healthcare data into seven major categories: electronic healthcare records (EHRs), social media, clinical text, genomic data, biomedical signals, biomedical images, and sensing data. Various data are grouped into a single category based on the nature of the data. For example, brain signal and heartbeat signal are grouped into biomedical signals. The biomedical data is fed into the Big Data technology for processing and stored in the Big Data environment. The advancement of Big Data technology makes the healthcare system easier than earlier conventional system. However, there are many barriers in modern biomedical systems [48]. Interestingly, some data are personal and extremely private, for example, genome. These data should not be exposed to the outer world and their privacy is to be maintained [19, 27]. Privacy becomes limited by deploying Big Data in genome [59]. In addition, most of the health data require privacy. Aziz et al. [3] emphasize on the secure and efficient genome data computation. Moreover, the large set of such kind of data transmission is also a key issue. The Big Data Smart Socket (BDSS) is an example of transferring large-scale biomedical data and provides an alternative mirror for data seamlessly [72]. On the contrary, the data discovery is also a prominent research area [46]. The metadata enhances the process of the data discovery. However, there are plenty of data indexing techniques available nowadays. With the technological advent, the biomedical data scientists apply computational and statistical approaches for data mining and machine learning to gain insight of the huge dataset. Precisely, the tools of Big Data are extremely helpful in the field of biomedical research, namely, Hadoop.

This chapter is structured as follows: Sect. 2 describes Big Data in the context of biomedical engineering. Section 3 exposes the role of Big Data analytics in biomedical data engineering. Sections 4.1 and 4.2 discuss about neurology and genome research using Big Data, respectively. Section 4.3 discusses the challenges and opportunities in cancer research. Section 4.4 discusses possible health center setup by Big Data. Section 7 discusses the future of Big Biomedical Data Engineering. Finally, Sect. 8 concludes the chapter.

2 Big Data

The Big Data is defined by Doug Laney using three V's: volume, velocity, and variety [35]. However, there are many V's to define the Big Data characteristics, namely, $V_3^{11} + C$ [52]. In simpler terms, the Big Data is enormous data handling which is impossible for a conventional system. The Big Data paradigm turns out as a game changer paradigm in many data-intensive fields, for example, environment, science, engineering, business, etc. [52]. The evolving use of Big Data paradigm creates many hope and possibilities in various data-intensive fields. The real-life application and system of Big Data are representing a large set of data in the heterogeneous environment, modeling and processing a plethora of data, and querying and mining a massive amount of data in databases and data warehouse [18]. The Big Data is employed in real-life systems due to the gigantic size of data that is stored, processed, monitored, analyzed, and visualized. For instance, Square Kilometer Array telescope data are exabytes [61]. However, data-intensive fields utilize Big Data technology to improve the organizational performance and revenue, like BDE. Undoubtedly, the Big Data is a better choice for BDE due to a mammoth volume of data that need to be analyzed [18]. The Big Biomedical Data Engineering (BBDE) requires visualization, analysis, processing capacities, and huge storage spaces [46]. The article [8] asks the question, "What is the reason for writing?" The considered environment may vary, but the answer is the same. However, the BBDE requires to store the data so that it is available for future use for the study of the diseases. It will help in finding a cure or in diagnostic process in the future. The answer coincides with an article [8]. Muade et al. [8] indicate that writing is the first phase of Big Data.

Figure 1 exposes a complete solution for a healthcare system using Big Data paradigm. The solutions are Hadoop-based solutions that are more common to other Big Data–based solution. The Hadoop is two tiered, namely, Hadoop Distributed File System (HDFS), a storage engine, and MapReduce, a programming engine. The HDFS is used to store biomedical data and MapReduce is used to define what to do with these stored data. The Hadoop stack is used to represent the Big Data technology. The selection of Hadoop tools depends on the purposes of designing the complete healthcare system. For example, HBase can be used for large-scale tabular data. As Fig. 1 depicts, the remote doctors/researchers play a vital role. The remote doctors are located in geographically different places as shown in Fig. 1. The Apache Spark is also another emerging tool for BBDE. In addition, it is easy to develop a framework for BBDE using Hadoop by collaborating between engineers and doctors. Most of the current solutions are Hadoop-based for clinical data in large-scale space, for instance, BigBWA [1] and GMQL [44]. Moreover, Dive is another tool to visualize biological data [11]. The technology is ready to solve the very complex dilemma of BBDE.

The Big Data technology is deployed to make a decision very correctly. Most of the time, we rely on the technology rather manual decision. However, there may not be an existing solution for a particular scenario of biomedical large-

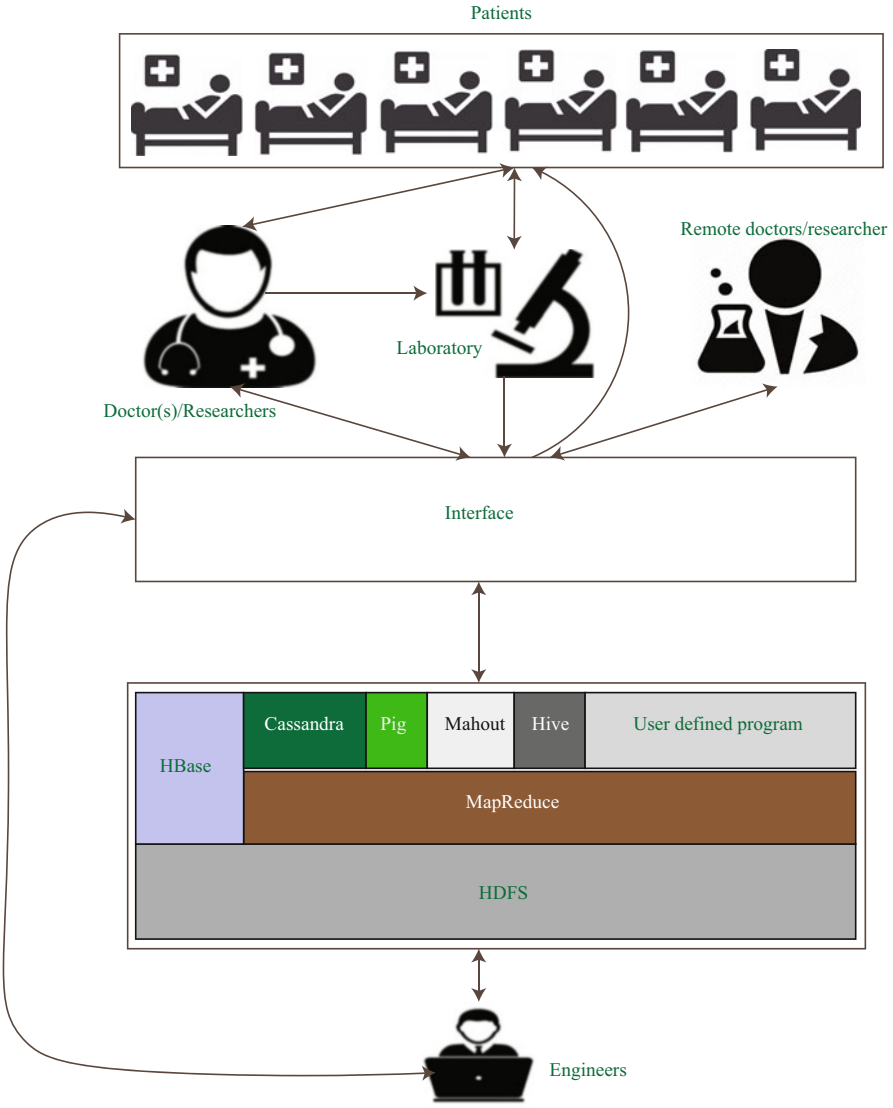


Fig. 1 The landscape of the Big Data solution of healthcare system

scale data. In this case, the problem is converted into the key/value to define the required action by the MapReduce program. The power of MapReduce lies in self-defining solution. In addition, the Big Data tools are ready to provide a solution to BBDE dilemmas. The open-source tools are HBase, Hive, Spark [49], Mahout, etc. [13]. These tools leverage heterogeneous medical data from heterogeneous sources [30]. The tools provide capabilities such as importing, exporting, comparing,

combining, and understanding data [40]. Bourne et al. [9] urged a better way of storing, processing, and managing the EHR data. Distributed Application Runtime Environment (DARE) is a gateway for scalable scientific data-intensive research [41]. Moreover, Scientific Workflow Management Systems (SWfMSs) are modern workflow systems for Big Data in distributed and parallel environment [39]. Another cloud-based workflow engine is developed by Szabo et al. [69].

3 Big Data Analytics

Big Data analytics (BDA) is a combination of Big Data and analytics [20]. The analytics refers to the logical method of analysis. The BDA provides a platform for the discovery of hidden important facts from data. The machine learning algorithms are used to implement the BDA. The BDA is categorized into five main subcategories, namely, descriptive analytics, predictive analytics, prescriptive analytics, decision analytics, risk analytics, and security analytics. These analytics are very useful in medical data analysis. The BDA helps in decision-making, analyzes large dataset properly, and reports the analysis results. In the past decades, healthcare data were stored in digital forms. The analytics can help in discovering hidden patterns on digitally stored medical data. For instance, the predictive analytics help in better disease forecast in biomedical system [6, 73]. Bates et al. [6] emphasized that the Big Data analytics improved the healthcare system [12]. A description-driven system enhances the biomedical data analytics [46], for instance, CRISTAL [10]. The description-driven system is a metadata system to reuse the stored data again and again in the future. Therefore, the metadata is stored alongside the health data. The metadata ensures the availability of the stored data. We may or may not know what kind of data is available in the databases. In this scenario, a descriptive query must be fired to get a similar case to retrieve these data from a very large set of data. Moreover, in most of the cases, we want to know the analysis report from healthcare data analytics about the possibilities of making a decision. The analytics are great tools to make a decision. Without the BDA, the decision-making is impossible in a large-scale environment. The large-scale medical data are applied not only in the decision-making process but also in forecasting, prescription, risk, and possibility analysis.

The purpose of Big Data analytics is to provide a logical analysis of the data. The data are enormous in size. It cannot be performed in a conventional way. Therefore, the Big Data analytics is used to analyze a plethora of data. The Big Data analytics dubs many problems to solve, for example, business, economy, science, etc. Shahand et al. [62] disclose the biomedical data analysis. GVSS performs analysis to detect dengue and flu [14]. Moreover, the most modern biomedical data analysis is cloud-based data analysis [23]. However, let us discuss the Big Data analytics by taking an example of cancer to understand, that is, Big Biomedical Data Analysis (BBDA). The BBDA is a merged term of Big Data, Big Data analytics, and biomedical data. This BBDA creates enormous solution in biomedical data

engineering. The BBDA is yet to develop. The taxonomy of Big Data analytics is discussed with respect to BBDA.

3.1 Descriptive Analytics

The descriptive analytics is a logical analysis of large datasets. A massive dataset has been explored to discover and gain insight. The descriptive analytics is categorized into two, namely, description (or exploration) and discovery analytics. The descriptive or explorative analytics is performed to gain in-depth insight on large-scale medical data and which provides an informative output. These data are explored to study, diagnose, and treat a patient. Discovery analytics discovers some hidden truth on very large-scale storage system which contains medical data of many years. However, this is very helpful in biomedical data analytics analysis reasons are unknown. For example, we do not know the patterns of a DNA. We would like to sequence and discover some new patterns (Fig. 2).

3.2 Predictive Analytics

The predictive analytics is a logical analysis on the massive amount of data where the future depends on those massive data. It is further subcategorized into three, namely, prediction or forecast, possibility, and indicative analytics. Now, when we perform DNA analysis, the DNA has the code of adenine (A), cytosine (C), guanine (G), and thymine (T). Prediction or forecast: what will happen if we alter some information? Possibilities: what is its future and what are the possible effects? And indication: where will it be beneficial?

3.3 Decision Analytics

The decision analytics is a logical analysis of the mammoth size of data to make a good decision. It is classified into three subcategories, namely, direct decision analytics, decision assistance, and decision possibility analytics. The Big Data analytics performs analysis and reports the results. Direct decision: is it cancer? Decision assistance: comparison among the noncancerous DNA cell and cancerous DNA cell. What are the patterns to declare it as a cancer? Decision possibilities: if it is a cancer, then what is the possible stage of the cancer? And if it is not a cancer, then what symptoms can it contain (it is decided that there is cancer)?

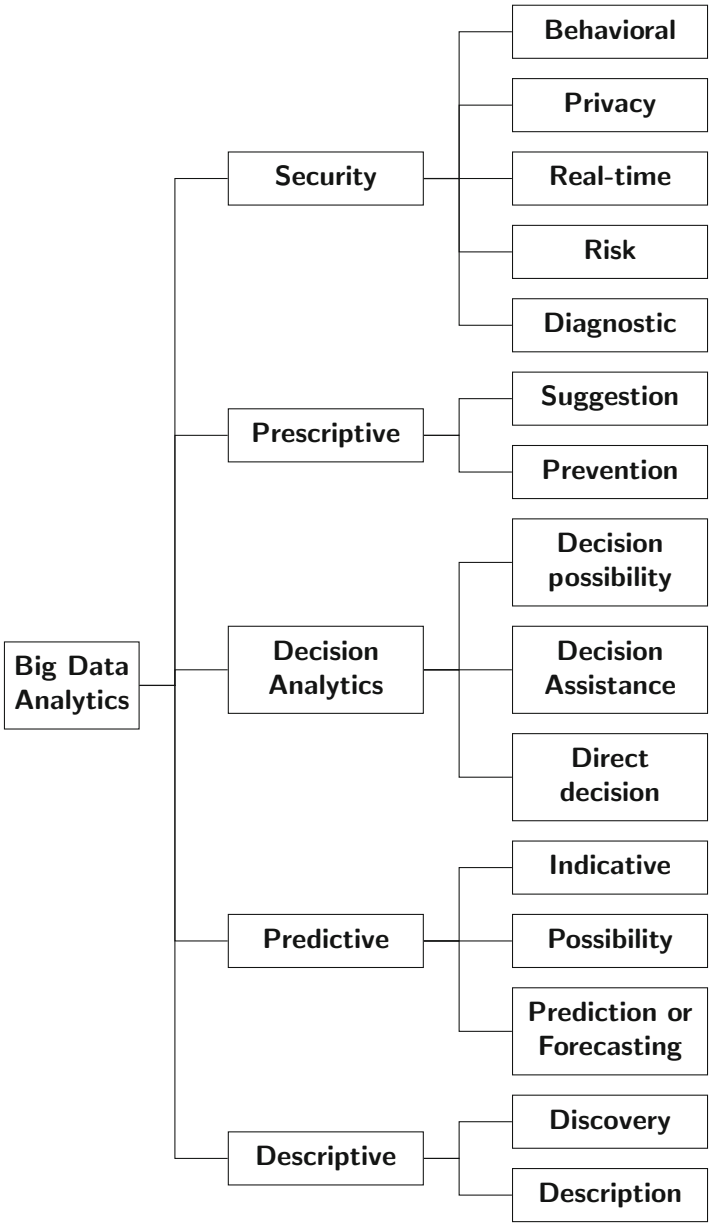


Fig. 2 Taxonomy of Big Data analytics

3.4 Prescriptive Analytics

The prescriptive analytics is a logical analysis of a huge dataset to fetch a suggestion. The prescriptive analytics also is further classified into two subcategories: suggestion (recommendation) and preventive analytics. Suggestion or recommendation: list of curing such kind of cancer. How has the curing process been done? Prevention: what is the reason for cancer?

3.5 Security Analytics

The security analytics is a logical analysis of huge data for security purposes. The analysis of the very large dataset is not possible in a conventional way. That's why the Big Data analytics is deployed to ensure security. The security analytics is classified into five subclasses, namely, diagnostic analytics, risk analytics, real-time analytics, privacy analytics, and behavioral analytics. Diagnostic: what is the type of cancer detected? Risk: is it false detection? Real-time: what is the current stage of the cancer after performing X treatment? Privacy: does the DNA made public or keep under privacy policy? Behavioral: what is the abnormality seen after performing X treatment?

4 Applications of Big Data Analytics in Medical Research

4.1 Neurology Research

The size of a brain is enormous to store and process. It requires very large-scale computing capability. The brain consists of neurons. The neurons of the human brain are 86 billion to compute which takes 17 million years in a conventional system [34]. Therefore, the Big Data is a solution to such mammoth-sized data. The data can be downloaded to analyze locally, but the data sizes are too big to do the task locally, and that is why, there is no point to move out from a cloud-based solution [43]. A neuroscientist manually tracks each axon from a very large dataset [67]. The neuron networks and nervous systems are very complex to analyze. Deploying Big Data helps in analyzing and visualizing those perplex neurons. For example, Big Graph can be deployed to analyze the relationship among the neuron network. Furthermore, deploying Big Data analytics is also very helpful in the analysis of such kind of gigantic database.

4.2 Genome Research

The genome contains all requirements of maintaining the structure of an organism. Genomes are made up of DNA. The DNA contains unique information to keep an organism's growth, health, and development. The genome-wide association studies (GWAS) analyze billions of single-nucleotide polymorphisms (SNPs), along with disease status [28, 59]. The analysis can be used to predict disease status. The Big Data creates an opportunity of data-driven prediction of diseases [24]. The key research challenge in a genome is the sequencing and alignment of genome data in large-scale space. A DNA contains ACGT code. Thus, human DNA contains more than three billions of such code. This code specifies every detail of a person and slight variation of this code can produce different species. Therefore, the information about detailed characteristics of any living things is encoded in the form of ACGT. A few such kind of DNA can form petabytes. It is reported that the size of genome data is petabytes [60, 68]. The DNA database sizes become enormous to store. It is impossible to manage the genome database without the Big Data technology. The Big Data analytics can play a vital role in the analysis of such mammoth size of a database.

4.3 Cancer Research

The Big Data and cancer research are very hot topic nowadays. The cancer research requires computational skills [28, 36, 64]. The data of cancer are very large to analyze, and therefore, the Big Data plays a vital role in cancer treatment, cancer prevention, and screening. A doctor can match available cancer trials with the patient sample rapidly even if the data size is huge [26]. The International Cancer Genome Consortium (ICGC) consists of two petabytes of cancer data in a 5-year period [68]. The genome sequences of tumors from patients with breast cancer are compared with the numerous available breast cancer genomes and searched for similar patterns in diverse cancer kinds which is a time-consuming process [43, 58]. In addition, a quote from Arend Sidow of Stanford University said, "If I could, I would routinely look at all sequenced cancer genomes. With the current infrastructure, that's impossible" [43]. Because a genome's size is 100GB that contributes tremendous size to a database, most of the clinical data are unstructured [33]. It requires Big Data to store, process, and analyze this plethora of data where 1,00,000 genomes of 75,000 patients have been studied by [50]. Moreover, the conventional database cannot cope up with the exponentially increasing size of the number of patients. The future lies within the Big Data technology; the sample data can easily be sequenced for better study and analyze a huge set of a sample of cancer patients. Chen et al. [15] develop a framework, called BUFAM, for breast cancer in the environment of highly heterogeneous biomedical big data. Moreover, biomarker is a new idea that is emerging for complex biomedical feature analysis

based on Big Data paradigm, like cancer [37, 74]. It is believed that engaging Big Data improves the screening, diagnosis, and treatment of tumor [2]. Fortunately, the National Institute of Health (NIH) reported that the cancer patient death rate declines nowadays.

4.4 Healthcare

Another aspect of the BBDE is biocuration [25]. The Big Data enhances the biocuration process through a well-established database to make a good decision. Howe et al. [25] state that the biocuration still lags behind the data generation in funding, development, and recognition. It requires Big Data technology in a large-scale environment. The Big Data assists in providing good health by exposing in-depth insights on causes and outcomes of a disease, precision medicine, and disease prediction and prevention [32]. The Big Data analytics are very useful in the prevention of epidemic disease. Mooney et al. [47] predict that the Big Data will be practicing in the future as today in an epidemic. The healthcare system must ensure eco-friendly diagnosis, screening, treatment, and prevention. Philip Bourne quotes:

> Our mission is to used data science to foster an open digital ecosystem that will accelerate efficient and cost-effective biomedical research to enhance health, lengthen life and reduce illness and disability. [7]

The Big Data helps in generating new knowledge in biomedical research and keep practitioner up to date [48].

5 Issues and Challenges

5.1 Data Gathering

The first challenge is gathering biomedical data. There is a diverse source of gathering biomedical data. However, gathering of biomedical data requires many years. Thus, collecting real biomedical data concerns not only money but also a time. The challenges are data collection of DNA of every citizen, and collection of data from every hospital, pathology, and research center. In addition, biomedical data gathering also requires human resources.

5.2 Data Storage

Let us assume the DNA database of the most populous country: India or China. Now, the government wishes to collect the DNA information of every person for various purposes, for instance, security. Now, collecting those data will take many years by the government. Imagine, what will be the database size? There is a requirement of a mammoth-sized data storage engine. Moreover, it also requires a creation of a different logical database which contains breast cancer, brain cancer, etc., separately depending on the disease. Now, these plethora of DNA data are used to study for diagnosing and treatment purposes. It makes the medical system easy if a system can analyze the gigantic database in a few seconds. However, deploying the Big Data analytics can make it possible to analyze all those data. But still, it takes time to analyze all those data. It will take huge manpower and processing to understand the entire DNA structure. Even today, the biomedical data engineering has a negligible amount of progress in research and yet to do much more. For instance, TRENCADIS is an effort on medical database creation [42].

Moreover, the medical information requires the metadata server to ensure the future accessibility of the data efficiently and effectively. The metadata defines how to store, where to store, and how to retrieve [51, 53]. Li et al. [38] implement metadata management systems for high-content screening. The data management requires the scalable metadata management system to cope up with ever-growing dataset in biomedical system. However, the metadata server is decoupled from data server for high manageability. Therefore, the data server does not affect the metadata server and vice versa. Moreover, the scalability is the biggest issue of healthcare system. The scalability can easily be achieved by decoupling metadata server from data server [51]. Besides, it also provides location-independent metadata management system.

5.3 Data Processing and Visualization

There is a call for a Big Data processing engine to process a jumbo-sized data. The biomedical data processing and visualization engine can be developed using MapReduce directly or using some famous framework, for instance, Spark. The data size is gigantic to process and visualize. The processing takes many days to many years in the conventional system and visualization cannot be done manually or using conventional system due to bulky size. The data are collected from diverse source and stitched together to present the data in a meaningful way. Therefore, there is always a call for engagement of Big Data technology to handle biomedical data.

5.4 Value

The collected data are enormous in size. Why are the data collected? Can we assign a worth to the collected data or simply dumping it? The Big Data always have concerns about extractions of worthy information from the collected data. The collected data is worthy only when we use those data for the betterment of the biomedical system. Thus, the involvement of Big Data in biomedical data always gives a worthy sense. The challenge lies in assigning value to the collected data and extracting value from the same data.

5.5 Academic Research

There are many countries which have a huge gap between doctors and engineers in the biomedical academic research fields. Introducing the BBDE course in both medical institutes and engineering institutes bridges the gap between the doctors and engineers. Thus, biomedical engineering can be enhanced. The research collaboration among engineers, doctors (medical), and industry people makes advancement in the biomedical data engineering. However, this is the most prominent issues in all countries. The collaborative research leads to better results. E-infrastructure is an example of a collaborative research [4].

5.6 Data Privacy

Most of the biomedical data are private data, and thus, privacy is the prime factor to be maintained in biomedical database. This privacy requirement creates an issue in creating open database worldwide. This is the reason the worldwide database creation requires privacy protection, and thus, the database cannot grow. Eric Schadt [59] emphasized the privacy of personal data in a quote:

> Genomic information has been the main focus of past debates on the protection of privacy and is subject to more legal regulations than other forms of high-dimensional molecular data such as RNA levels.

Moreover, Schadt also indicates that the Internet breaches the privacy of individuals. Earlier, the protection of privacy was easier than today's landscape of the BBDE.

5.7 Real-Time Processing

The modern healthcare system requires real-time profiling of patient which poses a big challenge to achieve. It is required to monitor continuously every parameter

of a patient automatically and remotely to enhance healthcare system [31, 70], for example, heartbeat. However, there are modern stream processing engines to implement the real-time healthcare system requirements, for example, Apache Spark. However, it is still called for a large-scale stream processing for healthcare systems.

6 Opportunity

There are abundant database available to perform research work. However, the privacy requirement becomes a barrier in creating open databases. There are numerous opportunities in biomedical research. Moreover, there are numerous biomedical databases available, namely, the International Cancer Genome Consortium (ICGC) [68], National Institute of Health (NIH), the Cancer Genome Atlas (TCGA), National Cancer Institute (NCI), National Human Genome Research Institute (NHGRI) [55, 56], CanSAR, NCI Genomic Data Commons, Cancer Genomics Hub, Pennsylvania Cancer Alliance Bioinformatics Consortium (PCABC) Biorepository, genome-wide association studies (GWAS) [28], Pancreatic Expression Database (PED), METABRIC repository, etc. [16, 50, 58]. There are also research opportunities in biocuration process, personalized medicine, cardiovascular care [57], MOOM [71], BigBWA [1], ENS@T-CANCER [65], SUPER-FOCUS [63], etc. Moreover, there are endless opportunities to work as a biomedical engineer. Orthology research is also another important field in data-intensive computation [66].

There are numerous opportunities in the field of genome research. The massive, open, online, and medicine (MOOM) is used to detect the genetic causes of disorders [71]. Topol [71] targets to collect nearly five million individual genomes for sequencing, which will contribute petabytes of data. However, collecting, storing, and processing the genome data are very time-consuming processes [3]. The BigBWA is a Hadoop-based solution for genome sequencing and alignment [1]. Moreover, the metagenomic shotgun sequencing is also another key research focus. The SUPER-FOCUS is an unannotated shotgun metagenome data sequencer [63]. Moreover, the GMQL is a Hadoop-based query language for abstractions of genomic region data and associated experimental, biological, and clinical metadata and interoperability between many data formats [44]. The GeneGrid provides a seamless integration of diverse data source to make strides of bioinformatics research work [29].

The opportunity for cancer research is provided by The Cancer Genome Atlas (TCGA) for better research, treatment, and prevention. The TCGA is the most famous framework for cancer research which is a result of a joint collaboration of the National Cancer Institute (NCI) and the National Human Genome Research Institute (NHGRI) [55, 56]. In addition, more cancer-related data are found, namely, CanSAR, NCI Genomic Data Commons, International Cancer Genome Consortium, Cancer Genomics Hub, Pennsylvania Cancer Alliance Bioinformatics Consortium

(PCABC) Biorepository, genome-wide association studies (GWAS) [28], Pancreatic Expression Database (PED), METABRIC repository, etc. [16, 50, 58].

7 Biomedical Future

The future of biomedical imaging will be 3D or beyond. To visualize these types of images requires huge computational power on a massive scale. Moreover, most of the clinical data will be recorded in the form of digital data. Therefore, the database size will continue to increase. However, there is also Moore's law. The Big Data will make an easier prediction, analysis, prevention, and curation. The cloud model also helps in reducing the cost of diagnosis, treatment, research, and analysis, but vendor lock-in is a problem too. However, treatment of a disease will become very cost-efficient in the future because biomedicine will depend on the data hosted on Big Data paradigm. In addition, the future of epidemiologist will require technical skills [47]. The knowledge of computer programming will be the sole criterion for an epidemiologist. Rare human resource of having a combination of science and computing knowledge for the BBDE [45]. A new data scientist will be required for the BBDE. Most of the healthcare system will be transferred to Big Data for research, prediction, prevention, etc. Those systems will be a computer program to assist the doctors automatically. Furthermore, the cancer database will continue to grow. Therefore, the death due to cancer will continue to decline. Moreover, the prior prediction of cancer disease will be possible in the near future. In the current scenario, the implication is data availability. However, this will not be an implication in the near future, but the technological singularities will arise.

8 Conclusions

The Big Data and biomedical data have a close relationship, and the merger forms a new paradigm called BBDE. This is an opportunity for the research community to work on this useful research. The future of the BBDE depends on Big Data paradigm. The treatment, prevention, and curing process are done using the Big Data technology. Moreover, there are many frameworks that are yet to be developed. The biomedical data are stored in digital forms. Thus, the research on biomedical data has become easier for engineers and doctors. The BBDE has many barriers to overcome. Moreover, the advancement of the Big Data technology creates an easier way to perform research work on the BBDE. In addition, the Big Data enables big biomedical data analytics and BBDE. BBDE suffers from storage issues, and thus, deploying Big Data technology can solve the issue. However, the BBDE is an emerging area of research for the research community and the welfare of human being.

References

1. Abuin, J. M., Pichel, J. C., Pena, T. F., & Amigo, J. (2015). BigBWA: Approaching the burrows-wheeler aligner to big data technologies. *Bioinformatics, 31*(24), 4003–4005.
2. Adams, J. U. (2015). Genetics: Big hopes for big data. *Nature, 527*(7578), S108–S109.
3. Al Aziz, M. M., Hasan, M. Z., Mohammed, N., & Alhadidi, D. (2016). Secure and efficient multiparty computation on genomic data. In *Proceedings of the 20th International Database Engineering & Applications Symposium* (pp. 278–283). New York: ACM. https://doi.org/10.1145/2938503.2938507.
4. Andronico, G., Ardizzone, V., Barbera, R., Becker, B., Bruno, R., Calanducci, A., Carvalho, D., Ciuffo, L., Fargetta, M., Giorgio, E., La Rocca, G., Masoni, A., Paganoni, M., Ruggieri, F., & Scardaci, D. (2011). e-infrastructures for e-science: A global view. *Journal of Grid Computing, 9*(2), 155–184. https://doi.org/10.1007/s10723-011-9187-y.
5. Baker, S., Xiang, W., & Atkinson, I. (2017). Internet of things for smart healthcare: Technologies, challenges, and opportunities. *IEEE Access*, (99), 1–1. https://doi.org/10.1109/ACCESS.2017.2775180.
6. Bates, D. W., Saria, S., Ohno-Machado, L., Shah, A., & Escobar, G. (2014). Big data in health care: Using analytics to identify and manage high-risk and high-cost patients. *Health Affairs, 33*, 1123–1131.
7. Bender, E. (2015). Big data in biomedicine: 4 big questions. *Nature, 527*(7576), S19.
8. Bonenfant, M., Desai, B. C., Desai, D., Fung, B. C. M., Özsu, M. T., & Ullman, J. D. (2016). Panel: The state of data: Invited paper from panelists. In *Proceedings of the 20th International Database Engineering & Applications Symposium* (pp. 2–11). New York: ACM. https://doi.org/10.1145/2938503.2939572.
9. Bourne, P. E., Lorsch, J. R., & Green, E. D. (2015). Perspective: Sustaining the big-data ecosystem. *Nature, 527*(7576), S16–S17. https://doi.org/10.1038/527S16a.
10. Branson, A., McClatchey, R., Goff, J. M. L., & Shamdasani, J. (2014). Cristal: A practical study in designing systems to cope with change. *Information Systems, 42*, 139–152. https://doi.org/10.1016/j.is.2013.12.009.
11. Bromley, D., Rysavy, S. J., Su, R., Toofanny, R. D., Schmidlin, T., & Daggett, V. (2014). Dive: A data intensive visualization engine. *Bioinformatics, 30*(4), 593–595.
12. Cassavia, N., Ciampi, M., De Pietro, G., & Masciari, E. (2016). A big data approach for querying data in EHR systems. In *Proceedings of the 20th International Database Engineering & Applications Symposium* (pp. 212–217). New York: ACM. https://doi.org/10.1145/2938503.2938539.
13. Chen, C. P., & Zhang, C. Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information Sciences, 275*, 314–347. https://doi.org/10.1016/j.ins.2014.01.015.
14. Chen, H. Y., Hsiung, M., Lee, H. C., Yen, E., Lin, S. C., & Wu, Y. T. (2010). GVSS: A high throughput drug discovery service of avian flu and dengue fever for EGEE and EUAsiaGrid. *Journal of Grid Computing, 8*(4), 529–541. https://doi.org/10.1007/s10723-010-9159-7.
15. Chen, H., Chen, W., Liu, C., Zhang, L., Su, J., & Zhou, X. (2016). Relational network for knowledge discovery through heterogeneous biomedical and clinical features. *Scientific Reports, 6*, 29915.
16. Clare, S. E., & Shaw, P. L. (2016). "Big data" for breast cancer: where to look and what you will find. *NPJ Breast Cancer, 2*, 16031.
17. Council, N. I. (2008). *Disruptive technologies global trends 2025. Six technologies with potential impacts on us interests out to 2025*. Accessed on 25 November 2017 from https://fas.org/irp/nic/disruptive.pdf
18. Cuzzocrea, A., Saccà, D., & Ullman, J. D. (2013). Big data: A research agenda. In *Proceedings of the 17th International Database Engineering & Applications Symposium* (pp. 198–203). New York: ACM. https://doi.org/10.1145/2513591.2527071.

19. Desai, B. C. (2014). The state of data. In *Proceedings of the 18th International Database Engineering & Applications Symposium* (pp. 77–86). New York: ACM. https://doi.org/10.1145/2628194.2628229.
20. Desai, B. C. (2014). Technological singularities. In *Proceedings of the 19th International Database Engineering & Applications Symposium* (pp. 10–22). New York: ACM. https://doi.org/10.1145/2790755.2790769.
21. Dunn, W., Burgun, A., Krebs, M. O., & Rance, B. (2016). Exploring and visualizing multidimensional data in translational research platforms. *Brief Bioinformatics*, bbw080.
22. Editorial. (2016). The power of big data must be harnessed for medical progress. *Nature, 539*(7630), 467–468. https://doi.org/10.1038/539467b.
23. Emeakaroha, V. C., Maurer, M., Stern, P., Łabaj, P. P., Brandic, I., & Kreil, D. P. (2013). Managing and optimizing bioinformatics workflows for data analysis in clouds. *Journal of Grid Computing, 11*(3), 407–428. https://doi.org/10.1007/s10723-013-9260-9.
24. Greene, A. C., Giffin, K. A., Greene, C. S., & Moore, J. H. (2016). Adapting bioinformatics curricula for big data. *Brief Bioinformatics, 17*(1), 43–50.
25. Howe, D., Costanzo, M., Fey, P., Gojobori, T., Hannick, L., Hide, W., Hill, D. P., Kania, R., Schaeffer, M., Pierre, S. S., Twigger, S., White, O., & Rhee, S. Y. (2008). Big data: The future of biocuration. *Nature, 455*(7209), 47–50.
26. Hoxha, J., & Weng, C. (2016). Leveraging dialog systems research to assist biomedical researchers' interrogation of big clinical data. *Journal of Biomedical Informatics, 61*, 176–184.
27. Huang, Z., Ayday, E., Lin, H., Aiyar, R. S., Molyneaux, A., Xu, Z., Fellay, J., Steinmetz, L. M., & Hubaux, J. P. (2016). A privacy-preserving solution for compressed storage and selective retrieval of genomic data. *Genome Research, 26*, 1687–1696.
28. Jiang, X., & Neapolitan, R. E. (2015). Evaluation of a two-stage framework for prediction using big genomic data. *Brief Bioinformatics, 16*(6), 912–921.
29. Jithesh, P. V., Donachy, P., Harmer, T., Kelly, N., Perrott, R., Wasnik, S., Johnston, J., McCurley, M., Townsley, M., & McKee, S. (2006). GeneGrid: Architecture, implementation and application. *Journal of Grid Computing, 4*(2), 209–222. https://doi.org/10.1007/s10723-006-9045-5.
30. Karasneh, Y., Ibrahim, H., Othman, M., & Yaakob, R. (2009). A model for matching and integrating heterogeneous relational biomedical databases schemas. In *Proceedings of the 2009 International Database Engineering & Applications Symposium* (pp. 242–250). New York: ACM. https://doi.org/10.1145/1620432.1620458.
31. Khazaei, H., McGregor, C., Eklund, M., El-Khatib, K., & Thommandram, A. (2014). Toward a big data healthcare analytics system: A mathematical modeling perspective. In *2014 IEEE World Congress on Services* (pp. 208–215). https://doi.org/10.1109/SERVICES.2014.45.
32. Khoury, M. J., & Ioannidis, J. P. A. (2014). Big data meets public health. *Science, 346*(6213), 1054–1055.
33. Khozin, S., Kim, G., & Pazdur, R. (2017). Regulatory watch: From big data to smart data: FDA's informed initiative. *Nature Reviews Drug Discovery, 16*(5), 306.
34. Landhuis, E. (2017). Neuroscience: Big brain, big data. *Nature, 541*(7638), 559–561.
35. Laney, D. (2015, February). Gartner predicts three big data trends for business intelligence. *Gartner*, *12*. Retrieved on December 10, 2016, from http://www.forbes.com/sites/gartnergroup/2015/02/12/gartner-predicts-three-big-data-trends-for-business-intelligence/
36. Levine, A. G. (2014). An explosion of bioinformatics careers. *Science*. https://doi.org/10.1126/science.opms.r1400143.
37. Li, G., Bankhead, P., Dunne, P. D., O'Reilly, P. G., James, J. A., Salto-Tellez, M., Hamilton, P. W., & McArt, D. G. (2016). Embracing an integromic approach to tissue biomarker research in cancer: Perspectives and lessons learned. *Brief Bioinformatics*, 1–13. https://doi.org/10.1093/bib/bbw044.
38. Li, S., Besson, S., Blackburn, C., Carroll, M., Ferguson, R.K., Flynn, H., Gillen, K., Leigh, R., Lindner, D., Linkert, M., Moore, W. J., Ramalingam, B., Rozbicki, E., Rustici, G., Tarkowska, A., Walczysko, P., Williams, E., Allan, C., Burel, J. M., Moore, J., & Swedlow, J. R. (2016)

Metadata management for high content screening in OMERO. *Methods 96*(Supplement C), 27–32 https://doi.org/10.1016/j.ymeth.2015.10.006, high-throughput Imaging.
39. Liu, J., Pacitti, E., Valduriez, P., & Mattoso, M. (2015). A survey of data-intensive scientific workflow management. *Journal of Grid Computing, 13*(4), 457–493. https://doi.org/10.1007/s10723-015-9329-8.
40. Lynch, C. (2008). Big data: How do your data grow? *Nature, 455*(7209), 28–29. https://doi.org/10.1038/455028a.
41. Maddineni, S., Kim, J., El-Khamra, Y., & Jha, S. (2012). Distributed application runtime environment (dare): A standards-based middleware framework for science-gateways. *Journal of Grid Computing, 10*(4), 647–664. https://doi.org/10.1007/s10723-012-9244-1.
42. Maestre, C., Segrelles Quilis, J. D., Torres, E., Blanquer, I., Medina, R., Hernández, V., & Martí, L. (2012). Assessing the usability of a science gateway for medical knowledge bases with TRENCADIS. *Journal of Grid Computing, 10*(4), 665–688. https://doi.org/10.1007/s10723-012-9243-2.
43. Marx, V. (2013). Biology: The big challenges of big data. *Nature, 498*(7453), 255–260. https://doi.org/10.1038/498255a.
44. Masseroli, M., Pinoli, P., Venco, F., Kaitoua, A., Jalili, V., Palluzzi, F., Muller, H., & Ceri, S. (2015). GenoMetric query language: a novel approach to large-scale genomic data management. *Bioinformatics, 31*(12), 1881–1888.
45. Mattmann, C. A. (2013). Computing: A vision for data science. *Nature, 493*(7433), 473–475. https://doi.org/10.1038/493473a.
46. McClatchey, R., Branson, A., & Shamdasani, J. (2016). Provenance support for biomedical big data analytics. In *Proceedings of the 20th International Database Engineering & Applications Symposium* (pp. 386–391). New York: ACM. https://doi.org/10.1145/2938503.2938540.
47. Mooney, S. J., Westreich, D. J., & El-Sayed, A. M. (2015). Epidemiology in the era of big data. *Epidemiology (Cambridge, MA), 26*(3), 390–394. https://doi.org/10.1097/EDE.0000000000000274.
48. Murdoch, T. B., & Detsky, A. S. (2013). The inevitable application of big data to health care. *JAMA, 309*(13), 1351–1352.
49. Nielsen, C. B., Younesy, H., O'Geen, H., Xu, X., Jackson, A. R., Milosavljevic, A., Wang, T., Costello, J. F., Hirst, M., Farnham, P. J., & Jones, S. J. M. (2012). Spark: A navigational paradigm for genomic data exploration. *Genome Research, 22*(11), 2262–2269.
50. Noor, A. M., Holmberg, L., Gillett, C., & Grigoriadis, A. (2015). Big data: The challenge for small research groups in the era of cancer genomics. *British Journal of Cancer, 113*(10), 1405–1412.
51. Patgiri, R. (2016). MDS: In-depth insight. In *2016 International Conference on Information Technology (ICIT)* (pp. 193–199). https://doi.org/10.1109/ICIT.2016.048.
52. Patgiri, R., & Ahmed, A. (2016). Big data: The v's of the game changer paradigm. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)* (pp. 17–24). Sydney: IEEE. https://doi.org/10.1109/HPCC-SmartCity-DSS.2016.0014.
53. Patgiri, R., Dev, D., & Ahmed, A. (2018). dMDS: Uncover the hidden issues of metadata server design. In *Progress in intelligent computing techniques: Theory, practice, and applications: Proceedings of ICACNI 2016* (Vol. 1, pp. 531–541). Singapore: Springer. https://doi.org/10.1007/978-981-10-3373-5_53.
54. Rider, A. K., & Chawla, N. V. (2013) An ensemble topic model for sharing healthcare data and predicting disease risk. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics* (pp. 333:333–333:340). New York: ACM. https://doi.org/10.1145/2506583.2506640
55. Robbins, D. E., Gruneberg, A., Deus, H. F., Tanik, M. M., & Almeida, J. (2013). TCGA toolbox: an open web app framework for distributing big data analysis pipelines for cancer genomics. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics* (pp. 62–67).

56. Robbins, D. E., Gruneberg, A., Deus, H. F., Tanik, M. M., & Almeida, J. S. (2013). A self-updating road map of the cancer genome atlas. *Bioinformatics, 29*(10), 1333–1340.
57. Rumsfeld, J. S., Joynt, K. E., & Maddox, T. M. (2016). Big data analytics to improve cardiovascular care: Promise and challenges. *Nature Reviews Cardiology, 13*(6). https://doi.org/10.1038/nrcardio.2016.42.
58. Saez-Rodriguez, J., Costello, J. C., Friend, S. H., Kellen, M. R., Mangravite, L., Meyer, P., Norman, T., & Stolovitzky, G. (2016). Crowdsourcing biomedical research: Leveraging communities as innovation engines. *Nature Reviews Genetics, 17*(8), 470–486.
59. Schadt, E. E. (2012). The changing privacy landscape in the era of big data. *Molecular Systems Biology, 8*(612), 1–3.
60. Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L., & Nolan, G. P. (2010). Computational solutions to large-scale data management and analysis. *Nature Reviews Genetics, 11*(9), 647–657.
61. Seife, C. (2015). Big data: The revolution is digitized. *Nature, 518*(7540), 480–481. https://doi.org/10.1038/518480a.
62. Shahand, S., Santcroos, M., van Kampen, A. H. C., & Olabarriaga, S. D. (2012). A grid-enabled gateway for biomedical data analysis. *Journal of Grid Computing, 10*(4), 725–742. https://doi.org/10.1007/s10723-012-9233-4.
63. Silva, G. G. Z., Green, K. T., Dutilh, B. E., & Edwards, R. A. (2016). Super-focus: A tool for agile functional analysis of shotgun metagenomic data. *Bioinformatics, 32*(3), 354–361.
64. Sinha, G. (2016). A career in cancer research? Computational skills wanted. *Science*. https://doi.org/10.1126/science.opms.r1600163.
65. Sinnott, R. O., Beuschlein, F., Effendy, J., Eisenhofer, G., Gloeckner, S., & Stell, A. (2016). Beyond a disease registry: An integrated virtual environment for adrenal cancer research. *Journal of Grid Computing, 14*(4), 515–532. https://doi.org/10.1007/s10723-016-9375-x.
66. Sonnhammer, E. L., Gabaldon, T., da Silva, A. W. S., Martin, M., Robinson-Rechavi, M., Boeckmann, B., Thomas, P. D., & Dessimoz, C. (2014). The quest for orthologs consortium: Big data and other challenges in the quest for orthologs. *Bioinformatics, 30*(21), 2993–2998.
67. Srinivasan, R., Li, Q., Zhou, X., Lu, J., Lichtman, J., & Wong, S. T. (2010). Reconstruction of the neuromuscular junction connectome. *Bioinformatics, 26*(12), i64–i70.
68. Stein, L. D., Knoppers, B. M., Campbell, P., Getz, G., & Korbel, J. O. (2015). Data analysis: Create a cloud commons. *Nature, 523*(7559), 149–151.
69. Szabo, C., Sheng, Q. Z., Kroeger, T., Zhang, Y., & Yu, J. (2014). Science in the cloud: Allocation and execution of data-intensive scientific workflows. *Journal of Grid Computing, 12*(2), 245–264. https://doi.org/10.1007/s10723-013-9282-3.
70. Ta, V. D., Liu, C. M., & Nkabinde, G. W. (2016). Big data stream computing in healthcare real-time analytics. In *2016 IEEE international conference on cloud computing and big data analysis (ICCCBDA)* (pp. 37–42). https://doi.org/10.1109/ICCCBDA.2016.7529531.
71. Topol, E. J. (2015). The big medical data miss: Challenges in establishing an open medical resource. *Nature Reviews Genetics, 16*(5), 253–254.
72. Watts, N. A., & Feltus, F. A. (2017). Big data smart socket (BDSS): A system that abstracts data transfer habits from end users. *Bioinformatics, 33*(4), 627–628.
73. Weil, A. R. (2014). Big data in health: A new era for research and patient care. *Health Affairs, 33*, 1110.
74. Zeng, T., Zhang, W., Yu, X., Liu, X., Li, M., & Chen, L. (2016). Big-data-based edge biomarkers: Study on dynamical drug sensitivity and resistance in individuals. *Brief Bioinformatics, 17*(4), 576–592.

Big Data Preprocessing: An Application on Online Social Networks

Androniki Sapountzi and Kostas E. Psannis

1 Introduction

Social networking data is a significant source of big data because they are voluminous, even from a single social network service (SNS), are mostly unstructured, and are evolving at an extremely fast pace. In big data theory, these dimensions are respectively called data volume, data variety, and data velocity. To mine insightful patterns from these data requires advanced data management and analysis methods, termed as Big Data Analytics. This is a complex process that demands expertise in gathering, processing and cleaning data, selecting proper analysis methods, understanding trade-offs of the chosen algorithms, and interpreting and exploiting the results.

An additional dimension of big data is that of data veracity [1] which came as a response to the weakness of messy data. Messy data are characterized by incompleteness, inconsistencies, and timeliness. In addition, data are unstructured, highly dimensional, ambiguous, or imbalanced, and they have to be preprocessed so as to satisfy the requirements of the algorithms used for analysis. The success of many machine learning algorithms depends highly on the preparation phase. Social networking data analysis faces all the abovementioned issues. Figure 1 depicts how the low data quality can affect all the stages of analytics process. Each process depicted in a rectangle is informed and affected by the previous process.

A. Sapountzi (✉)
Department of Behavioral and Movement Sciences, Vrije Universiteit, Amsterdam, The Netherlands
e-mail: a.sapountzi@vu.nl

K. E. Psannis
Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece
e-mail: kpsannis@uom.edu.gr

© Springer Nature Switzerland AG 2020

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_4

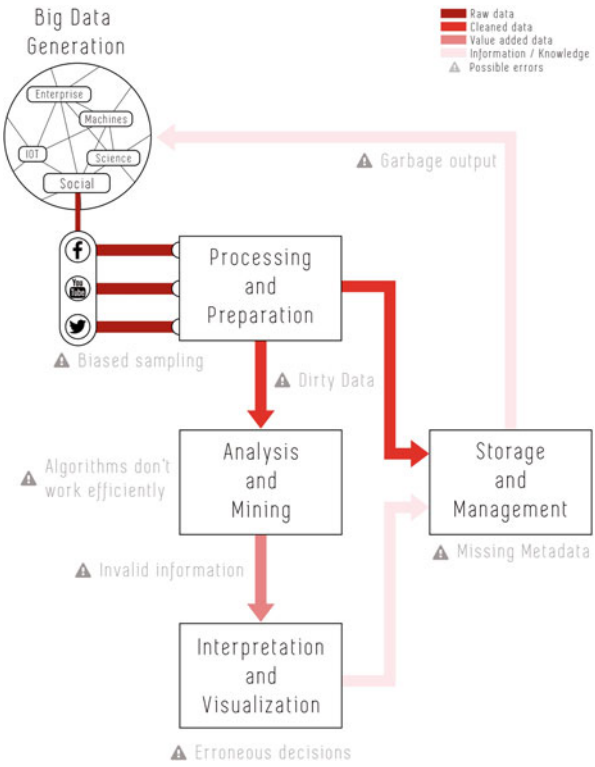


Fig. 1 The impact of low-quality data on each stage of big data analytics in online social networks

The purpose of data processing is both to revert the data to a format capable for the analysis process and to ensure the high quality of data. As the volume, variety, complexity, and velocity of data grow, data preprocessing becomes even more challenging. The absence of studies of data quality and preparation for online social network (OSN) data makes this process even more tedious.

Data cleansing and feature engineering are two of the involved tasks in a preprocessing pipeline that can play a vital role in improving model interpretability, prediction accuracy, and performance. Cleansing is a long-standing task in data analysis that deals with missing, inaccurate, and noisy data. Feature engineering is a task linked to machine learning algorithms and modifies, creates, or eliminates unnecessary features from the dataset. Both tasks require domain expertise and could reduce computational cost and complexity [2] of algorithms.

To the best of our knowledge, extant research regarding big data is primarily focused on analysis, whereas the preprocessing aspects remain largely unexplored. In this chapter, we aim to study and uncover issues of the preprocessing stage of big data by taking as an example content data created in OSNs. We define noise in the data source and provide a holistic overview and classification of preprocessing challenges and solutions. We illustrate both statistical and qualitative methods for data cleansing. We outline textual data preparation tasks and feature engineering

methods. Although the research is oriented toward social data networking analysis, the methods and algorithms reviewed are applied to any data source that shares similar data types.

The rest of this chapter is structured as follows. In Sect. 2, we present data types extracted from big data sources taking as an example that of OSN along with their analysis practices. The preprocessing stages and text are then introduced. The low-quality data related to them and the issues with regard to gathering and integration are briefly discussed. Qualitative and quantitative big data cleansing techniques are investigated in Sects. 3 and 4. Feature engineering and text preprocessing techniques are briefly discussed in Sect. 5. Contributions of a broad range of big data preprocessing frameworks and distributed frameworks for scalable analysis and preprocessing are presented in Sect. 6. Section 7 summarizes the main findings of this study and discusses the open issues raised for preprocessing big data in OSNs. It is worth noting that a single article cannot be a complete review of all the methods and algorithms. Yet, references cited can be used as guidance in narrower research directions in the field.

2 Data Quality Issues

There are many definitions of data quality and noise because both terms are domain dependent [3]. Generally, high-quality data are considered those that are representative for the indented problem or question being analyzed. Provided that the data are representative, data preprocessing tasks structure the data in a way that the analysis algorithms applied afterward can operate well enough so as to achieve high-accuracy results. Table 1 summarizes the application of the phases of data preprocessing for OSN.

2.1 Data Context

The study of online social networks (OSNs) is a quickly widening multidisciplinary area creating new research challenges for the understanding of the development and progression of socioeconomic phenomena. An SNS provides a specific type of service such as developing professional connections. The content and structural data occurred by this service are commonly represented in a large-scale graph G = (V, E), where V is a set of users or groups, called nodes which are connected by E, a set of relationships, called edges. A graph sometimes is defined by G = (V, E, A), where A is the adjacency matrix which shows which of the vertices are neighbors/adjacent to each other in the graph.

Big data sources, including that of OSNs, typically contain a tremendous amount of content and linkage data which can be leveraged for analysis. Depending on whether data are organized in a predefined manner, they are divided into structured

Table 1 Social networking data preprocessing tasks

Preprocessing task	Online social network
Noise treatment	Text informality
	Temporal ambiguity
	Spatial ambiguity
	Entity resolution
	Irrelevant data
	Outliers
	Mislabeled data
	Imbalanced data
Incomplete data	Content
	Nodes
	Links
	Attributes of nodes
	Labels of links
Data integration	Vocabulary schema
	Profile matching
Feature selection and scaling	Words
	Posts
	Tags
	Emoticons
	User profile
	Social relationships
	Interest relationships
	Activity relationships
	Time, date, and place
Feature extraction	Low-dimensional representation (e.g., word embedding)
	Annotation

and unstructured or semi-structured data. Content data are unstructured, while linkage data are graph structured.

The linkage data, also called graph data, is about patterns of interactions between the network entities (e.g., people, organizations, and items). They are divided into the following types:

(i) Activity-based relationships like “posted by” or “retweet”
(ii) Interest-based relations such as the number of likes or user-item ratings
(iii) Social-based relations such as number of common friends or the ratio of follower-followee, and network-structure features such as graph density

Content data, on the other hand, include multimedia, text, and clickstream data shared and created in the network. Analysis of posts, demographic and other user information, keywords, hashtags, and tags are all examples leveraged by social networking examples.

2.2 Data Analysis Types

Analysis practices make use of machine learning models which can be categorized into supervised, unsupervised, and reinforcement learning models.

Supervised is the case where the data used to train the model comprises examples both of the input vectors and their corresponding target vectors, whereas unsupervised is the case where the corresponding target vectors are not available. The reinforcement learning model is not common for social networking data analysis, and it will not be discussed any further.

Another distinction is between regression and classification. The former considers a continuous outcome variable while the latter a categorical one.

Prevalent analysis practices include the following:

1. Social network analysis (SNA)
2. Collaborative recommendation
3. Topic detection and tracking
4. Trend analysis
5. Sentiment analysis
6. Opinion mining

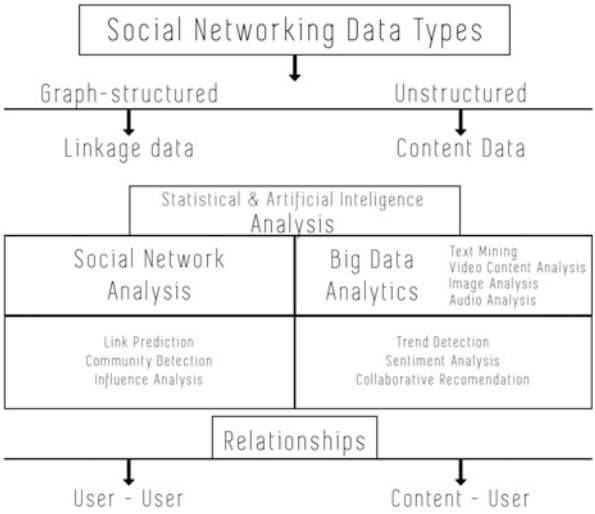
Supervised models are widely used in sentiment analysis and event detection while unsupervised learning in trend detection and collaborative filtering [4]. Clickstream and content data are enormously exploited in collaborative filtering, sentiment analysis, and opinion mining. Linkage data are extensively leveraged for social network analysis and collaborative filtering, and they are sometimes combined with content-based analytics [4]. Linkage and content data are usually grouped by geographical or temporal characteristics. Event detection, trend analysis, collaborative filtering, and SNA exploit frequently these features.

SNA does not fall into this categorization of models since it is a structured analysis for graphs. Other subsequent graph analytics techniques include information propagation, community detection, entity analysis, and link analysis. Table 2 summarizes the analysis practices and data structures in OSN.

2.3 Issues Solved by Preprocessing Tasks

Data preprocessing is divided into data cleansing, integration, transformation, and dimensionality reduction. It improves the overall generalization of the model by reducing variance (i.e., prevents overfitting) and reduces biased estimators (i.e., prevent underfitting). The aim of data reduction many times overlaps with that of transformation, and as such, there is no strict separation between the two. The two of them compose what is usually called feature engineering in a machine learning model.

Table 2 Social networking data analysis



Noise and missing values (MVs) are two long-standing issues on data instances. Building predictive or descriptive models on datasets containing missing or noisy values will affect the results. The magnitude of effect depends on the available data on hand and the processing requirements of the underlying analysis algorithm. To illustrate this with an example, naïve Bayes has a higher prediction accuracy on presence of missing data than decision trees. Data cleansing handles missing values, smoothes noisy data, and resolves inconsistencies and duplicates.

Noisy data are considered erroneous values, inconsistent data, and relevant data that get mixed with the irrelevant data. Noise is a term mostly being used in signals captured by sensors or machines. Random variation of brightness in a picture and high frequency in an electromagnetic signal are two examples of noise. Data inconsistencies commonly refer to the "noise" occurred because of erroneous entries by humans. The mismatched quantities between orders and sales of an inventory and between age and birth date of an individual are two examples of data inconsistencies. Noise and inconsistencies refer to the issue where different datasets that measure the same quantity do not agree with each other. It is worthy to note though that data inconsistencies are different to what is known as data outliers or anomalies, although the former can be seen as a generalization of the latter.

In structural analysis such as SNA, noise can be presented in terms of nodes. These include inactive nodes, artificial nodes, and duplicate nodes [5, 6]. Inactive nodes describe the fact of users who have an inactive account in social media, whereas artificial nodes represent spammers.

Analysis models cannot handle other values, e.g., blanks and Not A Number values, apart from numerical values. Additionally, missing data cause three main problems:

(i) Introducing a substantial amount of bias in the data generation process
(ii) Reducing the efficiency
(iii) Easiness of the implementation of analysis

The data generation process commands patterns of missing data which define the methods that should be used for resolving them. Specific data instances may be incomplete at random or not and may be caused by a human or a machine measurement error.

MVs in OSN are commonly caused by user-side values never edited in their profile or linkage data values that are not yet reflected in the network structure. Incomplete data could be distinguished in five forms for OSN: node, edge, attributes related to nodes, labels related to edges, and content components. MV in a structural analysis like SNA can have large negative effects on structural properties of the network [7–9].

Data integration covers the combination and loading of data residing in different sources that belong in the internal data ecosystem. An example includes merging data from different databases, such as a database with administrative data and dynamic data that come from the terminal of a client and being stored in the cloud.

Another challenge that big data has introduced is the high dimensionality in the feature or input space of unstructured or semi-structured data. Bellman referred to this phenomenon as the "curse of dimensionality" which denotes that as the dimensionality of the features rises, the amount of input data required to provide a reliable analysis grows exponentially [10]. The training time and memory usage then scale linearly with the number of input data points.

The feature space includes the independent variables, i.e., the attributes or otherwise called predictors, while target, class label, or outcome data include the dependent variable. The input space incorporates only the former or both. The literature sometimes refers to the feature space as the embedding space or as the vector space.

A large input space in terms of data instances can be handled with the utilization of scalable frameworks and architectures such as cloud computing, as briefly discussed in Sect. 6.1. A large feature space, on the other hand, is a much more sophisticated issue and introduces extra complexity to the algorithms. This is handled via feature engineering tasks, discussed in Sect. 5. Feature engineering shortens the training time of the algorithm which is a solution for overfitting.

2.4 Text Processing

Text can introduce inconsistencies, duplicates, and erroneous values, and it is handled via Natural Language Processing (NLP) techniques. Text analysis in online networks suffers additional problems because of the unstructured and often informal content posted in the network. The use of slang, abbreviations, and grammatically incorrect words should be taken into consideration. The fragmented and short

in length posts challenge even more the analysis algorithms. Providing structure to emoticons is another common preprocessing task. In case of text retrieval, URLs, stop words, quotations, punctuations, special characters, foreign words, extra spaces, and line breaks are always removed.

2.5 Entity Resolution

Entity resolution (ER) is needed to deduplicate and disambiguate the data. It is further discussed in subsect. 5.3.2. There are different semantic types both within edges and nodes in a source and across sources. For example, the network of Facebook in addition to friendship relationships between persons has got relationships of other types, such as person-photo tagging relationships. Combining data from multiple social networks causes a bigger expansion of relationships which sometimes lead in overlapping. For instance, when developing an interest graph (what people care about) by integrating data from multiple OSNs, there is ambiguity of words for the same interests such as "Likes" on Facebook or "Interests" in a LinkedIn profile. This data heterogeneity stems from the following differences in:

1. Relationship types and content sharing defined by an SNS
2. Terminology frameworks and the absence of a cross-domain vocabulary matching [11]

Duplicate nodes may occur as for example in a discussion network when person A replies to person B on multiple occasions. This makes the measurement of network metrics such as degree to be inaccurate. A solution to duplicate nodes is to be combined into a single weighted edge.

2.6 Data Extraction

Imperfect data acquisition process and communication failure while extracting data from the web are common problems in large-scale data retrieval.

The standard means of gathering data from the web including social networking data are the Application Public Interfaces (APIs). It is important to consider the traffic and networking aspects of the websites while extracting data in order to avoid communication failures which lead to incomplete data.

Providing some form of structure while extracting information from OSN is the first preprocessing step. This is done with the usage of lexicon resources and matching algorithms, or supervised classification models. There are also filtering rules provided by SNS API functionality and by data providers like DataSift, Gnip, and Spinn3r. These rules are commonly based on keywords, geo boundaries, conversation details, hashtags, phrase matches, and the activity type like share, comment, and like. A quality issue that arises from the extraction process is whether

the samples obtained via stream APIs and the filtering functions discard important data and bias the results.

Other irrelevant information that is considered noise in OSN and is omitted includes web robots, extensions of CSS, GTF, FLV, and the records with failed HTTP request.

3 Big Data Cleansing Approaches

The beforementioned issues can be handled via preprocessing tasks, implemented in the following:

(i) A qualitative way via the specification of rules and constraints
(ii) A quantitative way via the specification of statistical or machine learning models

Cleansing low-quality data is a hard-computational task that has been studied for decades. Cleansing data includes exploration and pattern recognition.

Understanding data dimensions and possible sources of errors is a first step toward developing a data cleansing technique [3], known as exploratory data analysis or understanding. Exploratory analysis is based on statistical measures such as frequencies, means, variances, and standard deviations used to collect statistics about individual attributes so as to understand the validness of the dataset and discover metadata. Regarding unstructured data, a searchable "map" of the metadata and full-text searches is created via exploratory analysis.

Pattern recognition includes usually unsupervised tasks like correlation and association rules. Pattern recognition and machine learning can be viewed as two facets of the same field.

Data cleansing is a multistage process that includes the following stages [12–14]:

1. Determine and/or identify data errors.
2. Correct errors by repairing or deleting them.
3. Document error examples and error types.
4. Modify data entry procedures to reduce future errors.

After data profiling and pattern detection, rules are derived for step (1), and then the data cleansing process iteratively runs steps (2) and (3). This process is also known as data munging in the case of predictive models. Effective big data munging requires the identification of a policy that specifies which correction has to be applied when an error can be corrected in several ways. Repair can be performed by using machines, human crowds, or a hybrid of both.

3.1 Error Identification

In step (1), errors can be specified either logically (qualitative) or statistically (quantitative) by knowledge bases [15], crowdsourcing [16], and domain experts or in a data-driven manner [17]. Table 3 illustrates big data preprocessing according to three variables: how, where, and by whom the data errors are detected and repaired. We analyze it gradually in this chapter.

Data-driven cleaning allows the rules to be learned from the dirty data itself and be validated incrementally as more data are gathered. Researchers [1] provide a big data quality management view that corrects data via a dependency model whose rules are learned by the data itself. Using information from the same OSN to cleanse and structure data produces more accurate results.

However, since data themselves are dirty, the need is to make these rules robust against outliers and to allow approximation [1]. Soft computing techniques are tolerant on imprecision and approximation.

Crowdsourcing [18] is used for labeling data for data cleansing and supervised learning. Active learning in crowdsourcing is often being used in combination with crowdsourcing, in which learning algorithms interactively query the user to provide labels to the "necessary" (i.e., unusual, uncertain) data points.

Knowledge bases and lexical resources have been extensively used to filter out noise [19–21]. WordNet is widely used for topic detection, while SenticNet is widely used for sentiment detection. The former covers semantic and lexical relations between terms and their meaning such as synonymy, hyponymy, and polysemy. SenticNet, on the other hand, includes the polarity of common-sense concepts. The limitation of using lexical resources in OSNs is that they do not contain many words or concepts being utilized in informal contexts. In addition to that, the development of non-English bases is needed. Many social networking data analyses have been done through the use of manually built keyword lexicons whose limitations for big data are apparent with the most obvious one to be their inability to scale.

The two approaches to cleanse and preprocess messy data are the qualitative, which follows the dependency theory, and the quantitative [1, 18]. Quantitative methods are listed as follows:

Table 3 Data cleansing techniques categorization

	How		
Who	Qualitative	Quantitative	Where
Knowledge base	Dependency rules: FD, CFD, DC	Natural language processing	Source (e.g., ETL)
Data		Statistics	Store (e.g., ELT)
Domain experts	Matching rules	Machine learning	
Crowdsourcing		Graph mining	

1. Statistics
2. Machine learning
3. NLP
4. Graph theory

3.2 Dependency Rules

The qualitative process uses logical reasoning over declarative data dependencies to detect errors and cleanse data. There are several classes of qualitative rules including functional dependencies (FD) and their extensions to conditional functional dependencies (CFD), and denial constraints (DC). Although these dependencies have been of paramount importance in database theory, they have been used in different fields like artificial intelligence and data processing.

Each rule usually targets a specific data quality error, which may be an instance feature or the whole dataset level, but interacting two types of quality rules may produce higher-quality results [22, 23].

Researchers [1, 12, 24, 25] refer to big data quality with a focus on CFD and FD because they specify a fundamental part of the semantics of data, and they are hence particularly important to real-life graphs. Finally, unlike relational data, graphs [24] and unstructured data typically do not come with a schema, and dependency rules have to specify not only the regularity between attribute values of the data points but also their topological structures.

Additionally, dependency rules like FD, CFD, and DC are considered as hard constraints where a specific set of conditions for the variables have to be satisfied. Data-driven cleansing approximated rules should be preferred against hard coded rules [1] in case of big data. The advantage of qualitative methods is the domain knowledge incorporation [1, 18]; quantitative techniques though are less prone to error introduced by domain knowledge.

3.3 Statistics and Machine Learning

Many qualitative techniques are amenable to a statistical analysis, and sometimes, there is an overlapping. There are also hybrid techniques [23, 26] that combine dependency rules with statistical methods. Qualitative techniques are more difficult to be implemented for big data sources except if there is an explicit set of rules and constraints that these should satisfy; hence, quantitative techniques are many times preferred. However promising, we do not cover graph theory for big data preprocessing apart from referring a few distinct examples.

Discovering dependency-based rules and computing a repair are cost prohibitive for big data. To cope with this, parallel scalable algorithms and distributed map-reduce processing may be used to lessen the issue of data volume; parallel

incremental algorithms, on the other hand, can deal with the velocity problem [1, 15]. Yet, rules are dependent both on the application and the data source hindering thus the scalability of a qualitative data cleanser to another application.

The quantitative approach relies traditionally on the use of statistical methods; recently, machine learning models are incorporated in this process [18]. Machine learning is closely related to statistics on that they both find structures and patterns in data. However, statistics emphasizes in low-dimensional problems, whereas machine learning in high-dimensional problems. Additionally, algorithms inspired by biological procedures found in nature belong to machine learning. Both machine learning and statistical models need to be modified in order to work for big data cases [27, 28].

Modifications of statistical models for big data should handle the unique features of incidental endogeneity, heterogeneity, spurious correlation, and noise accumulation [27–30]. Sophisticated model selection with modified regularized least square methods should handle incidental endogeneity. Efficient computation algorithms with regularization can deal with the heterogeneity to tackle the issue of statistical significance. Cross validation is a model selection technique. It subtracts the effects that spurious correlation has on feature selection and statistical significance. Feature engineering, discussed in Sect. 5, is proposed for addressing noise accumulation issues.

Many times, different terms in the areas of machine learning, statistics, or computer science are being used to describe the same entity. Table 4 depicts a few of these terms so as to help readers to have a broad idea of the terminology. Technically, these terms are not equivalent.

4 Machine Learning Algorithms for Preprocessing and Analysis

Machine learning models can produce false predictions in presence of noise and MVs.

In case of classification analysis, noise is divided into attribute and class noise. The latter includes contradictory examples (examples with identical input attribute values having different class labels) and misclassifications (examples which are incorrectly labeled). The objective of data extraction [19, 20], discussed in Sect. 2.6, is to distinguish relevant from irrelevant data. Supervised classifiers are widely used for this task.

In case of regression models, the independent variables are also considered noisy when there is dependence among them, namely, when they are not sampled independently from some underlying distribution.

Table 4 Different terminology used in machine learning, statistics, and computer science

Term	Similar terms
Observations	Data points
	Samples
	Instances
	Records
	Rows
Features	Independent variable
	Explanatory variable
	Attributes
	Columns
Target	Dependent variable
	Outcome variable
	Output
	Class or label (in classification)
Feature space	Vector space
	Embedding space
Data structure	Matrix
	Array
	Vector
	Tensor
	Graph
Penalization	Regularization
Estimation	Prediction
	Learning

4.1 Supervised Machine Learning

Predictive classifiers have been extensively studied and have shown remarkable success for analysis practices such as text classification, sentiment analysis, and event or topic detection [4, 19–21]. They are employed for preprocessing tasks such as classifying irrelevant data [6] and predicting missing values.

Supervised classification algorithms are affected from unnormalized data and missing values. Normalization is discussed in Sect. 5.1. Imputing missing values with the mean for all data points belonging to the same class is considered as a simple and effective strategy. A comparative study of imputing methods for supervised models is provided by [31].

4.1.1 Imbalanced Data

An additional issue is that of imbalanced data which occurs when there are more training examples for one class than another. This can cause the decision boundary to be biased toward the majority class.

Resampling techniques are used to balance the class distribution. The two main groups within resampling are under-sampling and oversampling methods [2, 32]. The first one creates a subset of the original dataset by eliminating the majority examples, whereas the second creates a superset of the original dataset by replicating or generating examples from existing ones. The advantageous part of resampling is that it is independent with the analysis algorithm applied afterward.

SMOTe is a recent oversampling technique that employs the Euclidean distance in nearest neighbors between data points in feature space.

Ensemble classification methods are capable of dealing with the imbalanced dataset problem [32, 33]. The core idea imitates the human desire to obtain several opinions before making any crucial decision. Combining different types of classifiers helps in improving predictive performance; this occurs mainly due to the phenomenon that various types of classifiers have different "inductive biases" [33].

Ensemble classifiers are divided into boosting, bagging, and stacking; and algorithms include AdaBoost, Rotation Forest, Random Forest, and LogitBoost. These are well suited when small differences in the training data produce very different classifiers like in decision trees.

Algorithmic modifications are also proposed to deal with the imbalanced data issue. Learning the features for a class using all training data being not in that class [34] is one such technique.

4.1.2 Algorithms and Preprocessing Requirements

Naïve Bayes is a probabilistic classifier that uses the Bayes theorem to predict the probability that a given feature (x) belongs to a particular class (c). Let us consider an example of classifying malicious mails, $c = malicious$, and a feature may be the presence of a text such as $x=$'*urgent action required*'. Equation (1) shows the posterior probability of $P(c|\, x)$ of the mail being malicious given the specific feature. The prior probability $P(c)$ is the initial belief of the mail being malicious before observing any data, P(x) is the marginal probability of observing that feature, and $P(x|\, c)$ is the likelihood of having x given c.

$$P\,(c|x) = \frac{P(c) * P\,(x|c)}{P(x)} \tag{1}$$

Given the naïve assumption of n number features being all independent, (1) is rewritten as (2) depicts

$$P\,(c|x) = \frac{P(c) * P\,(x_1|c) * P\,(x_2|c) * \ldots P\,(x_n|c)}{P(x)} \tag{2}$$

Because of the independence assumption, this classifier is highly scalable and can quickly learn to use high-dimensional features with limited training data. It has higher prediction accuracy on presence of missing data than many other classifiers like decision trees and neural networks. It can automatically ignore them while preparing the model and calculating the probability for a class value.

Support vector machines (SVM) is a kernel method and can be used for classification or regression. Each data instance is plotted as a point in an n-dimensional space with the value of each feature being the value of a particular coordinate. Then, the next step is to determine the decision boundary, which can best classify the different labeled classes. Equation (3) shows the optimization function of SVM.

$$\min_\theta C \left[\begin{array}{c} \sum_{i=1}^{m} y_i \left(-\log h_\theta \left(x_i\right)\right) + \\ \left(1-y\right)\left(\left(-\log\left(1-h_\theta\left(x_i\right)\right)\right)\right) \end{array} \right] + \frac{1}{2}\sum\nolimits_{j=1}^{n} \vartheta_j^2 \tag{3}$$

where $C = \lambda/m$, where λ is a regularization term such as ridge and m is the number of labeled data with x_i being the input and $y_i$the corresponding label, presented as a matrix of $\{(x_i, y_i))\}$ where $\boldsymbol{i} \in (1 \ldots m)$; $\mathrm{j} \in (1 \ldots n)$ is the weight for a specific feature and n x_i, n is the total number of features. Equation (4) calculates the linear hypothesis $h_\theta(x)$ as follows:

$$h_\vartheta(x) = \theta_0 + \theta_1 x_1 \cdots + \theta_n x_n \tag{4}$$

Reliance on boundary cases enables it to handle missing data. Heuristics and resampling may reduce the effect of imbalanced data in SVM modeling. Changing the value of C so that weight is inversely proportional to class is one simple solution. SVM has been used for filtering breaking news-related tweets [19] and for name resolution in a disambiguation framework [25].

Logistic regression is used to predict types of repairs needed to resolve data inconsistencies based in a set of statistics computed over the data and declarative rules [17]. It is a probabilistic, linear classifier that used to transform a linear combination of input features into a nonlinear output via the sigmoid function as shown in Eq. (5). It can also be used for missing values imputation.

$$\sigma\ (\mathrm{t}) = \frac{1}{1 + e^{-h_\theta(x)}} \tag{5}$$

LogitBoost and Logistic Model Tree algorithms are more appropriate algorithms for imbalanced datasets [6]. Logistics Model Tree combines logistic regression and decision tree learning, while LogitBoost combines AdaBoost and logistic regression.

The traditional algorithms cannot work for high number of features or examples. But, many attempts have been done to make it scalable for large data sets. Online stochastic gradient descent algorithm can respond to the data volume.

Decision tree is a supervised learning method used for classification and regression. It decomposes the data space into homogeneous groups based on independent variables' conditions, called as split points. The division is done recursively until the leaf nodes contain certain minimum numbers of records which are used for the purpose of classification. One of the benefits of decision trees is that input data does not require any scaling. However, they cannot work with missing values. Decision tree classifiers can handle imbalanced data better than other algorithms, but this always depends on the distribution of features in the space.

Decision trees are usually utilized with boosting, such as in logistic model tree, in order to reduce the bias and variance [6]. Random Forest and Rotation Forest use multiple decision trees. The latter is a state-of-the-art method where decision trees are independently trained via principal component analysis (PCA).

4.2 Unsupervised Machine Learning

There has been extensive work on unsupervised models and specifically those of anomaly or outlier detection, regression, and clustering for noise treatment and missing values imputation.

4.2.1 Imputation of Missing Values

A basic strategy to use incomplete datasets is to remove entire data instances, i.e., rows – known as listwise imputation and/or columns containing missing values. However, this many times comes at the price of discarding valuable information. Listwise deletion works well only when the data are missed at random. Regarding column deletion, if the variable with the missing value depends on at least one of the other variables of the model [35], it is not a good practice.

Inferring the missing values from the known part of the data is a better strategy, known as imputation. The inferred values may be constant or statistical measures. Conventional methods include marginal mean imputation and conditional mean imputation.

Cold-deck imputation selects the replaced value from central tendency measures such as the mean or mode of the feature. Marginal mean imputation is computing the overall mean to impute the missing value. For a categorical feature, a mode is used instead. In conditional mean imputation, the values for the missing rows are imputed conditioned on the values of the complete attributes that are most similar to it. To illustrate this with an example, a conditional imputation rule could state that the mean value for the "chest pain type" feature is imputed only if the value of "diagnosis" feature is absent. Although these are both simple and fast methods, they come with many disadvantages with the most obvious being the reduction of variance in the dataset. This in general may lead to biased estimators for the

predictive model. A more advanced method called hot-deck imputation has been proposed as an alternative [35].

Hot-deck imputation, by contrast, selects the replaced value from a similar data instance. Similarity is computed via proximity-based measures. This technique may yield biased results irrespective of the missing data mechanism, it may become less likely to find matches if the number of variables is large, and a random value is chosen when many instances are found to be similar.

Statistical imputation, also known as a sequential regressor, is a multivariate imputation technique that imputes the values based on all the feature dimensions of the dataset. The feature with the missing values is treated as the dependent variable of the regression function and the other features as independent variables. The estimator is fitted on the data and then predicts the missing value. This is done iteratively for each feature and repeated for a given number of rounds. The results of the final imputation round are returned.

These imputation methods usually are restricted to cases with a small number of features. The same is applied in large-scale linear regression with conventional missing values imputation methods like eyeballing and interpolation [30]. An optimized offline version of linear regression is proposed in [30] based on adding penalty factors to the original output value in order to diminish the error further. Researchers also enabled high-dimensional linear regression [36] with noisy, missing, and/or dependent data.

A popular unsupervised approach that uses proximity measures is the nearest neighbor method [37]. The relationship among features is considered when computing the degree of distance and can be implemented for high-dimensional data.

Distribution-based methods assign the value based on the probability distribution of the non-missing data. The expectation–maximization algorithm and heuristic methods are the most common and are used also for resolving noisy values. However, they can only be used for linear and log-linear models [35], and it is difficult to guarantee that the local optimum is close to a global optimum [36].

For categorical data, the following techniques are utilized for advanced imputation of missing values:

1. Multinomial inverse regression
2. Regression topic models
3. Penalized regression like lasso
4. Contextual anomaly detection
5. Latent factor models

Structural features of the network together with graph analysis algorithms can also be used as imputation methods [30]. Predicting missing attributes' values of users' can be done with a community detection algorithm proposed by [38].

There are imputation techniques for two-dimensional regression models, but only recently such techniques extended to high-dimensional data [30, 36].

4.2.2 Anomaly Detection

Anomaly detection deals with the identification of abnormal or unusual observations in the data by defining a decision boundary around normal observations. However, defining the region which separates outliers from normal data points is not a straightforward task, and it changes over time. Feature engineering choices have also to be carefully considered in order to not bias anomaly detection.

Outlier detection and novelty detection are both used for anomaly detection. Outlier detection is then also known as unsupervised anomaly detection and novelty detection as semi-supervised anomaly detection. Outlier detection includes the whole dataset, and it defines a decision boundary in the training data. Novelty detection defines a dataset of "good" data and in real time keeps or discards observations that fit the dataset. It is similar to "one-class classification," in which a model is constructed to describe "normal" observations. The difficulty lies on that the quantity of available "abnormal" data is insufficient to construct explicit models. SVMs have been used for novelty detection.

Distributed strategies for outlier detection in large data sets are Distributed Solving Set and Lazy Distributed Solving Set algorithms.

Duplicates are considered a form of noise. Matching rules and ER are common methods to address this issue, discussed in Sect. 5.3.

However, outliers in OSN may reflect real behaviors which necessitate an interpretable model in which explanations of "normal" data are to be provided.

4.3 Extracting and Integrating Data

Sanity checking is usually done after the extraction process in order to ensure that the desired, filtered data structure is gathered. A common cleansing task after extraction is to convert the data to the one format needed for the analysis algorithm. APIs exchange data mostly in the two formats of XML and JSON. These are semi-structured formats whose cleansing is largely unexplored [18]. Additionally, API's responses format and structure differ from one OSN to another demanding data conversion.

Social networking data are semi-structured and not consistent in "schema" level. This inconsistency grows when combining different social networks in order to perform cross-domain analytics, which are considered highly valuable. A solution [11] is the development of a unified conceptual data model that captures differences and similarities in structures and terminologies. Considering that there might be hidden relationship among features from different source, correlation-based algorithms can be used to find the most relevant relationship among features from different sources [30].

Dealing with the granularity and the hierarchies of data residing in different sources is a general issue for big data. Comprehending the scope, hierarchy, and

granularity [27] of the social networking data via domain-encompassing models is necessary in data integration.

4.3.1 Data Warehouses

Data warehouses are the mainstream database for integrating data. They require concise, entity-oriented schemas that facilitate the analysis of data. Data cleansing in these databases is a major part of the so-called Extract, Transform, and Load (ETL) process. With cloud and similar infrastructures, an alternative approach called ELT has emerged. Its advantage over ETL is that data can be loaded in their raw form and thus be used at any point in time. ETL can be viewed as "bring the code to the data," whereas ELT as "bring the data to the code" [27]. In any case, researchers can keep track of data provenance for different analysis tasks. The issue that arises from data warehouses is the timeliness dimension of data that having the data residing for a long time into a database is that they may become obsolete for a specific analysis. The data quality problems most often encountered in OSN and the preprocessing tasks related to them are briefly summarized in Table 1.

5 Feature Engineering and Text Preparation

Having a good representation of features is linked with the success of the machine learning models [50]. Feature engineering is composed of transformation and dimensionality reduction techniques.

5.1 Transformation of Features or Samples

Two common transformation techniques that deal with the range of values of features include discretization and encoding. The former converts the continuous features into their discrete counterparts, and it is necessary in algorithms that handle discrete only data, such as in probabilistic density estimators. Data encoding converts categorical values of features to their numerical counterparts; it is necessary to any algorithm since no algorithm can handle categorical values.

Aggregation is a simple exploration-transformation method that groups or summarizes data (e.g., computing the minimum of samples within a feature or creating a pivot table by combining three features). It is an informative step for checking the quantity of outliers or sparseness of features, or correlations in the data.

Feature scaling scales all samples within an individual feature to satisfy a particular property in the entire range of their values. Standardization ranges them to have unit variance and mean of zero, like the samples are drawn from a standard Gaussian distribution. In practice, however, when there is no interest about the shape

of a distribution, it centers the values by removing the mean of each feature and then scales them by dividing it with standard deviation. Sometimes, this is referred to as normalization. In the documentation of the scikit python library though normalization is the process that scales individual samples rather than features to have a unit form; this is useful when applying a dot product or computing the similarity between samples or creating a vector space model in text preparation.

There are other choices for feature scaling. They depend on the available characteristics of the data set and the analysis algorithm. For instance, if there are many outliers in the data, then standardization biases the information available in the data, while if there is a lot of sparsity, then centering the values can also harm the model. Nonlinear scaling is applied when the rank of the values along each feature has to be preserved.

Feature selection selects the most relevant features from the original dataset. It can be divided into lexicon-based that make use of knowledge bases, discussed in Sect. 3.1, and statistical methods. The latter include lasso and ridge and the selection of the value of the lambda hyperparameter in SVM depicted in Eq. (3), which controls the magnitude of the selected features. Decision tree estimator is another technique that computes the feature importance to the model's prediction. Recursive and backward elimination and rank algorithms like information gain and correlation coefficient are also employed. A feature selection survey is provided in [10].

5.2 Dimensionality Reduction

In the dimensionality reduction or otherwise called feature extraction, the original data gathered are replaced with a lower-dimensional approximate representation obtained via a matrix or multidirectional array factorization or decomposition. They can be divided into two groups, linear and nonlinear methods [10].

Linear feature extraction assumes that the data lies on a lower-dimensional linear subspace and projects them on this subspace using matrix factorization. The linear methods transform original variables into a new variable using the linear combination of the original variables. Some examples include the following:

- Principal component analysis (PCA)
- Linear discriminant analysis
- Probabilistic PCA
- Factor analysis
- Linear regression
- Singular value decomposition (SVD)

Nonlinear dimensionality reduction works in the following different ways:

(i) Kernel-based methods where a low-dimensional space can be mapped into a high-dimensional space so that a nonlinear relationship among the features can be found

Table 5 N-gram model

Unit	Sample sequence	Unigram BoW	Bigram BoW
Word	. . . As knowledge increases wonder . . .	. . . As, knowledge, increases, wonder, . . .	. . . As knowledge, increases wonder, . . .
Character	. . . to_be_or_not_to_be . . .	. . . , t, o, _, b, e, _, o, r, _, n, o, t, _, t, o, _, b, e, . . .	. . . , to, o_, _b, be, e_, _o, or, r_, _n, no, ot, t_, _t, to, o_, _b, be, . . .

(ii) Kernel PCA for nonlinear classification
(iii) Self-organizing maps or Kohonen, which creates a lower-dimensional mapping of an input by preserving its topological characteristics

Feature construction/extraction plays an important role often in reducing the misclassification error [2]. It creates new variables and increases the variety of data by constructing new features either on transforming the original data for dimensionality reduction or on addition of domain knowledge for algorithm performance improvement. The former is also known as data augmentation. A feature engineering task that combines weathers' data with citizens' transportations patterns could increase the performance of the algorithm if there is such a relation in the data. Adding noise such as changing the curvature of letters in a speech recognition task or blurring the image in an image recognition task includes two illustrations of data augmentation.

5.3 Natural Language Processing

The preprocessing of textual data extracted from a document starts with a tokenization step, namely, breaking a stream of text into words, phrases, symbols, or other meaningful lexical items that will create a dictionary of tokens.

5.3.1 Text Extraction

The most common representation of these tokens is the Bag of Words (BoW) model, also referred as N-gram model. An N-gram model via language units is shown in Table 5.

The BoW model correctly separates phrases, words that have meaning only when they occur together, in clusters that refer to the same categories, but it introduces spurious correlations [20]. That is, phrases are added to a cluster just because they share a word in common. To get rid of this spurious correlation, the words that appear together all the time (i.e., San Francisco) were merged, whereas words that appear both solely and in phrases (i.e., Barack Obama) were distinguished, and only the word that appeared more often was chosen (i.e., Obama).

Despite the wide use of BoW, there is a need for richer and distributed representations. Bag of Concept has been proposed as a replacement which makes use of deep learning. This is capable of capturing high-level structures of natural language and incorporating semantic relationships among them.

5.3.2 Feature Selection and Extraction

The vector space model is the data structure for a piece of text composed by a feature vector of terms and term weight pairs.

Word embedding is a feature extraction technique in NLP that uses distributed word representations. Words or phrases are mapped to vectors of real numbers resulting in a low-dimensional feature space. Methods that generate this mapping include neural networks, probabilistic models, and dimensionality reduction on the word co-occurrence matrix. The word2vec is a tool that takes a text corpus as input and produces the word vectors as output via neural networks. It first constructs a vocabulary from the training text data and then learns vector representation of words.

Normalized data are needed for the vector space model used in text classification and clustering contexts. Discretization is also common among the text processing community (probably to simplify the probabilistic reasoning); despite that, normalized counts (i.e., term frequencies) or TF-IDF valued features often perform slightly better in practice.

Feature selection algorithms applied in text preprocessing include TF-IDF, Entity Resolution, Mutual Information, Information Gain, and Chi square. These statistical analysis methods used to measure the dependency between the word and the target output like the category of a tweet as they are mentioned in Sect. 3.2. Semantic relations between words can be found via stemming and lemmatization. Both of them aim to reduce inflectional forms, but the former removes derivational affixes by applying a set of rules, whereas the latter applies morphological analysis of words with the use of a vocabulary. Stemming is just structural, whereas lemmatization is contextual and can consider synonyms and antonyms.

TF-IDF measures the significance of words in text. TF is the occurrence of the term appearing in the document: $dt_f = (tf_1, tf_2, tf_3, \ldots tf_n)$, where tf_i is the frequency of the i-th term of the document d and n is the total number of terms. IDF gives higher weight to terms that only occur in a few documents, and it is defined as the fraction: N/df_i, where N is the total number of documents. This means that common tokens will receive a lower score because they are terms that occur in several documents. TF-IDF is used to deal with typographical variations, e.g., "John Smith" versus "Smith, John," but does cannot take misspellings into account [6].

ER is used in detecting data duplicate representations for the same entities and merging them into single representations. It compares pairs of entities and evaluates them via multiple similarity measures. It is widely employed in NLP, but it is associated with high complexity. Using blocking techniques in order to place similar entity descriptions into blocks and thus only compare descriptions within

the same block is recommended as a solution [14, 39]. Matching dependencies is the corresponding qualitative technique where declarative similarity rules are specified. SVM [25] has also been proposed among other classification models for disambiguation. Cluster computing frameworks such as MapReduce and Spark are well suited for these techniques due to their inherent complexity.

In contrast to TF-IDF, ER does not ignore the sentence structure. It is used in repairing misspellings, name resolution [6], toponym resolution, duplicates (D-dupe), slang, acronyms, and grammar issues. In text deduplication, semantic-based ER measures how two values, lexicographically different, are semantically similar, while syntactic-based ER computes the distance between two values that have a limited number of different characters. Knowledge bases [15] can be useful in similarity and matching functions. Addressing slang words has also been approached via BoW and conditional random fields [40]. Correcting typographical errors is done by [5] via random walks on a contextual similarity bipartite graph constructed from n-gram sequences.

Named entity recognition (NER), part of speech (POS), and stemming are employed to resolve the ambiguities of time and location expressions [19–21]. NER detects an entity in a sentence (London) and assigns a semantic class on it (city). POS tagging is used to assign part of speech tags to a sequence of words and is widely used for language disambiguation tasks. Stochastic models of POS are used by researchers in [41, 42] for social media texts. The former research used conditional random field and the latter the first-order maximum entropy Markov model.

However, it is not easy to accurately identify neither named entities or part of speech words in social networked data which contain a lot of misspellings and abbreviations. POS and NER are trained on full-text documents instead of text extracted from OSN. Hence, researchers often resort in poorer representations like that of TF-IDF [20].

6 Preprocessing Frameworks

Platforms for distributed computing like cloud and grid together with distributed frameworks like Apache Spark and Hadoop MapReduce are used to deal with the high volume of data. They compute scalable, data-intensive machine learning algorithms for preprocessing data. Time-consuming algorithms though should be redesigned in order to allow for scalability and parallelization.

6.1 Distributed Frameworks

Cluster computing frameworks can be divided into online and offline processing. Apache Hadoop is a very popular implementation of MapReduce distributed

processing model which is based on offline processing. Although it supports the analysis of unstructured data, it cannot deal with streaming data and is not suitable for iterative processes. A map-reduce procedure can only conduct one pass over the data points while trying to optimize the execution of a single iteration which is done manually by tuning each map-reduce step.

Spark can deal with iterative processes effectively and efficiently compared to Hadoop's two-stage MapReduce paradigm. Its parallel execution model enables all data to be loaded into memory and avoid the I/O bottleneck which benefits the iterative computation. The execution plan is optimized via flexible directed acyclic graph-based data flows which can significantly speed up the computation of the iterative algorithms. Spark is the most common clustering framework used for in real-time big data analytics. Other frameworks that process incoming continuous data streams include Indoop, Muppet, and SCALLA.

Mahoot is the machine learning library provided by Hadoop. NLP tasks like deduplication, typographical error correction, normalization of dates, places and acronyms, and other word error correction that can be implemented on social media posts can be executed. Hadoop MapReduce has gained popularity implementing ELT processing. MLlib is the corresponding machine learning library of Spark. It supports TF-IDF, syntactic ER, word embeddings, and stop word removal among other functionalities.

Cloud computing makes available hundreds or thousands of machines to provide services such as computing and storage. Text processing, data preparation, and predictive modeling are all supported by cloud-based API services. Google Cloud Platform hosts machine learning algorithms for natural language processing like POS tagging and NER. An example of data parsing with Google NLP API is depicted in Fig. 2. Google Cloud Dataprep can be used to prepare data through basic statistical measures given that these are transformed in a relational form like BigQuery tables. Although cloud computing technology has been applied in the field of machine learning, there is still no real application to big data cleansing algorithms [43].

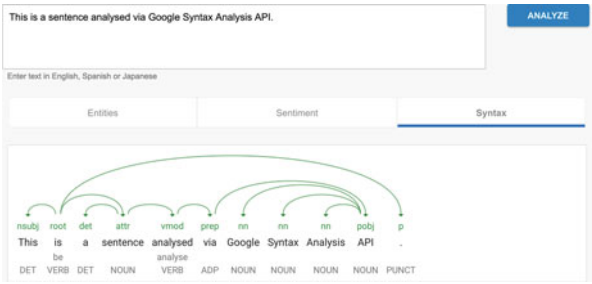


Fig. 2 Dependency parsed tree generated via Google NLP API

6.2 Preprocessing Frameworks for OSN

Table 6 categorizes big data preprocessing frameworks discussed in this chapter, according to the approach they follow, the data errors they are capable to handle, and the kind of service they can provide at social networks.

Analysis of opinionated text like sentiment analysis and opinion mining could incorporate preprocessing frameworks by [5, 44] to normalize text. In [44], researchers built a corpus with cleaned data that can harm sentiment analysis algorithms. Lemmatization, TF-IDF, and the removal of stop words and other noisy information to handle abbreviations and typographical errors are then employed. This is oriented to Twitter, while in [5], a text normalization system adaptable to any social media and language is proposed. It corrects typographical errors via unsupervised approach that learns the normalization candidates from unlabeled text data and maps the noisy form of the word to a normalized form.

Regarding deduplication, many frameworks have been proposed. Dedoop [39] deals with parallel deduplication via implementing ER on MapReduce, and cloud is used to handle the volume of big data. The ER workflow consists of three consecutive steps: blocking, similarity computation, and the actual match decision. In [14], a distributed big data deduplication system which utilizes MapReduce is also developed. The difference with Dedoop is that [14] provided a communication cost model. D-Dupe [45] is another ER tool oriented for social networks. It integrates data mining algorithms with an interactive information visualization interface that can be also used from non-experts.

Dependency rules have been utilized in big data preprocessing frameworks. BigDansing [13] deals with the volume by transforming functional dependencies, denial constraints, or user-defined function in order to enable distributed computations for a scalable data cleansing. It can run on top of most common general-purpose data processing platforms, ranging from DBMSs to MapReduce-like framework. NADEEF [22, 46] is a cleaning system that hosts ETL rules, CFDs, FDs, deduplication rules, and user-defined rules. Another data cleaning framework is LLUNATIC [47] which develops a parallel-chase procedure that detects violations of these rules and guarantees both generality and scalability. A generalized framework for repair computation given semantic qualitative rules is Katara [16]. It bridges crowdsourcing and knowledge bases to find semantic table patterns at the instance level via SPARQL queries. These are visualized as a labeled graph where a node represents an attribute and an edge the relationship between two attributes. An adaptive cleaning framework [17] capable of dealing with velocity is proposed for dynamic environments without fixed constraints. They presented a classifier that predicts the type of repair needed to resolve an inconsistency and automatically learns user repair preferences over time. A framework for missing consistent attributes that combines Bayesian inferential statistics and accuracy rules based on data semantics is proposed by [23]. Aetas system [26] combines FDs with probabilistic and outlier detection models on data modeled as dependency cubes.

Table 6 Big data preprocessing frameworks

	Approach				Distributed	Data errors			Service
	Dependency rules	NLP	Machine learning	Graph theory		MV's	Noise	Duplicate	
[6]		TF-IDF, ER	Boosting, Random Forest, Logistic model tree, Rotation Forest						Name Resolution
[38]				Community Detection, Betweeness centrality					Missing attributes
[23]	FD,CFD		Bayesian model						Deduplicate attributes values
[33]			Decision tree, Bagging, Random forest						Missing links
[30]			Linear regression		Spark				Missing numerical values
[39]		TF-IDF, ER			MapReduce				Entity Deduplication
[14]			ER		MapReduce				Distributed deduplication
[5]		ER		Graph random walks	Algorithm				Text normalization
[19]		NER, POS	SVM, Bayessian inference						Filter noise
[44]		TF-IDF			MapReduce				Text normalization
[17]	Dynamic FD		Logistic regression						Adaptive rules
[46]	CFD,FD, ER								Data cleansing Platform
[41]		POS		Conditional random field					POS tagger for Twitter
[42]		POS		Markov model	Algorithm				POS tagger for online conversational text

Big data quality frameworks proposed in [12, 48, 49] can be used to address the dimensions of data quality within big data. In particular, researchers [12] proposed a big data preprocessing system that aims to manage data quality through rules which can be user defined, auto-discovery, or domain related. Intrinsic and contextual quality dimensions are also classified in [12]. A big data processing architecture is presented by [49] that manages the quality of social media data by utilizing domain policy data rules and metadata creation to support provenance. Finally, in [48], a big data quality assessment framework whose hierarchical structure of data quality rules is composed of data quality dimensions and indexes is provided.

7 Conclusion

There are many data analyses that acknowledged the difficulties faced in dealing with either imperfect data or algorithmic limitations due to large-scale data. Data cleansing and feature engineering can play a vital role into eliminating these issues and improving the generalization of predictive models. Considering these aspects, we define noise in the source of social networks and explore new practices toward data quality and preprocessing tasks oriented toward big data analytics. Since text is a rich source for data in OSNs, we present natural language processing techniques.

The main technical challenges apart from the long-standing issues of incomplete and noisy data come from the unique discourse structure and grammar of the content, the computational complexity of data-intensive preprocessing algorithms, the poor representation of complex data, and the high-dimensional space. Imbalanced datasets, data integration, and analysis of network structures are also challenging issues which are only meagerly discussed in this chapter.

The following list outlines future interesting topics for research and new avenues proposed for dealing with these challenges:

1. Distributed and richer data representation
2. Distributed data preprocessing
3. Development of NLP tools trained in the data scope of social networks
4. Modified algorithms for imperfect data
5. Real-time preprocessing and noise detection
6. Active learning in crowdsourcing for label provision or validation of algorithmic output

The broader objective of this work is to set a beginning of developing a framework toward social networking data preprocessing in order to reach higher analysis insights and to alleviate the issue of developing automated tools for streamlining big data analytics. This chapter can be found helpful to any data scientist that conducts any type of data analysis, to newcomers that would like to obtain a panoramic view of the preprocessing task in big data sources, to researchers that will use social networks as a data source, and to researchers that are familiar with certain issues and algorithms and would like to enhance them.

References

1. Saha, B., & Srivastava, D. (2014). *Data quality: The other face of Big Data*. In *2014 IEEE 30th international conference on data engineering*, pp. 1294–1297.
2. Amin, A., et al. (2016). Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study. *IEEE Access, 4*, 7940–7957. https://doi.org/10.1109/ACCESS.2016.2619719.
3. Jagadish, H. V., Gehrke, J., Labrinidis, A., Papakonstantinou, Y., Patel, J. M., Ramakrishnan, R., & Shahabi, C. (Jul. 2014). Big data and its technical challenges. *Communications of the ACM, 57*(7), 86–94.
4. Sapountzi, A., & Psannis, K. E. (2016). Social networking data analysis tools & challenges. *Future Generation Computer Systems*. https://doi.org/10.1016/j.future.2016.10.019.
5. Hassan, H., & Menezes, A. (2013). *Social text normalization using contextual graph random walks* (pp. 1577–1586). Sofia: Association for Computational Linguistics.
6. Peled, O., Fire, M., Rokach, L., & Elovici, Y. (2016). Matching entities across online social networks. *Neurocomputing, 210*, 91–106.
7. Huisman, M. (2014). Imputation of missing network data: Some simple procedures. In *Encyclopedia of social network analysis and mining* (pp. 707–715). New York: Springer New York.
8. Kossinets, G. (2006). Effects of missing data in social networks. *Social Networks, 28*(3), 247–268.
9. Kim, M., & Leskovec, J. (2011). The network completion problem: Inferring missing nodes and edges in networks. In *Proceedings of the 2011 SIAM International Conference on Data Mining* (pp. 47–58). Philadelphia: Society for Industrial and Applied Mathematics.
10. Hira, Z. M., & Gillies, D. F. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics, 2015*, 198363.
11. Tan, W., Blake, M. B., Saleh, I., & Dustdar, S. (2013, September). Social-network-sourced big data analytics. *IEEE Internet Computing, 17*(5), 62–69.
12. Taleb, I., Dssouli, R., & Serhani, M. A. (2015). Big data pre-processing: A quality framework. *2015 IEEE International Congress on Big Data*, pp. 191–198.
13. Khayyat, Z., Ilyas, I. F., Jindal, A., Madden, S., Ouzzani, M., Papotti, P., Quiané-Ruiz, J.-A., Tang, N., & Yin, S. (2015). BigDansing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data – SIGMOD'15*, pp. 1215–1230.
14. Chu, X., Ilyas, I. F., & Koutris, P. (2016). Distributed data deduplication. *Proceedings of the VLDB Endowment, 9*(11), 864–875.
15. Fan, W., & Wenfei. (December 2015). Data quality: From theory to practice. *ACM SIGMOD Record, 44*(3), 7–18.
16. Chu, X., Morcos, J., Ilyas, I. F., Ouzzani, M., Papotti, P., Tang, N., & Ye, Y. (2015). KATARA. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data – SIGMOD'15*, pp. 1247–1261.
17. Volkovs, M., Chiang, F., Szlichta, J., & Miller, R. J. (2014, March). Continuous data cleaning. In *2014 IEEE 30th International Conference on Data Engineering* (pp. 244–255). IEEE.
18. Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data cleaning: Overview and emerging challenges. In *SIGMOD'16 Proceedings of the 2016 International Conference on Management of Data*, pp. 2201–2206.
19. Zhou, D., Chen, L., & He, Y. (2015). An unsupervised framework of exploring events on twitter: Filtering, extraction and categorization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
20. Sankaranarayanan, J., Samet, H., Teitler, B. E., Lieberman, M. D., & Sperling, J. (2009). TwitterStand. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems – GIS'09*, p. 42.
21. Ritter, A., Mausam, Etzioni, O., & Clark, S. (2012). Open domain event extraction from Twitter. In *Proceedings of the 18th ACM SIGKDD – KDD'12*, p. 1104.
22. Tang, N. (2014). *Big data cleaning* (pp. 13–24). Cham: Springer.

23. Cao, Y., Fan, W., & Yu, W. (2013). Determining the relative accuracy of attributes. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 565–576.
24. Fan, W., Wu, Y., & Xu, J. (2016). Functional dependencies for graphs. In *Proceedings of the 2016 International Conference on Management of Data – SIGMOD'16*, pp. 1843–1857.
25. Wang, P., Zhao, J., Huang, K., & Xu, B. (2014). *A unified semi-supervised framework for author disambiguation in academic social network* (pp. 1–16). Cham: Springer.
26. Abedjan, Z., Akcora, C. G., Ouzzani, M., Papotti, P., & Stonebraker, M. (Dec. 2015). Temporal rules discovery for web data cleaning. *Proceedings of the VLDB Endowment, 9*(4), 336–347.
27. Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: Issues and challenges moving forward. In *2013 46th Hawaii International Conference on System Sciences*, pp. 995–1004.
28. Fan, J., Han, F., & Liu, H. (Jun. 2014). Challenges of big data analysis. *National Science Review, 1*(2), 293–314.
29. Gandomi, A., & Haider, M. (April 2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*(2), 137–144.
30. Shi, W., Zhu, Y., Huang, T., Sheng, G., Lian, Y., Wang, G., & Chen, Y. (2016, March). An integrated data preprocessing framework based on apache spark for fault diagnosis of power grid equipment. *Journal of Signal Processing Systems, 86*, 1–16.
31. Poulos, J., & Valle, R. (2018). Missing data imputation for supervised learning. *Applied Artificial Intelligence, 32*(2), 186–196. https://doi.org/10.1080/08839514.2018.1448143.
32. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A review on ensembles for class imbalance problem: Bagging, boosting and hybrid-based approaches. *IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, 42*(4), 463–484.
33. Fire, M., Tenenboim-Chekina, L., Puzis, R., Lesser, O., Rokach, L., & Elovici, Y. (December 2013). Computationally efficient link prediction in a variety of social networks. *ACM Transactions on Intelligent Systems and Technology, 5*(1), 1–25.
34. Rennie, J., Shih, L., Teevan, J., & Karger, D. (2003) Tackling the poor assumptions of naive Bayes text classifiers. In *Proceedings of the ICLM-2003*.
35. Soley-Bori, M. (2013). *Dealing with missing data: Key assumptions and methods for applied analysis* (Vol. 4, pp. 1–19). Boston University.
36. Loh, P., & Wainwright, M. J. (2012). High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. *Annals of Statistics, 40*(3), 1637–1664.
37. Stekhoven, D. J., & Buhlmann, P. (2012). Missforest – Non-parametric missing value imputation for mixed-type data. *Bioinformatics, 28*(1), 112–118.
38. Mislove, A., Viswanath, B., Gummadi, K. P., & Druschel, P. (2010). You are who you know. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining – WSDM'10*, p. 251.
39. Kolb, L., Thor, A., & Rahm, E. (2012). Dedoop: Efficient deduplication with Hadoop. In *Proceedings of the VLDB endowment* (Vol. 5, p. 1878).
40. Singh, T., & Kumari, M. (2016). Role of text pre-processing in Twitter sentiment analysis. *Procedia Computer Science, 89*, 549–554.
41. Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., & Smith, N. A. (2010). *Part-of-speech tagging for twitter: Annotation, features, and experiments*. Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science.
42. Owoputi, O., Owoputi, O., Dyer, C., Gimpel, K., Schneider, N., & Smith, N. A. (2013). *Improved part-of-speech tagging for online conversational text with word clusters.* In *Proceedings of NAACL*.
43. Al-Hamami, M. A. H. (2015). The impact of big data on security. In *Handbook of research on threat detection and countermeasures in network security* (Vol. 3, pp. 276–298). Pennsylvania: IGI Global.

44. Nirmal, V. J., Amalarethinam, D. I. G., & Author, C. (2015). Parallel implementation of big data pre-processing algorithms for sentiment analysis of social networking data. *International Journal of Fuzzy Mathematical Archive, 6*(2), 149–159.
45. Bilgic, M., Licamele, L., Getoor, L., & Shneiderman, B. (2006). D-dupe: An interactive tool for entity resolution in social networks. In *2006 IEEE Symposium on Visual Analytics and Technology*, pp. 43–50.
46. Ebaid, A., Elmagarmid, A., Ilyas, I. F., Ouzzani, M., Quiane-Ruiz, J. A., Tang, N., & Yin, S. (2013). NADEEF: A generalized data cleaning system. *Proceedings of the VLDB Endowment, 6*(12), 1218–1221.
47. Geerts, F., Mecca, G., Papotti, p. & Santoro, D., 2014. That's all folks! LLUNATIC goes open source. *Proceedings of the VLDB Endowment, 7*(13), pp. 1565–1568.
48. Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the big data era. *Data Science Journal, 14*(May), 2. https://dx.doi.org/10.5334/dsj-2015-002.
49. Immonen, A., Paakkonen, P., & Ovaska, E. (2015). Evaluating the quality of social media data in big data architecture. *IEEE Access, 3*, 2028–2043.
50. Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78–87.

Feature Engineering

Sorin Soviany and Cristina Soviany

1 Introduction: Basic Concepts

Feature engineering represents the methodological framework that allows to design and generate informative and discriminant feature sets for the machine learning algorithms. These design and development tasks exploit the information and knowledge belonging to the application-specific domain in order to properly define, extract, and evaluate the more informative sets of variables that should be further processed and sometimes transformed during the more advanced stages of running the machine learning algorithms. The goal of these processing tasks is to provide a reliable information support for the proper decisions in various applications, either for predictive modeling, data clustering, or anomaly detection, depending on the application purposes.

A *feature* is a property or an attribute that is shared by data objects and different entities within the application domain. Such properties are usually evaluated using random variables that follow statistical distributions according to the problem constraints. The feature sets provided in applications are used during the following learning processes:

- In supervised learning, as information support for the predictive modeling, with historical data

S. Soviany
Communication Terminals and Telematics, National Communications Research Institute, Bucharest, Romania
e-mail: sorin.soviany@inscc.ro

C. Soviany (✉)
Features Analytics, Nivelles, Belgium
e-mail: cristina.soviany@features-analytics.com

© Springer Nature Switzerland AG 2020

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_5

- In unsupervised learning, in order to support the cluster analysis, with similarity measures

The features can represent and evaluate physical or logical properties of the entities or objects belonging to the involved domain. For example, in medical applications, the features can be extracted from images of the tissues, using image-processing techniques. Other features can be generated with various signal processing techniques, having different degrees of complexity.

The features are grouped into the following major categories, depending on the way in which these amounts are obtained in real use-cases:

- *Direct features* that represent the original measurements or observations provided in applications, as the raw data in the experiments. These features carry information close to the original data sources, but there is a potential drawback caused by the noisy data. This is why in many cases the original variables are further transformed or filtered in order to provide a more accurate data space for the application. The data cleansing and various data transformations are applied in order to improve the overall data quality and to ensure more useful features for the further modeling process.
- *New designed and extracted features* that are derived from the original variables (direct features) using different computational techniques, statistical amounts, and sometimes specific algorithms in order to provide more informative and discriminant features for the application. Some of these procedures are useful to enrich the overall data space of a certain problem (application domain), looking to discover the potentially hidden patterns within the available datasets, allowing to obtain a more useful representation of the required information.

The following operations are commonly required within this methodological framework *of feature engineering*:

- *The feature generation*: According to the definition given in [1], *feature generation* is the process that takes the input raw and sometimes *unstructured* data and defines features (informative variables) that are required in various *data analytic* processes, machine learning (either supervised or unsupervised), also to perform statistical analysis depending on the real application objectives.
- *The feature extraction*: The most general definition of *feature extraction* is as follows: the process that starts from the initial raw data with the corresponding measurements and observations and generates or derives informative and nonredundant variables describing the properties of the objects or entities. These variables should provide the most relevant information for the learning process, in order to ensure an optimal generalization capability of the classification models (in the cases of supervised learning). This definition is actually quite similar with that for the *feature generation* process. In many real cases the *feature generation* and *feature extraction* concepts have the same basic significance. *Feature extraction* can be seen as a *dimensionality reduction process*. This is the case for applications requiring image processing in order to provide the useful features. In these applications, the feature extraction is based on various image-processing techniques in order to get the required features for the application,

together with advanced techniques for further *dimensionality reduction* (such as principal component analysis [PCA] or linear discriminant analysis [LDA] that performs the *data space transformation* by data projection on new subspaces). According to [1], the generated features should be further transformed in order to ensure an enhanced data subspace with a reliable description of the original data complexity and sometimes with the potential to discover some hidden patterns within the original data. This is why in many design cases and corresponding developments, these transformations (PCA, LDA) are included within the feature extraction modules.

- *The feature selection*: This is the process that improves the relevance and the discriminant power of the data space (original variables together with the transformed features) by removing the redundancies and irrelevant variables (features) in order to preserve the information degree with a lower dimensional space. The process generates *a subset of relevant attributes or features* that should be further applied during the modeling for supervised or unsupervised learning, looking to ensure a suitable trade-off performance vs. complexity. The process is applied in order to reduce the complexity of the models (with a significant impact on the training speed) and also to properly manage the curse of dimensionality problem in order to design a suitable analytic data model.

The main challenge for working with features (direct or designed features, respectively) is given by the requirement to operate with *multidimensional data*. This is because in most cases the applications require to process several variables in order to ensure the required information for the final decision-making. These variables are grouped into *feature vectors* and define a multidimensional feature space in which the data points are placed. This is a vector space in which the data points should meet the typical properties as specified by the linear algebra rules (as concerning distances, inner and outer products, Hilbert space properties). Working with *multivariate distributions* still remains a complex task for many application domains, especially if the corresponding data spaces are high dimensional.

Another challenge is given the way in which *the hidden patterns within the original data can be retrieved within a multidimensional data space*. Some of the features (especially the new designed and transformed features) may exhibit a high informative degree ensuring a suitable relevance for further modeling goals (either for predictive modeling or for the cluster discovery in the unsupervised cases).

These measurable variables and the features that can be derived from them are used to separate and make distinction among several regions in which the data space of a specific problem can be partitioned. In real cases, this partitioning is actually not ideal, meaning that there are some overlapping regions, depending on the overall data quality and the relevance of the extracted or generated features (Fig. 1). For the example that is depicted in Fig. 1, there is a two-dimensional feature space in which two features are used to define the regions.

In the same general context of *feature engineering*, all the involved design and computational processes operate with *feature vectors* as basic data element representation and with the corresponding *feature spaces* the data items belong to [2].

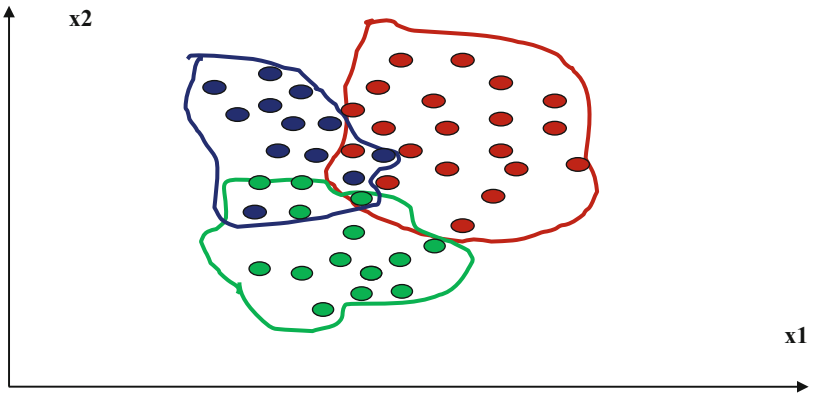


Fig. 1 Example of a two-dimensional feature space partitioning with overlapping

A *feature vector* is a mathematical representation of the generated (or extracted) features within a *multidimensional* data space, with a grouping that is according to the real use-case-specific criteria. The data elements are organized within a column vector *d*-dimensional (therefore having *d* components).

The *feature space* is the *d*-dimensional vector space in which the feature vectors are placed. It is specific for a given problem. The feature space contains all the variables (direct features) and the new extracted or designed features, together with their corresponding possible value domains.

For most applications, the feature space is *multidimensional* because a reliable decision for high-performance target requires exploiting the provided information from several variables and extracted features. However, there are cases in which the more relevant information can be retrieved from a small subspace; therefore, a small number of dimensions can provide the most useful information [2]. This is why in many situations the *feature engineering* methodology makes use of advanced *feature space transformation for dimensionality reduction*, together with automatic *feature selection* procedures using different algorithms that can be applied according to their performance and time constraints, especially for the use-cases with real-time requirements.

While working with a multidimensional data space, the probability density functions that describe the feature value repartition are associated with random variables that follow multivariate distributions laws.

In a *multidimensional feature space*, each *feature vector* represents a data point that describes the set of the attributes or properties for an object belonging to the application domain. The *feature space dimensionality* is given by *the total number of features* that are used within the overall modeling process; this amount includes the following:

- The *original variables* (drawn from the input measurements)

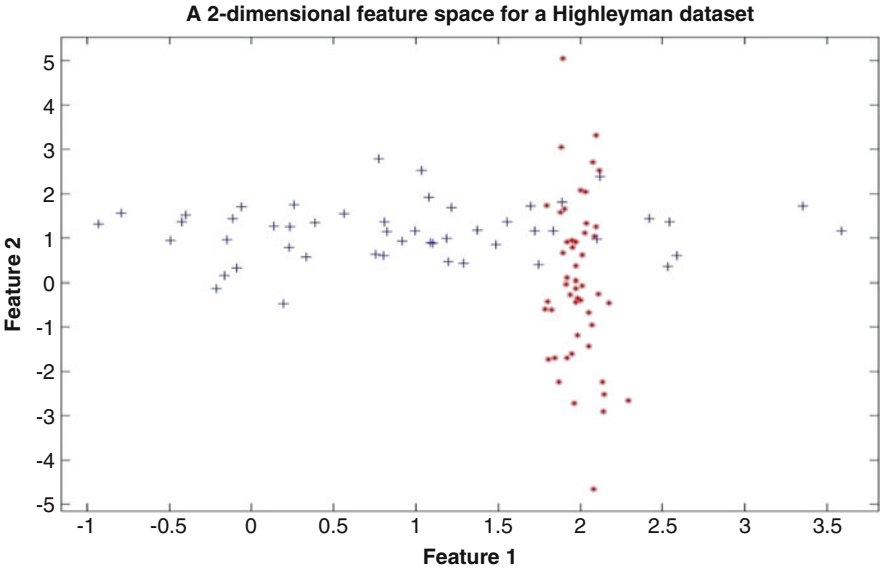


Fig. 2 A two-dimensional feature space example for a Highleyman dataset

- The *new extracted or generated and designed features* that depend on the particular use-case requirements

The data space visualization can be usually done using 2D and 3D Cartesian coordinate systems. Another example of such representation is given in Fig. 2. In such representations, each point corresponds to a given feature vector with two components. The example that is represented in Fig. 2 is drawn for a Highleyman two-class dataset with two generated features.

For the problems in which the total feature space size (*FSS*, or dimensionality *d*) is greater than 3, the data visualization is not possible for all the problem dimensions using the rectangular coordinate systems. An option could be a pairwise representation, therefore providing scatter diagrams for each of the most significant pairs of variables (features) in order to have a visual insight of their behavior.

Another option exploits the concept of *parallel coordinates* [3]. In this approach, each of the original variables or extracted/generated features is represented as a vertical axis, with an appropriate scaling according to the domain of the corresponding variable or feature. Therefore, for a case with $FSS = d$ features, the d-dimensional space can be visualized with a diagram that contains *d* parallel axes. A *feature vector* is represented with a polygonal line and not with a single point as in the rectangular coordinates. An example is provided in Fig. 3.

The feature spaces for real applications are not actually ideal, meaning that usually there is a certain overlapping among the regions in which the variables and/or generated features partition the full data space. The overlapping in the feature space can have one or several from the following causes [2, 4–6]:

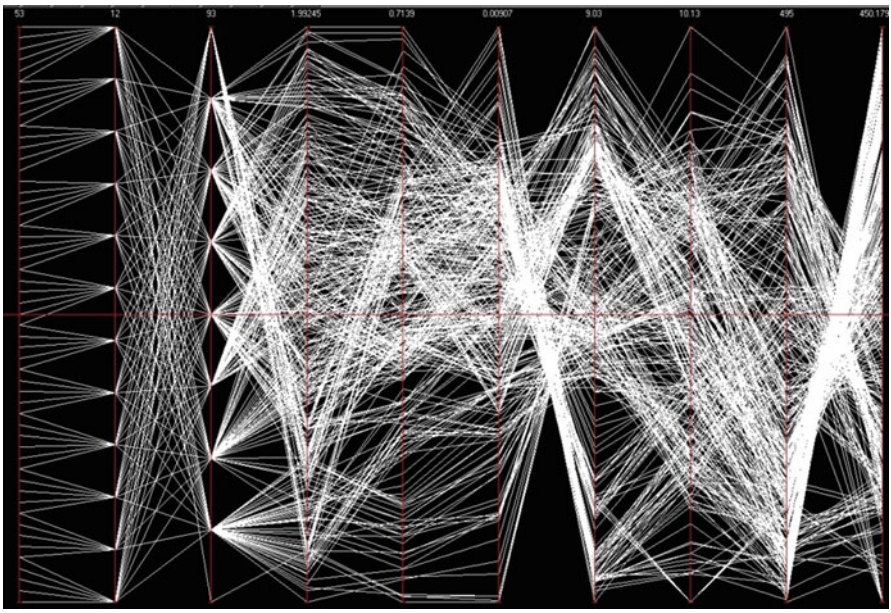


Fig. 3 An example of multidimensional data representation using parallel coordinates

- *A small dimensionality*: In this case, the total number of provided variables (direct features) and extracted features is too small and it cannot cover the overall range of useful attributes. This is why the design of new features can be useful in order to increase the required information degree and sometimes to enrich the data space allowing to discover the potentially hidden patterns in the data. However, there are cases in which the best information could be actually retained within less dimensions. The curse of dimensionality should be properly managed while performing *feature engineering*.
- *Features with poor relevance and low discriminant power*: This is about the variables and extracted features that are still not able to provide a comprehensive description of the full application domain with its entities and data objects. The features with weak relevance do not describe useful properties and attributes belonging to the overall application domain. The discriminant power of the features also depends on the algorithms that are used for feature extraction, selection, and other *feature engineering* operations.
- *The presence of outliers within the input measurements*: These cases are usually due to some improper conditions in which the data acquisition is performed, to the poor state of the sensors, or to some particular problems of the data sources. However, the outliers should only be removed if they do not represent the true application targets. Otherwise, for application cases that involve anomaly detection (including intrusion detection systems for the local network security, medical applications, and cases of fraud management for financial industry

especially for large financial markets), the outliers should be actually detected and recognized. The challenge is to do the accurate detection in real time, especially when a large amount of data must be processed.

Feature engineering, with its operational tasks and analytical and computational procedures, represents an important methodological framework with challenges especially for *unstructured data*.

In the *structured* case, the data collections are usually provided in tables or within relational databases. The data fields provide the input variables for the further modeling process and also the sources for the new feature extraction and design. For structured data the problems are associated with the number of records (or data samples) that should be processed in order to build a data space with a suitable discriminant power, especially if the required amount of data is huge (as in the case of databases containing millions of records). The problem of the amount of data that can be used to generate the most discriminant features is also important in applications with real-time constraints as concerning data analytics and decision-making. There are already available different optimization techniques that can be used to efficiently manage these issues in the structured case.

On the other hand, the *unstructured* case has several challenges to be approached within the overall data management process. There is a significant heterogeneity of data sources and representation formats, with the lack of standardization. The feature extraction is much more difficult because of these issues. The original observations (measurements or direct features) are not provided in a systematic and standardized way. The relevant and informative variables should be retrieved through a careful searching process and often with more sophisticated methodologies in order to get the meaningful information for the feature design, extraction, and selection.

The main steps for any machine learning-based application system design and development (with both supervised and unsupervised cases) include the *feature engineering* processes, as depicted in Fig. 4.

In the next sections, the following *feature engineering* operations and procedures are explained in relation to the overall context of *data summarization and modeling* and within the general framework represented in Fig. 4:

- *Feature design*
- *Dimensionality reduction and data space optimization* through
  - *Feature space transformation* with data projection methods (PCA, LDA)
  - *Automatic feature selection* with specific searching and optimization algorithms

These tasks are approached with various algorithms that can be optimized according to the application requirements as concerning the discriminant performance, execution speed, and particular constraints (including the real-time processing and decision-making).

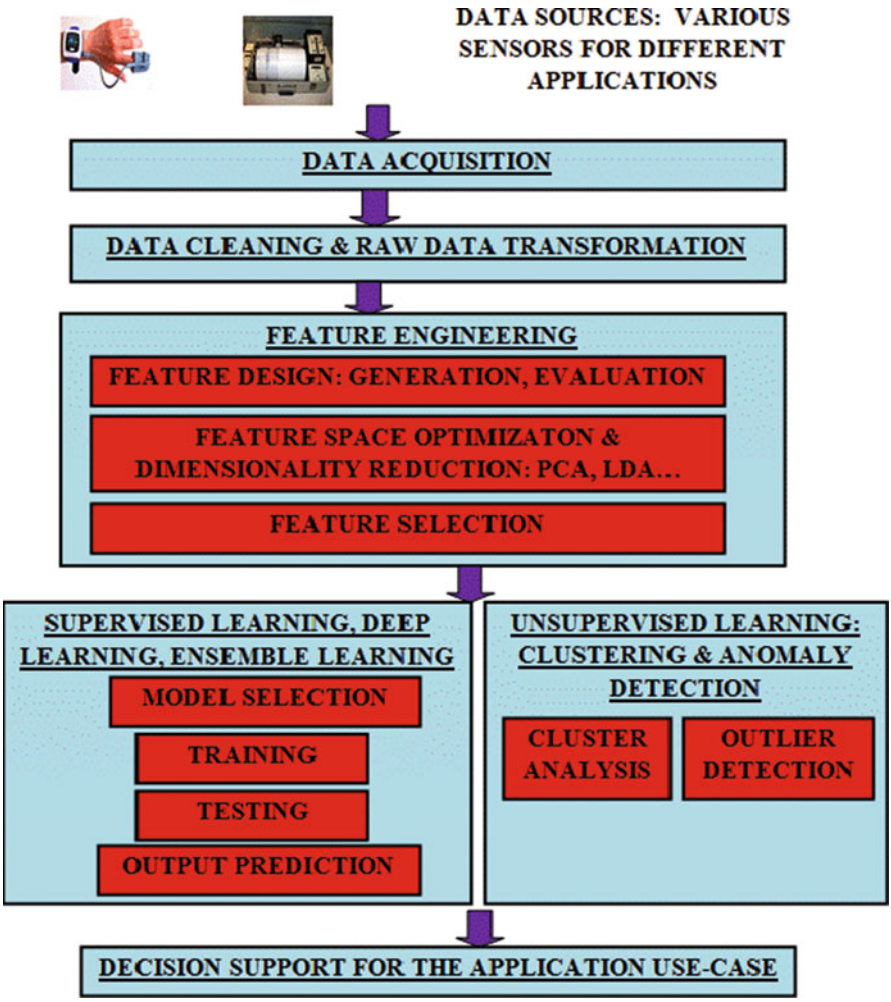


Fig. 4 *Feature engineering* within the overall methodology for data processing and analysis in machine learning-based applications

2 Feature Design

Feature design is the core of the overall *feature engineering* methodology. Its goal is to ensure a reliable set of features for the advanced modeling process, either in supervised, unsupervised, or semi-supervised cases.

The overall process includes the following operations starting from the input raw data samples:

- *Feature generation*, in which the raw data samples are transformed in order to find and retrieve the variables that describe the attributes or properties of the objects and entities belonging to the application domain. This activity requires analytical and computational tasks, depending on the involved data types.
- *Feature evaluation*, in which the original variables (direct features) and the new created features are evaluated in order to provide a first view concerning their potential discriminant power for class separation and also to support with additional information the modeling activities (in supervised and unsupervised learning, depending on the application objectives).

Depending on the application domain's specific requirements and constraints, the feature design process can be approached in an iterative way, especially when dealing with complex datasets, and the application requires many updates of the available datasets in order to keep the freshness of the information. This is the case of the face-based biometric recognition, in which it is recommended to periodically update the enrolled biometric templates of the subjects.

Feature generation is the *feature engineering* process in which the raw data samples are processed and transformed, in case, in order to provide the required features for the advanced modeling process. Feature generation is based on analytical and computational tasks ensuring the suitable descriptors of the most relevant attributes for the entities and objects belonging to the application domain.

The feature generation process can be more easily approached for *structured data*, because in this case the data fields or the table columns provide the required input variables (or direct features) for the more advanced processing steps in modeling.

In the *unstructured data* cases, the overall *data management* for *data summarization and modeling*, particularly the *feature engineering* tasks, should deal with the issues that are given by the heterogeneity of the data sources, data formats, and representation. These problems have significant consequences especially for the feature design operations. In these cases, the requirements are focused on finding the proper way in which the useful attributes can be retrieved and encoded into the required variables. The available approaches are strongly dependent on the provided data types, therefore on their sources in applications. Many applications require computationally intensive tasks with various signal processing techniques in order to provide the useful variables that should be able to properly describe the real properties of the objects and entities belonging to the application domain.

Another problem for the feature design and particularly feature generation task relates to the way in which the *categorical (nominal) variables* can be handled in order to provide the relevant information for the overall modeling process. In many use-cases, especially for the structured data collections, some of the provided data fields contain actually *categorical variables*, while the modeling design requires *numerical variables* and *derived features*. Therefore it is important to apply a reliable method in order to encode the categorical variables into numerical variables [7]. A *categorical variable* can only take one of a limited and fixed set of possible values, based on some qualitative properties of the data collection and application

domain objects. As examples of categorical variables, one can mention the gender of a person, his/her hair color, and the blood type.

Actually the values of *categorical variables* are associated with categories and group memberships, according to the properties of the application domain and its requirements. A *categorical variable* has values (names or labels) belonging to two or several categories, but without any intrinsic ordering relationships among its possible values [8].

Other variables can be ordinal that are quite similar to the categorical ones, but with the main difference that these variables can have certain ordering relationships among the possible values [8].

The *numerical variables* represent measurements or numerical (*quantitative*) observations from the data source in the application operational environment.

Several methodologies are available to deal with *categorical variables* and their conversion or encoding in order to be successfully applied during the learning process, and especially for the *predictive modeling* [9]:

- *Conversion to numerical variables*: In this approach, the given categorical variables are converted to numerical form using one of the following methods:
  - *Label encoding*, in which the nonnumerical labels are transformed into numerical labels with values between *0* and *No_classes-1*. This procedure cannot always ensure a predictive model with high performance.
  - *Conversion of the numerical bins to numbers*, in which a continuous variable is available for the entire dataset. This variable has some bins or intervals of values; these bins are converted into some fixed numerical variables. The conversion can be done in one of the following ways:

    Using the label encoding to perform the conversion, with care to handle the numerical bins as the multiple levels of a categorical (nonnumerical) feature. This approach does not ensure additional information for the overall learning process.

    Defining a new feature based on some statistics about the existing bins (e.g., the mean) or, in some cases, the most relevant value within each of the corresponding bins. In this way, a specific weighting can be provided for the levels of the categorical variable.

    Defining two additional features for the lower and upper limits, respectively. This modality can provide more information about the given bins,

- *Combination of levels*: In this approach different levels are combined in order to efficiently deal with cases of redundant levels for a categorical variable or with levels that have very low occurrence frequencies. The most used methods for the level combination are the following:
  - *Level combination based on business logic*, in order to combine the levels into groups according to the application domain or business experience. It is feasible for cases in which full domain knowledge are available.

  - *Level combination based on frequency or response rate*, approach in which the level combination processes also consider the same quantitative or statistical amounts, such as frequency and the distribution for each level (e.g., to combine the levels that have a frequency below a certain percentage of the total measurements).

- *Dummy coding*: This is a very used method for the categorical variable conversion to continuous (numeric) variables. This approach uses a "dummy" variable that is a duplicate variable representing one level of a categorical variable. This is encoded with 1 for the given level presence, or with 0 otherwise. A dummy variable is created in this way for each of the categorical variable levels. Sometimes, if the total number of levels is too large, one can firstly proceed to apply a certain method of level combination and then to apply dummy encoding. This approach is referred to as one-hot encoding.

Regarding the *feature evaluation*, this can be done using several overlapping measures in order to provide a first view about the real discrimination power of the generated features. Several amounts are available in order to describe and evaluate the data complexity and the true value of the features, even before performing the more advanced modeling steps for learning. The following categories of measures can be used in order to reduce the overall design complexity for the model selection and other advanced operations, ensuring the desired performance for the real application case:

- Measures of the overlapping between the classes within the data space, for individual features
- Measures of the class separation
- Measures of geometry and density
- Statistical measures

3 Advanced Feature Space Transformations for Dimensionality Reduction and Data Space Optimization

3.1 Feature Extraction and Dimensionality Reduction for Feature Space Optimization

Feature extraction with feature space transformation for dimensionality optimization is an advanced step in which the previously generated variables (direct features and/or the new features that are designed based on the original measurements) are further processed using specific transformation algorithms in order to build subspaces of the original feature space, for example, to maximize the overall variance, to optimize the feature space dimensionality, or to properly approach the complexity issues in the further modeling process.

The *dimensionality reduction* is a set of complex of analytical and computational methods that are applied in order to reduce the total number of variables and features to be used in application cases for modeling and learning operations [2, 10].

A key factor for the design of efficient and high-performance solutions in applications that require to process multidimensional data is how to extract and select the most useful variables in a proper way, with lower time and computational expenses. The dimensionality issues (related to the feature space size, *FSS*) have a significant impact on the overall design complexity but particularly on the selected model. In this framework, the *feature space optimization* according to a given *performance* vs. *cost trade-off* is often required for applications from many domains.

The *feature space optimization* means to find out the best trade-off between the *dimensionality reduction* and the degree of the discriminant power preservation or even improvement for the available data. This can be approached with two major methodologies:

- The *feature space transformations*, in which some data projections are applied in order to generate subspaces with enhanced discriminant properties and with reduced complexity. These operations can be done with unsupervised or supervised algorithms.
- The *relevant feature selection*, in which the most informative feature subsets are drawn from the original sets using certain evaluation criterion, while finally retaining the best features in respect to the application target.

The most used feature space transformations (PCA, LDA) are briefly described in this section. The feature selection is approached in the next subsection (4). In some cases these operations are sequentially applied, starting with the data projections (PCA and LDA); there are cases in which only one of these data projections is applied in order to properly enhance the data space. After the data projection, the automatic feature selection can be applied in order to remove the redundancies within the transformed feature space.

The *dimensionality reduction methods* through *feature space transformation* usually perform *the data projection on reduced feature subspaces*, but without taking into account any additional and specific performance criterion in order to retain the most suitable features as required in the real application case. Therefore, the desired and reduced feature set is determined using the components of the original set (direct and designed features), without any explicit feature selection step.

The general model of the *feature extraction for dimensionality reduction*, usually based on data *projection*, is represented in Fig. 5. In this process, all the input variables and designed features are used to build the transformed subspace; the features are generated using a specified data projection while retaining the desired components as combination of the original variables and features. The process is based on linear transformations that can be applied on the original space in order to adjust its dimensionality. These data projections can be used for exploration purposes, in order to get detailed information about the available dataset complexity,

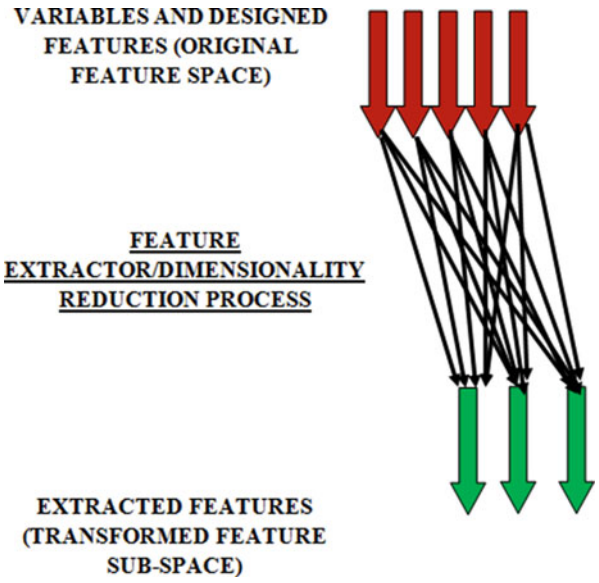


Fig. 5 Feature extraction process for dimensionality reduction

looking to find the existing hidden structures or patterns with potential to enhance the overall discriminant capacity.

Depending if the class membership information is considered within the data projection process, the most common feature space transformations are classified into the following main categories:

- *Unsupervised transforms*, in which the data projection through the corresponding linear transformation is performed without taking into account the class membership information. This is the case of a typical PCA (principal component analysis), in which the goals are to maximize the total variance within the dataset and to remove the correlated variables or features. Another transformation is ICA (independent component analysis), which retains the statistically independent features and variables. The statistical independence of the extracted features is a stronger condition than their correlation, but ICA is not required for any application, given its complexity.
- *Supervised transforms*, in which the class membership for the data to be transformed is considered. This is the case of LDA (linear discriminant analysis) and, for very particular situations, the supervised version of PCA. LDA is applied in order to maximize the class separation.

3.2 PCA (Principal Component Analysis)

3.2.1 Unsupervised PCA Common Version

Principal component analysis (PCA) or *Karhunen-Loève (KL) transform* is an *unsupervised linear* feature space transformation in which the feature extraction is performed without considering the class membership information for the original dataset [2]. In this approach, the dimensionality reduction is done by preserving the features with a specified variance degree that should be maximized. PCA determines a reduced linear subspace of the full feature space allowing to retain the highest fraction of the data variance.

PCA is based on a linear transformation of the original space. The transformation matrix is computed based on *statistical information describing the input data*, typically in an unsupervised way [2].

The *principal components* are determined using the original variables and features that must be projected on the reduced subspace. An automatic PCA-based feature extractor can use one of the following amounts [11]:

- The desired *number of the principal components*: An amount that directly measures the resulting space dimensionality
- The *fraction of the total variance* to be preserved in the projected data within the reduced subspace

If no parameter is provided to the feature extractor, then the process can only remove the correlated variables and features from the input datasets. This depends on the way in which the automatic PCA-based feature extractor is implemented.

The *basic PCA procedure* for the *feature space transforming* and *dimensionality reduction* performs the following operations [2, 10–14]:

- The *normalization of the input variables or features*, in order to ensure a homogeneous range of the variables and features. The reason for the normalization step is to prevent or reduce the cases in which variables and features with values belonging to large intervals could dominate those with significantly smaller ranges of values [10].
- *Computing the mean vector* μ *and the covariance matrix* $\sum$ for the full dataset. Actually, in some cases it is preferred to subtract the mean from the entire dataset, in order to achieve a dataset with null mean. This is because in the cases of random variables with null means, the correlation matrix $R_x$is equal to the covariance matrix Σ_x, according to [2]

$$\Sigma_x = R_x - E[x] \cdot E[x]^T \tag{1}$$

where

$$R_x = E\left[x \cdot x^T\right] \tag{2}$$

x is the input feature vector.

- The *correlation (or covariance) matrix eigenvector and eigenvalue computation* [11].
- The *selection of the eigenvectors that correspond to the k largest eigenvalues of the correlation (covariance) matrix*. These vectors define an orthonormal basis for the input data and actually they represent the desired *principal components* [10].
- Building *the linear transformation matrix A*, with the size *d x k*, where *d* is the dimensionality of the initial feature space. The columns of this transformation matrix are the orthonormal eigenvectors of the correlation (covariance) matrix [2].
- *The principal component sorting in descending ordering of the corresponding eigenvalues* and therefore in descending informative capacity of the features. The dimensionality reduction is ensured just by this descending sorting of the principal components, as much as it is followed by the lowest significance component removal (representing the variables and features with the lowest variance) [10].
- *The projection of the original data on the k-dimensional extracted subspace* (the subspace that is spanned by the eigenvectors corresponding to the *k* largest eigenvalues of the correlation or covariance matrix), according to [11]

$$y = \mathrm{A}^T \cdot (x - \mu) \tag{3}$$

where x is the data point (feature vector) in the original space, A is the linear transformation matrix, and y is the corresponding data point in the transformed subspace (with reduced dimensionality).

The main properties of PCA are as follows [2]:

- *The mutually uncorrelated feature extraction*, as much as the transformation is generated using the condition for uncorrelated variables. For the variables with null means, this condition is

$$E\left[y(i) \cdot y(j)\right] = 0, \forall i \neq j, i, j = \overline{0, \mathrm{FSS} - 1} \tag{4}$$

while for the variables with non-null means, the mean should be subtracted from the input vectors and the condition becomes

$$E\left[(y(i) - \mathrm{E}\left[y(i)\right]) \cdot (y(j) - \mathrm{E}\left[y(j)\right])\right] = 0, \forall i \neq j, i, j = \overline{0, \mathrm{FSS} - 1} \tag{5}$$

- *The approximation of the mean squared error (MSE)*: The eigenvectors are selected such as to ensure an overall minimized MSE, because they correspond to the largest eigenvalues of the covariance matrix. The error term that corresponds to the lowest eigenvalues is minimized.
- *The total variance maximization*: The extraction of the features that correspond to the k largest eigenvalues allows to maximize the total variance of the original data.
- *The entropy maximization*: The entropy is a measure of the randomness property of a process. In the case of a k-dimensional multivariable Gaussian process with null mean, the extraction of the k features corresponding to the highest eigenvalues also maximizes the process entropy. The variance and the randomness are related.
- *The dimensionality reduction*: PCA is a linear projection transformation that reduces the total dimensionality of a given dataset. This transformation is especially efficient for the cases in which the data points are distributed through a hyperplane. The decomposition of the covariance matrix according to its eigenvectors/eigenvalues is a reliable way to explore the dimensionality of the problem, actually the challenging dimensionality issues for the space in which the available data is placed. PCA projects the entire dataset on the directions that ensure the maximum variance.

There are two important problems that can be associated with PCA and that require specific or particularized approaches, depending on the application constraints and objectives:

- The fact that PCA *does not always preserve the class membership of the input data*, because it is an *unsupervised process* (in the common version)
- The management of *nonlinear cases*, given the fact that PCA is a *linear transformation* of the feature space

Despite its optimality as concerning the variance maximization and MSE approximation, PCA does not always lead to an appropriate *class separation* in the subspace with reduced dimensionality. Therefore, in case of PCA (KL), the dimensionality reduction is actually not optimized in respect to the class separability [2]. This fact can be seen in the following example.

Let's consider a dataset with difficult class separability. This dataset belongs to a medical use-case in which the features are extracted from the input image. The target is to accurately detect the cancer in an early stage. The original data space contains ten dimensions along which the data points are spread. During the overall design process, in the analytical phase PCA is applied in order to extract the most informative features (with the maximized variance) and to have a more detailed view about the potentially hidden patterns within data. Figure 6 shows the outputs of PCA, resulting in two principal components.

In this example, the PCA-based automatic feature extractor was applied using the parameter $frac = 0.9$, given the condition to retain 90% of the total variance.

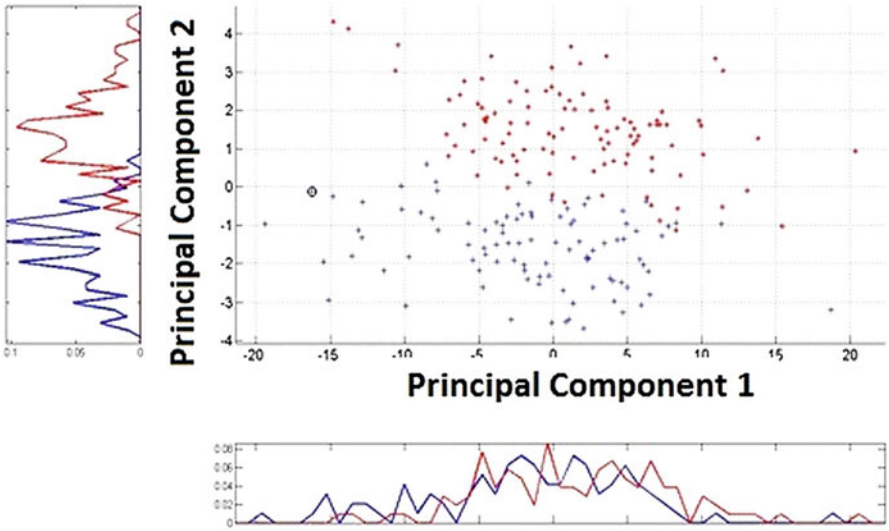


Fig. 6 The results of PCA on a difficult dataset

One can see that the first component principal (projected feature) has a higher overlapping degree and therefore it is not very useful for the modeling process.

A dataset is said to have a difficult separation property if the discrimination process among the problem classes has a high computational complexity and/or time expenses that exceed a fixed and allowed maximum threshold for the considered use-case. The difficult separation problem can be approached with particular transformations of the feature spaces. However, PCA cannot meet this constraint in some cases, given its unsupervised property. This is why PCA does not always help too much for the discrimination process performance enhancement. This usually happens if the intra-class variance of the data is significant.

3.2.2 Supervised PCA

This version of PCA includes a certain type of supervision in the transformation process, allowing to exploit *class membership* information during the feature extraction [11]. In the supervised PCA, the following weighted covariance matrix $\sum$ is used:

$$\Sigma = \sum_{i=1}^{C} P_i \cdot \Sigma_i \tag{6}$$

in which

C is the number of classes, according to the overall design specifications as given through the application requirements.

P_i is the prior probability that is assigned to the class i. A common approach is to use the relative frequency of the samples belonging to that class within the entire training dataset.

$\sum_i$ is the covariance matrix of the class i.

In real applications with high-complexity datasets and also with large amounts of data, even this supervised version of PCA may be not sufficient in order to improve the discriminant properties of the available datasets. Actually another challenge is the proper handling of the *nonlinear cases*.

3.2.3 Kernel PCA (the Nonlinear Case)

For many application domains there are more general cases in which the data generation mechanism presents a high degree of *nonlinearity* [2]. The application data points are placed within more complicated structures of the feature space, and in these cases. PCA may fail and generate incorrect results.

The solution is to consider *nonlinear techniques for dimensionality reduction*. One of these approaches is *kernel PCA*.

Kernel PCA is a version of the classical PCA transformation in which, starting from the original dataset, the overall process of dimensionality reduction is applied taking into account a *nonlinear mapping of the original input data points*. The main difference between the classic (linear) PCA and the nonlinear PCA is that in the last case the eigenvectors corresponding to the highest eigenvalues of the correlation (or covariance) matrix are not explicitly computed, but only the nonlinear data projections along those directions. In this approach, the basic structure in which the input data reside is not explicitly considered [2].

3.2.4 Observations About the PCA Usage

A significant benefit that PCA brings as concerning the feature space transformation for optimization and dimensionality reduction is that it provides additional information in order to perform more advanced data analytic tasks, for example, *to evaluate the degree of complexity of the available data*. This is true despite the fact that PCA does not always preserve the separability between classes.

As a practical consequence, if the available datasets can be evaluated just within the design phase of the application required system, then further modeling process may be significantly enhanced and even optimized by a proper selection of the classification models.

If PCA is applied in order to preserve a given variance fraction from the original data (e.g., 95%) and this process leads to significant dimensionality reduction (such as from 100 to 10 extracted features), then one can conclude that the original data are located into a multiplied copy structure; the informative components for

the discrimination are significantly limited. In such cases a *simple* classifier could be applied with good results (e.g., linear discriminant classifier or even Fisher classifier), especially if the available dataset does not contain a large number of examples. A *simple* model makes use of a small number of independent parameters in respect to the training set size (TSS), reducing the computational expenses.

If the resulting dimensionality is still high comparing to the initial number of variables and designed features (while preserving the same variance level as from the input data), then the discrimination problem is difficult, and this requires for a more complex classifier with high performances if it is trained with large datasets.

3.3 LDA (Linear Discriminant Analysis)

LDA (linear discriminant analysis) is an *unsupervised linear* feature space transformation in which the feature extraction is performed in order to reduce the dimensionality through a linear projection of the input variables and designed features, but preserving and even maximizing the class separation. This process exploits the *class membership information* about the input data.

The method finds a certain linear transformation w that can be applied to the input data in order to generate another dataset that only retains the components that maximize the *Fisher criterion*, given by the *ratio between inter-class and intra-class variance* [2, 4, 12, 15, 16]:

$$J(w) = \frac{{\sigma^2}_{\text{inter-class}}}{{\sigma^2}_{\text{intra-class}}} \tag{7}$$

The optimization of the feature spaces for *dimensionality reduction* by using such *transformations*, either supervised (LDA) or unsupervised (PCA), together with an additional step of *feature selection* (if required) is usually done using the following amounts [15]:

- The *training set size* (*TSS*) for the available data: The number of labeled examples per class within the used training set
- The *feature set size* (*FSS*): The number of features that are generated, extracted, and then retained after these advanced processes (PCA, LDA) and after a required *feature selection* process (based on optimal or nonoptimal approaches)

4 Feature Selection for Dimensionality Reduction

Feature selection is another dimensionality reduction process in which a more discriminant subset of features is drawn from the previously generated and extracted

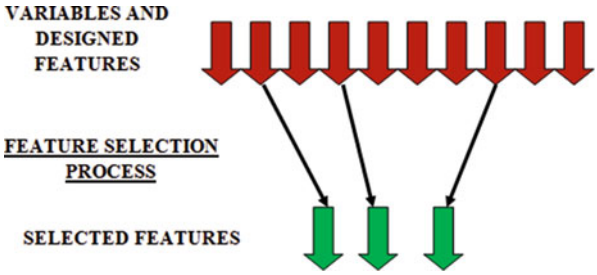


Fig. 7 Feature selection process for dimensionality reduction

feature set, in order to further adjust the dimensionality according to the problem-specific constraints but also to ensure the most discriminant information.

The *dimensionality reduction methods* through *relevant feature selection* usually filter the object (application items) properties in order to retain *the most relevant subset from the original set of attributes*. This is another approach to enhance the model's performance by looking at the data and the involved properties that are described by the original variables together with the new designed and extracted features.

The general model of the *feature selection for dimensionality reduction*, based on various *algorithms* and *evaluation criteria* (in the automatic approach), is represented in Fig. 7.

Typically not all the input variables and designed features are used to ensure the most relevant information for the modeling, either prediction or other specific application tasks. This is a selective approach in which the *dimensionality reduction* exploits the fact that not all the variables and even the new designed and generated features may have the same discriminant power.

On the other hand, *feature selection* can be seen as an optimization problem, looking to find the best solutions (*optimal subsets of attributes*) within an entire searching space of possible solutions.

Any automatic *feature selection* solution that can be designed and developed has the following components [5]:

- The *searching algorithm*: The sequence of operations that explore and analyze the space of the problem domain in order to identify a proper subset of solutions as concerning the relevance and desired dimensionality (if required) for the application target
- The *evaluation criterion*: The performance measure that is used in order to evaluate and compare the relevance and quality of the founded features with respect to the application purposes

4.1 Searching Algorithms for Feature Selection

The typical *searching algorithms* for *feature selection* that are currently used for various application cases being implemented in the existing automatic tools are grouped into the following major categories [2, 4, 5]:

- *Optimal algorithms*: These algorithms always lead to an optimal solution that is represented by the best feature subset with the highest discriminant capacity. The typical optimal algorithms are *exhaustive searching* and *branch-and-bound algorithm*. The last one can only be applied for strict monotonic criteria and not for the classification error.
- *Suboptimal algorithms*: These algorithms find relevant feature subsets that are not always the best in respect to the target, but they are customized according to the real application constraints and requirements (performance and execution speed). Among these algorithms, one can mention the most used ones: *individual selection*, *forward sequential searching*, *backward sequential searching*, *floating selection*, and *genetic algorithms*.

The *exhaustive searching* evaluates each feature subset that can be drawn from the initial set. In this process, a classifier is trained on the current subset, its generalization performance is evaluated, and finally the subset with the best performance is selected. The method is expensive because of the computational effort that requires a significant time to provide the final solution; this is true even for low-dimensional feature spaces. The number of the feature subsets that must be evaluated increases according to a combinatorial law with the number of the initial variables and features [5].

The *non-exhaustive searching* methods for automatic feature selection use more efficient searching algorithms as concerning their execution speed, in order to avoid the exhaustive searching. This requirement is important especially when dealing with applications in which huge amounts of data must be processed with real-time constraints. Among the most used *non-exhaustive feature selection algorithms*, the following approaches are commonly considered for real cases [5]:

- *Depth-first searching*
- *Breadth-first searching*
- *Sequential searching (hill climbing)*, with several versions depending on the direction in which the searching process is performed:
  - *Forward search feature selection (FSFS)*: Iterative adding of features to the currently optimal feature subset. This is a greedy method that locally finds the best solution [11]. The process starts with the best individual feature. In each step, the algorithm iteratively adds another feature to the current subset in order to enhance its performance. The process executes while the current performance can be improved and finishes when all the original features are explored. The resulting feature subsets are suboptimal. An option could be to select several features in each step.

- *Backward search feature selection (BSFS)*: Iterative removing of features from the currently optimal feature subset. The process starts by taking all the generated and extracted features. In each step the algorithm removes the features that decrease the performance given by the evaluation criterion. It stops when a feature removal does not further improve the current performance.
- *Floating searching*: A process that combines the forward and backward searching [2]. The forward searching and backward searching modalities are alternated for some fixed number of steps each other.

The *sequential searching* methods proceed by working on a currently optimal subset by adding or removing the features such as to improve the performance of the selected subset. A required condition for a *non-exhaustive* approach is to meet the monotonic condition for the criterion function that is used to evaluate the partial subsets of features. This means to ensure a continuous improvement of the performance for the feature subsets such as adding or removing features [5, 17].

In the *individual feature selection*, the optimization criterion is applied for each feature as separately taken. The feature is ranked according to the given criterion. In many real cases this method is convenient because of its execution speed, but its drawback is that it cannot exploit the potential correlation among the features. The speed advantage allows this method to be used as an initial step to select a first subset of candidate features within other feature selection algorithms [2, 11]. However, there are many application cases in which some features that are separately considered for the analytic and modeling processes only provide poor performances for discrimination, but if they are considered and applied together within the overall modeling process, the class separation is significantly improved.

The *random feature selection* evaluates an extended set that includes several randomly selected feature subsets, in order to find the best one among all the analyzed subsets. This approach can be used not only to design and implement more complex feature selection algorithms, for example, as support for genetic algorithms but also as an initialization phase of forward/backward searching. The last use-case is justified by the fact that FSFS is not very efficient if all features individually evaluated do not have a good discriminant power. The unreliable options for the initialization phases in greedy-based approaches cannot be changed [2, 11].

A more advanced approach for feature selection in machine learning is to use *genetic algorithms* in order to optimize the full search process to determine the most suitable feature subset solutions [18]. In many use-cases, for instance, in engineering applications, the genetic algorithms allow to find almost optimal feature subsets such as to meet the application constraints for dimensionality and performance. The genetic algorithm usage for *feature selection* is justified by the fact that this process can be seen as an optimization problem, looking to enhance the results of searching within an enlarge solution space for the given problem; in this case, the problem is to find the best subset of features from all possible subsets.

A *genetic algorithm* belongs to the category of evolutionary algorithms and uses the principles of natural selection in order to produce proper solutions for *optimization problems*. The design of these solutions is based on some bio-inspired

operators such as *mutation*, *cross-over*, and *selection*. The genetic algorithms use computational models for which the design is based on the evolution principles [19].

As stated in [18], one of the main properties of the genetic algorithms that justify their usage for *feature selection* in machine learning problems is their low sensitivity to the noise within the input data. When applying genetic algorithm in order to design and develop high-performance feature selection procedures, one should take care about the modality to represent the subspace of all possible feature subsets that can be derived from the original full feature set. Another issue is how to select the suitable evaluation function (the fitness function).

4.2 Evaluation Criteria for Feature Selection

The *evaluation criteria* for *feature selection* are amounts that are used in order to make decisions for maintaining or discarding a certain feature or feature subset. The most used criteria in the feature selection algorithms are as follows [2, 4, 5]:

- *Wrapper criteria*, in which the feature selection is based on the performance of a classifier (classification error rate). The classifier is trained using a certain dataset, its performance is evaluated using an independent dataset, and the evaluation result is used to maintain or to discard the features. This approach is often computationally expensive.
- *Filter criteria*, in which the feature selection process uses numerical indicators for the class separation degree. Among these amounts one can mention the following measures that are commonly applied [5, 14]:
  - *Mahalanobis distance* between two classes A and B:

$$D_M(\mathrm{A}, \mathrm{B}) = \sqrt{(\mu_\mathrm{A} - \mu_\mathrm{B})^T \cdot \Sigma^{-1} \cdot (\mu_\mathrm{A} - \mu_\mathrm{B})} \tag{8}$$

    where μ_A and μ_B are the mean vectors for the two classes and $\sum$ is the classes' covariance matrix. This measure has the following useful properties: scaling-invariance and exploiting the correlation of features. This evaluation criterion is usually applied in the following steps [20]:

    Modeling of each class with a Gaussian distribution
    Computing Mahalanobis distance for each pair of classes
    The criterion evaluation using the matrix that contains the computed distances

  - *The signal-noise report (Fisher criterion)*, defined as the ratio between inter- and intra-class variances:

$$J_F = \frac{(\mu_A - \mu_B)^2}{\sigma_A^2 + \sigma_B^2} \tag{9}$$

in which σ_A^2 and σ_B^2 are the variances of the two classes.

– *Kullback-Leibler divergence*, an amount that evaluates the overlapping or the closeness degree between the two classes' distributions. Kullback-Leibler divergence between the data distributions drawn from two classes (p_A and p_B) is defined according to [2]

$$D_{KL}(A, B) = \int p_A(x) \cdot \ln \frac{p_A(x)}{p_B(x)} dx \tag{10}$$

This amount has nonnegative values, but actually it is not a real distance, because it does not have the symmetry property.

References

1. ***. *What is the difference between feature generation and feature extraction?* https://datascience.stackexchange.com/questions/4903/what-is-the-difference-between-feature-generation-and-feature-extraction
2. Theodoridis, S., & Koutroumbas, K. (2009). *Pattern recognition* (4th ed.). Academic Press: Elsevier.
3. Inselberg, A. (2009). *Parallel coordinates. Visual multidimensional geometry and its applications*. Springer. ISBN 978-0-387-21507-5.
4. Devroye, L., Gyorfy, L., & Lugosi, G. (1997). *A probabilistic theory of pattern recognition*. Springer. ISBN 0-387-94618-7.
5. Polikar, R. (2006). Pattern recognition. In *Wiley encyclopedia of biomedical engineering*. http://users.rowan.edu/~polikar/RESEARCH/PUBLICATIONS/wiley06.pdf.
6. Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1). http://www.cs.bilkent.edu.tr/~saksoy/courses/cs551-Spring2009/papers/jain00_pr_survey.pdf.
7. Yang, L. *Encoding categorical features*. Towards Data Science. https://towardsdatascience.com/encoding-categorical-features-21a2651a065c
8. ***. *What is the difference between categorical, ordinal and interval variables?* Institute for Digital Research and Education. https://stats.idre.ucla.edu/other/mult-pkg/whatstat/what-is-the-difference-between-categorical-ordinal-and-interval-variables/
9. Ray, S. *Simple methods to deal with categorical variables in predictive modeling*. https://www.analyticsvidhya.com/blog/2015/11/easy-methods-deal-categorical-variables-predictive-modeling/
10. Han, J., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques* (3rd ed.). Amsterdam: Elsevier, Morgan Kaufmann Publishers.
11. ***. (2012, April). *PerClass Training Course: Machine Learning for R&D Specialists*, Delft, Netherlands.

12. Zhang, D., Song, F., Xu, Y., & Liang, Z. (2009). *Advanced Pattern Recognition Technologies with Applications to Biometrics*. Hershey: Medical Information Science Reference, IGI Global.
13. Shlens, J. (2009). *Tutorial on principal component analysis*. Center for Neural Science: New York University. http://www.cs.cmu.edu/~elaw/papers/pca.pdf.
14. Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). New York: Academic Press.
15. Soviany, S., Soviany, C., & Puşcoci, S. (2016, July 25–28). *A multimodal biometric system with several degrees of feature fusion for target identities recognition.* The 2016 international conference on security and management (SAM'16), Las Vegas, USA.
16. Tao Li, Shenghuo Zhu, Mitsunori Ogihara: *Using discriminant analysis for multi-class classification: An experimental investigation*, 2009. https://users.cs.fiu.edu/~taoli/pub/Li-discrimant.pdf
17. Webb, A. R., & Copsey, K. D. (2011). *Statistical pattern recognition* (3rd ed.). Chichester: Wiley.
18. Vafaie, H., & De Jong, K. ICTAI 1992: Proceedings of 4th International Conference on Tools with Artificial Intelligence; Arlington, Virginia, USA, November 10–13, 1992.
19. Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and Computing., 4*(2), 65–85.
20. Kapoor, S., Khanna, S., & Bhatia, R. (2010). Facial gesture recognition using correlation and Mahalanobis distance. *(IJCSIS) International Journal of Computer Science and Information Security, 7*(2). https://arxiv.org/ftp/arxiv/papers/1003/1003.1819.pdf.

Data Summarization Using Sampling Algorithms: Data Stream Case Study

Rayane El Sibai, Jacques Bou Abdo, Yousra Chabchoub, Jacques Demerjian, Raja Chiky, and Kablan Barbar

1 Introduction

Subsequent paragraphs, however, are indented. Data streams are large sets of data generated continuously and at a rapid rate in comparison to the available processing and storage capacities of the system that receives them. Thus, these streams cannot be fully stored. That is why we have to process them in one pass without storing them exhaustively. However, for a particular stream, it is not always possible to predict in advance all the processing to be performed. It is, therefore, necessary to save some of these data for future processing. These stored data constitute the "summaries." Several techniques can be used for the construction of data stream summaries, among them the sampling algorithms.

In this chapter, we present a study of these algorithms. Firstly, we introduce the basic concepts of data streams, windowing models, as well as the data stream applications. Next, we detail the different sampling algorithms used in streaming

R. El Sibai
Al Maaref University, Faculty of Engineering, Beirut, Lebanon
e-mail: rayane.elsibai@mu.edu.lb

J. B. Abdo (✉)
Faculty of Natural and Applied Sciences, Notre Dame University, Deir El Kamar, Lebanon
e-mail: jbouabdo@ndu.edu.lb

Y. Chabchoub · R. Chiky
Institut supérieur d'électronique de Paris, Issy-les-Moulineaux, France
e-mail: yousra.chabchoub@isep.fr; raja.chiky@isep.fr

J. Demerjian · K. Barbar
Faculty of Sciences, LaRRIS, Lebanese University, Fanar, Lebanon
e-mail: jacques.demerjian@ul.edu.lb; kbarbar@ul.edu.lb

© Springer Nature Switzerland AG 2020

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_6

environments, and we propose to qualify them according to the following metrics: the number of passes, memory consumption, skewing ability, and complexity.

This chapter is organized as follows. We present in Sect.2 the basic concepts of data streams. We discuss several application domains of data streams in Sect. 3. Section 4 is dedicated to data stream management systems. In Sect. 5, we introduce a detailed study of the summarization techniques used in streaming environments. In Sect. 6, we compare the performance of several sampling algorithms. Section 7 presents future research directions and discusses two research axes related to data sampling. Finally, Sect. 8 concludes the chapter.

2 Data Stream Basic Concepts

2.1 Definition

A data stream is an infinite sequence of tuples generated continuously and rapidly with respect to the available processing and storage capacities. Golab et al. [1] define a data stream as follows:

> *A data stream is a real-time, continuous, ordered (implicitly by arrival time or explicitly by timestamp) sequence of items. It is impossible to control the order in which items arrive, nor is it feasible to locally store a stream in its entirety.*

Several other definitions of data streams have been presented in the literature. Their common characteristic is that they all rely on the main features of these streams, namely [2]:

- Continuous. The tuples arrive continuously and sequentially.
- Fast. The data arrive at a high speed compared to the processing and storage capacities available in the system that receives them.
- Ordered. The order of the data is often defined by timestamp which can be either implicit (data arrival time) or explicit by a timestamp contained in the data.
- Unlimited volume. The size of the data stream is potentially unbounded and can be very large. The exhaustive storage of all the received data is not possible. For instance, one gigabyte of records per hour is generated by AT&T (the largest provider of local and long-distance voice and xDSL services in the United States) [3].
- Volatile. Once the data are processed, they will be discarded, and there is no possibility to treat them another time unless they have been stored in memory. The latter is very small compared to the data stream size. Therefore, immediate processing of the data is required and has to be fast enough to achieve the response time requirement.
- The uncertainty of the data. Some data of the stream may be missing, duplicated, or prone to errors. This is due to external factors such as network congestion, the hardware problem in measurement instruments, etc.

- Push type. The sources of the data are not controllable. They are programmed to send regular measurements. The input rate can vary widely from a data stream to another. Also, some data streams have irregular input rates, while others are highly bursty such as the HTTP traffic streams [4] and the local Ethernet traffic streams [5].

2.2 Data Stream Structure

The form and type of data belonging to a stream depend on the application that led to the data generation. Two types of data can be distinguished: quantitative data and qualitative data. Quantitative data includes the data whose representation is in the numerical form. They usually come from measurements. Qualitative data concern the data represented by specific values from a discrete set of possible values. For example, the weight of the human is a qualitative data that can be represented by the labels low, medium, and high. Depending on the form of the data, the data stream can be represented by three types. In structured data streams, the tuples arrive as records that respect a specific relationship schema including the fields' names of the tuples and the associated values. These tuples arrive in an ordered manner which is often determined by the timestamp of the tuple. The data of a semi-structured stream are heterogeneous sets of weakly structured data; they arrive in the form of XML tags or RDF. In unstructured data streams, the data have different structures. Currently, more than 85% of all business information are unstructured data [6]. These data include e-mails, surveys, Web pages, PowerPoint presentations, chats, etc. WebCQ [7] is a data stream management system for managing unstructured data streams. Its purpose is to monitor the pages on the Web in order to detect and to report the interesting changes that occur to the users.

2.3 Time Modeling and Windowing Models

Data streams are infinite. They must be processed in an online manner, and the data stream management system (DSMS) must provide fast responses to continuous requests while respecting the data stream arrival rate. Thus, windowing models were introduced in the formulation of continuous requests in the DSMS. Data windowing models are based on the principle of cutting the stream into successive portions, and they are used to limit the amount of data to be processed. With the use of windows, at any time, a finite set of stream tuples can be defined and used to respond to the query and produce the corresponding results. Windowing models can be classified according to the time modeling fashion. The temporal aspect of a data stream can be modeled in two manners: the physical time, also called temporal time, and the logical time, also called sequential time. The physical time is expressed in terms of date while the logical time is expressed in terms of the number of elements. One can

notice that, with the logical time model, it is possible to know in advance the number of elements in the window. This number is unknown for the physical window model when the stream rate is variable. Alternatively, each of these two types of windows can be defined by its two boundaries. According to the start and end dates of the window, we can distinguish:

- Fixed window. When using the fixed window model, the stream is partitioned into nonoverlapping windows and the data are preserved only for that part of the stream within the current window. The boundaries of this type of window are accurate and absolute.
- Sliding window. With the sliding window model, the boundaries of the window change over time. Each time an item is added to the window, the oldest element will come out. The queries are periodically performed on the data included in the last window.
- Landmark window. The start date of this window is fixed, while the end date is relative. The size of the window increases gradually as new elements of the stream arrive. For instance, a window between a specific date and the actual date is of type landmark.

3 Data Stream Application Domains

The field of data streams is a subject of growing interest in the industrial community. This interest is reflected by the growing number of applications and industrial systems that continuously generate data streams [1, 8]. These applications are heterogeneous and quite diverse, although their main purpose is the supervision and the control of the data. A typical application of data streams is to study the impact of the weather on the traffic networks. Such a study is useful for analyzing and predicting traffic density as a function of weather conditions [9]. The analysis of weather data is also used to predict weather conditions. Indeed, several weather indicators, such as temperature, humidity, air pressure, wind speed, etc., are indicative of the weather. Through the classification and learning of these data, several models can be derived and used to predict future weather conditions [10]. Social networks also provide more and more data streams that can be exploited in many areas. For instance, TwitterMonitor is a real-time system that allows the detection and analysis of emerging topics on Twitter. The results are provided to the users who in turn interact with the system to rank the detected trends according to different criteria [11]. Data streams can be found in other applications as well, such as website logs [12] and medical remote monitoring [13, 14]. We discuss in the following various other applications:

3.1 Sensor Networks

Wireless sensor networks (WSN) are a special type of ad hoc networks. They use compact and autonomous devices, called sensor nodes. These nodes collect and transmit their observations autonomously to other nodes or to the central server directly. Sensor networks are used in many application fields [15, 16], especially in the domain of electrical energy monitoring [17]. Currently, several electrical energy providers are using smart sensors. These latter send in a continuous manner their observations about the users' electricity consumption to the information systems of the electricity suppliers to which they are linked. The recorded data are in the form of streams. The analysis of these data streams makes it possible to detect several anomalies such as the overconsumption of the energy or the failure in a household appliance. PQStream is a data stream management system designed to process and to manage the stream data generated by the Turkish Electricity Transmission System [18]. The system includes a module for continuous data processing, where several data mining methods such as classification and clustering are applied to the data, a database to store the obtained data analysis results, and a graphical user interface (GUI).

3.2 Financial Analysis

The financial analysis is one of the major application of data streams. Previously, the analysis of financial data was intended to assess the probability of a financial crisis of a company. Nowadays, the analysis of these data involves a wide variety of users, including the commercial providers, banks, investors, credit agencies, and the stock market, among others. In this context, several data mining operations can be applied to financial data, such as fuzzy logic techniques, machine learning, neural networks, and genetic algorithms [19]. The purpose of these operations is to study the impact of one market on another one, monitor the conformity and consistency of the trading operations and improve their performance, and, last but not least, trigger warning signs of changing trends [19]. For instance, Tradebot [20] is a search engine that allows for quantitative analysis of financial data and trading performance and the design of strategies and implementation of scientific experiments to evolve the theory of commerce and, ultimately, work with traders to improve the trading system.

3.3 Network Traffic Analysis

Several real-time systems for the analysis of network traffic have been designed. They aim to infer statistics from the network traffic and to detect critical conditions

such as congestion and denial of service attacks. GigaScope [21] allows monitoring Internet traffic with an SQL interface. Tribeca [22] is a stream-oriented DBMS designed to monitor and analyze the network traffic performance. Tribeca has a query language that can be written and compiled by the users to process the data streams coming from the network traffic. Gilbert et al. [12] proposed Quick-SAND to summarize the network traffic data using the sketches. In this context, one can search, for instance, for the clients who consumed the most bandwidth of the network. The analysis of Web traffic has also several interesting applications and serves several purposes [23]:

- Rank the n most accessed pages during a specific period of time in order to optimize the loading time of these pages.
- Analyze the behavior of the visitors of a particular page or website and identify and determine the distinct users among them.
- Analyze the traffic generated by social networks.

4 Data Stream Management

Traditional database management systems (DBMSs) allow permanent storage and efficient management of the data by exploiting their structure. A query language is used to query the data and retrieve information. However, due to the emergence of data streams, new challenges related to data processing have appeared. These issues are mainly due to the infinite volume of the stream and its very high arrival rate. These new constraints make the use of the DBMSs inadequate. As a result, the traditional data storage and analysis systems need to be revised to allow processing of data streams, hence the emergence of data stream management systems. We detail in the following the main constraints related to the processing of data streams.

A DSMS is supposed to meet the constraints of data streams and the needs of the applications that generate these data by having characteristics related to both the functionality and the performance [2].

- Processing continuous queries. In database applications, the queries are evaluated in a finite environment on persistent data. In such applications, the data does not change as long as the current query is not answered. On the contrary, in streaming applications, data keep growing and the whole environment is fully scalable. As a result, the queries are persistent; they must be executed continuously on volatile data. Also, it is important that the DSMS has a highly optimized engine to handle the volume of data.
- Data availability. The DSMS must ensure the availability of the data at all times. It is also supposed to deal with the system failures and must consider the eventual arrival delay of the data. Due to the eventual long delays in the arrival of data, the operations may be blocked. To avoid such a situation, a maximum delay time (time out) can be specified. Thus, queries that can be blocking are processed in a timely manner even if the data is not complete.

- Infinity of the data. The DSMS must be able to handle the huge volume of the data stream. The use of the load shedding techniques [24] and the summary structures are possible solutions to reduce the load on the system.
- Resistance to stream imperfections. The system must handle the imperfections of the data. In the real world, the data often contain noisy, erroneous, duplicate, and missing values. The DSMS has to deal with these issues.

Several DSMSs have been developed in recent years. These systems are distinguished by the query reformulation languages, the procedures used to represent the streams, and the type of application they are designed for. Thus, some DSMSs have a generalist vocation, such as Aurora [25], TruViso [26], Medusa [27], Borealis [28], TelegraphCQ [29], and StreamBase [24], while others are intended for a particular type of application, such as GigaScope [21], NiagaraCQ [30], OpenCQ [31], StatStream [32], and Tradebot [20].

5 Data Summarization Using Sampling Algorithms

In streaming environments, the data arrive continuously, often at a high rate, and the system that receives the data may not have a sufficient memory to store them exhaustively. Thus, data stream processing implies reducing the size of the data by maintaining and storing a summary of the data in the memory. Sampling algorithms are used to construct a data stream summary. An effective summary of a data stream must have the ability to respond, in an approximate manner, to any query whenever the time period investigated. The purpose of the sampling algorithms is to provide information concerning a large set of data from a representative sample extracted from it. Data stream sampling algorithms are based on the traditional sampling techniques. These techniques require accessing all the data in order to construct the sample, also called summary. However, in the streaming context, this condition is not guaranteed because of the infinite size of the stream. Thus, the sampling algorithms have to be adapted to the streaming context using the windowing models presented in Sect. 2.3.

In the following, we present in detail the different sampling algorithms proposed in the literature to construct a data stream summary.

1. *Simple Random Sampling (SRS).* The SRS algorithm [31] is the most used sampling algorithm. It is simple and gives a random sample. It consists of sampling the data in a random manner, where each item of the data has the same probability p of being selected. Data sampling can be with or without replacement. With the SRS with replacement, the sample may contain duplicate elements since each item of the data may be selected twice or more, whereas, with the SRS without replacement, each item can be selected only once which makes this type of sampling more accurate and convenient.
2. *Systematic Sampling*. Let n be the sample size and N the size of the data. Systematic sampling algorithm divides the data into n groups, each one of size

$k = N/n$. Then, it chooses a random number $j \in [1, k]$ and adds the following elements to the sample: j, j + k, j + 2k, j + 3k ... [31]. Systematic sampling algorithm has several advantages, the sample is easy to be built, and it is faster and more accurate than simple random sampling algorithm since the sampled elements are spread over the entire data [31]. One drawback of this algorithm is its lack of randomness in the sample. In fact, the sampled elements are periodically selected which can impact the quality of the sample. If the original dataset presents a periodicity close to the value k, the sample cannot represent the original dataset, and it will be biased in such case.

3. *Stratified Sampling.* Stratified sampling algorithm [31] is a simple random sampling where the original data is divided into homogeneous subgroups, called strata, according to one or more predefined criteria. A sample will then be extracted from each subgroup by applying the simple random sampling algorithm. Through stratification of the data, the stratified sampling algorithm reduces the sampling error and ensures a high level of representativity of the sample compared to the simple random sampling algorithm. The more the groups are heterogeneous with each other and homogeneous internally, the more the sampling accuracy is high. The use of this type of sampling is beneficial in several cases, for example, when it is desired to highlight a specific subgroup within the data and ensure its presence in the sample. Stratified sampling is also used to represent the smallest, extreme, or rare subgroups of the data in the sample. Since each element of the original data must be associated with a subgroup beforehand the sampling, the construction of the sample using the stratified sampling algorithm will be more expensive than that with the simple random sampling algorithm. The choice of the sample size within each stratum and the choice of the number of strata are the principal issues encountered with the Stratified sampling algorithm.
4. *Weighted Random Sampling (WRS) Without Replacement.* A general summary must be representative of the entire stream. In some cases, this condition is not satisfied. In fact, some data may be overrepresented or underrepresented in the sample. Subsequently, the statistical inferences and conclusions drawn from this sample will be unreliable. This problem is known as "nonresponse" in the survey theory. In such a situation, a correction of the sample is suggested to overcome the lack of representativeness of the sampled data. One of the correction solutions is the weighting of the survey. In contrast to the SRS where all the data have the same probability of being included in the sample, the WRS samples each item with a probability that depends on its associated weight [32].

   Efraimidis et al. [32] introduced two types of WRS: WRS-N-P and WRS-N-W. Their difference lies in the way of calculating the sampling probability of the data. WRS-N-P is the WRS without replacement with defined probabilities. The sampling probability of each item e_k is proportional to the relative weight w_k of the item and is calculated as follows:

$$p_k = \frac{\alpha \times w_k}{\sum_{i=1}^{k} w_i}$$

where α is the size of the sample.

WRS-N-W is the WRS without replacement with defined weights. The sampling probability of each item e_k is proportional to its relative weight w_k with respect to the weights of all not sampled items. The sampling probability p_k of the item e_k is therefore calculated as follows:

$$p_k = \frac{\mathrm{w}_k}{\sum_{i \subset e-S} w_i}$$

where S represents the sample.

A-Chao [32, 33] is a reservoir-based WRS algorithm without replacement and with defined probabilities. It proceeds as follows to construct and to maintain a sample of fixed size k: the first k items of the stream are added to sample with a probability $p = 1$. For each new incoming item, A-Chao adds it to the sample with the probability given by eq. 1 while deleting randomly another item from the sample. The acceptance/rejection algorithm [34] samples each item of the data with a probability proportional to its weight. The weights are associated in a random manner to the items. To construct a random sample of size $k = 1$, the algorithm generates a random number $j \in [1, n]$ and samples the item with index j with a probability equal to w_i/w_max where w_{max} is the maximum weight among all the items. If the item j is not sampled, the process will be repeated until an item is selected. A-Res presented in [35] is a reservoir-based WRS algorithm without replacement. It maintains a sample of fixed size k. Firstly, it adds the first k items to the sample with a probability $p = 1$. Then, for each sampled item, a key is calculated as follows:

$$key_i = u_i 1/wi$$

where u_i is a random value $\in [0, 1]$. After that, for each new incoming item e, calculate its key and find the smallest key l in the sample. If the key of item e is greater than l, replace the item having the smallest key in the sample by the item e. The main concern with these three algorithms is the lack of a policy for the definition and the updating of the weights. The weights are generated in a random manner and only once.

5. *Reservoir Sampling.* The goal of the reservoir sampling algorithm [36] is to maintain a uniform random sample of a fixed size k from the entire data stream, without requiring a priori knowledge of the stream size. Firstly, the algorithm adds the first k received elements of the stream to the reservoir (sample), each with a probability equal to 1. After that, with the arrival of new elements, the algorithm adds each element e to the sample with a probability $p = k/i$, where i is the index of e, while deleting a random element from the sample. This algorithm is simple and suitable for the streaming environment as it is executed in one pass. The main concern with the reservoir sampling algorithm is that the sample becomes irrelevant over time. In fact, the more recent the elements, the less likely they are to be included in the sample.

6. *Backing Sampling.* Backing sampling [37, 38] is a uniform random sampling algorithm based on the reservoir sampling algorithm [36]. It was designed to overcome the problems of expired data in a sliding window, which are not handled by the reservoir sampling algorithm. The algorithm starts by adding the first k received elements of the stream to the reservoir with a probability equal to 1. Then, the algorithm skips a random number of elements and add the next element to the reservoir with a probability equal to the sampling rate. Another random number of items is skipped and so forth. The aim of backing sample algorithm is to maintain a sample containing only the unexpired elements of the stream. For this purpose, two bounds are defined: the upper bound K representing the maximal size of the reservoir and the lower bound L representing the minimal size of the reservoir. When an element expires, the algorithm removes it from the sample if it was present. Successive deletions of the expired elements lead to the decrease of the size of the reservoir. Therefore, at each time the size of the reservoir becomes smaller than the lower bound, the sampling process will be reinitialized and a new reservoir containing k elements will be reconstructed.
7. *Concise Sampling.* Gibbons et al. [39] proposed the concise sampling algorithm to construct and to maintain a concise representation of the data, where each element that occurs several times in the original data will be represented in the sample as a pair <*value*, *count* > , where *value* is the value of the item and *count* in the number of occurrences of this item in the original data. If an item occurs only once, it will be represented as a singleton value. The concise sampling algorithm defines a *footprint* for the sample which represents the size of the sample to be stored in the memory. This size is defined as the number of the items in the sample and the corresponding *count* value. At first, all the received items are added to the sample with a probability $p = 1$. As new items arrive, the *count* value of each item in the sample will be updated, and the new incoming items will be added randomly to the concise sample. If the size of the sample exceeds the predefined *footprint*, the algorithm proceeds to reduce the size of the sample either by deleting singleton items in a random manner or by decreasing randomly the *count* values of the items in the sample.
8. *Chain-Sample.* Under sampling over a sliding window, the main difficulty is how to maintain a representative sample of the window. In fact, expired elements must be replaced in the sample in case they are present. Assuming that recent data are more important than older data, Chain-sample [40] maintains a sample of fixed size $k = 1$ over a logical sliding window. For a sample of size $k > 1$, the algorithm is repeated k times. At the first stage, the algorithm selects an element e_i from the first window with a probability $Min(i, n)/n$, where i is the index of e_i in the stream and n is the size of the window. Then, a random replacement element r is selected from the group of elements with indexes going from $i + 1$ to $i + n$ and it will replace e_i when this latter expires. When r arrives at the current window, it will be stored and a random replacement element is chosen from the elements going from $r + 1$ to $r + n$ and so on.

Chain-sample algorithm has the disadvantage of generating a sample containing duplicate elements in case $k > 1$.

9. *Priority Sampling.* In addition to the chain-sample algorithm, Babcock et al. [40] proposed the priority sampling algorithm to construct and to maintain a uniform sample over time-based sliding windows. The key idea of the priority sampling algorithm is to assign a random priority $p \in [0, 1]$ for each incoming item of the stream and to select the item having the highest priority in the window.
10. *Random Pairing (RP) Sampling.* The RP sampling algorithm [41, 42] builds and maintains a uniform sample of fixed size over a sliding window. RP retains three measures on each window: $c1$ which represents the number of expired items that are included in the sample, c2 which counts the number of expired items that are not included in the sample, and d which depicts the number of all expired items ($d = c1 + c2$). At each time a sampled item expires, RP removes it from the sample. When a new item arrives at the window, it can be added to the sample depending on the value of d. If $d = 0$ (the expired element in the window is not included in the sample), the addition of the new item to the sample follows the reservoir sampling algorithm [36]. On the contrary, if $d > 0$, the new item will be added to the sample with a probability equal to $\frac{c_1}{c_1+c_2}$. Following this step, the values of $c1$, $c2$, and d are updated.
11. *StreamSamp.* StreamSamp [23] is a progressive sampling algorithm based on the simple random sampling algorithm. As soon as they are received, the items of the stream are sampled with a fixed sampling rate p. When the predefined sample size k is reached, StreamSamp associates the order 0 to the sample and stores it and constructs a second sample of the same size k and so on. As the size of the stream increases, the number of samples of order 0 also increases. When this number exceeds a given bound, StreamSamp proceeds to fuse the two old samples of order 0 into one sample of size k by performing a simple random sampling of rate $p = 0.5$. The newly obtained sample is of order 1 and so on.
12. *Distance-Based Sampling for Streaming Data (DSS).* DSS [43] maintains a sample of a fixed size k and updates it each time a new element is added to the stream. The algorithm manages the insertions into the sample so that the difference between the sample and the stream is minimal. This difference is defined by:

$$\sum_{A \in i} \mid f\,(A, S_0) - f\,(A, S) \mid$$

where I denotes the set of frequent itemsets. Initially, the first k items are added to the sample $S_0$with a probability equal to 1. DSS uses the notion of ranking to manage the insertions in the sample. The element with the highest ranking is the one that its deletion from the sample engenders an important increase of the difference between S_0 and S. On the contrary, the element with the lowest ranking denoted LRT is the element of the sample whose presence or

absence from the sample will almost not affect the difference between $S_0$and S. Initially, two tables for the ranking of the elements contained in both $S_0$and S are initialized. The classification of the elements in each of the two tables is calculated from the equation:

$$Dist = Dist\,(S_0 - t, S)$$

where t is the element.

When a new element t is added to the stream, two distances will be calculated:

$$D_{without}t = Dist\,(S_0,\ \ S + t) \text{ and } D_{with}t = Dist\,(S_0\ \ + t - LRT,\ \ S + t)\,.$$

If $D_{without}t > D_{with}t$, the presence of t in $S_0$is favored, and it will replace LRT in the sample. If $D_{with}t > D_{without}t$, the element will be skipped. DSS is very expensive; each time a new element is added to the stream, the distance *Dist* has to be calculated and the set of frequent itemsets of the entire stream must be again computed.

The quality of a summary depends on the sampling algorithm used to construct it. The effectiveness of a data stream sampling algorithm can be qualified according to the following metrics [44]:

- The number of passes over the data. One of the main constraints that a data stream sampling algorithm must satisfy is the single pass on the data. In fact, since it is impossible to store all incoming data for further processing, the data stream has to be treated on the fly and without prior storing the entire data [45]. Hence, it is important to make sure that the sampling algorithm makes only one pass over the data to sample them.
- Memory consumption. A data stream is by definition infinite. Most sampling algorithms use an always increasing sample size. One can cite [31]. The sample size is often proportional to stream size. It depends on the sampling rate which is set according to the required accuracy. With a high sampling rate, very few information about the original dataset will be lost. However, this requires more resources, in particular, memory usage to store the sample. Some other sampling algorithms (such as reservoir sampling [36] and StreamSamp [46]) use a fixed bounded memory independent of the stream size. In this case, the sample is always updated to replace old elements.
- Skewing ability: It represents the possibility to give more chance for some particular items to be selected and added to the sample. It can be based on the content of the item or its timestamp. Besides, skewing the data must comply with a certain well-defined policy. This latter will define how to associate the weights and how to calculate and update them over time according to the objective of the application.
- Complexity. Sampling algorithms have to be fast enough to deal with the high rate of current data streams. Therefore, a low complexity is required to reduce the execution time and the CPU charge of the sampling algorithm. This criterion is particularly important when implementing the sampling algorithm on devices

Table 1 Comparison of data stream sampling algorithms

Sampling algorithms	Number of passes	Skewing ability	Memory consumption
Simple random [31]	1	No	Unbounded
Systematic [31]	1	No	Unbounded
Stratified [31]	1	No	Unbounded
Weighted random [35, 32]	1	Yes	Unbounded
Reservoir [39, 51]	1	No	Bounded
Distance based [43]	>1	No	Bounded
Concise [39]	1	No	Bounded
Backing [37, 38]	>1	No	Bounded
Chain-sample [40]	1	No	Bounded
Priority [40]	1	Yes	Bounded
Random pairing [41, 42]	1	No	Bounded
StreamSamp [23]	1	Yes	Bounded

Adapted from [48]

with limited resources such as sensors. Implementing low complexity sampling processes on WSN devices is a solution to conserve the energy consumption and the CPU usage and to deal with the memory and time constraints [47].

A brief comparison of these algorithms is presented in Table 1. Three metrics are considered to evaluate the effectiveness of these algorithms: number of passes over the stream, skewing ability, and memory consumption.

6 Performance Evaluation of Sampling Algorithms

Data stream sampling is intended to build a sample on which the future data analysis tasks will be performed. The effectiveness of the sample depends on several parameters: the used sampling algorithm, the chosen sampling rate k/n, and the window size n if the sliding window model is adopted. Given a stream of items, the goal is to find the most relevant sampling technique to use and the right parameters k and n to choose.

We present in this section a performance comparison of the following sampling algorithms: simple random sampling (SRS), deterministic sampling, and chain-sample algorithms in terms of execution time and sampling accuracy. The detailed study was presented in our previous work [44].

We evaluate in Fig. 1 the computational resources of the three algorithms in terms of execution time needed to summarize a given stream, for different sampling rates. The size of the window n is fixed to 10. Regarding the chain-sample algorithm, we use the "Inverting the selection for high sampling rates" version. It is the improved version of the Chain+ algorithm that reduces the execution time for high sampling rates [49]. The results show that the deterministic sampling algorithm has the

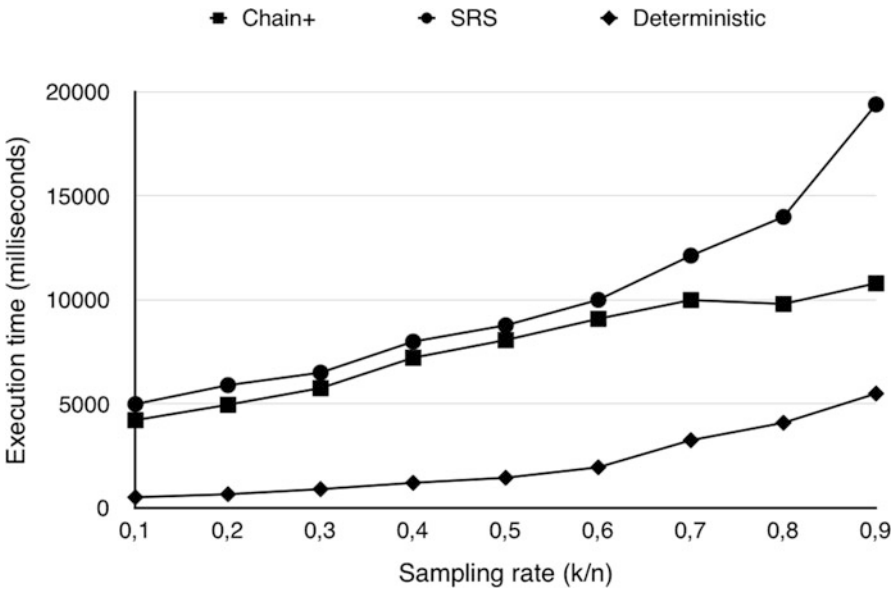


Fig. 1 Execution time of the deterministic, simple random, and chain+ sampling algorithms over a purely sliding window, for different sampling rates k/n [44]

smallest execution time compared to SRS and chain algorithms, as it is the simplest sampling algorithm. Regarding SRS, the redundancy in the sample occurs when the sampling rate k/n is>0.1. This happens when the same item is selected many times to be added to the sample on the same sliding window. This problem becomes more severe when k is close to the size of the window n. To avoid this duplication, and to provide exactly k distinct items in each sample, the selection procedure must be repeated until selecting an item that is not already present in the current sample. This condition adds a considerable overhead in terms of execution time, especially when the sampling rate is high (k is close to n). The difference in the execution time for SRS and chain algorithms increases with the increase of the sampling rate k/n. This difference becomes clearer when k/n is greater than 0.5. This is due to the use of the "Inverting the selection for high sampling rates" strategy which reduces the collision rate to be equal to that of a sampling rate of $1 - k/n$ when k/n exceeds 0.5 and, thus, reduces the execution time of the chain-sample algorithm.

When evaluating a query on the most recent data of the stream, we only have a set of samples built on the windows of this stream. Thus, the response to the query can be provided by estimation. We study in this section the impact of the sampling on the quality of the estimation and we limit the analysis to the MEAN aggregation query.

Let m be the true mean calculated from the original data in a given window. The empirical mean $\overline{X}$ of the window sample is an unbiased estimator of m, calculated as follows:

$$\overline{X} = \frac{1}{k} \sum_{i=1}^{k} e_i$$

where k is the size of the sample and e_i is the value of the item of index i in the sample. The accuracy of the mean estimation is quantified by the relative error given by:

$$error = \mid \frac{m - \overline{X}}{m} \mid \times 100\%$$

There are two parameters that affect the sampling accuracy: the sampling rate k/n and the window size n. Actually, we can achieve a sampling rate p while sampling with different sample sizes k and window sizes n, such that:

$$p = \frac{k}{n} = \frac{x \times k'}{x \times n'}$$

where x is a positive integer.

We study in Fig. 2 the impact of the data periodicity on the accuracy of the estimation. The data are sampled using the chain-sample, SRS, and deterministic sampling algorithms with a very small sampling rate. In fact, in degraded environments such as wireless sensor networks (WSN), the sensors are devices with limited resources in terms of battery power, CPU, memory, and bandwidth [50]. Because of memory limitations, WSN devices cannot store a lot of information. However, each node must compute, process, and transmit its data to other nodes or to the central system. Implementing the sampling process on WSN devices and using a low sampling rate can be a solution to deal with memory and time constraints.

Figure 2 depicts the sampling accuracy for SRS, chain, and deterministic sampling algorithms in such degraded environments. As shown in this figure, the deterministic sampling algorithm has the highest estimation error, and SRS and chain-sample algorithms give both similar results. This can be explained by the fact that when the period of the data is smaller than the sampling rate, the sampled data with the deterministic sampling algorithm will be unrepresentative of the window as a whole. In our case, the selected data often belong to a peak of water consumption, leading to a high error of the mean estimation. On the contrary, the chain-sample and the SRS algorithms are more representatives because the elements are selected randomly in an independent manner of the data periodicity. One can notice that the size of the window has an impact on the sampling accuracy of the three sampling algorithms. The decrease of the error for a high window size can be explained by the law of large numbers.

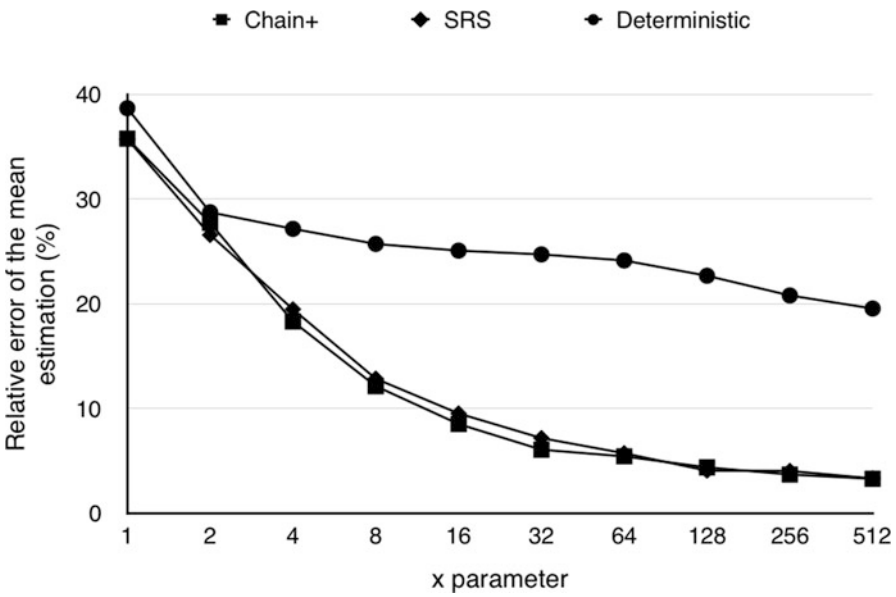


Fig. 2 Mean estimation error of the deterministic, simple random, and chain+ sampling algorithms for a sampling rate of 1 observation per 12 h, for different window sizes [44]

7 Future Directions

Many research axes related to data summarization using sampling algorithms are identified. They can be summarized as follows.

7.1 Sampling Impact on Data Stream Statistical Inference

Data stream sampling is intended to build a sample on which the future data analysis tasks will be performed. The effectiveness of the sample depends on several parameters: the used sampling algorithm, the chosen sampling rate k/n, and the window size n if the sliding window model is adopted. In this light, several studies have been dedicated to examining the impact of data sampling. Mai et al. [51] presented an in-depth study of the effect of data sampling on anomalies detection of traffic measurements coming from high-speed IP-backbone networks. Several sampling methods were applied to sample the IP packet traces. The authors assessed the impact of sampling on port scan detection, volume anomaly detection, and data characteristics, namely, the variance. The impact of IP packets sampling was also addressed in [52–54]. In [55], the authors studied the impact of sampling on the analysis of tweets generated on social networks. This analysis includes the study of

tweet volume, tweet distribution, and user influence measured by his comments and retweet activities.

7.2 Adaptive Sampling

In the sampling algorithms considered in this chapter, the sampling rate is constant during the execution of the algorithm. In some use cases, it is useful to adapt dynamically the sampling rate to some varying conditions such as the available computational resources as proposed in [56].

8 Conclusion

Data streams are volatile; once expired, they are no longer available for analysis. This makes it impossible to evaluate any undefined query before the arrival of the data, while new requirements may appear after the arrival of the stream. In this case, the data stream management system cannot answer new queries. One solution to overcome this problem is to store an extract of the stream in a compact structure, called summary. In this chapter, we presented the basic concepts of data streams, windowing models, and application domains. Thereafter, we detailed the different sampling algorithms used to construct a data stream summary, and we focus particularly on their drawbacks. Finally, we studied the effectiveness of three sampling algorithms in terms of the accuracy of the provided response, and the time needed to update the summary.

References

1. Golab, L., & Ozsu, M. T. (2003). Issues in data stream management. *ACM SIGMOD Record, 32*(2), 5–14.
2. Gabsi, N. (2011). *Extension et interrogation de résumés de flux de données*. PhD thesis, Télécom ParisTech.
3. Chakravarthy, S., & Jiang, Q. (2009). *Stream data processing: A quality of service perspective: Modeling, scheduling, load shedding, and complex event processing* (Vol. 36). New York: Springer.
4. Crovella, M. E., & Bestavros, A. (1997). Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking, 5*(6), 835–846.
5. Leland, W. E., Taqqu, M. S., Willinger, W., & Wilson, D. V. (1993). On the self- similar nature of Ethernet traffic. *ACM SIGCOMM computer communication review, 23*, 183–193.
6. Blumberg, R., & Atre, S. (2003). The problem with unstructured data. *DM Review, 13*(42–49), 62.
7. Liu, L., Calton, P., & Tang, W. (2000). Webcq-detecting and delivering information changes on the web. In *Proceedings of the ninth international conference on information and knowledge management* (pp. 512–519). New York: Association for Computing Machinery.

8. Golab, L., & Ozsu, M. T. (2003). *Data stream management issues–a survey*. Technical report, Apr. 2003. http://db.uwaterloo.ca/~ddbms/publications/stream/streamsurvey.pdf
9. Gietl, J. K., & Klemm, O. (2009). Analysis of traffic and meteorology on air-borne particulate matter in Münster, Northwest Germany. *Journal of the Air & Waste Management Association, 59*(7), 809–818.
10. Bartok, J., Habala, O., Bednar, P., Gazak, M., & Hluchy, L. (2010). Data mining and integration for predicting significant meteorological phenomena. *Procedia Computer Science, 1*(1), 37–46.
11. Mathioudakis, M., & Koudas, N. (2010). Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD international conference on management of data* (pp. 1155–1158). ACM.
12. Gilbert, A. C., Kotidis, Y., Muthukrishnan, S., & Strauss, M. (2001). *Quick-sand: Quick summary and analysis of network data*. Technical Report, Dec. 2001. https://citeseer.nj.nec.com/gilbert01quicksand.html.
13. Sachpazidis, J. (2002). @ home: A modular telemedicine system. In *Mobile computing in medicine, second conference on mobile computing in medicine, workshop of the project group MoCoMed*. GMDS-Fachbereich Medizinische Informatik & GI-Fachausschuss 4.7 (pp 87–95). GI.
14. Brettlecker, G., & Schuldt, H. (2007). The osiris-se (stream-enabled) infrastructure for reliable data stream management on mobile devices. In *Proceedings of the 2007 ACM SIGMOD international conference on management of data* (pp. 1097–1099). New York: Association for Computing Machinery.
15. Liu, J., Liu, J., Reich, J., Cheung, P., & Zhao, F. (2003). Distributed group management for track initiation and maintenance in target localization applications. In *Information processing in sensor networks* (pp. 113–128). Heidelberg: Springer.
16. Gurgen, L. (2007). *Gestion à grande échelle de données de capteurs hétérogènes*. PhD thesis, Grenoble, INPG.
17. Abdessalem, T., Chiky, R., Hébrail, G., Vitti, J. L., & GET-ENST Paris. (2007). Traitement de données de consommation électrique par un système de gestion de flux de données. In *EGC* (pp. 521–532).
18. Küçük, D., İnan, T., Boyrazoğlu, B., Buhan, S., Salor, Ö., Çadırcı, I., Ermiş, M. (2015). *Pqstream: A data stream architecture for electrical power quality*. arXiv preprint arXiv:1504.04750.
19. Kovalerchuk, B., & Vityaev, E. (2000). *Data mining in finance: Advances in relational and hybrid methods* (Vol. 547). New York: Kluwer Academic Publishers.
20. https://traderbotmarketplace.com.
21. Cranor, C., Gao, Y., Johnson, T., Shkapenyuk, V., & Spatscheck, O. (2002). Gigascope: High-performance network monitoring with an SQL interface. In *Proceedings of the 2002 ACM SIGMOD international conference on management of data* (pp. 623–623). ACM.
22. Sullivan, M., & Heybey, A. (1998). A system for managing large databases of network traffic. In *Proceedings of USENIX*.
23. Csernel, B. (2008). *Résumé généraliste de flux de données*. PhD thesis, Paris, ENST.
24. Tatbul, N., Cetintemel, U., Zdonik, S., Cherniack, M., & Stonebraker, M. (2003). Load shedding in a data stream manager. In *Proceedings 2003 VLDB conference* (pp. 309–320). San Diego: Elsevier.
25. Abadi, D. J., Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., & Zdonik, S. (2003). Aurora: A new model and architecture for data stream management. *The VLDB Journal, 12*(2), 120–139.
26. http://www.truviso.com/, 2004.
27. Cetintemel, U. (2003). The aurora and medusa projects. *Data Engineering, 51*, 3.
28. Abadi, D. J., Ahmad, Y., Balazinska, M., Cetintemel, U., Cherniack, M., Hwang, J.-H., Lindner, W., Maskey, A., Rasin, A., Ryvkina, E., et al. (2005). The design of the borealis stream processing engine. *Cidr, 5*, 277–289.

29. Chandrasekaran, S., Cooper, O., Deshpande, A., Franklin, M. J., Hellerstein, J. M., Hong, W., Krishnamurthy, S., Madden, S. R., Reiss, F., & Shah, M. A. (2003). TelegraphCQ: continuous dataflow processing. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data* (pp. 668–668). ACM.
30. Chen, J., DeWitt, D. J., Tian, F., & Wang, Y. (2000). Niagaracq: A scalable continuous query system for internet databases. *ACM SIGMOD Record, 29*, 379–390.
31. Chao, M. T. (1982). A general purpose unequal probability sampling plan. *Biometrika, 69*(3), 653–656.
32. Olken, F., & Rotem, D. (1992). Maintenance of materialized views of sampling queries. In *Proceedings of eighth international conference on data engineering* (pp. 632–641). IEEE.
33. Efraimidis, P. S., & Spirakis, P. G. (2006). Weighted random sampling with a reservoir. In *Information processing letters* (pp 181–185)
34. Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS), 11*, 37–57.
35. Phillip, B. (1997). Gibbons, Yossi Matias, and Viswanath Poosala. *Fast incremental maintenance of approximate histograms. In VLDB, 97*, 466–475.
36. Gibbons, P. B., Matias, Y., & Poosala, V. (2002). Fast incremental maintenance of approximate histograms. *ACM Transactions on Database Systems (TODS), 27*, 261–298.
37. Gibbons, P. B., & Matias, Y. (1998). New sampling-based summary statistics for improving approximate query answers. *ACM SIGMOD Record, 27*, 331–342.
38. Babcock, B., Datar, M., & Motwani, R. (2002). Sampling from a moving window over streaming data. *In Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms, 2002*, 633–634.
39. Gemulla, R., Lehner, W., & Haas, P. J. (2006). A dip in the reservoir: Maintaining sample synopses of evolving datasets. In *Proceedings of the 32nd international conference on Very large databases* (pp. 595–606). VLDB Endowment.
40. Gemulla, R. (2008). *Sampling algorithms for evolving datasets*. PhD thesis, Technischen Universitat Dresden Fakultat Informatik.
41. Dash, M., & Ng, W. (2006). Efficient reservoir sampling for transactional data streams. In *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)* (pp. 662–666). IEEE.
42. El Sibai, R., Chabchoub, Y., Demerjian, J., Chiky, R., & Barbar, K. (2018). A performance evaluation of data streams sampling algorithms over a sliding window. In *Communications conference (MENACOMM), IEEE Middle East and North Africa* (pp. 1–6). IEEE.
43. Muthukrishnan, S., et al. (2005). Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science, 1*(2), 117–236.
44. Csernel, B., Clerot, F., & Hébrail, G. (2006). Datastream clustering over tilted windows through sampling. In *Knowledge discovery from data streams* (p. 127).
45. de Aquino, A. L. L., Figueiredo, C. M. S., Nakamura, E. F., Buriol, L. S., Loureiro, A. A. F., Fernandes, A. O., & Coelho, C. J. N., Jr. (2007). A sampling data stream algorithm for wireless sensor networks. In *2007. ICC'07. IEEE international conference on communications* (pp. 3207–3212). IEEE.
46. Mai, J., Chuah, C.-N., Sridharan, A., Ye, T., & Zang, H. (2006). Is sampled data sufficient for anomaly detection? In *Proceedings of the 6th ACM SIGCOMM conference on internet measurement* (pp. 165–176). New York: ACM.
47. El Sibai, R., Chabchoub, Y., Demerjian, J., Kazi-Aoul, Z., & Barbar, K. (2016). Sampling algorithms in data stream environments. In *2016 IEEE first International Conference on Digital Economy Emerging Technologies and Business Innovation (ICDEc)* (pp. 29–36). IEEE.
48. El Sibai, R., Chabchoub, Y., Demerjian, J., Kazi-Aoul, Z., & Barbar, K. (2015). A performance study of the chain sampling algorithm. In *2015 IEEE seventh international conference on intelligent computing and information systems (ICICIS)* (pp. 487–494). Cairo: IEEE.
49. Brauckhoff, D., Tellenbach, B., Wagner, A., May, M., & Lakhina, A. (2006). Impact of packet sampling on anomaly detection metrics. In *Proceedings of the 6th ACM SIGCOMM conference on internet measurement* (pp. 159–164). New York: ACM.

50. Pescapé, A., Rossi, D., Tammaro, D., & Valenti, S. (2010). On the impact of sampling on traffic monitoring and analysis. In *Teletraffic congress (ITC), 2010 22nd international* (pp. 1–8). Piscataway: IEEE.
51. Hu, Z., Liu, J., Zhou, W., & Zhang, S. (2016). Sampling method in traffic logs analyzing. In *2016 8th international conference on intelligent human-machine systems and cybernetics (IHMSC)* (Vol. 1, pp. 554–558). Piscataway: IEEE.
52. Xu, K., Wang, F., Jia, X., & Wang, H. (2015). The impact of sampling on big data analysis of social media: A case study on flu and Ebola. In *Global communications conference (GLOBECOM), 2015 IEEE* (pp. 1–6). Piscataway: IEEE.
53. Schinkel, M., & Chen, W.-H. (2006). Control of sampled-data systems with variable sampling rate. *International Journal of Systems Science, 37*(9), 609–618.
54. Liu, L., Calton, P., & Tang, W. (1999). Continual queries for internet scale event-driven information delivery. *IEEE Transactions on Knowledge and Data Engineering, 11*(4), 610–628.
55. Zhu, Y., & Shasha, D. (2002). Statstream: Statistical monitoring of thousands of data streams in real time$_{**}$ work supported in part by us NSF grants iis-9988345 and n2010: 0115586. *In VLDB'02: Proceedings of the 28th international conference on very large databases* (pp 358–369). Elsevier.
56. McLeod, A. I., & Bellhouse, D. R. (1983). A convenient algorithm for drawing a simple random sample. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 32*, 182–184.

Fast Imputation: An Algorithmic Formalism

Devisha Arunadevi Tiwari

1 Introduction

Missing value imputation is a serious problem reported by data analysts. Many contemporary methods used in the past simply work by replacing missing values by observed variables or delete them causing discontinuity in data and disrupt its homogeneity. Some methods show near-far imputation which significantly fluctuates the distribution moments like mean, variance, skew and kurtosis of the dataset. Researches in the past stab to overcome this flaw, as the issue became serious for reliability of time series prediction through randomized imputation, but fall deficit for reasonable results. Time series prediction is an endogenous event which requires influential composition of data characteristics subjective to strong inference in a given large dataset. But available imputation techniques are hardly suggestive and reliable for complex operations like time series prediction and rare for forecasting, which is an exogenous event, fully dependent on outcome of time series prediction. The data collection is fully dependent on statistics available on the World Wide Web.

The World Wide Web has been a burgeoning platform of information availability and has become a resource of knowledge. This has led to development of e-commerce and online markets combating the real shops up to limits of being used only as a rural market. The online markets have placed abandon options and high availability of products providing users with a bundle of engulfing choices [1]. The growth of information availability on the web has put a platform for facilities like recommendation services to ease the users in generating choices and opinions for product search. The e-commerce domain started with recommender systems as a

D. A. Tiwari (✉)
Department of Computer Science and Engineering, Sipna College of Engineering and Technology, Amravati, Maharashtra, India

© Springer Nature Switzerland AG 2020

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_7

tool for summarization of products and services and filtering the information as per the likes of the user. The recommender systems were designed to store the history of likes and purchase of a certain product through opinions from community users. The technique of recommendation was pioneered from the concept of search engines. Recommender systems include information retrieval techniques but search engines are basically information filtering models. The information retrieval models use ranking algorithms to fetch needful data based on user interest context. A typical recommender system uses ranking schema to ensemble item of interest in the order of the ratings given by the user. Different methods to design recommender systems exist based on the application where these will be used. The widely used techniques to design recommender systems are collaborative filtering and content-based filtering. Also, knowledge-based recommendation is the most prominent technique nowadays [2].

In a collaborative filtering technique, the users in the neighbourhood like the family, friends and colleague suggest and share their opinions for item promotion. This creates a cluster of people with similar likes and genre [3]. The reverse approach for a precise recommendation also used popularly is item-based clustering in which a set of users who purchased the same item is listed under user of same genre. Also, a user-item pair history forms the basis in future recommendations. The collaborative filtering algorithms are designed preliminarily from neighbourhood cluster shaping based on user genre. The next step includes forming similarity measures from cluster similarity. Most applications used this technique. Various algorithms [4–7] have been developed and used in recommender systems available on different web applications. The details of some of them have been discussed in the following section.

The information domain consists of users' preferences in the form of ratings in huge variants. The preferences given by the user are fetched in the form of a feature vector and represented like a triple (user, item, and rating). The data in the triple can be available in different scales and implicit representation. The task of the collaborative filtering recommender systems is to compute the unknown ratings (user, item, '?') when a set of known (user, item, rating) triple is given. The following example gives more clarity.

Suppose the information domain has a set of items like I = {Product1, Product2, Product3} and a set of users like U = {User1, User2, User3} and the following as shown in Fig. 1 is available.

In the given piece of information, the task is to find the values of the unknowns as a judgement of the existing values given by a certain user. Suppose User1 has a same genre as that of User2 because both have given same ratings to Product3. Also, User3 has same genre as that of User3 because both have given the same ratings

Fig. 1 Rating matrix based on 5-point rating scale

	Product1	Product2	Product3
User1	3	?	2
User2	?	4	2
User3	3	4	5

to Product1. Based on these sets of judgements, a similarity value is computed to predict the ratings of the unknowns. The task of finding the missing values requires a predictor. When the information is dense, it is easier to compute the rating predictions of the missing values. Sometimes, the information available is sparse. So, collaborative filtering recommender systems are the better solution for dealing with overloaded online information spaces. The collaborative filtering recommender works in two phases here. It computes the missing values from the known values using mathematical predictors. Secondly, it computes the recommendation wherein a list of products the user needs is produced from best ranks.

The recommendation generation can be treated as a missing value imputation problem. The dataset consists of two partitions, namely, the users and the item. The mapping of user to a particular item is defined as rating. Hence, the process of recommendation is mathematically the problem of solving unknown mappings in a bipartite graph problem, as illustrated in Fig. 2.

The k-nearest neighbour algorithms are one of the preliminarily available algorithms to implement collaborative filtering recommender systems. The algorithms work by forming the clusters of similar users and using a suitable voting algorithm to compute recommendation. It requires cluster formation to congregate the information under one head. There are different techniques to ensemble users based on their genre. In this method, the centroid is chosen as the initial point to form a cluster. The number of clusters formed depends on 'k' number of centroid chosen. In collaborative filtering recommender systems, 'k' is chosen on the number of genre, a typical user-repository has. There are different techniques to formulate the mean vector, also called the 'centroid' here. The clustering phase defines the accuracy in a typical user-specific collaborative recommender design. There are plenty of data mining techniques available for preprocessing the clusters and cluster space reduction. The next task is to find similarity measure between two feature vectors. There are plenty of data mining techniques to compute distance between two feature vectors in assessment of similarity. Each technique uses a suitable

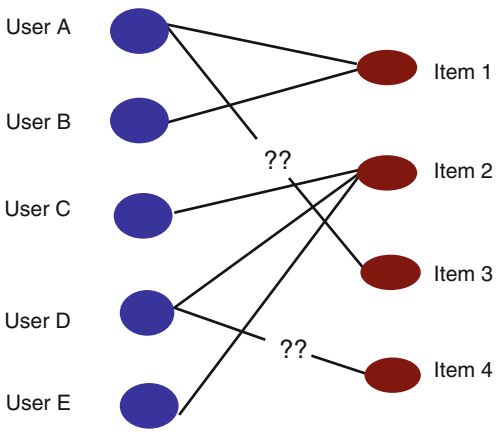


Fig. 2 Heterogeneity in missing pattern

distance metric. A discussion of these metrics and their application background has been propositioned below in detail.

1. *Euclidean Distance:* The Euclidean distance computes the root of the square difference between co-ordinates of pair of objects:

$$Dist_{AB} = \sqrt{\sum_{k=1}^{m} (A_i - B_i)^2}$$

The Euclidean as a metric is used when the data is available in a raw form and is unprocessed. This metric is disadvantageous in situations where new objects keep on adding to the system, so the distance calculation method remains unaffected since newly added objects are treated as outliers. But the problem is the distance is affected in scale as the dimensions keep changing. So, two different cluster analysis attempts will give vague results.

2. *Cosine Distance:* The cosine distance is one less than the cosine of the included angle between two points. Here, the points are treated as vectors. Given an m-by-n data matrix, then the cosine distance between the vectors x_s and x_t are defined as follows:

$$D_{st} = 1 - \frac{X_s X_t'}{\sqrt{\left(x_s x_t'\right)\left(x_s x_t'\right)}}.$$

This method is a very common approach in design of collaborative filtering recommenders. The user-item matrix is treated as vector here. The cosine angle between two vectors of two different users is calculated using the cosine similarity formula illustrated above. It is used beneficially in non-parametric regression analysis. But it is not useful when the radius of the cluster is divergently large with dynamic values.

3. *Correlation Distance:* The correlation distance is the measure of dependence between two vectors. Given any two m-by-n data matrix, the correlation distance between two vectors A_s and A_t is calculated as follows:

$$D_{st} = 1 - \frac{\left(\mathrm{A}_s - \bar{\mathrm{A}}_s\right)\left(\mathrm{A}_t - \bar{\mathrm{A}}_t\right)'}{\sqrt{\left(A_s - \bar{A}_s\right)\left(A_s - \bar{A}_s\right)'}\sqrt{\left(A_t - \bar{A}_t\right)\left(A_t - \bar{A}_t\right)'}}$$

This distance matrix is used in scenarios when the genre of two users is partially known. Hence, the genre is calculated from correlation distance with the partly matching user as a predictive analysis method. This is the situation which normally occurs during a new user problem also known as the cold start problem.

4. *Manhattan Distance:* Manhattan distance is just an advancement of Euclidean distance in which the absolute differences between co-ordinates of object pairs are computed in Cartesian geometry:

$$D_{\text{st}} = |A_s - A_t| + |\ B_s - B_t\ |$$

5. *Chebyshev Distance:* It is also known as the maximum value distance. It is computed as the absolute magnitude of the differences between the co-ordinates of the objects. It is used as a metric to compute the multidimensional distance between two known-unknown pairs:

$$D_{\text{chebyshev}} = \max\ |A_i - B_i|$$

6. *Pearson Correlation Similarity:* The Pearson correlation coefficient is used to determine the strength and relationship between two variables. If there is a strong relationship between two variables, then the value is positive one. If there is an opposite relationship between two variables, then the value is negative one. If there is no relationship, then the value is zero:

$$P_{\mathbf{ST}} = \frac{\text{Covariance (S, T)}}{\delta_{\text{S}}\delta_{\text{T}}}$$

where 'δ_S' is the standard deviation of S and 'δ_T' is the standard deviation of T.

Figure 3 depicts the categorization of the variables used in the problem of missing value imputation. A qualitative and quantitative analysis is needful for correct identification of variable category since the orientation is to tune the propensity predictor attribute.

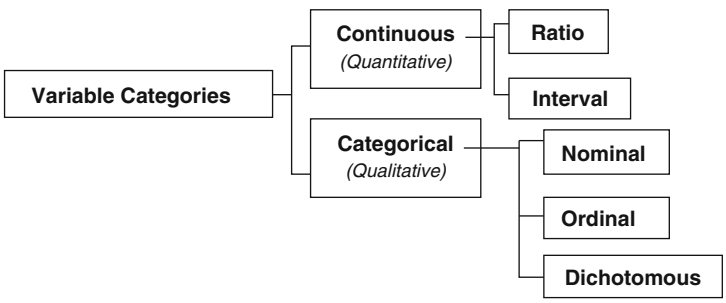


Fig. 3 Variable categorization

2 Literature Review

In real-life application scenarios like medical analysis, astronomical data, biochemical data and many more, data analysis is needful for research work. But data originating from sensor sources is usually incomplete or missing some values. This occurs due to the presence of some non-responsive items at the time of data collection. The data obtained this way is a set of information useful for analysis domain and gives biased results in different cases. Also, the proportion and patterns of missing values are one of the key factors for its usage as an analysis data. There are some attributes which provide information and some attributes which help in analysis over the dataset. The informant attributes are characterized as observed variables and the variables with missing values are termed as missing values. According to these characteristics, the data is grouped into three categories (missing at random) MAR, missing completely at random (MCAR) and not missing at random (NMAR) [8]. The three cases are dealt individually by treating the dataset as a partition of observed and unobserved variables. When the dataset has missing values in both the observed data section and the unobserved data section, then the missing value problem is identified as missing completely at random. The missing at random is identified as the case in which the occurrence of missing values is fully random and does not depend on the existence of observed data values. The stringent problem for quantitative research is determining the MNAR case wherein the probability of missing is fully dependent on the missing variables only [9–11]. Techniques to deal with imputation mostly use statistical inferences to determine observed variables. This causes MAR condition to pose in MCAR plausibly. Past researches have qualitatively emphasized some methods to check the presence of MCAR and MAR conditions. Little and Schenker (1995) designed a multivariate test to analyse MCAR. Also, Carpenter and Goldstein (2004), Horton and Kleinman (2007) and White et al. (2011) proposed that it is not possible to determine existence of MAR conditions based merely on the existence of observed data. But Diggle et al. (1995) and Tabachnick and Fidell (2012) presented that the chances of occurrence can be hypothesized by a t-test on complete data versus missing data. In past decades [11–29], the missing value identification involved methods to delete the missing data like list-wise deletion and pairwise deletion but the APA task force (Wilkinson and the Task-Force on Statistical Inference 1999) posed a restriction in usage of these methods as they bias the dataset completely.

3 Headway in Statistical Imputation Approaches

Imputation techniques aim to focus on quality and correctness of imputation. This depends on the values acquired from observed variables. Also, the key perspective in accurate imputation is in deciding which variables to treat as indicator variables. The filling of missing values is dealt through targeting the observed

variables and learning their characteristics. The developments in the field of data analysis presented plenty of methods to deal with statistical imputation [24, 25, 28]. In data mining, the data is identified by its features into various types like univariate, bivariate, multivariate, ordinal, etc. The field of statistics provides various techniques to find values of unknown variables based on their types. Some of the methods used for statistical inference of observed variables to target statistical missing value imputation methods as a review have been discussed.

Statistical inference is difficult to work in situation wherein the indicator variable is bivariate in nature and values are missing in pairs. But methods like mean and mode imputation and covariance-based imputation have been useful over univariate data. In [11], k-ranked covariance is used to estimate missing values. The absolute diagonal covariances are calculated over non-missing value samples and are ranked. The imputation values are estimated from the set of ranked covariances. The method outperforms over KNN clustering methods. On strict mathematical backgrounds, random fields have proven over stochastic process with discrete set of points spatially correlated and are estimated using function valued random variables. Single dimensional Gaussian process with probability density function is used as an autocorrelation function through a Fourier transformation on power values of predictive variables. In [9], a pairwise Gaussian random field has been researched and evaluated to perform linear inference-based imputation. It uses a pair of strongly related variables in Gaussian field of random. The method produces results of imputation during learning as compared to Gaussian Markov random field.

4 Requisites for Imputation Reliability

Estimating relationship amongst all attributes in the dataset is one of the key factors improving the reliability of imputation methods. Myriad approaches use mean- and mode-based statistics for imputing the target amongst the cluster of tuples on same attribute. Some methods advocate using all possible value APV-based imputation. The method works but at the stake of increased space-time complexity. Most of the imputation algorithms use regression calculation amongst assumption attributes. The regression-based approaches work only by finding linear statistical relationship [24, 25, 28] for predicting the missing value. In this case, a specific missing data can be imputed predicatively but if used on diverse set of tuples having unknowns on same attribute will strongly bias the dataset. Some of the techniques like hot deck and cold deck imputation seemingly use mean/mode imputation but they differ from the other cases that they use current observed variables and survey variables. Past researches exhibit that only few methods exist to deal with missing value problem. One of the major challenges is imputing on character dataset. Data scientists have recently advocated data predictors for imputing in character datasets. Some of the properties an exact imputer should possess have been briefly projected below.

4.1 Property-1. Imputation Fairness

The dataset should have statistically balanced records over various primary attributes in the dataset. The stoichiometric analysis of each data slice should give 1–1 ratio of data volume and attributable magnitude post-imputation.

4.2 Property-2. Composition Ratio Should Be Relative Ratio of Observed Set

The distribution of missing values should be estimated from a very large sample of dataset. This property addresses the issue of missing ratio per record and per variable. Multiple imputation methods when incorporated over sparse variable can bias a certain quantity of records beholding that variable. The biasing of imputation can sometimes be effective/advantageous or defective/hazardous spoiling the proximity of overall analysis dependent on such dataset. This usually occurs when the MCAR assumption is violated, consequently causing biased estimates in the resulting dataset.

The key properties [12–21] of the dataset include:

1. The centre of the data
2. The spread amongst the spread of the data
3. The skewness of the data
4. The probability distribution of the data trails
5. The correlation amongst the elements of the dataset
6. Whether or not the elements of the data are constant over time
7. The presence of outliers in the data

An exploratory data analysis is useful pre-imputation and post-imputation to match the minimum composition factor needed for hypothesis testing. The means of the two populations, the one obtained pre-imputation and the one obtained post-imputation in dataset, forms the basis of inferential analytics.

Due to this, it is very essential to precalculate the composition ratio of missing patterns, breadthwise and depth-wise. The theoretical underpinnings of these properties have been statistically proven in the following mathematical foundation [16]. A few of the well-known attempts to deal with missing data include: hot deck and cold deck imputation, list-wise and pairwise deletion, mean imputation, regression imputation, last observation carried forward, stochastic imputation and multiple imputation.

In simple stochastic imputation, instead of replacing a value with a mean, a random draw is made from some suitable distribution, provided the distribution is chosen appropriately; consistent estimators can be obtained from methods that would work with the whole dataset.

This is very important in the large survey setting where draws are made from units with complete data that are ‘similar’ to the one with missing values (donors). There are many variations on this hot deck approach. Implicitly they use non-parametric estimates of the distribution of the missing data: they typically need very large samples.

Although the resulting estimators can behave well, for precision (and inference) account must be taken of the source of the imputations (i.e. there is no ‘extra’ data). This implies that the usual complete data estimators of precision cannot be used. Thus, for each particular class of estimator (e.g. mean, ratio, percentile) each type of imputation has an associated variance estimator that may be design based (i.e. using the sampling structure of the survey) or model based or model assisted (i.e. using some additional modelling assumptions). These variance estimators can be very complicated and are not convenient for generalization.

The two processes are equivalent [6]. The former process was advantageous in the past when only tables of test statistics at common probability thresholds were available. It allowed a decision to be made without the calculation of a probability. It was adequate for classwork and for operational use, but it was deficient for reporting results.

The latter process relied on extensive tables since the computational support was not always available. The explicit calculation of a probability is useful for reporting. The calculations were trivially performed with contemporary softwares based on formula-based worksheet computations [8–11, 30].

The difference in the two processes applied to the radioactive suitcase example below:

- The Geiger counter reading is 10. The limit is 9. Check the suitcase.
- The Geiger counter reading is high. 97% of safe suitcases have lower readings. The limit is 95%. Check the suitcase.

The former report is adequate but the latter gives a more detailed explanation of the data and the reason why the suitcase is being checked. It is important to note the difference between accepting the null hypothesis and simply failing to reject it. The *fail to reject* terminology highlights the fact that the null hypothesis is assumed to be true from the start of the test; if there is a lack of evidence against it, it simply continues to be assumed true. The phrase *accepts the null hypothesis* may suggest it has been proved simply because it has not been disproved, a logical fallacy known as the argument from ignorance. Unless a test with particularly high power is used, the idea of *accepting* the null hypothesis may be dangerous. Nonetheless the terminology is prevalent throughout statistics, where the meaning actually intended is well understood.

5 Comprehensive Analysis of Inferential Algorithms

Data analysis field had been using vivid approaches for data imputation over decades. The former approaches used include expectation-maximization and full information maximum likelihood to deal with missing data of various categories especially in MCAR type of missing value datasets. The expectation-maximization is a three-step method in which the latent variables are identified first using conditional distributional mathematics and then the expected value is computed using a log likelihood function. In [31], a decision tree-based approach has been combined with expectation-maximization method. The prediction tree with missing value is constructed using binary classification from all observations. Then the missing values are computed using linear regression tree pruning. The empirical results prove that the method is workable for all three categories of data, namely, MAR, MCAR and NM. A state-of-the-art missing value imputation techniques have been discussed in Appendix.

6 Algorithmic Formalism

I. *Probing the Missing Pattern Using Inclusive Analysis Strategy:*

1. Analyse the pattern of missing using a look-up propeller to tune the regressor.
2. Search the data frame to find predictor variable.
3. Allocate role as a predictor and direction to propel, forming a propeller vector.
4. Specify a lower bound as a minimum allowable limit for imputation.
5. Specify an upper bound as a maximum allowable boundary of imputation.
6. Find the distribution of variables.
7. Compute variance in between time series data frames in every iteration.

II. *Reconstruction:*

1. Compute variance and validate for the boundary condition.
2. Compute total variance and its matching propeller in the heterogeneous cluster.
3. Compute relative increase in variance, if any to jump to next cluster in the set of heterogeneous clusters.

III. *Recompilation:*

1. Perform relative efficiency to gauge the amount of missing information as a ratio of number of imputations performed/number of imputations not computed.
2. Perform repetitive iteration until all clusters are done.

A good rule of thumb is to have the number imputations, at least equal the highest FMI percentage as being proposed in the model.

7 Problem Modelling

7.1 Score Prediction

The traditional methods for missing value imputation techniques were useful for data analysis and surveillance. But recently, these techniques have proven their worth on platforms like score prediction, market analysis, flight-schedule computing, result prediction, weather forecasting, astronomical numerology and satellite data imaging. Many researchers in this field have led a plethora in development of techniques of imputation workable on diverse platforms. But authors in [8, 9, 30, 32, 33] proposed a novel spatial imputation technique on satellite data wherein an image reflects spatial data. Spatial imputation technique based on machine learning algorithms has been evaluated to recover original image from partially filled image. Computer scientists also propound usage of imputation techniques on trash recovery and storage failure to reconstruct the data from partly known values [10, 11] and researched a novel technique of imputation for trash pickup logistic management based on iterative K-NN. The authors exploited grey relational grade analysis [17–21] for distance computing between the predictor variable and target value.

Let us treat the records in the missing value dataset as missing value tuple and relate them to observed value tuples. Then try to find out the parametric description of values under associator attributes, predictor attributes and auxiliary attributes. Taking the problem to mathematical formalisms, let us model the problem as a linear inequality. Then generate a factor tuple from a propensity pointer triple (associator, predictor, auxiliary). Pick the propensity matcher triple from the missing value record.

Finding a parametric description of the solution set of a linear system is the same as solving the system. The missing value imputation is treated as the linear inequality solving problem here. AI technique of using associator, predictor and auxiliary variable as the meta-heuristic for producing plausibility tuple is followed next.

If A←B means A is dependent on B, then B is given the role as a predictor. To find this, we need some statistics which proves A is dependent on B. This is computed using simultaneous equation solver. The set of dependent variables is related with outcome variable.

Let the factors of this tuple be denoted by f_1, f_2, $f_3 \ldots f_n$ on similar cluster analysis. Later on, in the imputation cycle, if T_1, T_2, $T_3, \ldots T_n$ is the target tuple, then we simulate the vector relationship of the target tuple in picture as Fig. 4 illustrates an ideology of missing columnar predictor attributes versus missing of attributes in row. The aim is to portray the problem domain in finding missing values

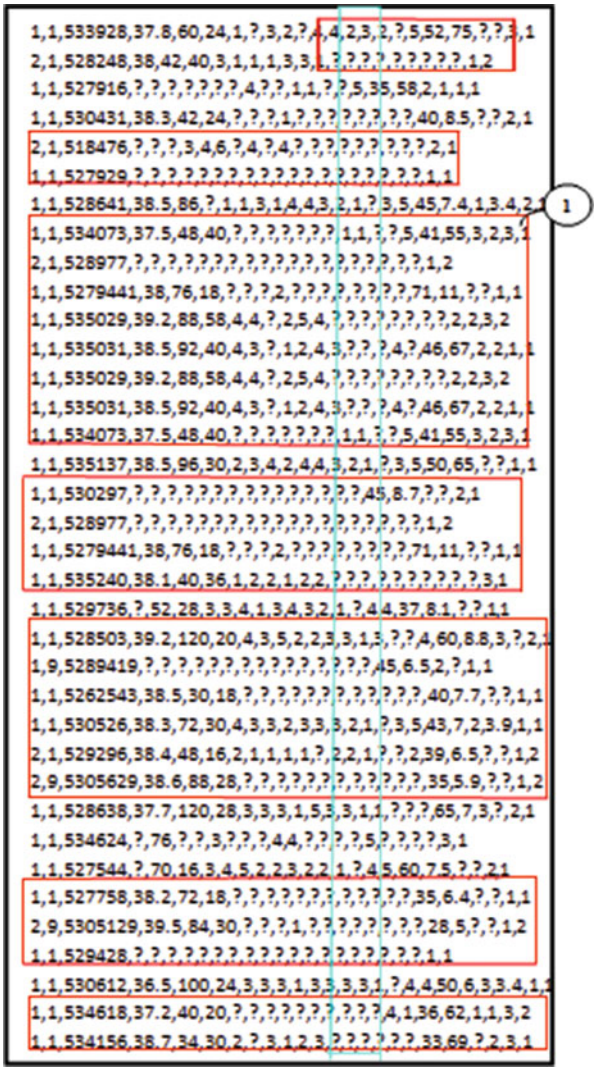


Fig. 4 Non-homogeneous pattern of missing

if the data frames to be clubbed belong to multi-source heterogeneous foundations as illustrated in Fig. 4, showing non-homogeneous pattern of missing:

$$T_1.f_1, T_2.f_2, T_3.f_{3.....}T_n.f_n$$

Thus, a system of regression equation is formulated to address the problem in the above example using similar cluster analysis.

8 Missing Value Imputation Using Similar Cluster Analysis

A method to imputation of continuous variables of missing at random using the method of simulated scores1. The system of regression equations has, as dependent variable for each equation, the variable to be 'imputed if missing' and has on the right-hand side all the other variables:

$$Y_1 = X\gamma 11 + Y2\gamma 12 + Y3\gamma 13 + \cdots + Yp\gamma 1p + \varepsilon_1 \quad (1)$$

$$Y_2 = X\gamma 21 + Y1\gamma 22 + Y3\gamma 23 + \cdots + Yp\gamma 2p + \varepsilon_2 \quad (2)$$

$$Yp = X\gamma p1 + Y1\gamma p2 + Y2\gamma p3 + \cdots + Yp - 1\gamma pp + \varepsilon_p \quad (3)$$

The $\gamma 11$, $\gamma 21$, ..., $\gamma p1$ are scalars or ($k \times 1$) vectors depending on X being a single column or a ($n \times k$) matrix, while all the other γ-s are scalars and the ε-s have a cross-equation multivariate normal distribution. Equation (1) represents a system of simultaneous equations in structural form. The jointly dependent variables Y appear also on the right-hand side of the equations, while the variables X play the role of 'exogenous' variables. Such a system is obviously under identified, as it violates the order condition for identification 4 (e.g. Greene, 2000, Sec. 16.3.1): infinite sets of γ-values would be observationally equivalent. It is therefore useless (or impossible) to apply estimation techniques suitable for simultaneous equation systems, like two- or three-stage least squares, full information maximum likelihood, etc. The implications and consequences of the two propositions, proofs and a more detailed discussion can be found in Calzolari and Neri (2002). Working on the structural form system (1) has the advantage of being computationally simple as well as rather intuitive. The discussion on convergence of the iterated imputations with fixed parameters (Sect. 2) and proposition 1 ensure that we can get exactly the same results if we work directly on the reduced form, estimating its parameters directly by (01s) and using such an estimated reduced form for imputation. However, even if the estimation phase would be simple (even simpler than for (1)), the imputation phase would be much more complex. For each pattern of missing data we should, in fact, specify the appropriate imputation function, with pseudorandom errors that should be conditional on the Y-s observed in that pattern. Since there are 2p possibly different patterns of missings, the technical solution would be very hard. Also, there would be no substantial simplifications over the exact maximum likelihood approach, where up to 2p different conditional densities should be specified, according to which Y-s are observed in each pattern. That's why it is preferable to work, in practice, with the structural form [34–39].

So we have a first set of completed values for Y1 and we attach it as an additional column to X. We then regress the next variable with fewest missing values against X and the completed Y1 and use the Ols estimated coefficients and variance for an imputation step that completes Y2. Going on, the first round ends when all the missing values are completed. As the Srmi's model [6, 40] authors put in evidence, the updating of the right-hand side variables after imputing the missing values depends on the order in which we select the variables for imputation. Thus, the imputed values for Yj involve only (X, Y1 ... Yj − 1), but not Yj + 1 ... Yp. For this reason the procedure continues to overwrite the imputations for the missing values iteratively. In any iteration after the first round, we always have complete data for all variables, part of which are observed and the other part have been imputed in the previous iteration. Nevertheless we can estimate each equation separately by Ols as in Srmi approach. After coefficients have been estimated by Ols, we compute from residuals the estimate of the (p × p) variance-covariance matrix, say Ψb, differently from the Srmi method; the Cholesky decomposition of the matrix Ψb to produce vectors of pseudorandom numbers for imputation is used, thus considering also covariances besides variances. When a value of Y1 is missing, we impute the value obtained from the right-hand side of the first equation in (1) where: the γ - s are at the previous iteration estimated value, the value(s) of X is (are) observed, the values of the Y on the right-hand side are in any case complete (some of them are observed; the others have been imputed in the previous iteration) and the value of ε1 is 'jointly' produced with ε2, ..., with εp by the pseudorandom generator with a cross-equation variance-covariance matrix equal to the last estimated Ψb. The same is done for the second equation in (1), filling missing values of Y2 and so on. Repeated cycles continue until convergence on the estimated parameters has been achieved. This is illustrated completely in the flowchart (Fig. 5).

9 Problem Statement

The missing value imputation is an uncertainty modelling problem in computational intelligence. Identifying similar records in different data slices can indicate a link across the database which helps facilitate merging heterogeneous record structures.

9.1 Objective

To fill missing values intelligently so that the dataset remains equitable and gives correct result for predictions.

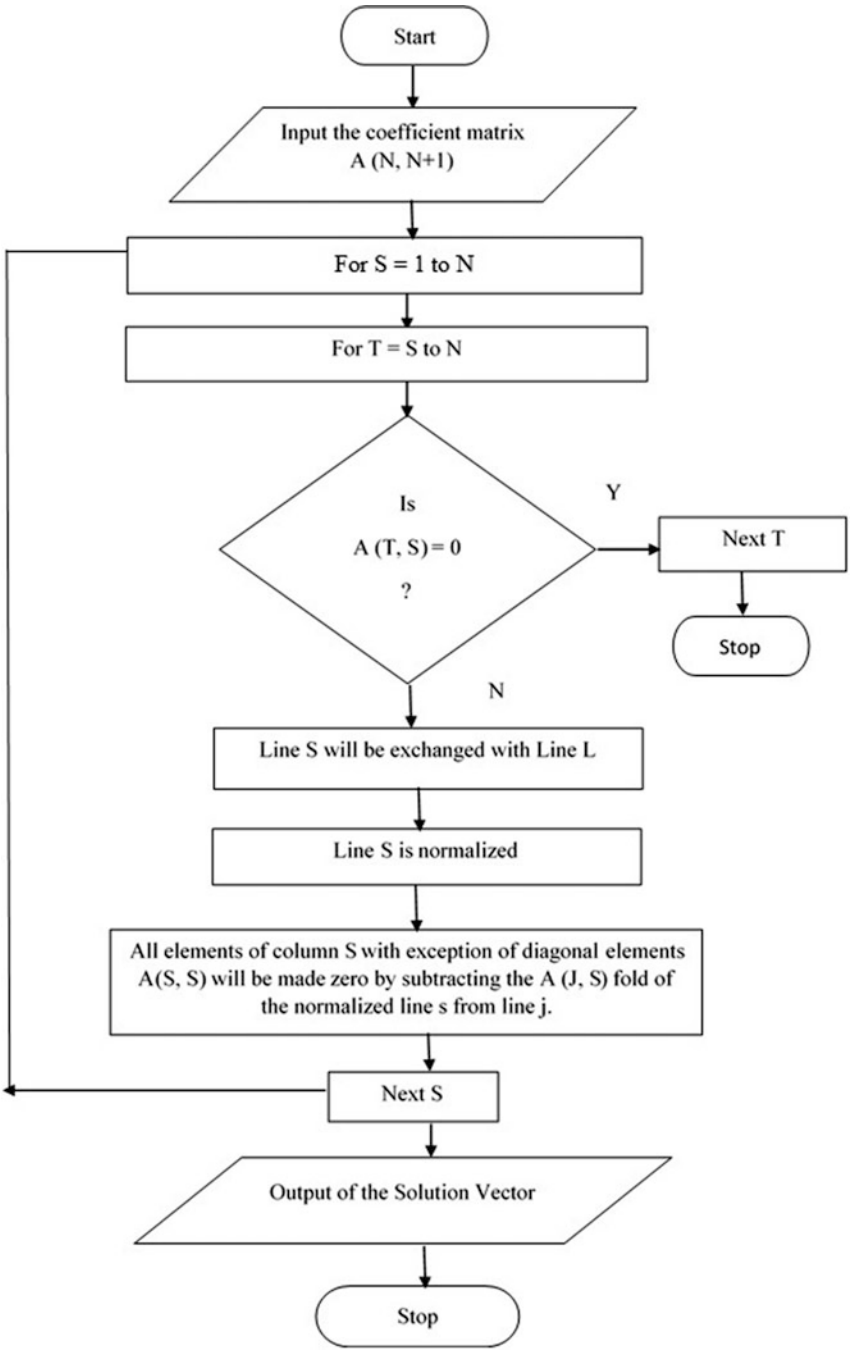


Fig. 5 Flowchart for factor analysis

9.2 Part A: Data Engineering

Imputing with mean, mode and median diminishes any correlations involving the variable(s) that are imputed. This is because we assume that there are no relationships between the imputed variable and the measured variable.

According to the rules of data analysis, the properties for univariate analysis should be reflected in multivariate analysis. In totality, the following measures should be incorporated to fill the missing values:

1. Identify the nature of the dataset.
2. The cause of the missing values plays a major role in deciding whether the missing values can be deleted or filled with mean, median or mode in the near-far data frame, so identifying the cause and pattern of missing is the crucial beginning.

9.3 Part B: Feature Engineering

The conditional attributes are demolished in a typical dataset having combinations of multiple features. A concise feature engineering should expound the quintessential features. Some of the primary categories in a weather attribute can be sunny, rainy, windy, cloudy and snowy. Similarly, the genre of a video game can be elucidated using adverbs like action, adventure, fighting, puzzle, racing, role playing, shooter, simulation, sports and strategy. Feature engineering should traverse through the values of the conditional attributes using tree-based model and calculate entropy of each conditional attribute to acknowledge the homogeneity for choosing the predictor variable.

9.4 Part C: State-of-the-Art Imputation Techniques

Some of the commonly used imputation techniques are KNN[37] imputation for categorical data. Bayes' rule-based theorem is used for imputation in multivariate longitudinal data structure. It uses inference-based system to fill missing values using Rubin's rule[14, 15, 28] on functionalism of posterior distribution. Bayesian's multiple imputation uses Markov chain Monte Carlo algorithm to iteratively draw samples from the parameter vector. The missing pattern in most of the observations is unknown so posterior predictive distribution cannot be simulated directly. So, the Markov chain Monte Carlo algorithm uses Gibbs sampling algorithm to draw missing value starting from the initial vector and deducing iterative convergence to create stochastic sequences. It covers imputation uncertainty but with a periodically lagged convergence for larger iterations.

Composition factor is calculated using divergence and similarity as the two metrics. To analyse the disparity of monotone and non-monotone missing data frame, standard deviation is calculated first. The dataset whose deviation is smaller will have more balanced distribution. To estimate the similarity between two data frames, cosine similarity is calculated. Constituting divergence 'D' as the lower bound (D, $\leq$) in the data frame and similarity 'S' as the upper bound (S, $\geq$), the composition scale is postulated as follows:

(i) *Weighted Approximation # 1*

The lower bound parameter divergence 'D' is determined by:

$$(\boldsymbol{Df}) = \sqrt{\frac{\mathbf{1}}{\boldsymbol{n-1}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{a_{i-}}\overline{\boldsymbol{a}})^{\mathbf{2}}}$$

(ii) *Weighted Approximation # 2*

The upper bound parameter similarity 'S' is determined by:

$$\mathbf{Sim}\,(\boldsymbol{a_i}, \boldsymbol{a_{i=1}}) = \frac{\sum_{\boldsymbol{k}}^{\boldsymbol{n}} \boldsymbol{a_{ik}}_{\cdot}\ \boldsymbol{a_{i+1k}}}{\sqrt{\sum_{\boldsymbol{k}}^{\boldsymbol{n}} \boldsymbol{a_k^2}} \cdot \sqrt{\sum_{\boldsymbol{k}}^{\boldsymbol{n}} \boldsymbol{a_{i+1}^2}}}$$

wherein 'a' is the conditional attribute in the data frame of the data slice under consideration and 'i' is the index of association from infimum to supremum.

(iii) *Weighted Approximation # 3*

The composition factor of the conditional attribute in the targeted data frame can be determined as follows:

$$\%C_a = \frac{\boldsymbol{g}^a}{\boldsymbol{g}^t} \times 100$$

where $\boldsymbol{g}^a$ represents gravity of the attribute 'a' in the dataset and $\boldsymbol{g}^t$ represents the gravity of total attributes in the dataset.

The approximation # 1 and approximation # 2 are used to partition the dataset into data slices during the stoichiometric analysis of large voluminous disparate database. The approximations # 1 and 2 are assigned weights as per the indicator coefficients obtained during global statistical analysis (GST) of the database. The weighted approximations # 1 and 2 are seriously important when the dataset is a complete case (CC) having missing values completely at random. The weighted approximation # 3 is an intuitive factor to overcome imputation uncertainty so that predictive distributions are deductible. The data slicing operation can be effectively performed using the above framed approximations to spot the patches of monotone missing and non-monotonous missing.

9.5 Part D: Filling Strategy

The insightful analysis of state-of-the-art methods deduces the following strategies for filling the missing values along with best suit algorithm to deal with varied types of datasets.

Strategy #1
Most of the statistical approaches in data science can be used starting from the most fundamental one like:

(i) Iterate to impute.
(ii) Use a predetermined formula to fill the missing values.
(iii) Filling missing values using Yate's method.
(iv) Determine a global value in-far to impute.
(v) Calculate a class-wise average value to impute.

Strategy # 2
Inference-based filling can be used especially when most of the records in the data slice belong to complete case (CC) with optimistic approach not to discard any missing record and avoid data loss compensations. The following data mining algorithms can be applied to infer the values to complete the incomplete values.

(i) AANN – autoassociative neural networks
(ii) PCA-NN – principal component analysis neural network model
(iii) Genetic algorithm along with combination of decision trees
(iv) ANCOVA estimates of the residual sum of squares and covariance matrix

Strategy # 3
Predicting a missing value for best-fit approach uses machine learning algorithms to impute the missing. But machine learning algorithms destroy the original distribution of the dataset during several phases of imputation. Owing to the fact that some types of attributes demand machine learning algorithm for imputation, the following can be best used:

(i) C4.5 can be best used for conditional attributes with discrete and continuous values.
(ii) Grey-based KNN iterative imputation can be best used for categorical and continuous attributes.
(iii) Autoregressive integrated moving average model is used to predict the missing value from the relationship extracted through exploratory analysis.
(iv) Feedforward neural network uses historical past-to-future pairs to predict the missing value as a best-fit substitute.

Strategy # 4
Computational intelligence paradigms use measures of association as the propensity attribute for correlatively filling the missing values. The computational intelligence deploys neural networks, fuzzy systems and evolutionary computation techniques for deriving the substitute of missing values. It identifies relationships and patterns

amongst the feature attributes based on subject-object cases. The CI techniques overcome the prediction-based methods in which the consecutive series of missing data is not taken into consideration, hence giving unsatisfactory results. The CI methods use condition monitoring and incremental learning to intelligently fill the missing data. It uses three parameters vigilance, learning rate and choice to build a model for a requisite subject-object paired attribute and intelligently fills the values. The following are used in CI:

(i) Fuzzy adaptive resonance theory ARTMAP uses conditional monitoring and supervised learning.
(ii) MTBF-MTMF is a mean-time-between-failure to mean-time-to-failure mapping to deduce missing values, used to recover values from sensor data.

9.6 Part E: Missing Data Recovery

Substitution and missing data recovery are two different approaches to mitigate the missing value problem. Substituting a best-fit value is called missing value imputation in data science. It involves finding a value from the observed cases in the given database. But not all attributes can be filled with data discovered from the history. Some databases like the ones mostly used in disease diagnosis of a patient, weather forecasting, notary, tax calculation and finance sector are highly risky to go for the best-fit approach in imputation. Determining the complexity of this problem, the standardized way to categorize imputation techniques is as follows:

(i) *Recompile*
(ii) *Reconstruct*
(iii) *Exploratory filling*
(iv) *Opinion-based filling*

Missing data is grouped into two categories depending on the type of its origin, namely, missing data from known processes and missing data from unknown processes. The missing data from known processes is majorly due to error in data entry or sometimes due to failure on completing the questionnaire. The missing data from unknown processes is due to refusal of the respondent to fill highly sensitive quantities or unwillingness to disclose data for some attributes. This is also sometimes the cases where the respondent does not have enough knowledge to answer some of the attributes asked in the questionnaire.

Hence, recompilation can be done to infer missing values typically for the data originated from known processes because such missings are only due to error in filling missing values. Algorithm approaches to substitute values for this category of data cannot be beneficial to most of the extent of missingness. Similarly, value reconstruction can be used to complete the incomplete values through pattern discovery. Law of duality can be used to find majority population and recompile values for the leftover group as a subsidiary through flood fill.

Extensive approaches using deep learning can be used to apply exploratory fill for missing data originated for unknown sources typically as such data is missing due to lack of knowledge or difficulty to disclose the data. But exploratory data imputation for filling unknown values cannot be the best-fit. It can only be a comprised value and the dataset post-imputation can only give approximate near-far predictions. This is due to the fact that some values cannot be explored at all.

Opinion-based filling is a good substitute to feature attributes which have been left empty by the subject due to lack of knowledge. A majority opinion collection for subject-to-object case can be used as an experience collection and filling approach to complete the missing data and make the database suitable for predictions.

9.7 Part F: Mitigation of Missing Values

The missing value problem and filling with the best-fit can be mitigated using the following practical approaches:

- Some quantities can be filled based on the opinion collection approaches in deep learning.
- Some quantities can be inferred or deduced from prime attributes of the complete cases (CC).
- Some quantities can be recompiled and values can be reconstructed to compensate the data loss.
- Some quantities which are highly sensitive cannot be imputed because these can give negative or extremely biased results of predictions and classifications in the post-imputed dataset.

10 Methodology

The adaptive resonance theory is a type of neural network technique developed by Stephen Grossberg and Gail Carpenter in 1987. The significance of this technique is that it regulates newly learned information without discarding the old information. The adaptive resonance property is exploited to predict and fine-tune the missing value in the proposed methodology and a radial basis function approximation is put to fasten the clustering process.

The proposed algorithm is a fast fuzzy ARTMAP. It uses supervised adaptive resonance theory architecture with included fuzzy logic and factor-based weighted approximation. The proposed algorithm uses K-NN clustering wherein the values of ‘K’ are incrementally chosen using Fibonacci sequence as the weighted approximation. The increasing values of ‘K’ in every iteration aim to build a large Fibonacci spiral cluster using Fibonacci initial values and factor retracements of the probable neighbourhood. It is a nearest neighbour approach to iteratively acquire

new neighbour and add the previous set of factor retracements forming a spiral cluster of extensively large neighbourhood. The idea behind this is to overcome the dimensionality reduction problem for databases having many key attributes, each highly dependent on the other, and the input to each quantity is an aggregation of data from multiple sources. The Fibonacci radial basis function for layer-1 of fast fuzzy ARTMAP is shown below.

The relationship analysis is first performed on the dataset to find correlated subject-object pairs and accordingly the given dataset is divided into data slices. Each data slice is projected one after the other in the form of the scatter plot to observe inter-set and intra-set similarity. The data slice, as shown in Fig. 6, having conditional attributes with known subject-object pairs versus unknown subject-object pairs is projected to find partial mapping if it exists iteratively in all slices of the given database.

The centroid of the Fibonacci spiral is calculated from the Euclidean distance between the highest similarity data slice. Factor analysis is performed by framing the equations for correlations. Spearman's correlation coefficient is used to rank the values for ordinal variables and Pearson's correlation coefficient is used to evaluate the linear relationship between the continuous variable quantities. Keeping the factor value as the prime magnitude, the centroid is plotted and the value of 'K' is set based on the degree of similarity between the units in layer-1 and layer-2 in the radial basis function-based feedforward neural network. The layer-1 is the factor analysis field of the fast fuzzy ARTMAP. The layer-2 is the correlation analysis field of the fast fuzzy ARTMAP. The layer-3 is the Gaussian elimination to prune extraneous quantities and fine-tune the final predicted missing value.

The layer-2 is the angular distance amongst the two neighbourhoods. In this, the majority select for value fixing of 'K' is denominated using angular distance itself and not like contemporary K-NN where 'K' is chosen based on rank-based majority population. The factor value is the vigilance coefficient incorporated in this fast fuzzy ARTMAP. The learning rate is aligned to weighted majority scheme. A spiral trail of grouping is done based on heteroskedasticity values. The reason behind choosing heteroskedasticity value is to incorporate the choice factor for fast convergence in ARTMAP.

The missing value imputation in the proposed fast fuzzy ARTMAP is treated as the combination selection and regression problem. Here in the given expression, 'n' is the number of neighbours and 'r' is the number of data slices:

$$C_r^n = n(n-1)(n-2)\ldots\ldots(n-r-1)/r! \tag{4}$$

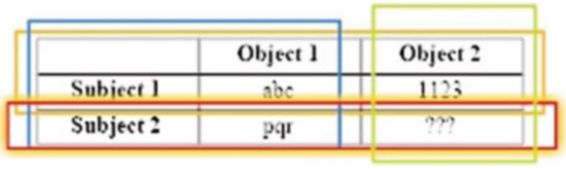


Fig. 6 Slicing for relationship analysis

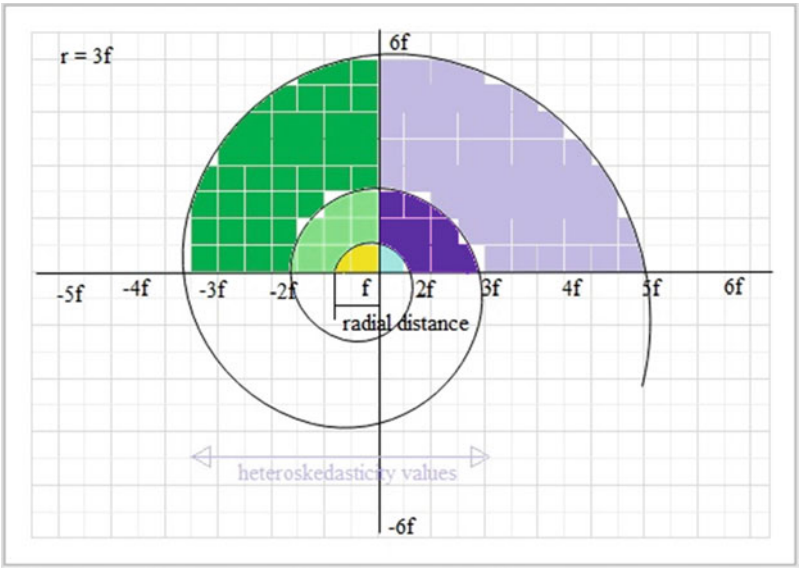


Fig. 7 Illustration of clustering using radial distance 'f'

$$C_r^n = \frac{n!}{r!\,(n-r)!} = \frac{P_r^n}{r!} \tag{5}$$

The statistical regression is used to determine the strength of relationship between the dependent quantities (Y); here 'Y' is the missing value to be computed and the other changing quantities in the ensemble in the combination function. The multiple regression incorporated to quantify 'Y' is given by:

$$Y = a + b_1x_1 + b_2x_2 + b_3x_3 + \ldots . b_1x_1 + u \tag{6}$$

where 'Y' is the quantity to be predicted and is the dependent variable:

$X_{i..n}$ is the variable that is used to predict 'Y' and is the independent variable.
'a' is the intercept, the mean of cumulative radial coefficient between the neighbours.
'b' is the slope, the distance between two hops in the neighbourhood obtained in the factor analysis phase in layer-1.

11 Approximation

The radial basis function is used as an approximation over fuzzy ARTMAP. The objective here is to save the time in exhaustive computation for finding the best-fit estimate. The learning ability of the ARTMAP is improved using intelligent

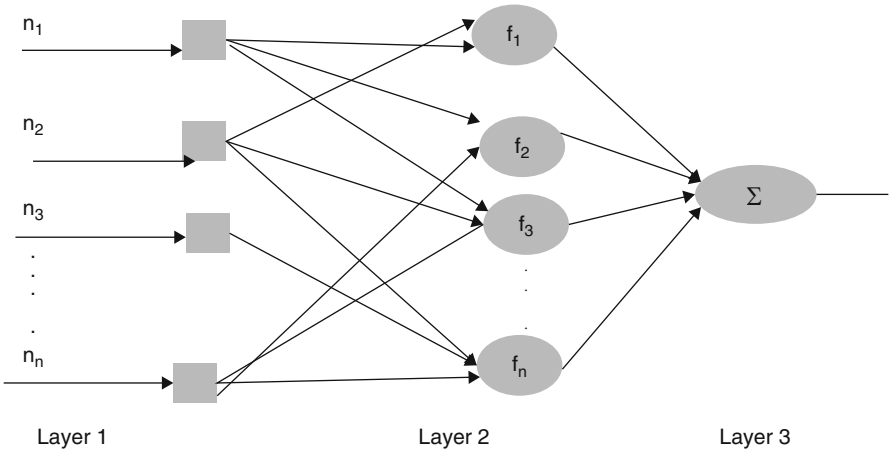


Fig. 8 Radial basis function 'f'

layers L1 and L2. Layer-3 is a core synthetic intelligence component which boosts knowledge engineering to build a trail of synthesized values from spiral chain of neighbourhood. A fuzzy set is a partially ordered set having infinite values, in contrast to crisp set which either belongs to a set or else not. Hence, convergence in predicting the missing values is time consuming and requires a lot of computational effort. The radial basis function along with layer-3 Gaussian elimination is used to impute the final prediction fast. This approximation is illustrated in Figure 7 wherein the radial basis function is used which is illustrated in Fig. 8. The following features of the fuzzy ARTMAP are exploited to postulate radial basis fast fuzzy ARTMAP:

(i) It has two layers for feedforward neural network. Layer-1 and layer-2 decide the value of 'K' in each iteration.
(ii) The layer-3 is hidden and performs Gaussian elimination for final value prediction. The Gaussian elimination is incorporated with its benign property of fast convergence.
(iii) The output nodes implement linear summation functions to produce output of multiple regression.
(iv) Layer-1 and layer-2 are the learning component of the ARTMAP.
(v) The feedforward networks incorporate vigilance factor analysis as the vigilance component and the subject-object pair selection as the choice component.

12 Proposed Algorithm

The proposed algorithmic formalism contributes distance computation algorithms to infer predictor attributes from heterogeneous sources of mixed data frames. The algorithm is based on balanced minimum evolution, which is the theoretical

underpinning of the principle of neighbour joining. The proposed algorithm improves underestimation problem induced due to inaccuracy of variable categorization by performing topologically moving using a pointer propeller as a look-ahead vector. The first implementation of the proposed algorithm includes similarity-based clustering using propensity propeller.

Algorithm 1: Fast Fuzzy Associative Resonance Theory Algorithm for missing data imputation using Radial basis Clustering Function

Input : Given DataSlice with missing values
Output : Imputed Missing Values
Read the input data
Perform Factor Analysis
Read the next input Slice
forall (records 1 → N) do
Categorize the data structure of the records
Perform Factor Analysis using the Spearson's Rank Correlation Coefficient 'ρ',

Compute Raw Score using Pearson correlation coefficient,

$$r_s = \rho_{r_{gx}, r_{gy}} = \frac{\mathrm{cov}(r_{gx}, r_{gy})}{\sigma_{r_{gx}} \sigma_{r_{gy}}}$$

where

P denotes the optimized Pearson coerrelation coefficient applied on rank subject-object pairs,
$\mathrm{Cov}(r_{g_x}, r_{g_y})$ is the covariance of the rank subject-object pairs,
$\sigma_{r_{gx}}$ and $\sigma_{r_{gy}}$ are the standard deviations of the rank variables subject-object pairs.

end
forall(dataslice [0][1] → dataslice[N][N])
← Initialize $P_{proximity} := r_S$

Calculate the equi-angular distance between consecutive radials using polar equation,

$$r = ae^{\theta \cot b}$$

Collect[i][r] = Collect[i][r]+ next
Sort(Collect[i][r])
Pick first K entries where K is current $P_{proximity}$
Fetch the labels of first K entries
if regression return Mean(K) entries using eq.3
else if classification return Mode(K) entries using eq.3
end

13 Simulation Results

The experiments are done using IBM's SPSS software. The radial basis function clustering is developed and evaluated on python using pandas, seaborn and TensorFlow using Scikit-learn. The movie lens 1B synthetic dataset is taken into consideration for evaluation process, as it is best suitable for homogeneity analysis. The users to rate different genres of movie belong to divergent platforms. The matching score between the target sample and the cluster, denoted as score (DS_i, C_j), is calculated using cosine similarity. Movie titles are downloaded from themoviedb.org. The userids are anonymized and are consistent between ratings.csv and tags.csv. The ratings are in between 0.5 star to 5.0 star. Same userid means uniformity in consistency in rating. A synthetic data slice has been peeled off from the original to test the composition of prediction values. This is illustrated in Table 1 and Table 2.

Table 1 Some examples of movie lens 1B synthetic dataset

Userid	Genre	Movieid	Rating
A3R5OBKS7OM2IR	Action	0179803	5
AH3QC2PC1VTGP	Adventure	0179211	4.5
A3LKP6WPMP9UKX	Comedy	0179999	4
AVIY68KEPQ5ZD	Fantasy	0179207	3.5
A1CV1WROP5KTTW	Horror	0179330	3
AP57WZ2X4G0AA	Mystery	0179437	2.5
A3NMBJ2LCRCATT	Sci-fi	0179778	2
A5Y15SAOMX6XA	Thriller	0179893	1.5
A3P671HJ32TCSF	(No genres listed)	0179260	1

Table 2 Experiment results of filling attribute rating's values

Supposing that the values of the attribute 'rating' are missing				Results of guessing the values of 'rating' based on RBFimpute			
Userid	Genre	Movieid	Rating	Userid	Genre	Movieid	Rating
A3R:IR	Action	0179803	5	A3R:IR	Action	0179803	4.89
AH3:GP	Adventure	0179211	4.5	AH3:GP	Adventure	0179211	4.16
A3L:KX	Comedy	0179999	4	A3L:KX	Comedy	0179999	4.23
AVI:ZD	Fantasy	0179207	3.5	AVI:ZD	Fantasy	0179207	3.45
A1C:TW	Horror	0179330	3	A1C:TW	Horror	0179330	2.67
AP5:AA	Mystery	0179437	2.5	AP5:AA	Mystery	0179437	2.04
A3N:TT	Sci-fi	0179778	2	A3N:TT	Sci-fi	0179778	1.89
A5Y:XA	Thriller	0179893	1.5	A5Y:XA	Thriller	0179893	1.56
A3P:SF	(No genres listed)	0179260	1	A3P:SF	(No genres listed)	0179260	0.94

14 Discussion and Future Works

The experimental results indicate that the fast fuzzy ARTMAP is beneficial for filling missing values and persistence of data composition. If the data slice is less than 50% incomplete, it cannot be used to guess the missing values. The RBFimpute function gives fast convergence towards the most comprehensive values as a substitute for the missings. It takes into account the holistic information by tracing through the high rank neighbourhood in the form of logarithmic spirals. The future work of the proposed RBFimpute will be a comparative research with well-known algorithms and its performance analysis on extremely large datasets with minimum 10 M size.

A.1 Appendix: Review of State-of-the-Art [41–46] Imputation Algorithms

Endogenous/ Exogenous Imputation Methods	Data Charac-teristics	Missing Pattern	Dataset Sparsity	Missing Value Estimation	
				Mathematical Model Used	Rate of Reliable Impu-tation(RRI)
Missing Value Imputation with lagged correlation	Multivariate	MAR, NMAR	Medium	Fourier Transform based Imputation for time series data	Slow convergence but approximate to nearly equal target value
Iterative KNN Imputation based on Grey Relational Analysis	Multivariate	MAR	Low	Co-relational Referential Sequences with Compared Observations	83% approximate to the target value
MVI using Reinforcement Programming	Univariate	MAR	Low	Value propagation using Probability for Exploration rate	Approximate to the target value with minor error.
Imputation using copula	Univariate	MAR	Low	Conditional Distribution using Gaussian Copula	Approximate to the target value with minor error.
Imputation using Decision Tree and Expectation Maximization	Multivariate	MAR	Low	Combined approach of Decision tree and expectation maximization	Slow convergence with nearly approximate imputation value.
Imputation using user preference genre and SVD	Grouped	MAR	High	Genre based KNN Clustering followed by Singular value based Matrix factorization	Lowest Mean Absolute error (MAE) over traditional approaches in imputation prediction
MVI using Cluster Based LLS method	General	MAR	Medium	Grouping similarity using regression analysis based on similar genes and Local Least Square method	Lowest ratio of Davies Bouldin index for varying datasets

(continued)

Endogenous/ Exogenous Imputation Methods	Data Charac-teristics	Missing Pattern	Dataset Sparsity	Missing Value Estimation	
				Mathematical Model Used	Rate of Reliable Impu-tation(RRI)
Cluster based KNN for microarray data	General	MAR	Medium	Grouping similar cluster based on genre as a parameter using K Nearest Neighbour clustering	Davies Bouldin Index measure shows sufficient ratio of clustering.
Index Measure Imputation based on Machine Learning	General	MAR	Medium	K-folds based decision tree classification followed by cross folds mean computation.	Significant range of imputation in interval 75 to 98%
MVI using Bayesian Network	Multivariate	MAR	Medium	Continuous variables grouped from Gaussian distribution are used for missing value imputation using posterior probability	Significant achievement of missing value imputation as compared to contemporary methods.

References

1. Astuti, T., & Nugroho, H. A., & Adji, T. B. (2016). The impact of different fold for cross validation of missing values imputation method on hepatitis dataset. In *14th International Conference on Quality in Research (QiR) 2015 – Conjunction with 4th Asian Symposium on Material & Processing (ASMP)2015 international conference of saving energy in refrigeration and air-conditioning ICSERA 2015* (pp. 51–55).
2. Pattanodom, M., Iam-On, N., & Boongoen, T. (2016). Clustering data with the presence of missing values by ensemble approach. In *2016 Second Asian conference on defence technology.* (pp. 151–156).
3. Rahman, S. A., Huang, Y., Claassen, J., & Kleinberg, S. (2015). Imputation of missing values in time series with lagged correlations. In *IEEE international conference on data mining workshops ICDMW* (Vol. 2015–January, no. January, pp. 753–762).
4. Zhu, M., & Cheng, X. (2015). Iterative KNN imputation based on GRA for missing values in TPLMS. In *ICCSNT 2015* (pp. 94–99).
5. Rachmawan, I. E. W., & Barakbah, A. R. (2015). Optimization of missing value imputation using reinforcement programming. In 2015 *International electronics symposium* (pp. 128–133).
6. Afrianti, Y. S., Indratno, S. W., & Pasaribu, U. S. (2015). Imputation algorithm based on copula for missing value in time series data. In *Proceedings of 2014 2nd international conference on technology, informatics, management, engineering and environment TIME-E 2014* (pp. 252–257).

7. Wu, S., Chang, C., & Lee, S. (2015). Time Series Forecasting with Missing Values. In *2015 1st International conference on industrial networks and intelligent systems* (pp. 151–156).
8. Dong, Y., & Peng, C.-Y. J. (2013). Principled missing data methods for researchers. *Springerplus, 2*(1), 222.
9. Cai, Z., Jermaine, C., Vagena, Z., Logothetis, D., & Perez, L. L. (2013). The pairwise Gaussian random field for high-dimensional data imputation. In *Proceedings of – IEEE 13th international conference on data mining ICDM*, pp. 61–70.
10. Honda, K., Nonoguchi, R., Notsu, A., & Ichihashi, H. (2011) *PCA-guided k-means clustering with incomplete data* (pp. 1710–1714).
11. Sehgal, M. S. B., Gondal, I., & Dooley, L. (2001). *K-ranked covariance based missing values estimation for microarray data classification* (pp. 274–279). Kitakyushu: Fourth International Conference on Hybrid Intelligent Systems.
12. Blake, C., & Merz, C. (1998). *UCI repository of machine learning databases*.
13. Caruana, R. (2001, January). A non-parametric EM-style algorithm for imputing missing value. In *Artificial intelligence and statistics*.
14. Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: Wiley.
15. Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B, 39*, 1–38.
16. Zhang, S. C., et al. Optimized parameters for missing data imputation. In Q. Yang & G. Webb (Eds.), *PRICAI 2006. LNCS (LNAI)* (Vol. 4099, pp. 1010–1016). Heidelberg: Springer.
17. Huang, C.C., Lee, H.M. (2002, September) An instance-based learning approach based on grey relational structure. In: Proc. of the UK workshop on computational intelligence (UKCI-02), Birmingham.
18. Lakshminarayan, K., et al. (1999). Imputation of missing data in industrial databases. *Applied Intelligence, 11*, 259–275.
19. Brown, M. L. (2003). Data mining and the impact of missing data. *Industrial Management & Data Systems, 103*(8), 611–621.
20. Lichman, M. (2013). *UCI machine learning repository*. Irvine: University of California, School of Information and Computer Science.
21. Saar-Tsechansky, M., Provost, F., & Caruana, R. (2007). Handling missing values when applying classification models. *Journal of Machine Learning Research, 8*, 1217–1250.
22. Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Sieb, C., Thiel, K., & Wiswedel, B. (2007). KNIME: The Konstanz information miner. In *Studies in classification, data analysis, and knowledge organization (GfKL 2007)*. Berlin: Springer.
23. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *SIGKDD Explorations, 11*(1), 10–18.
24. R Core Team. (2014). *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing.
25. Honaker, J., King, G., & Blackwell, M. (2011). Amelia II: A program for missing data. *Journal of Statistical Software, 45*(7), 1–47.
26. Orczyk, T., Porwik, P., & Bernas, M. (2014). Medical diagnosis support system based on the Ensemble of Single-Parameter Classifiers. *Journal of Medical Informatics and Technologies, 23*, 173–180.
27. Wozniak, M., & Krawczyk, B. (2012). Combined classifier based on feature space partitioning. *International Journal of Applied Mathematics and Computer Science, 22*(4), 855–866.
28. Little, R. J. A., & Rubin, D. B. (1987). *Statistical analysis with missing data*. New York: Wiley.
29. Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. Boca Raton: Chapman and Hall/CRC.
30. Vateekul, P., & Sarinnapakorn, K. (2009). Tree-based approach to missing data imputation. In *ICDM IEEE international conference on data mining workshops* (pp. 70–75).
31. Rahman, G., & Islam, Z. (2014). *iDM I: A novel technique for missing value imputation using a decision tree and expectation-maximization algorithm*, no. March (pp. 8–10).

32. Chen, M. H. (2010). Pattern recognition of business failure by auto-associative neural networks in considering the missing values. In *ICS 2010 – International computer symposium* (pp. 711–715).
33. Wang, Z. H. (2010). Numeric missing value's hot deck imputation based on cloud model and association rules. In *2nd International Workshop on Education Technology and Computer Science (ETCS)* 2010 (Vol. 1, pp. 238–241).
34. Insuwan, W., Suksawatchon, U., & Suksawatchon, J. (2014). Improving missing values imputation in collaborative filtering with user-preference genre and singular value decomposition. In *2014 6th international conference on Knowledge and Smart Technology (KST)* (pp. 87–92).
35. Houari, R., Bounceur, A., Tari, A. K., & Kecha, M. T. (2014). Handling missing data problems with sampling methods. In *2014 International conference on advanced networking distributed systems and applications* (pp. 99–104).
36. Keerin, P., Kurutach, W., & Boongoen, T. (2013). An improvement of missing value imputation in DNA microarray data using cluster-based LLS method. In *2013 13th international symposium on communications and information technologies* (pp. 559–564).
37. Keerin, P., Kurutach, W., & Boongoen, T. (2012). Cluster-based KNN missing value imputation for DNA microarray data. In *IEEE international conference on systems, man, and cybernetics* (pp. 445–450).
38. Madhu, G., & Rajinikanth, T. V. (2012). A novel index measure imputation algorithm for missing data values: A machine learning approach. In *2012 IEEE international conference on computational intelligence and computing research*.
39. Ohashi, O., & Torgo, L. (2012). Spatial interpolation using multiple regression. In *2012 IEEE 12th international conference on data mining* (pp. 1044–1049).
40. Miyakashi, Y., & Kato, S. (2011). Missing value imputation method using Bayesian network for decision-making on HCR. In *Distribution* (pp. 379–384).
41. Ozkan, H., Pelvan, O. S., & Kozat, S. S. (2015). Data imputation through the identification of local anomalies. *IEEE Transactions on Neural Networks and Learning Systems, 26*(10), 2381–2395.
42. Sridevi, S., Rajaram, S., Parthiban, C., SibiArasan, S., & Swadhikar, C. (2011). Imputation for the analysis of missing values and prediction of time series data. In *2011 International conference on recent trends in information technology* (pp. 1158–1163).
43. Porwik, P., Sosnowski, M., Wesolowski, T., & Wrobel, K. (2011). A computational assessment of a blood vessel's compliance: A procedure based on computed tomography coronary angiography. In E. Corchado, M. Kurzyński, & M. Woźniak (Eds.), *HAIS 2011, Part I. LNCS* (Vol. 6678, pp. 428–435). Heidelberg: Springer.
44. Doroz, R., & Porwik, P. (2011). Handwritten signature recognition with adaptive selection of behavioral features. In N. Chaki & A. Cortesi (Eds.), *CISIM 2011. CCIS* (Vol. 245, pp. 128–136). Heidelberg: Springer.
45. Foster, K. R., Koprowski, R., & Skufca, J. D. (2014). Machine learning, medical diagnosis, and biomedical engineering research – Commentary. *Biomedical Engineering Online, 13*, 94.
46. Bernas, M., Orczyk, T., & Porwik, P. (2015). Fusion of granular computing and k –NN classifiers for medical data support system. In N. T. Nguyen, B. Trawiński, & R. Kosala (Eds.), *ACIIDS 2015. LNCS* (Vol. 9012, pp. 62–71). Heidelberg: Springer.

A Scientific Perspective on Big Data in Earth Observation

Corina Vaduva, Michele Iapaolo, and Mihai Datcu

1 Introduction

The Earth Observation (EO) community is growing year after year, reaching the point where its needs and challenges became part of our everyday activity, no matter the application domain and industry. With a limited contribution in the early 1960s, remote sensing technologies bring now a great impact in almost all economic sectors, including security, land surveying, agriculture, forestry, mining, urbanism or even social media. The space industry turned from a list of few players to a diverse and expanding ecosystem.

The progress has been possible with the support and commitment of international institutions and agencies. There were many initiatives to provide funding through dedicated programmes in order to push for technological breakthroughs and generate new sets of data and measurements capable to create insights on the Earth. About hundreds of space-borne missions and more than 2000 space-borne sensors

C. Vaduva (✉)
Research Center for Spatial Information, University "Politehnica" of Bucharest (UPB-CEOSpaceTech), Bucharest, Romania

M. Iapaolo
European Space Agency Centre for Earth Observation (ESA-ESRIN), Frascati, Italy
e-mail: michele.iapaolo@esa.int

M. Datcu
Research Center for Spatial Information, University "Politehnica" of Bucharest (UPB-CEOSpaceTech), Bucharest, Romania

Remote Sensing Technology Institute, German Aerospace Center (DLR), Oberpfaffenhofen, Germany
e-mail: mihai.datcu@dlr.de

© Springer Nature Switzerland AG 2020

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_8

were developed over the past decades, bringing the EO community at the edge of a new era, that of Big Data [1].

The new possibilities that arise in the EO domain are widely amplified by the technological evolution in the non-space field. EO data can easily be harmonized with smart sensor data, UAV imagery, ground-based measurements and crowdsourcing knowledge and amplified through artificial intelligence and cloud computing. This fact unlocks a multidimensional stack of diverse and correlated information, allowing scientists to develop an unprecedented understanding on the Earth physical parameters and their evolution over time. This great amount of unexplored data stimulates a sharp development both in science and technology.

In order to speed up science, open data sharing is encouraged at international level (e.g. the G8 Open Data Charter or the Open Science Earth Observation call). A wider use of open, discoverable and shareable data is promoted worldwide to increase societal and economic benefits [2]. With respect to satellite data, NASA made public Landsat data corresponding to more than 40 years of acquisition, while the European Commission made the Sentinel constellation available free of charge.

Data openness increases the value of Big Data in EO field: first, because it is not at hand to acquire (just a limited number of institution afford it, in terms of knowledge and capabilities) and, second, due to the uniqueness of data capturing physical features at a great scale. In order to prepare for the impending initiatives and be able to exploit their advantages, one must fully understand the main characteristics, the "Vs" of Big Data. Definitely, EO archives now host zettabytes of data (a huge *volume*) that are generated daily or even faster (high *velocity*) by a wide range of sensors (increased data *variety*). The *value* of information contained in – and that can be potentially extracted from – such data quantity is immense, although its *veracity* shall be carefully assessed to emphasize uncertainty and incompleteness during the acquisition process, and its *variability* confirms the need of a dynamic approach due to increased number of inconsistences. Big Data from space is after all a collection of signals and measurements of the Earth physical parameters. Given the lack of universal methods to provide consistent and accurate results as regards the data's intended use, the scientific community will argue against *validity* of such Big Data. Finally, EO data interpretation lies on *visualization*, an inconvenient characteristic, as most of the data features are not always perceivable by the human eye, while the procedures are data driven, and on a *vocabulary*, defining clear and unambiguous encoding for the extracted information (features, data models, ontologies, taxonomies, semantics).

However, the true meaning of Bid Data analysis, and challenge at the same time, comes with the need to simultaneously process and analyse data coming from different sources. A model must be defined in such way that relevant information single out of a big volume of heterogeneous data. Machine learning has proven able to overcome in most cases the issues of Big Data (caused by the Vs) and facilitate the meaningful analysis.

The information theoretical approach, till recently only a parallel field to information retrieval, opens new conceptual approaches for understanding, modelling and solving the challenges of the transformation data-information-knowledge-decision. Information-based exploitation such as spatio-temporal and long-time-

series data analytics, data cube structures, smart data management, information extraction technologies, computational intelligence, platforms and services are turning into solid instruments capable to empower the EO domain and unlock new applications. Their seamless integration with cloud environments exposes higher capability for storage, manipulation, transmission and processing of new streams of data.

Although incredibly fast evolving, science and technology will persist in being a step behind with respect to Big Data analysis without a deep understanding of its main features with respect to the EO data particularities. The need to outpass these challenges and benefit from the value of the data motivated the EO community to come together and stimulate interactions between researchers, engineers, users, infrastructure and service providers, with the main goal of widening the awareness, and share experience. The *Big Data from Space (BiDS)* Conference aims at promoting the potential of using Big Data and widening competences, fostering the sharing of expertise across various domains (EO, space science, telecommunications, etc.) [3]. The process of digital transformation and data revolution will maximize scientific and socio-economic value. This aspect shall be also emphasized by the adaptation of such digital revolution into advantages for space in Europe [4].

The upcoming evolution of information retrieval technologies is driven by its application areas and its use. While the applications in general share a common perspective referring to scientific, environment and societal issues, the ecosystem of contributors to the field includes several categories of users. From scientists and developers to large organizations and end-users, they all bring specific, but mutually dependent, influences to the constantly expanding process of Big Data exploitation. The development cycle includes indisputable proof of concept, dedicated tools and key applications.

This chapter provides an extensive examination of the outcome of the *Big Data from Space (BiDS)* Conference, underpinning the critical aspects, as well as its implications for the EO community, of the Big Data field. The second chapter presents a comprehensive understanding of all the challenges that one must face in the attempt to harness the information included in EO data archives. The third chapter deepens into the analysis of Big Data. In the fourth chapter, the authors describe data lifecycle and review the developed technologies. Chapter "Feature Engineering" resumes the existent EO exploitation platforms and the envisaged applications. The public awareness with respect to these technologies and their drawbacks are described in chapter "Data Summarization Using Sampling Algorithms: Data Streams Case Study". The last chapter anticipates the perspectives on Big Data from a scientific point of view, and this chapter derives and summarizes the main conclusions.

2 Understanding the Challenges of Big Data from Space

Big Data from space (BiDS) refers to Earth and space observations, or spacecraft housekeeping and telemetry data, collected by space-borne or ground-based sensors. It includes EO data, as well as other space-related applications, e.g. satellite

navigation or satellite telecommunications. All of this is qualified as being Big Data, but the field goes beyond the Big Data usual interpretation, adding specific challenges to the process.

An important particularity of EO images that should be considered refers to their "instrument" nature. They are sensing physical parameters and they are often sensing outside of the visual spectrum. As, for instance, a typical EO multispectral sensors acquire "images" in eight spectral channels, covering the visible and infrared spectra, or the synthetic aperture radar (SAR), "images" are represented as complex values representing modulations in amplitude, frequency and phase of the collected radar echoes.

Consequently, the semantic aspects are much broader and much more difficult to be formalized. The EO data content "meaning" comprises many facets. They carry quantitative information about physical, geometrical or other types of attributes of the observed scene. The instrument (sensor) and image formation parameters lend understandability to the data. And they also refer to geographical names as well as to geomorphologic, tectonic, political categories. They have cartographic symbols and, last but not least, have ubiquitous names. Further "semantics" will carry this complex meaning.

As already introduced in the previous chapter, Big Data from space are characterized by [5–8]:

- Sheer *volume* of sensed data: archived data are currently reaching the zettabyte scale.
- High *velocity*: new data are acquired almost on a continuous basis and with an increasing rate, moving towards real data streams.
- High *variety*: data are delivered by a series of devices and sensors, acting over various frequencies of the electromagnetic spectrum, in passive and active modes.
- *Veracity*: sensed data shall be associated with qualified uncertainty and accuracy measurements; the data source is not always consistent to the use case.
- *Variability*: it refers to unpredictable data dynamics due to increased number of inconsistencies generated, e.g. by cloud coverage and artefacts, multiple data sources causing heterogeneity at different dimensions and various speeds.
- *Validity*: the lack of universal methods to provide consistent data quality as regards its intended use.
- *Visualization:* key particularity given the nature of EO data and an inconvenient feature for anyone who wants to complement the lack of perception by means of visual analysis to interpret EO data.
- Their ultimate *value*: depends on the capacity to effectively extract relevant information and meaning from the archives.
- *Vocabulary*: a consequence of data complexity trying to structure the EO data and define clear rules for information extraction (features, data models, ontologies, taxonomies, semantics).

This paradigm (known as the Big Data Vs paradigm) permits to identify the key features and forthcoming challenges related to Big Data, as summarized in Table 1.

Table 1 Big Data key features and challenges

	Traditional data	Big Data from space
Volume	Gigabyte (10^9)	Terabyte (10^{12}) or petabyte (10^{15}) and beyond
Velocity	Per hour, per day	Faster, real-time data streams
Variety	Structured	Semi-structured or unstructured, coming from heterogeneous sources
Veracity	Centralized/straight and reliable data sources	Distributed/uncertain and incomplete data source
Variability	Anomaly/outliers	Increased number of inconsistencies/spatio-temporal data
Validity	Reference datasets, consistent data quality	Subjective and application-based reference datasets, lack of awareness over Earth physical evolution
Value	Easy/low value	Difficult/high value
Visualization	RGB data/real data	Collection of signals, microwave-RGB-IR data/complex data
Vocabulary	Data's structure, data models, ontologies, taxonomies, semantics	Features, data models, ontologies, taxonomies, semantics

3 Deepen into Big Data Analysis

As users of EO data well know, image and data repositories are much too large to be scanned or analysed thoroughly by humans. Practical approaches to successfully use such data and imagery require automatic procedures for significant image retrieval from a large repository [9]. It is not feasible for a human to look at tens of thousands of individual images or data instances to determine which are relevant and which are not.

Therefore, search engines and data mining became fields of study that have arisen to seek solutions for automatic extraction of information from EO repositories and other related sources that can lead to knowledge discovery and the creation of actionable information. Knowledge discovery is among the most interesting research trends; however, the real challenge is to combine machine intelligence with the power and potential of human intelligence, this being a primary objective in the field of Human-System Interaction (HIS) [10]. The goal is to go beyond the today methods of information retrieval and develop new concepts and methods to support end-users of EO data to interactively analyse the information content, extract relevant parameters, associate various sources of information, learn and/or apply knowledge and to visualize the pertinent information without getting overwhelmed [11]. In this context, the synergy of HSI and information retrieval becomes an interdisciplinary approach in automating EO data analysis.

At the same time, the need for timely delivery of focused information for decision-making is increasing. Therefore, EO data, which are not immediately usable, require chained transformations before becoming the needed "information

products" (easily understandable and ready to use without further manipulations) offered as on-demand or systematic services. Different entities may perform these transformations using their own processes, which require specific knowledge, experience and, possibly, data or information from domains other than EO. Today, these information products are primarily developed in a semi-automated fashion by experts or specialized companies operating in specific application domains.

By using systems which can learn and/or apply knowledge, the time for information extraction will decrease and the overall efficiency will be improved. Automatic or semi-automatic data mining (DM) techniques enable fast identification of the relevant subsets among large quantity of images, support the expert's interpretation or even directly provide extracted information. This procedure can also empower petabytes of (archived or new) data processing, as opposed to current techniques that systematically process only limited quantities. In this context, DM is an interdisciplinary approach for automatic remote-sensing analysis that draws on signal/image analysis, pattern recognition, artificial intelligence, machine learning, information theory, databases, semantics, ontologies and knowledge management. It includes novel concepts and methods to help humans to access and discover information in large image archives to rapidly gather information about courses of action.

More in detail, the DM techniques include the following aspects:

- Data cleaning/artefact removal
- Databases
- Challenges in image understanding
- Data models
- Information vs algorithmic information theory
- Semantics
- Information mining and knowledge discovery

The general domain of data mining, regardless of the data type (texts, data records or images) and its applications, hinges on the quality of the data. Every source of data is potentially a source of errors: misspellings, redundancy, contradictory values, inconsistency, etc. This may restrain direct access to accurate and consistent data. Such artefacts disturb all the functionalities of data warehouses: learning and inference clustering, which are used to make decisions from the data and where the correctness of the data is necessary to avoid erroneous or contradictory conclusions. Hence, each data analysis process requires support for data cleaning in order to detect all major errors and inconsistencies. However, data correction is often impossible since a large part of the information may be destroyed. Recent technological progress in satellite image acquisition prompted data providers to create large databases of images fed everyday by a huge amount of data (100 GB/day for a typical satellite), where errors are impossible to annotate manually. Moreover, the increased radiometric resolution nowadays makes some artefacts invisible to human perception. Thus, we need algorithms to automatically detect the artefacts introduced in these images. Because of the complexity of the image formation process, the variety of these errors is large; they may be generated

by sensors (calibration, saturations and sampling), electronics, transmission (noise) and processing (filtering and compression).

The content-based data retrieval is a two-level hierarchy process. First level refers to primitive feature extraction and basic descriptor composing, while the second level of annotation is based on relevance feedback or active learning procedures relying on the low-level feature definition. Thus they can be applied only to homogeneous media [12].

The content retrieval or heterogeneous data mining requires the integrated use of databases, images or sensor signal descriptors, learning paradigms, visual and semantic capabilities [13]. Besides the "semantic gap" and "sensory gap" [14, 15], the main challenge refers to the fundamental difference between a traditional database record and signals as signs. The relation between the data content as a sign and its information expressed at semantic level is generally given by a similarity function [16]. However, the relation of similarity is too generic to represent meaning (semantics). Therefore, the heterogeneous data exploration requires specific interaction with the users, in the form of a dialogue that enables the definition of semantics adapted to the user conjecture.

The recommended concept is based on the elaboration of methods for a complete retrieval model able to treat heterogeneous data [10, 17]. The models aim to represent the relation among homogeneous media content via low-level features and various levels of aggregation using similarity set measures. To extract the joint content, i.e. the common concepts in multimodal data, it generally used an active learning process.

The envisaged models are in the class of multilayer Bayesian hierarchical models, with one layer of latent variable. The latent layer is understood as a representation of the topics encountered in the data. Such a Bayesian hierarchical formalism may allow an unambiguous and causal definition of semantic concepts in the heterogeneous data. However, the inference of the latent topics can be extremely complex or not tractable.

Alternatively, this could be accomplished by using separated models for each data modality [11]. A study topic will be how to exploit the advantage of classes of simple conjugate topic priors [18, 19]. The cross types of data models will be studied as bridges at various levels of abstraction between the unimodal models. The model parameters will be learned in a double loop of active learning, involving relevance feedback in the various mode spaces. For the optimization of the joint multimode content extraction will be studied using mutual information-based and related methods.

The motivation underpinning this concept is the discussion and proposal of a unitary solution, from the perspective of the information theoretical approaches. The concept assumes that content extraction and communication is mainly assured by the fact that "the distinction between the perception of information as signals, signs or symbols, in general, does not depend on the form in which the information is presented, but rather on the context in which it is perceived".

The intention is pushing towards assimilating the coding theory with a simple principle of semantic compositionality; the meaning of a whole is enriched by the

meanings of its parts together with the manner in which these parts were combined [20, 21]. In this frame, the basic evolution steps from Shannon communication channel to a semantic communication will be studied. The data content will be coded as a meaningful message, a semantic code [22]. The subsequent procedure is finding an appropriate adaptation process in the human-computer interaction that leads to achieve a common mode at communication; i.e. the computer behaviour is similar to humans. To achieve this desiderate, the hierarchical Bayesian information representation may become a base for a semantic communication channel. The model is similar to the "classical" communication channel, but may involve several levels, i.e. chained communication channels. For example, the two-level content retrieval can be modelled as first "channel" having as input the data, and output an abstract vocabulary of data classes, and a second "channel" having as input the "vocabulary" and output the semantic code, i.e. the semantics used in dialogue with the user. Since these models are known, or can be learned, the coding of semantics, and thus the query in the heterogeneous database, can be optimized and reduced to very simple schemas. Generative models and deep learning paradigms will be considered for generation of conceptual representations, and alternatively latency will be analysed to extract the meaningful content.

Willing to obtain the "semantic communication channel" a series of conceptual theoretical solutions should be discussed and studied, as, for instance:

- Establish a deep learning process as generalized communication channel coping with the issue of content, knowledge and relevance.
- Define "the error" in the transmission of content and knowledge.
- Define a code (signs, symbols) and data representations to support optimal transmission of content.
- Formulation of channel capacity to include a function of semantic relevance.
- Formulation of rate-distortion theory for data content including the user's conjecture.
- Formalize contextual information, also subjective conjectures, and use it in the prediction and encoding process to communicate data content.

Few years ago information and algorithmic information theories revealed a new perspective for quantifying the information contained in data [23]. The Shannon information theory based on a statistical approach cannot describe the information content of a unique entity. Instead, the algorithmic information theory measures information in an isolated entity by quantifying how a computer could create it [24]. New proper concepts are now redefined based on the bridges opened by a new understanding of the probabilistic inference frame, Occam razor or complexity. These are clarifying and open new perspectives to the idea that information content and compression are closely related concepts. Recently an essential concept born from the algorithmic information theory is the compression-based similarity measure. This general similarity metric circumvents the non-computable Kolmogorov complexity, comparing "entities" just by estimation of compression factors. It determines the amount of information shared by any two entities. Being a parameter-

free method it opens the perspective to be adapted to different data models by considering different codecs.

This discussion assumed the lossless compression methods. However, in the case of heterogeneous data, the methods of lossy compression have an innate association with content retrieval. For instance, the dictionaries or the codebooks extracted from the images could be compared through distortion measures to find the minimum distortion match to an observed signal. This opens a new perspective for new definitions of the "relevant information", and besides it would facilitate matching across heterogeneous data [25, 26].

For content retrieval the communication channel model can be used. The source is the ensemble of heterogeneous data, and the information contained in the data is a message. For an infinite number of queries, the union of query answers is the received message such that the transmitted information is contained in the index used. For an interactive exploration, only a subset index information is used. Consequently, in order to transmit sufficient information to the index and in order to satisfy any user's interests, non-redundant information should be used. This concept should lead to a lossy source coding of the database such that relevant and non-redundant information is transmitted to the index.

Therefore, the index is contained in the resulting code and is equivalent to a dictionary. The dictionary is computed from the dataset. Then, a coder using this dictionary should code well each entity of the data, i.e. its content. The resulting content retrieval will be composed of a dictionary and the coded entities. This is a two-part representation similar to MDL approach. The coding scheme is lossless and the dictionary is a lossy representation of the data. It contains the relevant information in the sense of Kolmogorov. The field is yet to be developed and various topics arise, such as data abstraction based on data compression [27], algorithmic information theory for modelling the intrinsic content of very large volumes of data or encoding based on formalized contextual information.

4 Data Lifecycle and Developed Technologies

Big Data from space is now recognized as an emerging domain, also given the recent sharp increase in all three main dimensions of Big Data: volume, velocity and variety. Fortunately, this increase is paralleled by new developments related to Big Data in other fields, and it is at the same time enabled by technological breakthroughs in hardware and software developments, e.g. high-performance computing, global memory capacity, cloud networking and storage, global connectivity worldwide, data science disciplines, etc. In addition, the recent multiplication of open access initiatives towards Big Data is giving momentum to the field, widening substantially the spectrum of users as well as the awareness among the general public and offering new opportunities for scientists and value-added companies.

When talking about Big Data from space, it is evident that various space-related domains and applications are concerned (Earth Observation, space science, satellite

operations, satellite telecommunication, satellite navigation, etc.). Synergies and cross-fertilization opportunities between the various domains and applications shall therefore be fostered and exploited to address common Big Data issues and to achieve the maximum benefit from research and technology development activities in the field. The contribution from each domain shall be developed and extended. In order to enable the development of common cross-cutting solutions, which require diverse approaches due to the data characteristics (the so-called the Big Data Vs paradigm presented above), a common model is deemed as necessary. Expending inputs from the analysis of use cases and reference architectures collected during the past Big Data from Space Conferences, a vendor-neutral and technology-agnostic conceptual model has been derived ([8, 28]), based on the entire Space Data Lifecycle. The model is based on four conceptual layers (e.g. the lifecycle steps):

1. Acquisition: data collection and acquisition process
2. Organization: data storage, organization, management and dissemination
3. Analysis: data processing, analysis and visualization
4. Provision: Generation and provision of value-added information and services for decision-making

This model is graphically presented in Fig. 1.

The data and information flow (circulating across the four steps presented in the figure) must be supported by suitable information and communication technology (ICT) resources and infrastructures, e.g. suitable architectures to optimize data storage, processing and analysis, data and software frameworks for scalable archiving and processing, algorithms for exploratory and systematic analysis of massive data, approaches and methods to provide the basis for reproducing, guaranteeing and protecting data and results across the entire data lifecycle, etc.

The model therefore represents a common basis for:

– Eliciting cross-cutting enabling technologies
– Deriving the corresponding technology requirements
– Highlighting the areas requiring special attention
– Defining the processes to achieve a seamless deployment of developed technologies

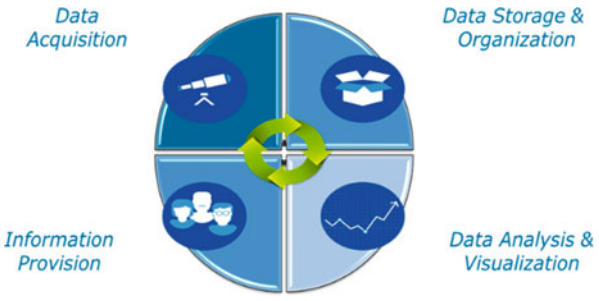


Fig. 1 Big data lifecycle

- Setting up the mechanisms to generate and deliver meaningful and reliable information
- Facilitating the identification of priorities and opportunities for business development
- Helping in the definition of legislation frameworks to guarantee interoperability, data protection and intellectual property rights (IPRs)

4.1 Data Acquisition

This is the first step of the data lifecycle, where sensed data are collected and acquired. On-board processing systems are of course crucial in this phase; next generation of on-board instruments will collect an increasing quantity of information at increasing rates, due to the recent technological improvements in sensors and data processing capabilities. The huge amount of data generated on-board is competing with the limited channel resources available for the transmission of data to the ground. The result of this scenario is that the importance of on-board payload data reduction methods is increasing and becoming critical in the framework of a spacecraft design. Two main areas are being considered in this field: data compression and compressive sensing.

Concerning data compression, the algorithms standardized by the Consultative Committee for Space Data Systems (CCSDS) are widely used and are quite mature, but the challenges posed by the Big Data era demand for more sophisticated algorithm to be used on-board, both lossy and lossless. Recent advances in the lossless compression have proved that it is possible to achieve better compression performances, using dictionary-based or recursive-based algorithms.

Regarding the compressive sensing (CS), signals with the intriguing characteristics which permit to apply this technology (sparse signals) can be mathematically represented by an integral transformation (IT) of the signal itself, where the number of non-trivial elements is less than that originated by a non-sparse signal. The sparse mathematical representation admitted by the signal can be made accessible to a sensor, provided that a dedicated subsystem performs the involved IT before its focal plane. Compressive sampling can bring many latent advantages to hyperspectral satellite imaging, since high spectral and spatial resolution can be attained with fewer detectors, lesser memory capacity and narrower downlink bandwidth. All these aspects could be relevant for addressing related Big Data issues. The main advantage of CS is that compression takes place before the signal sampling, hence avoiding the acquisition of large volumes of data followed by standard signal compression. The possible impact of CS could be remarkable, motivating new investigations and research programs regarding this emerging technology. The principal disadvantage of CS is instead the intensive offline data processing that leads to the desired source estimation.

The raw housekeeping data acquisition, done either through file transfer operations or CCSDS links, is by itself currently not a major Big Data challenge.

These data however are expanded on ground to compute engineering values, apply calibration curves and combine them to create new derived parameters. This explosion on the ground causes challenges in terms of volume and velocity and affects the following ground segment (GS) functional components: long-term storage repositories, offline analysis and reporting and data dissemination.

The entire ground segment is currently undergoing a major evolution, with the development of the new European Ground System-Common Core (EGS-CC), which aims at providing a modern and flexible monitoring and control platform. Even if EGS-CC data acquisition pattern and interfaces are very similar to the current GS, Big Data technologies (such as Hadoop and HBase) have been selected to serve as the basis of the EGS-CC archive component. As a result, cross-domain applications (which are GS systems not fully covered by the EGS-CC framework) are moving in the direction of common multi-mission services, further exacerbating the volume and velocity challenges and introducing at the same time challenges related to variety and veracity.

For what concerns the acquisition process of Earth Observation (EO) payload data, the current main challenges to be addressed are as follows: (1) the satellite downlink capacity, to cope with increasing data rates and volumes and the high-performance data transfer, from receiving stations to processing and archiving facilities; (2) advanced archiving methods, through, e.g. distributed repositories and mirroring sites, which shall ensure optimum performance while maintaining security in a multi-user environment; and (3) advanced dissemination methods, which include open and flexible data access mechanisms, multicast data distribution architectures across national and network domain boundaries, dissemination networks capable to scale with the number of partner sites and to satisfy the requirements for contractually agreed service levels and design of thorough data logistics.

Also current space science missions (e.g. Gaia) and future ones (e.g. Euclid) are as well calling for the introduction of new delivery mechanisms able to cope with the increasing demands on data flow. As part of project initiatives like the Science Operations RAW Data Archive (RAWDAR), Big Data technologies are expected to play a more prominent role in the next generation of data acquisition systems. Built around Big Data processing engines, sophisticated mission retrieval strategies will allow increased speed, ease of use and advanced analytics.

4.2 Data Organization, Management and Storage

In this area, system and software architectures play a central role. Scale requirements are the main drivers for the trade-off that needs to be done in order to select the right distributed software, the data and the deployment architecture that actually cannot be considered separately in Big Data systems. Scale requirements can be addressed by both vertical and horizontal scaling architectures: vertical scaling, which consists in adopting faster processors and bigger disks as workload

or storage requirements increase, and horizontal scaling, which consists instead in achieving high performance, storage capacity and availability by partitioning and replicating datasets across a clusters of servers. Vertical scaling is typically adopted by SQL database technology and has limitations concerning Big Data availability and performance requirement. As a matter of fact, normalized and strong consistency systems are being replaced by schema-less, intentionally de-normalized and weak consistency data models, designed to scale horizontally across clusters of servers.

Nevertheless, even schema-less data model-based systems require a conceptual data model to be established at the early phase, in order to specify system components, roles and relationships, independently from any physical implementation. Once specified, then the actual data model can be selected in order to fulfil the project (or domain)-specific requirements.

The main four types of NoSQL data models are as follows:

1. Key-value store, using a hash tables of keys and values
2. Document store, storing documents made up of tagged elements
3. Column store, where each storage block contains data from only one column
4. Graph, consisting of a network database that uses links and nodes to represent and store data

For example, in the case of ground segment data management, current solutions relying on SQL database technologies work well under certain data volume and velocity thresholds; however, it gradually starts to impose severe limitations (e.g. non-linear performance and discard storing of certain elements). These limitations have started to affect current missions where important trade-offs had to be made, such as for data backup and replication or scope and time range of decoded parameter data. Horizontal scaling technologies and NoSQL models have been selected also for several ground segment components (including EGS-CC) in order to mitigate these limitations.

In the Earth Observation field, the organization, storage and management of data is a very crucial aspect, since it represents the necessary background for successive data (pre)processing, delivery and dissemination. The main challenges to be currently addressed are in this case:

– The development of new technologies for data discovery and access, in order to allow, e.g. instant access to large volumes of full-resolution EO datasets, access or download of tailored coverages already prepared for specific applications, content-aware access methods, control and grouping of associated data, etc.
– Improvement of standards and their implementations (e.g. WCS, WCPS, FTP, HTTP, etc.) or development of new ones to harmonize online data access to very large and distributed EO data holdings; in this view, it is very important to promote the development of interoperable functional components, permitting to cross-link different systems (e.g. it is done in the context of the Open Geospatial Consortium (OGC)).

– Maintenance and operation of long-term curated data archives, including historical, heterogeneous and auxiliary data, processing tools and related documentation.
– Efficient data ingestion, preparation and retrieval mechanisms.

In this field, the term *data cube* is recently receiving increasing attention, as it has the potential of greatly simplifying "big Earth data" services for users by providing massive spatio-temporal data in an analysis-ready way. However, there is still a lot to be done for consolidating the data cube modelling, query languages, architectures, distributed processing, standardization and active deployment at the petabyte scale (and beyond).

Also in space science, data storage and management is at the core of all activities. Raw data retrieved from the Mission Operations Centre (MOC) provides the foundation for the science pipeline in charge of generating high level science products consumed by the scientific community. Already, the Gaia data archive is expected to be of the order of petabytes, too large for the general scientific community to download for analysis. Moreover, new missions like Euclid push the limits of storage requirements beyond the 175 PB.

In order to cope with these challenges, data distribution, data federation and data discovery technologies available in the Big Data arena represent a leverage point to act upon. The Science Data Archives shall act as central access points, offering a facade that allows details of the internal implementation of the repository to be abstracted away. Behind the scenes, an orchestration of services will enable fast integration of multiple datasets. The simplified interface will enable the provision of visualization and analysis tools to the scientists.

4.3 Data Processing and Analysis

As a general aspect, data-intensive system architecture choices shall be carefully decided according to the specific system (Big Data) requirements:

– Write-heavy workloads, where a possible solution would be the data partitioning and distribution to spread write operations across disks and data replication to provide high availability; however, data partitioning and distribution and data replication introduce availability and consistency issues to be addressed.
– Variable request loads, where systems need to be optimized in terms of performances and resources and also to respond to increase resource requirements (e.g. adding processing capacity to share loads and releasing resources when loads drop).
– Computation-intensive data analytics and/or long-running requests that perform complex analytics on significant portions of data collections, where software and data architectures require tactics to partition simultaneously between low-latency requests and requests for advanced analytics on large data collections.

– High availability for horizontally scaled deployment, where hardware and network failures inevitably occur and distributed software and data architectures must be resilient.

In addition, the quantum computing technology is now emerging, investigating on computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data. The quantum computing uses quantum bits, which can be in superpositions of states, instead of binary digits (bits) used by traditional digital computing (e.g. for public key cryptographic systems).

For the ground segments, the current trend is to provide generic interfaces and platforms that allow missions to deploy and run their custom processing algorithms and models – which could be very complex and usually considered mission specific – without the need to worry about the underlying technology and infrastructure. This challenge is facilitated by the use of generic processing distributed frameworks (e.g. MapReduce or Apache Spark). GS components focus mainly on extreme high availability (including several monitoring and alert components), write-heavy workflow (ratio of incoming and outgoing data is extremely unbalanced) and variable request loads with very unpredictable read frequencies (however with demanding performance requirements). Some of the specific applications are implemented using machine learning approaches (e.g. deep neural networks, support vector machines, etc.); these technologies require a lot of data, also made available from previous missions or recorded in the past and computation power to build the models. Among them we recall: advanced data visualization for performance assessment, enhanced monitoring functions to provide early warning on the anomalies (i.e. on the way to happen), enhanced diagnostics to understand the cause of a given anomaly or to perform characterization of an observed event, prediction models to perform on-ground forecasting and what-if analysis, analysis of flight control parameters, etc.

In the case of EO applications, data processing and analysis is affected by all the aspects of the five-V paradigm. Currently, the main challenges to be addressed are as follows:

– The development and set-up of petabyte-scale environment for scientists and researchers to manipulate and analyse huge datasets (including multi-mission series of satellite data, high-resolution climate models, etc.)
– Development of advanced analysis algorithms and powerful visualization tools to trigger and support the analysis (now being called visual analytics methodologies)
– Development and set-up of effective e-collaboration and knowledge-sharing approaches, as well as networks for research and education

Among emerging initiatives, the virtual research environments (VRE), for example, provide specific set of tools and protocols to facilitate access, analysis and exploitation of vast datasets, promoting interoperability of components, standardization of interfaces and harmonization of architectural patterns.

4.4 Provision of Value-Added Information

As described in the previous sections, achieving competency in the Big Data field is a process that requires not only building up the analytics capability but also setting up the right organization to make the most of the opportunity given by Big Data assets. Big Data systems analytics capability strongly depends on the extreme coupling of the designed architecture of the distributed software, the data and the actual deployment infrastructure; such a synergy is a key point in producing Big Data added value. Successful proven analytics teams include expertise on:

- Data science, which provides expertise in statistics, correlations and quality
- Hardware and software to deploy solutions for collecting, cleaning and processing the data
- Business analysis, which identifies and prioritizes the problems worth solving and the business relevance of identified patterns and anomalies

For example, in the context of ground segment systems all received data are extremely valuable and capabilities such as replication chains, gap checks and restore mechanisms are necessary. Data are also stored in multiple layers and lower layers can be repopulated from the raw data if necessary. The ground segment does not provide by itself final data products; however it provides means to make data from multiple sources (spacecraft telemetry, commanding, ground station parameters, flight dynamics, space debris, mission planning, etc.) available for further processing. One big challenge created by this approach is the correlation and combination of data elements from distinct deployments of the ground segment and their data preservation after mission completion, e.g. to allow for valuable data comparisons with similar future missions. This can be achieved with the use of a common Big Data storage platform that can scale horizontally on-demand.

Also in the Earth Observation domain, data processing and analysis represent the underlying layer to finally produce and deliver the value-added information (sometimes also called the actionable information). Currently, the main challenges to be addressed are as follows:

- Deriving value-added information from a large variety of heterogeneous sources, including in situ data, auxiliary information, etc.
- Implementing methodologies and tools to constantly update value-added products as soon as new data become available
- Developing suitable mechanisms and services to efficiently deliver the requested information

5 Earth Observation Exploitation Platforms and Applications

Although defined as a unitary concept on the subject of a fast-growing volume of heterogeneous and complex data, Big Data is actually a sum of interconnected issues and challenges that one must overcome in order to completely understand, exploit and benefit from the advantages of such amount of information. The scientifical and technical contributions presented during the BiDS 2017 conference focus on narrow solutions for specific topics, from data valorization and preservation to on-board processing to ground-based image and data analysis, including downstream exploitation. Nevertheless, the great power of Big Data in the context of EO lies in the ability to link all the technologies in view of solving complex use cases in the everyday life while considering all the aspects involved. Sustained by the technological progress, the EO community seems ready to address such scenarios. Major contributions emerged in the shape of dedicated infrastructures and platforms, as reflected in the conference proceedings. The basic underlying concept for all these platform implementations is the paradigm change of "bringing the users to the data", instead of the traditional approach, used in the past years, of "delivering data to the users". The interest covers a very large variety of applications, which aim at a comprehensive understanding of our planet dynamics, and goes towards very large scale, multidimensional data analysis, including space, temporal and content information. A wide number of dedicated platforms were introduced. Most of them are dedicated computing (virtual) environments, for public and/or private use, offering data assets together with a collection of tools to manipulate them and serve the purpose, providing support to various EO communities and related institutions. In the following paragraphs, this chapter will present the main platforms recently made available, highlighting their characteristics.

5.1 Overview of Platform Implementations

At first, this chapter indicates the main three platforms aiming at a wide, general use, developed by public or private means. The French Space Agency (CNES) decided to highlight the relevance of a global environmental monitoring and encouraged the development of the Platforme d'Explotation des Produits Sentinels (PEPS), as part of the French Sentinel Collaborative Ground Segment (CGS). The platform provides full access to all the Sentinel products based on an open and free data policy [29]. Airbus introduced an online platform designed precisely for image processing on the cloud; on top of this, OneAtlas offers access to a global very high (0.5–1.5 m)-resolution map, constantly updated with every new image acquisition [30].

Narrowing the selected dataset, without losing the concept of Big Data though, VITO presented a platform dedicated to the exploitation of Proba-V and SPOT-VEGETATION EO data archives, together with derived vegetation parameters from

the Copernicus Global Land Service. This gives rise to the concept of a Mission Exploitation Platform (MEP) [31], a fully operational service dedicated to the Proba-V mission and auxiliary missions related to vegetation monitoring.

Despite the trend of data openness, some technological developments are kept at the institutional level. In its attempt to cope with the Big Data shift towards the free, full and open character, the Joint Research Centre (JRC) defines the concept of Earth Observation Data and Processing Platform (JEODPP). The platform is dedicated to fulfil the needs of the JRC institutional projects relying on geospatial data analysis in the context of their policy support activities. Moreover, the platform copes with the needs and requirements of users with different backgrounds and various programming languages. JEODPP offers the possibility to interact directly with the image data [32].

The Research, Technology Development and Innovation Unit of the European Union Satellite Centre combined Big Data assets with cloud computing and machine learning for the development of new solutions to access, process, analyse and visualize both satellite imagery and collateral data, as social media data from open sources. The platform is considering the full data lifecycle, with the objective of extracting from available data all the relevant information in the space and security domain [33].

MUSCATE – the data and service infrastructure of THEIA, French Land Data Centre – is now operational at the CNES Computing Centre. The platform is dedicated to the exploitation of imagery provided by SPOT, Landsat and Sentinel-2 satellites. It has the ability to catalogue and distribute up to 1600 products daily, in order to assist monitoring human and climate impact worldwide. MUSCATE is intended for the use of scientific community, but mostly for the public policy actors [34].

Another highly successful data analysis system handling petabytes of environmental data is JASMIN, used in partnerships by universities and industry in the UK. JASMIN provides virtualized services to optimize the use of computing resources available, large archives of EO and climate science datasets, a community cloud, tools and interfaces to shared environmental data formats. For the past 6 years, this cluster enabled users to access data, add new data and exploit their own data manipulation techniques [35].

The increased EO data flow and volume generated by the huge number of available satellite missions turns out to be an extremely difficult test for both scientific and industrial EO players. Many efforts were channelled towards integrated environments while building easy to use web platforms to support management of large archives, near-real-time data streams and advanced monitoring tasks for the entire EO community. Coping with different sources and types of data is a frequent issue in the Big Data domain. To this aim, a multi-cloud EO processing platform was developed as a building block which streamlines cloud vendor solutions. It acts as an integrated environment for various ICT technologies and data [36]. The versatility of its EO data management, processing and analysis tools was proven in support of cost-effective shoreline protection, against increasing sedimentation, erosion and flood risk. The MI-SAFE platform uses intelligent Geographical Information

System (GIS), recognized international standards for the interoperability (OGC) and the Google Earth Engine to extract global coastal vegetation and elevation products. MI-SAFE contributes to the sustainability of costal nature-based solutions by enabling advanced modelling while increasing awareness among the engineering community [37].

Another good example of a specific virtual research environment (VRE) has been established through the Technology and Atmospheric Mission Platform (TAMP), which provides access to multiple data sources (including simulated data generated by models), flexible processing tools for analysis and inter-comparison (e.g. for calibration and validation purposes), visualization tools and social-like portal for information sharing, tailored for scientific users of atmospheric missions [38].

This brings us directly to the emerging concept of the Thematic Exploitation Platforms (TEPs). This initiative, started by ESA in 2014 in the context of the Copernicus Programme, which includes the Sentinel missions, along with the Copernicus Contributing Missions (and possibly some Third Party Missions), aims at creating an ecosystem of interconnected exploitation platforms, addressing a variety of environmental monitoring fields:

- TEP Geohazards (GEP), enables the exploitation of EO data in support of geohazard management
- TEP Costal (C-TEP), provides insight on the coastal area dynamics, enabling data-intensive research
- TEP Forestry (F-TEP), focuses on generating value-added information products related to forest areas
- TEP Hydrology (H-TEP), a friendly environment assisting hydrology modelling and mapping
- TEP Polar (P-TEP), tackles the challenges of polar ecosystems to monitor, predict and mitigate changes
- TEP Urban (U-TEP), drives the innovation and maximization of societal benefits in the urban areas
- TEP Food Security (FS-TEP), uses EO information to uphold sustainable food production

In short, a TEP is a collaborative, virtual work environment providing access to EO data and the tools, processors and information and communication technology (ICT) resources required to work with them, through one coherent interface. As such the TEPs may be seen as a new ground segment operations approach, complementary to the traditional operations concept.

The Forestry TEP [39] responds to the worldwide need for forestry remote sensing services and it is built around the design of a one-stop-shop platform, which includes computing infrastructure, access to proper data, value-adding functionalities and the possibility to upload private applications.

The Urban TEP enables an operational, modular and highly automated processing chain based on modern information technologies and EO services to derive actionable intelligence from a wide variety of EO data [40]. The capabilities to successfully access, process and analyse mass data streams are embedded in

powerful infrastructure to meet the needs of environment science, planning and policy related to the global urbanization process.

A similar approach leads to the generation of a platform tackling smart, data-intensive agriculture and aquaculture application. The Food Security TEP integrates a federation of instruments and data from space and in situ measurements to provide both local- and regional-scale analysis for farmers and scientific-related communities [41].

The interest expands towards the coastal waters. The Co-ReSyF platform proposes a synergetic cloud-based framework to increase the use of EO data into socio-economic costal research activities responding at the same time to the needs of inexperienced scientists, EO experts and costal specialists [42].

Water, in general, is a sensitive matter. It can generate tremendous issues whether it lacks (causing draught) or plentiful (leading to floods). The need of a platform to incorporate hydro-meteorological data, EO imagery and ground-based measurements is obvious. Close to the data, the processing algorithms will enable fast, cost-effective, meaningful assessments of water resources, predictions and mitigation plans in case of flooding. The DFMS platform proves the efficiency of such approach for the Ugandan drought and flood mitigation service [43].

A more complex environment is set in the frame of the EO4wildlife project. Multidisciplinary scientists (e.g. marine biologists, ecologists, ornithologists, EO experts) get connected to a wide range of data sources (e.g. EO data, Copernicus Marine Environment Monitoring Service, catalogues of animal-driven information, the AVISO catalogue for the altimetry domain) and dedicated tools focusing on implementing analytic workflows to model animal behaviour by means of environmental observations [44].

In addition to these integrated solutions addressing worldwide subjects of interest, well-targeted services relying on Big Data (EO related in general, but not limited to it) have been developed: cloud-based geoinformation service for ground instability detection and monitoring (Rheticus service, [45]) and SAR altimetry processing for CRYOSAT-2 and Sentinel-3 data (G-POD SARvatore service, [46]). They benefit of the same large heterogeneous data archives, visualization and processing tools, web user interface and cloud environment.

Such a new approach, fostered by the large number of specific "platform" implementations, is inducing a parallel mindset change. Generic tools aiming at facilitating the use of computing infrastructures are being developed. A representative example is the Automated Service Builder (ASB), a generic platform that enables a dynamic, scalable environment to automatically execute various processing chains when new data become available. ASB is a universal agnostic solution that helps users to define, configure and run algorithms over globally distributed processing and data resources [47], independently from the specific deployment platform.

Finally, it is worth to notice that similar approach is emerging in other fields, different from EO. As an example, the European Space Agency (ESA) is building its own platform at the European Space Operations Centre (ESOC) for housekeeping and telemetry data analysis and exploitation. The Analysis and Report System

(ARES) handles in a simplified manner telemetry packets and parameter information, history of failure events and spacecraft and mission control system events and performs an offline evaluation of data analysis systems at ESOC. Through its outcome, the system allows mitigation of operational restrictions and technical limitation and enables the assessment of future opportunities of new data analysis techniques and algorithms [48]. This example demonstrates the exciting possibilities of synergies across different domains, which are deemed to be essential to address Big Data challenges.

5.2 Characteristics of the Platform Concept

This short survey highlights the interest on integrated solutions for EO data analysis activities. ESA fuelled the general focus through the Thematic Exploitation Platform (TEP) initiative. The global monitoring from space activities will turn into an expanding operational capability by reinforcing the long-term EO data archives with additional insight, such as in situ networks and models. As expected, handling this amount of data will rather necessitate a collaborative effort from the entire community. In order to facilitate the process, a ground segment operations concept must be defined in such a way that data and tools individually owned by users to move to a common environment and turn them into a coherent engine to work right beside the data archives. The cloud environment provides the right means to make the most out of existing technologies while upgrading the entire process through the data lifecycle:

– Data users will no longer depend on the ICT ownerships costs. TEP will provide straight access to both data and ICT infrastructure, without the need of unnecessary transfer and data download. This kind of environment reduces the resource usage, allowing for additional connections and processes.
– Data analysis will not be bound on the Internet connectivity, as the process can run constraint-free. The connections is only required to lunch the process and visualize the results. The amount of data transfer is thus consistently reduced.
– Data sharing gets faster, as the results; once computed, they could be stored at the side of the data. No additional transfer or data replication “at home” is required; anyone can access the data through the web interface.

The approach encourages sharing and user collaboration in view of standardized exploitation, infrastructure-free progress and long-term sustainability while securing the intellectual property rights (IPR). More specifically, TEPs are complex environments sustained by the involvement of several players, with the main goal to provide users with an adequate set of capabilities to solve their applications. Although configurable according to specific roles of the platform, the full set of functionalities includes:

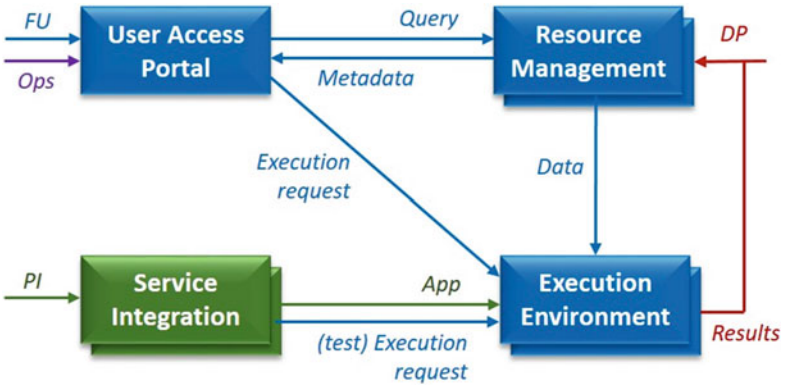


Fig. 2 General overview of an EP architecture [49]

- Data discovery
- Data management
- On-demand processing
- Massive processing
- Development and integration system
- Processing service management
- Collaboration tools
- Documentation and support tools
- Services marketplace
- Single access portal
- Operator interface

The architecture of a generic EP (Fig. 2) at a glance can be split in four macro-components, meaning elements comprising similar or related functionalities, and generate homogeneous services [49]: *user access portal* includes the interfaces for both the users and system operators; *resource manager* deals with system resources, from storage to allocation and queries regarding the database; *service integration* facilitates external plugins of algorithms or software into the platform as a new service to respond to an applications; and *execution environment* supplies the environment to run processing services, being in contact with the user through the access portal at all times.

A generic EP is usually not used in the generic shape. They are tailored to a particular scope based on regional analysis or, more often, on thematic exploitation [50]. The results are considered the Thematic Exploitation Platforms (TEPs), which have been presented in the previous paragraph. TEPs are considered quite a mature technology, since several pilots were successfully implemented. Each platform addresses the needs of a specific group in the EO community. Even if the group is sharing a common goal or theme, each individual in part brings a different expertize and unique set of tasks. The role of TEPs is to provide the right tools and graphical interfaces for an improved and effective activity concerning each individual.

Nevertheless, there are situations where the interest of a group meets that of another group. For instance, hydrology models and mapping near coastal areas might require inputs from both H-TEP and C-TEP communities, which means that interoperability between TEPs and other platforms becomes desirable. The Sentinel database available in the PEPS platform can be an important asset in the monitoring process of a certain area. To this aim, the natural way for scientific and technical evolution goes towards a network of data and service providers to further streamline the usage of EO data. Such an experiment is currently in progress, aiming at a smooth interconnection between C-TEP and PEPS platforms [51].

6 Public Awareness and Drawbacks

Somehow, the Big Data paradigm is forcing a new EO-based innovation to emerge. Complementary data access solutions are required to bring together multiple sources and types of data, shaping a scenario of interconnected exploitation platforms. Managing data at a global scale introduces an extremely innovative operation concept where the users meet both data and the necessary resources (Fig. 3). The exploitation platforms enable a virtual open and collaborative environment bringing together EO and non-EO data, dedicated software and ICT resources, namely:

Fig. 3 Concept of the EO Big Data exploitation

- Data centres
- Computing resources and hosted processing
- Collaborative tools for preprocessing, data mining or other user tools
- Concurrent design and test bench functions
- Application shops and market place functionalities
- Communication tools and documentation
- Accounting tools to manage resource utilizations

With this increasingly and sophisticated pool of resources at hand, the collaboration within the EO community is progressing towards a universal framework. As such, the actors of the EO value chain, and implicitly the users of this comprehensive environment, refer to the same extent to remote sensing researchers and engineers, public institutions, infrastructure and service providers, geoscience end-users, general public, education and media. EO turns into a key element for a wide range of applications fields, enabling new projects and opportunities that will attract ever more players to the EO community worldwide. These interconnections are essential for a continuous expanding field and they allow research groups to benefit from appropriate infrastructure and services, as well as of other's experience.

This pool of dedicated tools and technologies gathers around the Big Data from space, demanding for common management rules. Through its "V" characteristics, Big Data is governing the users' expectations to evolve, demanding the EO Big Data market to grow as well. The goal is to enable global-scale exploitation of data from space in a manner that provides full, free, open and continuous access. Based on the extensive collaboration, sharing, networked governance, affiliation and openness for business, multi-source funding initiatives are stimulating innovation and science with respect to Big Data from space utilization.

As a consequence, the large number of data sources (including the data provided by the Sentinels) reflects in the large number of applications solved. In the past years, innovative EO know-how passed from inception studies and proof of concept to mainstream consolidated technologies, such as Docker containers or Jupyter Notebooks.

The road towards interconnected exploitation tools and universal dedicated platform is driven by the European Union policy to provide free, unlimited access to Copernicus data (8 petabytes per year) [52]. In order to avoid network limitation and ensure a secure access to the data for the EO community, service providers shared their infrastructure to distribute the data either through direct access to the data archive or sales through dedicated resellers.

The first data hubs distributing open EO data (particularly acquired by the Sentinel constellation) have been made available by ESA for individuals, as well as for institutions. However, in order to avoid Internet connection overload, restrictions were considered. We introduce some of the most common data access channels:

- Copernicus Sentinel Data Hub: provides complete, free and open access to Sentinel data under self-registration and up to ten concurrent downloads to ensure download capacity for all users [53].

– Collaborative Data Hub: two nodes available, with no self-registration possible. The first one permits up to ten concurrent downloads with a rolling policy to retain the latest data for 1 month. The second node has no download restrictions, but the data is retained only for 9 days [54].
– International Access Hub: special access to a 21-day rolling archive for international partners under agreement with the European Commission [55].

However, data access and availability are currently misleading. Raw data is rather useless for most of the early adopters. EO data require dedicated selection and processing to become actionable and fit the purpose of the intended application. For most of the times, a complete data access interpretation process requires a minimum level of technical background. The proper tools for this part, advanced and intuitive, are still to be improved and integrated into the exploitation platforms. For instance, ESA's Research and Service Support (RSS) service [56] offers support to ease EO data exploitation, but within a limited reach (the services included in that environment).

As expected, the role for innovation goes to research institutions and universities. Yet, the lack of appropriate infrastructure and services (e.g. federated access, troubleshooting, cloud-based computing or storage service) hampers the advancements. In other words, groundbreaking concepts need technology in order to become tools within reach, as much as the EO community craves for the development of Big Data processing methods.

To this scope, the European Commission has launched an initiative to create a Copernicus Data and Information Access Services (DIAS) platform that will lead to a digital revolution in the field of Big Data from space [57]. This new EO data-based ecosystem is intended to fully support innovation while expediting transition from research to business. The Big Data framework is complemented by a collaborative environment enabling full coordination and provide mutual support among all the users of the EO community to achieve their goals and bring together their efforts towards progress. This initiative will help particularly the scientific community, as they usually lack of integrated infrastructure. Based on the "user to the data paradigm", DIAS will provide the right means to access the data and efficient local processing [58] while empowering the researchers with the right capabilities (e.g. discovery and visualization frame, integration and support layer) to develop tools for data analysis and add new and flexible value chains. The final purpose is to manage the design and configuration of exploitation platforms, which build virtual workspaces for scientists to access and process EO data, collaborate and share results.

However, besides all the advantages of having such amount of data and resources at hand, there is a risk that users can get overwhelmed by the situation. This fully equipped and ready to use environment can boost innovation and development, but it may also create difficulties in selecting the proper data and resources. While the first case is favourable to the scientific community, the second example will lead to the uprise of an EO application marketplace and industrial development. The Atos Codex SparkInData (SID) project provides confirmation through a case

study. The proposed user-oriented platform promises to overcome any technical and economic barriers and assist users in extracting added value from EO Big Data for their activities while helping data and service providers to monetize their products [59].

7 Perspectives

7.1 Market Perspectives

Various space-related domains and applications are generating and using Big Data, with increasing volumes and data rates. Earth Observation is one of them, but synergies and spin-in/spin-off technologies and methods are extremely required. Below we report some examples of market perspectives in various fields:

- Earth Observation: it includes all the aspects, from the implementation of sensors to the generation, transmission, acquisition, storage, handling, processing, dissemination and exploitation of increasing volumes of data. Main needs are to support users in Earth Sciences (e.g. climate monitoring), public health services (e.g. natural disaster monitoring) and commercial markets of EO data (e.g. crop monitoring for food management); to improve data access policies (standards, interfaces, processing environments, federation of infrastructures, etc.); foster outreach of Earth Observation and/or space-related business; and help industry in setting the goals and improving competitiveness.
- Satellite Telecommunications: this domain is characterized by ubiquity and global mobility, i.e. services provisioning across borders, with pan-European and global service coverage area, fast services deployment and coverage of remote telecom-wise isolated areas. They offer resilient connectivity and support emergency and disaster relief. Complementarity to terrestrial connectivity is also offered by native support of multicast (one too many) and broadcast (one to all) with respect to unicast. In machine-to-machine (M2M) and Internet of things (IoT), thousands or millions of sensors can be remotely monitored and controlled via satellite and contribute the Big Data.
- Satellite Navigation and Integrated Applications: user receivers can both contribute and use Big Data, for example, to offload calculation of positions by using cloud computing, to receive satellite ephemeris, to exchange traffic information in Google maps, etc. GNSS satellite constellations and related ground and user segments, such as GPS (USA) or Galileo (EU) ecosystems, are enabling technologies for generating positioning information associated to Big Data. The costs of manufacturing, deploying and operating Galileo first and second generations are representing almost 1 B€ per year. According to the latest GNSS Market Report from GSA [60], the GNSS user equipment market (e.g. chipsets, navigation devices, professional receives, traffic information systems) is approximately 75 B€, of which 53% concerns location-based services (LBS,

including wearable devices) and 38% concerns road market. Overall, GSA estimates that the global revenues enabled by GNSS applications and services represent currently more than 250 B€. Furthermore, new value-added services in the 5G era involve Big Data from different sensors. Some specific Big Data from space cases, such as weather data or other environmental sensor data, are crucial for automotive/transportation, energy, agro-food and other verticals.

Satellite Ground Segment: this domain is characterized by an increasing volume of data coming from design, testing, integration, validation and operation of spacecraft and from associated supporting ground assets, such as mission control, ground stations, simulators, EGSE, etc. Clearly detected trends include the provision of performance assessment, early detection of anomalies, mitigation of risks, cost reduction and taking measures that maximize science return. In addition data analytics for system situation awareness – related to the monitoring of both space segment and ground segment – will play an important role for increasing the level of automation and autonomy in operations.

7.2 European Strategic Interest

Concerning the EO domain, the promise of Big Data is massive, but getting into this business is not going to be an easy task as scaling up requires large investment in infrastructure for storage, computation and acquisition of data. The success of this emerging industry will not only be driven by large organizations (who can afford large and powerful infrastructures) but also by small ones through collaboration around a distributed ecosystem of small data analytics, also designated as information products, which could be focused on specific markets only (e.g. agriculture). They could make use of the freely available datasets through various government programs (like the Landsat, Sentinels, MODIS, etc.) and establish partnerships with giants like ESRI and SAP to gain customer access by using the visualization layer of these established platforms and integrating with their technology stack as the corresponding data layer. Big Data analytics can provide the satellite EO industry with means to engage with customers more insightfully; the Northern Sky Research (NSR) report [61] is expecting services, defence and intelligence, and managed living resource verticals to be its biggest vertical markets. The success of Big Data analytics in the satellite EO industry will not be defined by just volume and velocity of data, which can be achieved by adding more sensors in orbit, but more so by the variety and veracity of the datasets being consumed to build the final product. It is not cost-efficient for a single satellite operator to address all the Vs required for the EO Big Data business (volume, velocity, variety, veracity, variability, validity, value, visualization, vocabulary), which is where smaller EO data aggregators can come into picture, especially to add the required variety, veracity and vocabulary in data. From the picture depicted above, there is a clear European strategic interest in supporting science and commercial markets of EO

data, as well as knowledge and information extraction and management. Suitable IT and EO-specific infrastructures, relying on European strengths in both fields, shall be put in place.

7.3 Advancing the Scientific Development

The abundance of available open data and low-cost powerful computation is now a certainty. The BiDS 2017 conference revealed that EO is coming fast from behind and joins the Big Data community. EO opens a whole new perspective for science, as well as for the applications in general, giving the EO industry the momentum from business development (e.g. information geospatial services, oil and gas, constructions) to public sector (e.g. infrastructure evaluation, advanced monitoring of natural landscape, environment assessment) and academia (e.g. advanced research and scientific innovation, training new skills, creating competences). New means to address the user challenges are brought to light based on a large-scale analysis. The spatial (wide coverage), the temporal (repeated acquisitions) and the structural (ability to separately measure Earth's radiation reflections) dimensions, along with the contextual information (non-EO data), create awareness on the complex dynamics of the processing shaping our planet. The full understanding is bound by the definition of new data models, hence the rising of information extraction using advanced machine learning techniques that outpass traditional data mining approaches. Deep learning and convolutional neural networks are trained to gradually discover not only patterns inside the data but also the relationships between them, such that in the end, to be able to highlight those homogeneous structures representative and meaningfully relevant for a certain application. Algorithmic advancement is assisted by technological breakthroughs and standards to ensure interoperability of mass processes in Big Data. The interest for data cubes and multidimensional array representations is growing, as well as for intelligent management systems to ensure that data is consistently organized while remaining easy to access. Various research initiatives are targeting the collection and exploitation of in situ and crowdsourced data. The picture is completed by laboriously attempts to define and set up large reference annotated databases to support data model validation and put the basis of solid experience to propel further learning. Considering the openness and sharing approach for data and infrastructure, all of these topics are expected to gain considerable attention through the EO scientific community, especially because dedicated programs are planned to be lunched. Collaborative research is rising.

Up to now, defined theories and completed experiments focused on individual events, with no possibility to correlate outputs and gained experiences. Nevertheless, the great volume of scientific data captured at the moment in such a diverse manner on a 24/7 basis, along with data models and information resided through application-driven computation, is likely to reside forever in a live, extensively publicly accessible, curated state for the purposes of continued analysis [62]. In the

frame of EO, such processing is able to place a specific object or event precisely in space and time, provide quite complex information and predict through inference its evolution. Previous experience will improve learning, against usual drawbacks (e.g. sparsity, veracity, variability, validity). This kind of progress in the field of machine learning depends on an interdisciplinary approach centred on computer science and statistics, putting the basis for artificial intelligence and data-intensive science.

Scientific discovery enters a new era. As Big Data prevails, the general opinion moves away from the idea of generating hypothesis before collecting the data to confirm or contest the theory. New data science approach shifts from simple analysis tools to complete knowledge discovery frameworks [63]. Machine learning identifies patterns and uses previous knowledge to provide analytics of increasing complexity able to serve as descriptive (describe what happened), predictive (anticipate what will probabilistically happen) or prescriptive (recommendation on what to do to achieve goals) tools. This cognitive ability enables intelligence on computers, generating similar human reasoning (e.g. perceiving, reasoning, learning, interacting with the environment, problem solving). Nevertheless, the real boost on Big Data analysis is provided by deep learning techniques, which are able to go deeper into the structure of the data and understand its nature. Even if it requires a consistent training set and heavy computation, it copes better with the variety of data sources. The success of data science in EO is somehow anticipated, as already proven to be a solution for Internet-scale data-driven applications.

Turning large amount of data into actionable information using just models derived straight from the data could represent an adequate process. However, the results obtained on a dataset in general are more likely to produce misleading conclusions outside the test area. A shortage of representative samples in a second experiment reduces the modelling performance. Any new application, with a slightly changed interest, will no longer match the previous training. Consequently, a new paradigm appears. It combines the capability of data science models to learn patterns from large data, without ignoring information automatically generated through theoretical hypothesis. Called theory to guided data science (TGDS), the focus is on scientific consistency as an essential component for learning generalizable models [63].

8 Conclusions

Our planet is an unceasing source of tremendous information only partially recorded and stored in huge archives. The EO process is struggling to capture as many aspects as possible from the Earth evolution, in terms of structure or land cover transformations. Remote sensing is a thriving field driven by the technological development, as demonstrated by the multitude of satellite missions that were lunched over the years. Each sensor is built to capture different aspects on our planet. In consequence, the impact of EO is growing day by day along with a large

community of researchers, engineers, end-users, infrastructure and service providers involved in the process.

Massive spatio-temporal observation data made from space or ground over the years compose what it is called EO Big Data and share similar challenges as the well-known term of Big Data: volume, velocity, variety, veracity, variability, validity, value, visualization and vocabulary. This paradigm triggers novel information theory-based algorithmical approaches as well as technological breakthroughs in terms of both hardware and software. On top of this, the trend to standardize the activity is accelerating the development while enabling the adoption of an open frame for data and infrastructure sharing.

The field is deeply supported through several initiatives at institutional, national and even international level. Already a tradition, the conference of Big Data from space gathers together the main players (universities, research institutes, industrial companies, national organizations and agencies) to introduce their most recent contributions to the field. The proceedings shows an undergoing sharp development with numerous innovations covering all the way from intelligent sensors to data science [5]. These proceedings consist of a collection of 126 short papers addressing the whole lifecycle of EO Big Data from data valorization and preservation to on-board processing, down to image and data analysis, including downstream exploitation.

A perfect environment is born to scale up existing algorithms and introduce new ones in view of generalizing previous data models and discover new insight on the dynamics, physical parameters and land evolution. Users are expected to gain enhanced understanding on Earth phenomena and predict future transformation. Apart of the economic impact, the EO Big Data is expected to revolutionize the perception on our surroundings.

Acknowledgements The authors wish to thank Pierre Soille and Pier Giorgio Marchetti, the editors of the BIDS 2017 Proceedings, for their efforts in collecting and organizing the vast material resulting from the conference.

References

1. Kramer, H. J. (2002). *Observation of the earth and its environment, survey of missions and sensors* (4th ed., p. 1514). Berlin: Springer, 522 illustrations, 857 tables, ISBN: 3-540-42388-5.
2. Mathieu, P. P., Borgeaud, M., Desnos, Y. L., Rast, M., Brockmann, C., See, L., Fritz, S., Kapur, R., Mahecha, M., & Benz, U. (2017). The earth observation open science program. *IEEE GRSM*, June 2017, pp. 86–96.
3. http://www.bigdatafromspace2017.org/
4. http://www.esa.int/About_Us/Digital_Agenda/The_ESA_Digital_Agenda_for_Space
5. Preface to Proceedings of the 2016 conference on Big Data from Space (BiDS'16), https://doi.org/10.2788/854791, 2016.
6. Hilbert, M. (2016). Big data for development: A review of promises and challenges. *Development Policy Review, 34*(1), 135–174.

7. IBM Big Data & Analytics Hub web site. http://www.ibmbigdatahub.com/
8. Borne, K. (2014). Top 10 big data challenges a serious look at 10 big data V's. *Technical Report.*https://www.mapr.com/blog/top-10-big-data-challenges-look-10-big-data-v
9. M. Lew, et. al., "Content-based multimedia information retrieval: State-of-the-art and challenges", ACM Transactions on Multimedia Computing, Communication, and Applications, 2, pp. 1–19, 2006.
10. Daschiel, H., & Datcu, M. (2005). Human machine interaction for image information mining. *IEEE Transactions on Multimedia, 7*, 1036–1046.
11. Bratasanu, D., Nedelcu, I., & Datcu, M. (2012). Interactive spectral band discovery for exploratory visual analysis of satellite images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5*(1), 207–224.
12. Zhou, X. S., & Huang, T. S. (2003). Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems, 8*, 536–544.
13. Datta, R., Joshi, D., Li, J., & Wang, J. (2008, April). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR), 40*(2), 1–60, article no. 5.
14. Smeulders, A. W. M., Worring, M., Santini, S., Gupta, A., & Jain, R. (2000, December). Content-based image retrieval at the end of the early years. *IEEE Transactions Pattern Analysis and Machine Intelligence, 22*(12), 1349–1380.
15. Vaduva, C., Georgescu, F., & Datcu, M. (2018). Understanding heterogeneous EO datasets: A framework for semantic representations. *IEEE Access, 6*, 11184–11202.
16. Rasmussen, J. (1983). Skills, rules, and knowledge: Signals, signs, and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 257–266.
17. Datcu, M., & Seidel, K. (2005). Human Centered concepts for exploration and understanding of images. *IEEE Transactions on Geoscience and Remote Sensing, 43*, 601–609.
18. Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003, March). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.
19. Vaduva, C., Gavat, I., & Datcu, M. (2013, May). Latent Dirichlet allocation for spatial analysis of satellite images. *IEEE Transactions on Geoscience and Remote Sensing, 51*(5), 2770–2786, part 1.
20. Jégou, H., et al. (2011, January). Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(1), 117–128.
21. Gueguen, M., & Datcu, M. (2008). A similarity metric for retrieval of compressed objects: Application for mining satellite image time series. *IEEE Transations on Knowledge and Data Engineering, 20*, 562–575.
22. Hwang Juang, B. (2011). Quantification and transmission of information and intelligence-history and outlook. *IEEE Signal Processing Magazine, July, 2011*, 91–101.
23. Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission, 1*, 1–7.
24. Li, M., et al. (2004). The similarity metric. *IEEE Transactions on Information Theory, 50*, 3250–3264.
25. Watanabe, T., et al. (2002). A new pattern representation scheme using data compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*, 579–590.
26. Cerra, D., & Datcu, M. (2012). "A fast compression-based similarity measure with applications to content-based image retrieval", journal of visual communication and image representation. *Elsevier, 23*, 293–302.
27. Rissanen, J. J. (1965). A universal data compression system. *IEEE Transactions on. Information Theory, 29*, 656–664.
28. Crichton, D., Doyle, R., Lazio, J., Jones, D., Castillo-Rogez, J., & Sotin, C. (2014). Considerations for space observing and analysis systems in the Big Data Era. In *Proceedings of 2014 big data from space conference*, ESA-ESRIN, Frascati, Italy, 12–14 November 2014.
29. Duprat, S. et al. (2017). PEPS – the French Copernicus collaborative ground segment. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 134–137).

30. Gabet, L. et al. (2017). OneAtlas, Airbus Defence and Space digital platform for imagery. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 146–140).
31. Goor, E., Dries, J., & Daems, D. (2017). PROBA-V mission exploitation platform. In *2017 Big data from space conference (BiDS'17),* Toulouse, 28–30 November 2017 (pp. 150–153).
32. P. Soille et al. (2017). The JRC Earth Observation data and processing platform. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 271–274).
33. Albani, S. et al. (2017). A platform for management and exploitation of big geospatial data in the space and security domain. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 275–278).
34. J. Donadieu et al. (2017). MUSCATE: a versatile data and services infrastructure compatible with public cloud computing. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 279–282).
35. Massey, N. et al. (2017). Evolving JASMIN: high performance Analysis and the data deluge. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 287–288).
36. Romeo, A. et al. (2017). Cloud based Earth Observation data exploitation platforms. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 233–235).
37. Calero, J. S. et al. (2017). Fast MI-SAFE platform: foreshore assessment using space technology. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 247–250).
38. Natali, S. et al. (2017) Virtual exploitation environment demonstration for atmospheric missions. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 236–238).
39. Hame, T. et al. (2017). Forestry TEP responds to user needs for sentinel data value adding in cloud. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 239–242).
40. Balhar, J. et al. (2017). Monitoring urbanization with big data from space – The urban thematic exploitation platform. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 243–246).
41. Bach, H. et al. (2017). Food security – Thematic exploitation platform: Big data for sustainable food production. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 461–464).
42. Terra-Homem, M. et al. (2017). The coastal waters research synergy framework, for unlocking our potential for coastal innovation growth. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 453–456).
43. Lavender, S. et al. (2017). Application of earth observation to a Ugandan drought and flood mitigation service. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 469–472).
44. Castel, F. et al. (2017). EO4WILDLIFE: a cloud platform to exploit satellite data for animal protection. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 465–468).
45. Samarelli, S. et al. (2017). Rheticus: A cloud-based geo-information service for ground instabilities detection and monitoring based on fusion of Earth Observation and Inspire data. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 473–476).
46. Benveniste, J. et al. (2017). SAR altimetry processing on demand service for CRYOSAT-2 and Sentinel-3 at ESA G-POD. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 477–480).
47. Valentin, B. et al. (2017). ASB – A platform and application agnostic solution for implementing complex processing chains over globally distributed processing and data resources. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 142–145).

48. Santos, R. et al. (2017) Combing small housekeeping data lakes into a shared big data infrastructure at ESOC – Achievements and future evolution. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 283–286).
49. https://tep.eo.esa.int/news/-/blogs/exploitation-platforms-open-architecture-released
50. https://tep.eo.esa.int/home
51. Clerc, S. et al. (2017). Interconnecting platforms via WPS: Experience from the CTEP/PEPS connection. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 344–347).
52. Probst, L., Frideres, L., Cambier, B., Duval, J. P., Roth, M., & Lu-Dac, C. (2016). *Space tech and services, big data in earth observation.* European Union, February 2016.
53. https://scihub.copernicus.eu/
54. https://colhub.copernicus.eu/
55. https://inthub.copernicus.eu/
56. Cuccu, R. et al. (2017). Earth observation data exploitation in the era of big data: ESA's research and service support environment. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 340–343).
57. European Commission, "*Functional Requirements for the Copernicus Distribution Services and the Data and Information Access Services (DIAS)*", http://ec.europa.eu/DocsRoom/documents/20510/attachments/1/translations/en/renditions/pdf
58. Schick, M. et al. (2017). EUMETSAT, ECMWF & Mercator Ocean partners DIAS. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 138–141).
59. Emery, J. et al. (2017). ATOS codex sparkindata: Promoting user uptake of an Earth Observation application marketplace. In *2017 Big data from space conference (BiDS'17)*, Toulouse, 28–30 November 2017 (pp. 483–486).
60. https://www.gsa.europa.eu/market/market-report
61. Satellite-Based Earth Observation, 8th Edition, NSR Report, September 2016.
62. Hey, T., Tansley, S., & Tolle, K. (2019, October). *The fourth paradigm: Data-intensive scientific discovery*. Redmond: Microsoft Research.
63. Ganguly, A., Shekhar, S., Samatova, N., & Kumar, V. (2017, October). Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering, 29*(10), 2318–2331.

Corina Vaduva received the B.S. and Ph.D. degrees in Electronics and Telecommunications from the University "Politehnica" of Bucharest (UPB), Romania, in 2007 and respectively in 2011. In 2010, she performed a 6-month internship at the German Aerospace Center (DLR) in Oberpfaffenhofen, Germany, as a visiting Ph.D. student in the Remote Sensing Technology Institute (IMF). In 2012, she returned at DLR as a Visiting Scientist for 2 months. She is a research engineer with the Research Centre for Spatial Information, UPB, since 2007. Her research fields include Earth Observation image processing and data mining based on information theory, information retrieval, data fusion, change detection, temporal analysis in image time series and semantic annotation for very high-resolution image understanding and interpretation. She was Chair at the ESA Conference on Big Data from Space (BiDS) in 2016 and 2017. In 2010, Dr. Vaduva received a competition prize in the DigitalGlobe 8-Band Research Challenge.

Michele Iapaolo received the Master's Degree in Environmental Engineering and the Ph.D. in Geoinformation from the Tor Vergata University of Rome in 2002 and 2006, respectively. Since 2005 he is working at the European Space Agency (ESA-ESRIN) in the area of Research and Technology Development (RTD) for the ground segment. Activities carried out during more than 10 years mainly concern the fields of Image Information Mining, Machine Learning and Information Extraction applied to satellite imagery, the development of tools and infrastructures for Earth Observation data management, processing and distribution, user service provision and development of standards and semantic tools supporting the archiving and use of EO data. Since 2014 he has been involved in the organization (as part of both the Organising and Program

Committees) of the “Big Data from Space Conference” (BIDS2014, BIDS2016, BIDS2017) and from 2017 in the Big Data Technology Harmonisation process promoted by ESA to coordinate and harmonize European initiatives in the Big Data field.

Mihai Datcu (SM’04–F’13) received the M.S. and Ph.D. degrees in electronics and telecommunications from the University Politehnica of Bucharest (UPB), Bucharest, Romania, in 1978 and 1986, respectively. Since 1981, he has been a Professor in electronics and telecommunications with UPB. Since 1993, he has been a Scientist with the German Aerospace Center (DLR), Munich, Germany. Currently, he is a Senior Scientist and Image Analysis research group leader with the Remote Sensing Technology Institute (IMF), DLR, a coordinator of the CNES-DLR-ENST Competence Centre on Information Extraction and Image Understanding for Earth Observation and a Professor with Paris Institute of Technology/GET Telecom Paris. From 1991 to 1992, he was a Visiting Professor with the Department of Mathematics, University of Oviedo, Oviedo, Spain, and from 2000 to 2002 with the Universitouis Pasteur, and the International Space University, both in Strasbourg, France. From 1992 to 2002, he was a Longer Invited Professor with the Swiss Federal Institute of Technology ETH Zürich, Zürich, Switzerland. In 1994, he was a Guest Scientist with the Swiss Center for Scientific Computing (CSCS), Manno, Switzerland, and in 2003, he was a Visiting Professor with the University of Siegen, Siegen, Germany. His research interests include Bayesian inference, information and complexity theory, stochastic processes, model-based scene understanding, image information mining, for applications in information retrieval and understanding of high-resolution SAR and optical observations. Dr. Datcu is a member of the European Image Information Mining Coordination Group (IIMCG). In 1999, he received the title “Habilitation á diriger des recherches” from Universitouis Pasteur, Strasbourg, France.

Visualizing High-Dimensional Data Using t-Distributed Stochastic Neighbor Embedding Algorithm

Jayesh Soni, Nagarajan Prabakar, and Himanshu Upadhyay

Data visualization is a powerful tool and widely adopted by organizations for its effectiveness to abstract the right information, understand, and interpret results clearly and easily. The real challenge in any data science exploration is to visualize it. Visualizing a discrete, categorical data attribute using bar plots, pie charts are a few of the effective ways for data exploration. Most of the datasets have a large number of features. In other words, data is distributed across a high number of dimensions. Visually exploring such high-dimensional data can then become challenging and even practically impossible to do manually. Hence it is essential to understand how to visualize high-dimensional datasets. t-Distributed stochastic neighbor embedding (t-SNE) is a technique for dimensionality reduction and explicitly applicable to the visualization of high-dimensional datasets.

Contrary to PCA (principal component analysis), t-SNE is a probabilistic technique and not a mathematical one. Fundamentally it looks at the original data, which is fed as input into the algorithm and checks the best-fit representation of the data using fewer dimensions by matching distributions across dimensions. It is a machine learning algorithm for visualization. It transforms high-dimensional data into two or three dimensions for display. Precisely, it models in such a way that similar data points are joined with nearby locations, and different objects are joined with distant points. The t-SNE algorithm has been used in computer security research, music analysis, cancer research, and bioinformatics for visualization. It is used to

J. Soni (✉) · N. Prabakar
School of Computing and Information Sciences, Florida International University, Miami, FL, USA
e-mail: jsoni@fiu.edu; prabakar@fiu.edu

H. Upadhyay
Applied Research Center, Florida International University, Miami, FL, USA
e-mail: upadhyay@fiu.edu

© Springer Nature Switzerland AG 2020

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_9

visualize high-level representations learned by an artificial neural network. In this chapter, we analyze the hyperparameter optimization of the t-SNE algorithm on high-dimensional data using basic and advanced machine learning frameworks such as scikit-learn and popular deep learning TensorFlow. We also discuss issues and challenges related to t-SNE. In the end, we discussed the practical implementation of t-SNE using python on a novel example.

1 Introduction

With advances in data collection and storage technology, organizations and businesses generate a massive amount of data. Also, it results in high-dimensional datasets when different attributes are added. The first step is to visualize their structure, finding meaningful complex relationships and trends in data to gain useful information. The key idea is to produce a representation of the data in such a way that the human eye can gain insight into their structure and patterns.

Traditional data-handling applications fail to deal with big data due to its complexity and variety. Insights generating and understanding become challenging with an increase in data volume and variety. Large data sizes along with high dimension create issues such as algorithmic instability and high computational cost. It is challenging to identify the patterns and trends in big data due to high visualization rendering time.

Given these challenges, to optimize the visualization process, compression and reduction techniques should be used. Before making models with machine learning algorithms, feature extraction and feature selection should be applied. To reduce the number of attributes, dimensionality reduction techniques are used that generate the principal variables. To reduce the number of data points, numerosity reduction techniques are used. Smaller representations of data are estimated by parametric models and nonparametric models. Dimensionality reduction and numerosity reduction techniques improve operational efficiency involved with massive datasets by lowering computational and storage costs [1, 2]. Finally, dimensionality reduction techniques help in improving machine learning algorithm accuracy. The goal of this chapter is to study the impact and compare popular dimensional reduction algorithms on visualizing high-dimensional data.

The rest of the chapter is summarized as follows. Section 2 describes the related work, Sect. 3 explains the t-SNE algorithm, Sect. 4 compares t-SNE and PCA on the accessible MNIST dataset, Sect. 5 describes the use cases of t-SNE, Sect. 6 differentiates t-SNE and PCA, Sect. 7 describes the common fallacies of t-SNE, and, finally, we conclude in Sect. 8.

2 Literature Review

2.1 Dimensionality Reduction Technique

Dimensionality reduction technique develops a mathematical model that preserves the vital characteristics of the data while reducing the complexity of the data [3]. Depending on the structure of the underlying manifolds, it can be divided into linear and nonlinear methods. Linear dimensionality reduction algorithms learn linear structures, whereas nonlinear dimension reduction algorithms can study complex structures. Linear methods perform linear transformations and project the data into a lower-dimensional space. This approach is computationally faster and easy to execute. Linear methods while being computationally efficient might miss nonlinear structures in the data. To preserve neighborhood geometry, nonlinear methods map data points to lower-dimensional subspaces. Multidimensional scaling [4] infers the differences between data points in high-dimensional space while transforming it into low-dimensional space. To accelerate computation, many techniques have been proposed [5]. Isomaps [6, 7] build the model of manifold connectivity to preserve local neighborhood structures. But, these methods heavily rely on neighborhood structures and are computationally very intensive. Deep learning techniques such as neural networks [8–10] and auto-encoders [11] are heavily used for reducing the dimensions. An artificial neural network [12] is used to model the complicated relationship between input and output. It simulates the functional and structural features of biological neural networks. It is internal structure changes with the change in external input. While the deep learning model still gives impressive results, it lacks the theoretical aspects. Also, for time series data, there are many dimensionality reduction algorithms. One such method is a wavelet [13], which explores the frequential content of the signal. Further, feature extraction and feature selection methods can be used to perform linear and nonlinear dimensionality reduction techniques. We focus basically on feature extraction techniques.

2.2 Background on Dimensionality Reduction and Feature Extraction

The feature extraction method creates new features by transforming data from high dimensions to low dimensions while feature selection keeps essential features and discards the rest.

2.2.1 Feature Extraction Techniques: Principal Component Analysis (PCA)

Principal component analysis (PCA) reduces the dimensionality of a dataset while keeping the variation present in the features of the dataset [14]. PCA generates principal components which are sets of linear combinations of uncorrelated variables by transforming correlated variables. It achieves that by selecting the direction where the variance between data points is highest and then projects the original data in that direction. With this, it creates a scatter plot in two dimensions and that makes data easy to interpret.

PCA uses the eigenvectors of the covariance matrix to calculate the weights of the principal components [15]. The highest amount of variance is explicated by the first principal component since it has the least sum of the squared distances between the data points. The next principal component is calculated by extracting the residuals from the dataset. Subsequently, the successive principal component explains much less variance. Since the variance explained by the first principal component is highest, it is sufficient to select and it thus reduces the dimensionality of the problem. PCA, being an unsupervised learning algorithm, does not consider resultant variables. There are some challenges with PCA. First, it needs data to be normalized since it is sensitive to the scale of the variables. Without scaling, features on the most significant scale can overlook the new principal components. PCA needs a threshold to be tuned for the cumulative explained variance.

2.2.1.1 Limitations of PCA

PCA being a linear algorithm fails to interpret complex polynomial relationships between features. Linear dimensionality reduction algorithms place different data points far apart in a low-dimensional subspace compared to their original data points. Linear dimensionality reduction algorithms fail to represent similar data points close together to visualize high-dimensional data with fewer dimensions. Local methods map adjacent data points in high dimension to nearby points in the low-dimensional subspace. Global methods attempt to preserve geometry, i.e., retain the geometrical distance among data points. Most of the nonlinear dimensionality reduction techniques except t-SNE fail in retaining both the global and local aspects of the data.

2.2.2 Feature Extraction Techniques: t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed stochastic neighbor embedding (t-SNE) is a dimensionality reduction method suited for transforming high-dimensional data into low-dimensional space for visualization. It includes two main phases. In the first phase, t-SNE generates a probability distribution so that different points have a low probability of being

chosen over pairs of high-dimensional points while nearby points have a higher probability of getting selected. In the second phase, t-SNE minimizes the Kullback-Leibler deviation between the two distributions by calculating the probability distribution over the data points in the low-dimensional space. Kullback-Leibler deviation measures the difference between two probability distributions. Since t-SNE is a nonlinear algorithm, it gives better visualization even when the underlying relationship is nonlinear. It does that by preserving the relative distance between observations and subsequently creates plots such as a scatterplot, assisting in visualization. t-SNE works well on finding patterns in the dataset. Unfortunately, the patterns from t-SNE are difficult to interpret [16]. t-SNE is a dimensionality reduction algorithm rather than a clustering algorithm. The generated features are different because it transforms high-dimensional data into a lower-dimensional space. Thus, it is hard to make any implication based solely on t-SNE output. However, the t-SNE output can become the input feature for classification and clustering algorithms [17]. t-SNE is considered a black box-type algorithm since it doesn't give similar output with successive runs.

3 Algorithmic Details of t-SNE

t-SNE is an improved version of the stochastic neighbor embedding (SNE) algorithm.

3.1 Algorithm

Stochastic neighbor embedding (SNE) calculates Euclidean distances in high-dimensional space. Further to represent similarities, it transforms calculated Euclidean distances into conditional probabilities. Conditional probability $p_{j|i}$ is the similarity between data point x_i and data point x_j such that it would elite x_j as its neighbor if neighbors were selected under a Gaussian distribution centered at x_i. For nearby points, $p_{j|i}$ is relatively high, whereas for broadly separated data points, $p_{j|i}$ will be almost infinitesimal. Scientifically, the conditional probability $p_{j|i}$ is given by

$$p_{j|i} = \frac{\exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

where σ_i is the Gaussian variance centered on data point x_i.

Step 1:

The algorithm calculates the shortest distance between the points and converts it into the probability of resemblance of points. The similarity between data points is the conditional probability that x_j would be picked by x_i as its neighbor if neighbors were nominated under a Gaussian (normal distribution) aligned at x_i.

Step 2:

SNE uses the gradient descent algorithm to reduce the sum of Kullback-Leibler (KL) divergence. The SNE cost function retains the local aspects of the data points. Optimizing this cost function is computationally inefficient. Therefore, to reduce the sum of the difference in conditional probabilities, t-SNE uses the symmetric form of the SNE cost function. It does that with gradients.

Step 3:

t-Distribution's variance is selected as the parameters and such variance is centered over data point x_i in high-dimensional subspace. Since the density of the data varies, it is unlikely that there is a single optimal value for all data points. With an increase in the variance, entropy of the distribution increases. t-SNE yields P_i with a user-specified perplexity value. The perplexity function is defined as

$$\mathrm{Perp}\,(P_i) = 2^{\mathrm{H}(P_i)}$$

where the Shannon entropy of point P_i is $H(P_i)$ which is defined as

$$H(P_i) = -\sum\nolimits_j K \ \log \ K$$

where $K = P_{j|i}$.

The perplexity is a measure of an adequate amount of neighbors. Typical values of perplexity are between 5 and 50 and are relatively robust. To minimize the cost function, gradient descent is used. And it is the resulting force created between the point $\mathrm{y_i}$ and all other points y_j.

3.2 Space and Time Complexity

For lower and higher dimensions, the algorithm minimizes the sum of their probabilities by computing pairwise conditional probabilities. Since this requires a lot of computations, this algorithm needs additional computational resources.

t-SNE requires more computational power when applied to larger datasets such as dataset with more than 10,000 observations since its space and time complexity is quadratic in nature.

3.3 Effective Use of t-SNE

t-SNE plots can sometimes be ambiguous, although it is extremely useful for visualizing high-dimensional data. We can learn to use it efficiently by exploring how it performs in simple cases.

We will walk through a sequence of simple examples to exemplify what t-SNE diagrams can and cannot display. Maintaining attention between global and local aspects of the dataset is one of the features of t-SNE. A tuneable parameter for balancing attention is known as "perplexity." This parameter guesses the total number of close neighbors for each data point. Resulting pictures are often affected by the perplexity value. One must analyze multiple plots with different perplexities to get the most from t-SNE.

3.3.1 Hyperparameter Changes the Whole Show

We consider clusters in a 2D plane to make things as simple as possible, as shown in Fig. 1. (For clarity, the two clusters are color coded.) Figure 2 shows the t-SNE plots for five different perplexity values.

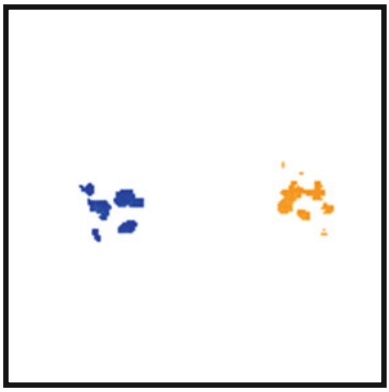


Fig. 1 Clusters in a 2D plane

Fig. 2 t-SNE plots

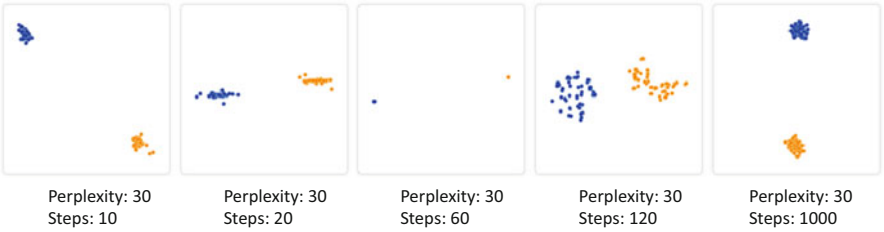


Fig. 3 t-SNE plots on different runs

Although with very different shapes, the above diagrams do show these clusters when the perplexity values in the range (5–50). Outside that range, patterns get a little strange. Local variations dominate with perplexity 2. The algorithm performs poorly with perplexity 100 which indicates that the perplexity value should be smaller than the total number of data points.

We use a learning rate of 10 with 5000 iterations for each of the above plots, and we see a stable point by step 5000. The most important thing is to repeat until reaching a stable stage.

The images in Fig. 3 show five different rounds at perplexity 30. After 10, 20, 60, and 120 stages, we can see point-like images of the cluster layouts with apparent one-dimensional space. We would see a t-SNE plot with strange "pinched" shapes if the process was stopped a little early. There is no fixed number of steps that give a stable result. Different datasets require a varying number of runs to converge.

A question arises whether the same hyperparameters with different runs produce the same results or not. Multiple runs give the same global shape in this simple two-cluster example. From now on, shown results are run with 5000 iterations, which are enough for convergence in the (relatively small) instances. We will show runs with different ranges of perplexities since that makes a big difference.

3.3.2 Cluster Sizes in a t-SNE Plot

What if the two clusters have different sizes and different standard deviations (Fig. 4)? Figure 5 shows t-SNE plots for the data points, where one group is ten times as dispersed as the other.

Astonishingly, the two clusters look about the same size. The t-SNE algorithm expands dense clusters, and contracts sparse ones, to make cluster sizes even.

3.3.3 Distances Between Clusters Might Not Mean Anything

What about distances *between* clusters? Figure 6 shows three Gaussians of 50 points each, one pair being five times as far apart as another pair.

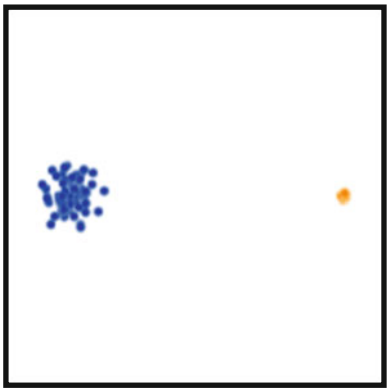


Fig. 4 Cluster with different std. deviation

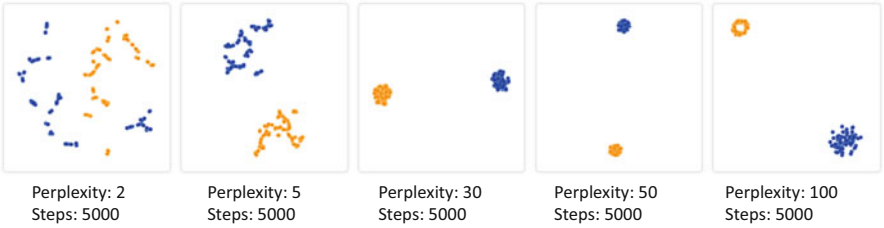


Fig. 5 t-SNE plots for the cluster with different std. deviation

Fig. 6 Distant clusters of 50 points each

In Fig. 7, at perplexity 50, we see global geometry. The clusters look intermediate for lower perplexity values. We see the correct global geometry with the perplexity of 100, but one of the clusters seems, deceptively, much smaller than the others. To understand global geometry, one should not set perplexity to 50 always, since the perplexity has to increase with an increasing number of points in each cluster. Figure 8 shows the t-SNE diagrams of three different clusters with 200 points depicted in Fig. 9 (Gaussian distributed) each, instead of 50. Now we notice that none of the trial perplexity values gives a good result.

Fig. 7 t-SNE plots on distant clusters of 50 points each

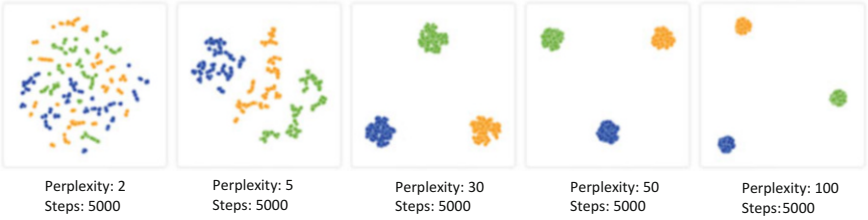
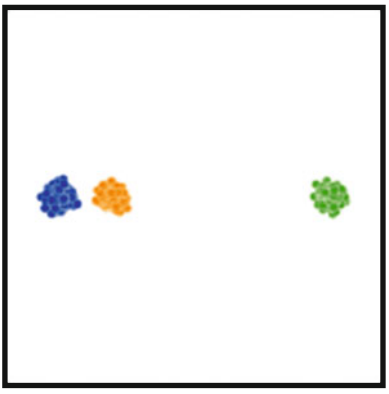


Fig. 8 t-SNE plots on distant clusters of 200 points each

Fig. 9 Distant clusters of 200 points each



It requires to fine-tune perplexity to see global geometry. Real-world datasets have varying numbers of clusters, each with different numbers of features. Since perplexity is a global parameter, we have to try with varying values of perplexity to capture distances across all clusters. It is an exciting area of research to solve this problem. The underlying meaning is that distances between well-separated clusters in a t-SNE plot may mean nothing.

3.3.4 The Necessity of More Than One Plot for Topology

Containment is one of the most straightforward topological properties. Figure 10 shows two clusters of 75 points in 50-dimensional space. One is 50 times more tightly isolated than the other. Samples are taken from symmetric Gaussian distributions. The "small" distribution is contained in the large one.

t-SNE greatly overstate the size of the smaller group of points at perplexity 30. At perplexity 50, there's a new occurrence; the outer group becomes a circle, as the plot tries to represent the fact that all its points are at equidistance from the inner group (Fig. 11).

The trefoil knot shown in Fig. 12 is an example of how multiple runs affect the results of t-SNE. Figure 13 shows the results of five runs each at perplexity 2. The algorithm preserves the intrinsic topology. It introduces artificial breaks in four of the runs and ends up with four different solutions. We also observe that the first and third runs are far from each other.

Figure 14 shows five runs at perplexity 50; however, it gives results that (up to symmetry) are visually identical.

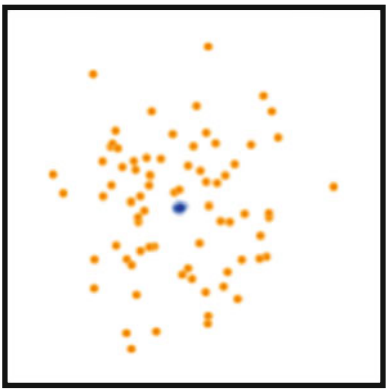


Fig. 10 Two clusters in 50-dimensional space

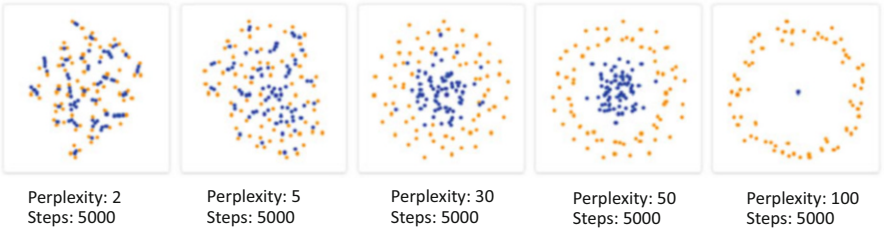


Fig. 11 t-SNE plots on two clusters in 50-dimensional space

Fig. 12 Trefoil knot

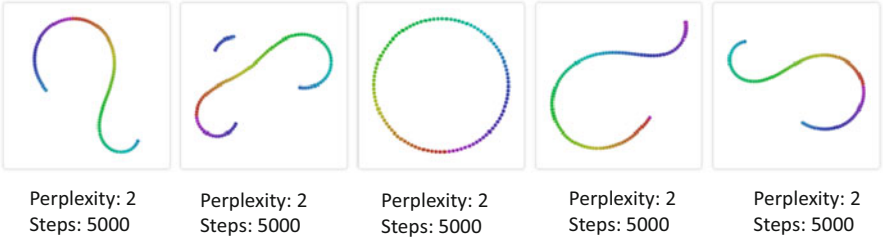


Fig. 13 t-SNE plots on trefoil knot with perplexity-2

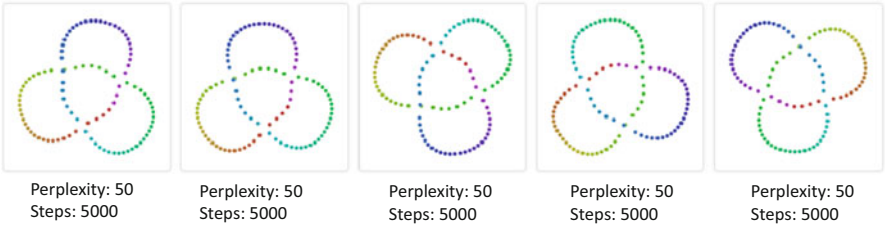


Fig. 14 t-SNE plots on trefoil knot with perplexity-50

4 t-SNE Versus PCA on MNIST Dataset

MNIST is handwritten digit dataset where each digit is of 28 × 28 pixel, as shown in Fig. 15.

For example, the number "1" can be represented in an array as shown in Fig. 16.

We get a 28 × 28 array since each image has 28 by 28 pixels. We then flatten each array into 784 ($28_{*}28$) dimensional vector. The values in the array are between zero and one describing the pixel intensity.

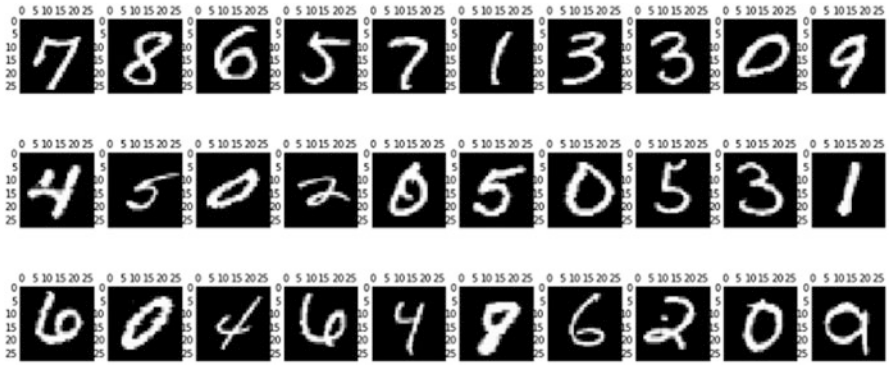


Fig. 15 MNIST dataset

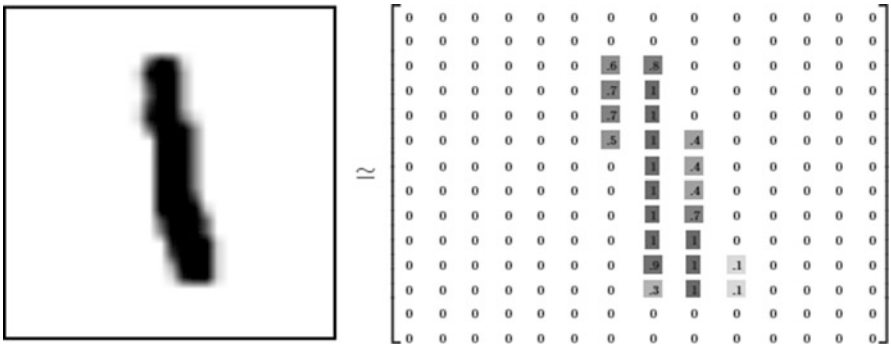


Fig. 16 A part of the pixel view of a single digit in MNIST dataset

4.1 Dimensional Reduction Using PCA on MNIST

Since it is hard for humans to visualize data with more than three dimensions, let us generate the first two principal components from the original 784 dimensions. And we will see how much of the variation is explained by those principal components.

Now, we can create a scatterplot of the first two principal components and color them w.r.t. digits. If we are fortunate, the same type of digits will be clustered together in groups, which would mean that with the first two principal components, we can classify the specific types of digits.

From the graph in Fig. 17, we can see the two components provide some information, especially for specific digits, but not enough that can set all of them apart.

Despite minor successes like these, it is hard to visualize the MNIST dataset with PCA.

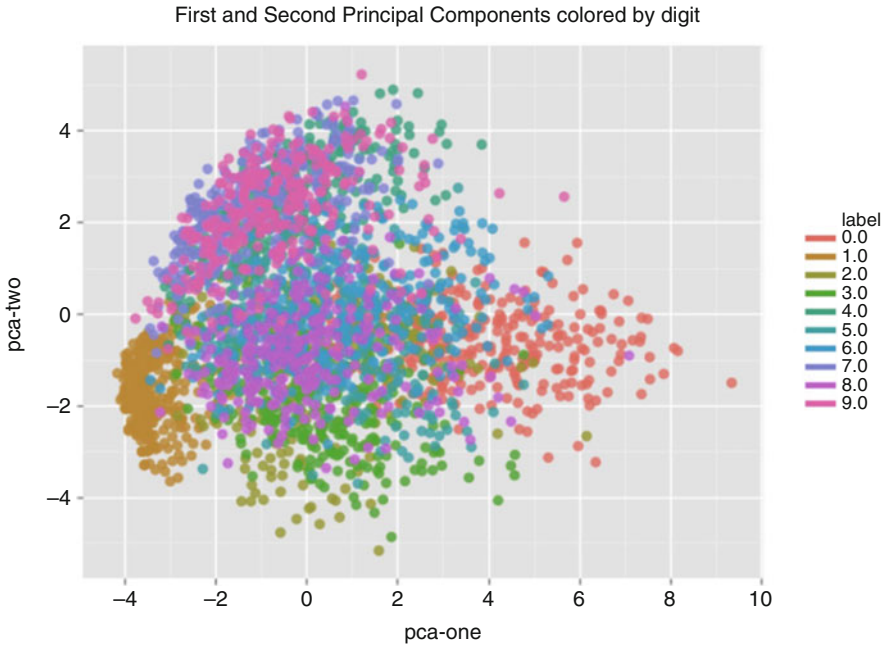


Fig. 17 PCA on MNIST dataset

4.2 Dimensional Reduction on MNIST Using t-SNE

For every point, t-SNE tries to make all points have the same number of neighbors by constructing a perception of neighbors with nearby points. t-SNE is a graph-based visualization.

This is a noteworthy improvement compared to PCA visualization, where the digits are clustered in their respective clusters. t-SNE is effective at revealing clusters and subclusters in data. It does an impressive job of finding clusters and subclusters in the data but is prone to get stuck in local minima (Fig. 18).

4.3 t-SNE Usage

1. *Data Scientist:* t-SNE is the black box-type algorithm for the data scientist. Also, it gives different results with different runs. The best use of this algorithm is for experimental data analysis, where we can easily visualize the hidden structure of the data. Furthermore, it can become the input parameter for clustering and classification algorithms.

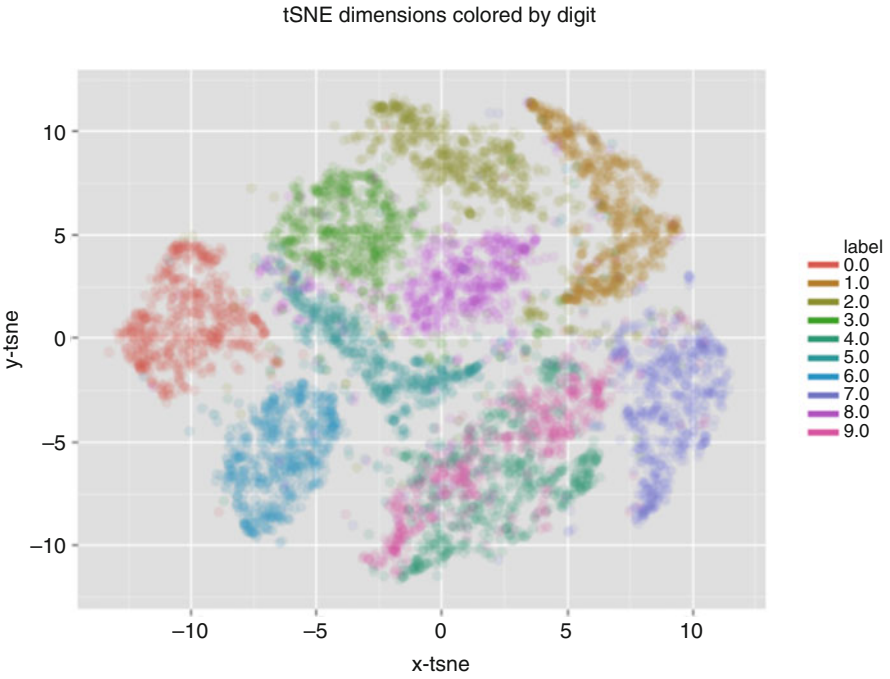


Fig. 18 t-SNE on MNIST dataset

2. *Deep Learning/Machine Learning Engineer:* Reduce the dataset to two or three dimensions using a holdout set for stacking, and stack this with a nonlinear function. Then, to get enhanced results, use XGBoost on t-SNE vectors. In terms of research and performance enrichments, this algorithm provides good openings for data science enthusiasts. Few researchers used linear functions to reduce the time complexity of the algorithm. But an optimal solution is still required. Exploring t-SNE for image processing and NLP problems has a broad future scope.

5 Application of t-SNE

t-SNE has been used extensively in many cases. The following are a few examples:

1. *Facial Expression Recognition (FER)*

For FER, algorithms like decision tree, random forests, NNs, AdaBoost, and other classifiers are used for the facial expression classification after deducing low-dimensional subspace from high-dimensional data using t-SNE.

Table 1 SVM and AdaBoost accuracy w.r.t PCA, SNE, and t-SNE

	PCA	SNE	t-SNE
SVM	73.5%	89.6%	90.3%
AdaBoost	75.4%	90.6%	94.5%

In one example, researchers used t-SNE to decrease the dimension of the Japanese Female Facial Expression (JAFFE) database and further applied AdaBoost for facial expression classification. Experimental results showed in Table 1 depicts that the t-SNE provided better performance than traditional algorithms [18–20].

2. *Identifying Tumor Subpopulations*

To extracts the 3D distribution of biomolecules from tissue, mass spectrometry imaging (MSI) is used. t-SNE can reveal tumor populations that are statistically related to patient survival in primary tumors of breast cancer. t-SNE clusters used for survival analysis provide significant results [21].

3. *Wordvec-Based Text Comparison*

Dialectal properties such as plurality, semantic notions, and gender are captured by word vector representations. A two-dimensional plot can be computed where like words can be semantically close to each other using dimensionality reduction techniques. This helps researchers to visualize text-based data more like a geographical map [22].

6 t-SNE Versus PCA

Even though both t-SNE and PCA have their benefits and drawbacks, some crucial dissimilarities between t-SNE and PCA can be distinguished as follows:

- t-SNE takes several hours on large datasets and is computationally expensive while PCA finishes in seconds or minutes.
- Sometimes, t-SNE may give different results with different runs with the same hyperparameters; hence, several plots have to be observed before making any evaluation, whereas PCA gives the same results.
- t-SNE is a probabilistic method while PCA is a mathematical one.
- PCA fails to interpret the complicated polynomial relationship between features due to its linear characteristics while t-SNE easily captures the polynomial relationship.
- Due to the application of the Gaussian kernel in the high-dimensional space, t-SNE performs a smooth border function on the structure of the data. It determines the regional neighborhood scope separately for each data point depending on the dataset's local concentration.

7 Common Fallacies of t-SNE

Below are a few mutual mistakes to evade while interpreting the behavior of t-SNE:

- The same hyperparameters may produce different results with different runs.
- The perplexity value should be less than the number of points for the algorithm to execute correctly. The recommended perplexity value should be between 5 and 50.
- One should not rely on the cluster sizes in any t-SNE plots to evaluate statistical measures such as standard deviation and dispersion. This is because of t-SNE contracts sparser clusters and expands denser clusters to have uniform cluster sizes and that make t-SNE to produce clear plots.
- Diverse perplexity values give different cluster shapes.
- In a dataset with different numbers of elements and clusters, a single perplexity value cannot optimize distances for all clusters. Since perplexity changed the global geometry, distances between clusters may change.
- Before making any assessment to analyze the topology, various plots have to be witnessed, and it should not be based on a single t-SNE plot.
- Random noise also has patterns, so several runs with different sets of hyperparameters must be tested before determining if a pattern is present in the data [23].

8 Conclusion

Dimensionality reduction is a well-established area, and we're only scraping the surface here. It's easy to think that one of these techniques is better than the others, but they all have their own characteristics. It's difficult to preserve all the structures while mapping high-dimensional data into low-dimensional data. So, any approach has to sacrifice one property to keep another, and trade-offs have to be made. PCA preserves the linear structure and t-SNE preserves neighborhood structure. These techniques give us a way to visualize high-dimensional data in different contexts.

References

1. Bellman, R. E. (1961). *Adaptive control processes: A guided tour* (p. 197). Princeton: Princeton University Press.
2. Negrel, R., Picard, D., & Gosselin, P. -H. (2014). Dimensionality reduction of visual features using sparse projectors for content-based image retrieval. In *IEEE International Conference on Image Processing*, Paris, France.
3. Pavel Pudil, J. N. (1998). Novel methods for feature subset selection with respect to problem knowledge. In H. M. Huan Liu (Ed.), *Feature extraction, construction and selection: A data mining perspective* (The Springer international series in engineering and computer science) (Vol. 453, pp. 101–116). New York: Springer.

4. Bae, S.-H., Qiu, J., & Fox, G. (2012). High performance multidimensional scaling for large high-dimensional data visualization. In *IEEE transaction of parallel and distributed system*.
5. Ingram, S., Munzner, T., & Olano, M. (2009). Glimmer: Multilevel MDS on the GPU. *IEEE Transactions on Visualization and Computer Graphics, 15*(2), 249–261.
6. Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323.
7. Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.
8. Wang, W., Huang, Y., Wang, Y., & Wang, L. (2014). Generalized autoencoder: A neural network framework. In *IEEE conference on computer vision and pattern recognition workshops*.
9. Soni, J., Prabakar, N., & Upadhyay, H. (2019, December 05–07). Behavioral analyses of system call sequences using LSTM Seq-Seq, cosine similarity and Jaccard similarity for real-time anomaly detection. In *Proceedings of the 2019 International Conference on Computational Science and Computational Intelligence (CSCI'19)*, Las Vegas, NV.
10. Soni, J., Prabakar, N., & Upadhyay, H. (2019, May 15–17). *Deep learning approach to detect malicious attacks at system level*. In *WiSec'19: Proceedings of 12th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, Miami, FL, USA.
11. Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing, 184*(C), 232–242.
12. Soni, J., Prabakar, N., & Upadhyay, H. (2019). *Feature extraction through deepwalk on weighted graph.* In *Proceedings of the 15th International Conference on Data Science (ICDATA'19)*, Las Vegas, NV.
13. Rioul, O., & Vetterli, M. (1991). Wavelets and signal processing. *IEEE Signal Processing Magazine, 8*, 14–38.
14. Jolliffe, I. (2002). *Principal component analysis* (2nd ed.). New York: Springer-Verlag.
15. Reid, S., (2014, October). Dimensionality reduction techniques. *Turing Finance* [Online]. Available http://www.turingfinance.com/artificialintelligence-and-statistics-principal-component-analysis-and-self-organizing-maps/
16. Kwon, H., Fan, J., & Kharchenko, P (2017), *Comparison of principal component analysis and t-Stochastic neighbor embedding with distance metric modifications for single-cell RNA-sequencing data analysis*, bioRxiv.
17. Yi, J., Mao, X., Xue, Y., & Compare, A. (2013). *Facial expression recognition based on t-SNE and AdaboostM2*. In 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing.
18. Van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.
19. Soni, J., Prabakar, N., & Kim, J.-H. (2017). Prediction of component failures of telepresence robot with temporal data. In *30th Florida Conference on Recent Advances in Robotics*.
20. Soni, J., & Prabakar, N. (2018). Effective machine learning approach to detect groups of fake reviewers. In *Proceedings of the 14th International Conference on Data Science (ICDATA'18)*, Las Vegas, NV.
21. Abdelmoula, Balluff, B., Englert, S., Dijkstra, J., Reinders, M. J. T., Walch, A., McDonnell, L. A., & Lelieveldt, B. P. F. (2016). Data-driven identification of prognostic tumor subpopulations using spatially mapped t-sne of mass spectrometry imaging data. *Proceedings of the National Academy of Sciences, 113*(43), 12244–12249.
22. Heuer, H. (2015). Text comparison using word vector representations and dimensionality reduction. In *Proceedings of the EuroSciPy*, pp. 13–16.
23. https://distill.pub/2016/misread-tsne/

Active and Machine Learning for Earth Observation Image Analysis with Traditional and Innovative Approaches

Corneliu Octavian Dumitru, Gottfried Schwarz, Gabriel Dax, Vlad Andrei, Dongyang Ao, and Mihai Datcu

1 Introduction

Today we are faced with impressive progress in machine learning and artificial intelligence. This not only applies to autonomous driving for car manufacturers but also to Earth observation, where we need reliable and efficient techniques for the automated analysis and understanding of remote sensing data.

While automated classification of satellite images dates back more than 50 years, many recently published deep learning concepts aim at still more reliable and user-oriented image analysis tools. On the other hand, we should also be continuously interested in innovative data analysis approaches that have not yet reached widespread use.

We demonstrate how established applications and tools for image classification and change detection can profit from advanced information theory together with automated quality control strategies. As a typical example, we deal with the task of coastline detection in satellite images; here, rapid and correct image interpretation is of utmost importance for riskless shipping and accurate event monitoring.

If we combine current machine learning algorithms with new approaches, we can see how current deep learning concepts can still be enhanced. Here, information theory paves the way toward interesting innovative solutions.

In the following, we will describe the goals and implementation steps of the European research project ExtremeEarth funded by the European Union [2] that aims at a reliable interpretation of satellite radar images with the help of current and upcoming developments in machine learning and artificial intelligence. While

C. O. Dumitru (✉) · G. Schwarz · G. Dax · V. Andrei · D. Ao · M. Datcu
Remote Sensing Technology Institute, German Aerospace Center (DLR), Weßling, Germany
e-mail: corneliu.dumitru@dlr.de; gottfried.schwarz@dlr.de; gdax.its-m2017@fh-salzburg.ac.at; vlad.andrei@tum.de; mihai.datcu@dlr.de

© Springer Nature Switzerland AG 2020

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_10

existing techniques for land, water, and ice cover monitoring often rely on optical satellite missions and manual data interpretation to create up-to-date Earth surface cover charts, these already established techniques do not always allow the fully automated generation of highly accurate charts without manual interaction.

On the other hand, the already existing experiences with satellite images and interactions with the user communities have already led to the definition of local image content categories [3] that should be recognized by any new image analysis systems. Since traditional image processing techniques mainly rely only on pixel brightness statistics and local neighborhood relationships, we cannot expect that all the human knowledge and expertise being available for manual and interactive image interpretation can be included in image interpretation by fully automated systems unless these systems provide access to geophysical databases, machine learning capabilities, and artificial intelligence.

The approaches being described below shall remedy this situation and demonstrate the necessary functionality and how to test the correctness of the results. Thus, already existing expert knowledge can be coupled with up-to-date satellite images. We expect that this type of image analysis systems will become a prevalent system approach for future climate and land cover monitoring tools.

In the following, we describe the design principle of a self-contained software package for radar data interpretation for two use cases (i.e., one agricultural/flooding scenario in Romania and one polar ice scenario near Greenland), where we aim at efficient and reliable solutions for each single scenario. Thus, the selected algorithms differ slightly despite their common goal of radar image data interpretation. As we aimed at general solutions, we did not have to observe specific constraints with respect to hardware platforms or programming languages. Instead, our goal was to provide a basic toolbox allowing us to interpret and understand radar satellite images as made available by the European Sentinel-1 satellites. Besides image analytics of routinely processed satellite data, our software package also had to provide the generation of reference data with known content and characteristics (so-called benchmark data). At the moment, our complete software package supports two research projects funded by the European Union: the ECOPOTENTIAL project [4] and the upcoming ExtremeEarth project [2]. These two projects are typical examples of current and future Earth observation research.

As radar images are characterized by rather peculiar imaging phenomena, we finally selected three algorithms for image analysis: normalized compression distance (NCD) [5], auto-encoders as being used in machine-learning applications [6], and the well-known classic Canny detector [7]. The main motivations for selecting these algorithms were, in the case of NCD, its direct applicability without any need for application-dependent feature selection and the successful implementation of a complete set of routines that run very fast, where we only had to resort to the processing of sequences of adjacent image patches. We did not find too many applications of NCD for radar images; therefore, we adapted a robust algorithm by Burrows and Wheeler [8] to our needs. In contrast, we additionally selected auto-encoders as they already have been analyzed by many authors, they are very often exploited in current machine learning applications, they

have a clear theoretical background, and they can be controlled easily by individual parameter settings. Finally, we also propose Canny detectors as reliable workhorses in image processing. They already have found numerous applications, for instance, for land/sea separation in satellite images. In our case, we only had to adapt some parameters for efficient ice/sea separation.

This chapter describes typical use cases of SAR satellite images where one wants to extract content descriptions from radar data, such as images of agricultural areas or Artic ice cover. This chapter is organized as follows: Section 2 presents the characteristics of our image data, while Sect. 3 outlines our image analysis methodology to generate reference data; Section 4 describes the use of normalized compression distance for change detection exploiting the data from Sect. 2 and reference data from Sect. 3; Section 5 presents a method based on auto-encoders for feature extraction and takes this input to feed a Support Vector Machine and a k-NN classifier; Section 6 explains a methodology for the detection of coastlines and ice using polarimetric radar data; finally, Sect. 7 contains a summary of this chapter and future perspectives. This chapter ends with an acknowledgment and a list of references.

2 Data Set Description

In the field of radar remote sensing, there are many satellites for different applications. In this chapter, for our proposed charting applications, we chose a typical Synthetic Aperture Radar (SAR) satellite system, namely, Sentinel-1 [9]. The Sentinel-1 mission comprises a constellation of two satellites, operating in C-band for SAR imaging. Sentinel-1A has been launched on April 1, 2014, while Sentinel-1B has been launched 2 years later on April 25, 2016.

SAR has the advantage of operating at wavelengths not impeded by thin cloud cover or a lack of solar illumination and can acquire data over large areas during day or night time with nearly no weather condition restrictions. The repeat period of each satellite is 12 days, which means every 6 days there may be an acquisition by one of the two satellites.

The selection of the two scenarios is in close relation with two European projects funded by the European Union, namely, ECOPOTENTIAL [4] that focuses on a targeted set of internationally recognized protected areas in Europe and ExtremeEarth [2] that plans to develop analytics techniques and technologies for Copernicus satellite data with information and knowledge extraction, and exploiting this in the frame of Thematic Exploitation Platforms (TEPs) of the European Space Agency (ESA) [10]. For demonstration of advanced image analytics in a data science environment, we selected one scenario from the first project, namely, the Danube Delta in Romania, and one scenario from the second project, namely, Belgica Bank in the north-east of Greenland.

The Danube Delta is the second largest river delta in Europe and is the best preserved one on the continent [11]. The Delta is formed around the three main

channels of the Danube, named after their respective ports Chilia (in the north), Sulina (in the middle), and Sfantu Gheorghe (in the south). The greater part of the Danube Delta lies in Romania (Tulcea County), while its northern part, on the left bank of the Chilia arm, is situated in Ukraine (Odessa Oblast). Its total surface is 4152 km^2 of which 3446 km^2 is in Romania. The waters of the Danube, which flow into the Black Sea, form the largest and best preserved delta in Europe. In 1991, the Danube Delta was inscribed on the UNESCO World Heritage List due do its biological uniqueness.

In contrast, Belgica Bank is an ice-covered area along the north-east coast of Greenland (around 79°N and 16°E). The ice in this area can be locally classified as first-year ice or multiyear ice; in addition, ice development, ice ridges, hummocks, actual ice edge positions, leads within the ice cover, floating icebergs, etc., can be identified and charted. The classification of different ice types is supported by well-known polarimetric image analysis techniques. For further details about ice features occurring in different seasons, see [12, 13].

For image analysis, we selected Sentinel-1 level-1 images of Interferometric Wide (IW) swath mode and its GRDH (Ground Range Detected High resolution) option containing medium-resolution data. Their pixel spacing is 10 m, and the ground range resolution is 20 m; the data is acquired in dual polarization VV and VH for Danube Delta data and HH and HV for Greenland. The images were taken from ascending or descending orbit branches, with an incidence angle of 39° for both test areas.

For the first area, the Danube Delta, 15 processed Sentinel-1 images (with a size of 25,300×16,697 pixels) acquired within a period of 7 months from November 20, 2015, till May 18, 2016, cover the autumn, winter, and spring season, while for Belgica Bank area, two processed Sentinel-1 images (with a size of 25,673×16,641 pixels) were used, one from April 17, 2018, and one from June 16, 2018, covering the winter and summer season. All images were tiled into a series of patches with a size of 128×128 pixels and classified into several categories by using our existing semantic annotation catalogue [14] or by using the MANICE catalogue [12].

3 Ground Truth Data Generation

3.1 Motivation

In order to generate ground truth data, we are using the EOMiner tool which was developed by us in an ESA-funded project [15]. EOMiner is a semantic annotation and knowledge discovery tool based on a kernel Support Vector Machine (SVM) and statistical decisions using a multi-scale partition of the input data. The tool operates interactively, starting with an image classification at a coarse scale, discarding non-interesting data, and continues at finer scales. In contrast to conventional approaches, it accelerates learning by up to two orders of magnitude.

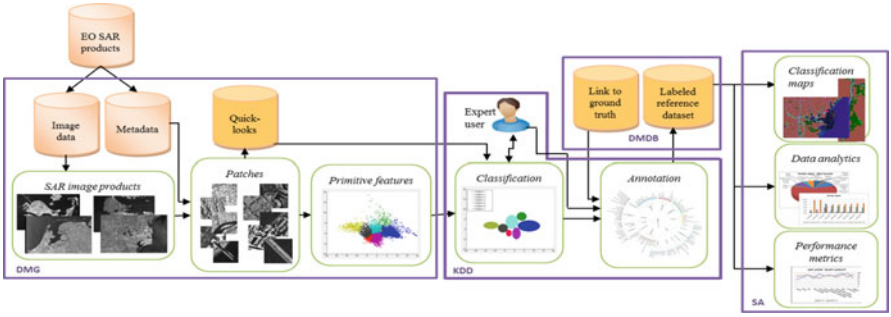


Fig. 1 The system used to generate ground truth data

3.2 Methodology

We used EOMiner to generate large Earth Observation (EO) image benchmark data sets with Sentinel-1 and with other satellite images [16]; EOMiner is composed of several modules which offer different functionality (see Fig. 1).

Its data model generation (DMG) component consists of a processing chain that extracts information items from an EO product and its metadata. The main functionality of DMG is metadata extraction, tiling an image with multiple spatial grid sizes, basic feature extraction, and generation of high-resolution quick-look images for visualization purposes. All this information is structured and saved into our data mining database. The available feature extraction methods are patch/tile-oriented and based on Gabor and Weber Local Descriptor methods [17, 18]. Each image can be tiled with different grid sizes.

The Data Mining Database (DMDB) provides the data model for actionable high-speed access to the extracted information, called data model generation (DMG). The DMDB component manages data handling, storage, administration, and some of the processing for all system components. It is based on the publicly available relational database MonetDB [19].

The additional Knowledge Discovery in Databases (KDD) component is based on machine learning methods to explore the image data, to classify their content, and to define semantic labels. The semantic definition is an active learning process, based on training data selected interactively by the user; the algorithm provides a prediction of the results. The annotation can be performed with different image tile sizes (e.g., 256×256, 128×128, or 64×64 pixels). This is achieved by the implementation of a cascaded learning algorithm [20]. For both target areas of Section 2, we generated two levels of annotation [16] with 11 categories for the Danube Delta and eight categories for Belgica Bank.

The remaining module is our Statistical Analysis (SA) tool that generates classification maps of each image and statistical values of the retrieved categories in an image. This module also has the possibility to generate data analytics as well as performance metrics.

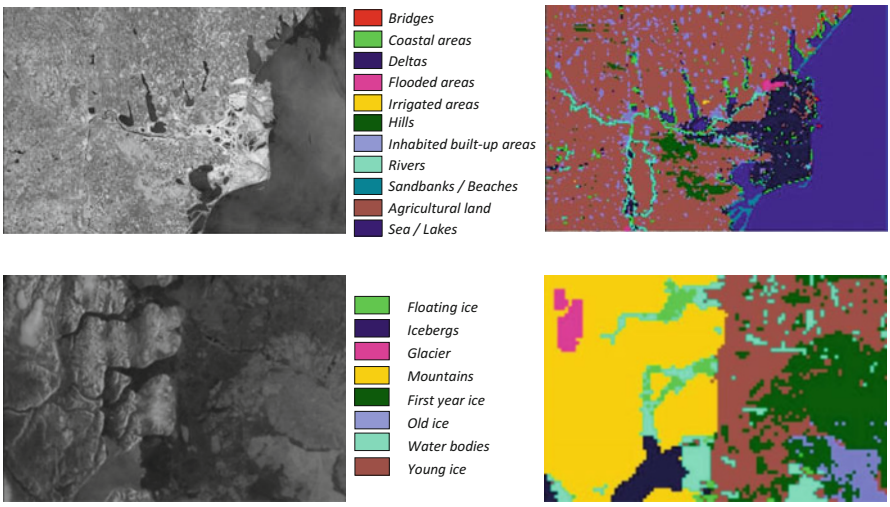


Fig. 2 (**a**) Sentinel-1A quick-look view (left) and classification map (right) for an image of the Danube Delta. (**b**) Sentinel-1A quick-look view (left) and classification map (right) for an image of Belgica Bank

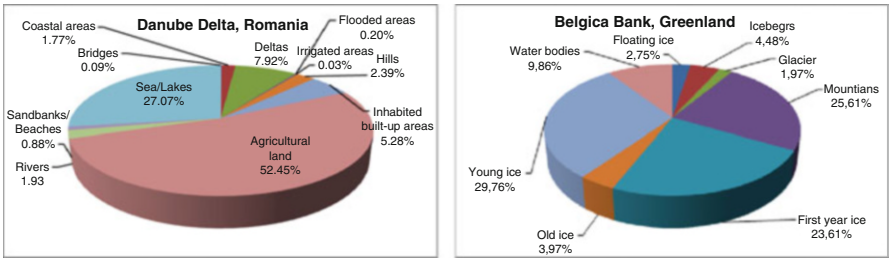


Fig. 3 Diversity of categories identified from a single image of the Danube Delta (left) and of Belgica Bank (right)

3.3 Experimental Results

By using EOMiner, we are able to generate a reference ground truth data set for both areas of interest. Figures 2 and 3 show the classification maps of each area of interest together with the diversity of categories identified from a single image. The reference data for the Danube Delta was generated using data acquired on May 18, 2016, and for Belgica Bank with data acquired on April 17, 2018.

3.4 Conclusions

The proposed methodology is able to generate reference data sets, using different input sources (e.g., TerraSAR-X, Sentinel-1, Sentinel-2, and Sentinel-3, but also

other satellite data in GeoTIFF format such as Worldview and Landsat). For our two target areas, the input source is Sentinel-1, and the generated data has an accuracy of about 0.95 (in precision/recall) [16].

4 Normalized Compression Distance for Change Detection

4.1 Motivation

In the past years, many natural disasters had a strong impact on the affected areas, including long-term damages that can be seen in satellite images. Furthermore, the climate itself is changing, which has a considerable impact on the environment [21]. Therefore, the goal is to detect these changes, such as flooded areas or melting polar ice in a satellite image time series, and to create binary change maps, as well as undirected graphs of an affected area showing the (dis-)similarities between local target classifications [21]. In the following, we describe a systematic approach to generate and classify changes in a sequence of overlapping satellite images.

4.2 Methodology

In order to create a *binary change map (BCM)* of a distinguishing region, several processing steps have to be made. The first phase is a local classification, for instance, being based on the normalized compression distance (NCD) described in [5]. This method to create a similarity distance between selected data patches ("strings") uses a compressor to classify the data depending on their similarity. This approach uses the Kolmogorov complexity and is defined as follows:

$$\mathrm{NCD}\,(x,\,y) = \frac{C\,(x,\,y) - \min\left\{C_{(x)},\,C_{(y)}\right\}}{\max\left\{C_{(x)},\,C_{(y)}\right\}},$$

where C is the length of the compressed string, for instance, x. Furthermore, the authors of [22] showed that when the strings are replaced with patches from an image, the NCD can detect changes between image pairs. Consequently, one can calculate a binary change map (BCM) where white stands for changes and black means no changes.

The method to calculate a BCM as well as the corresponding similarity graph requires the following processing steps:

Step1*: *Preprocessing: In order to cut out a common region of interest (ROI), all images of a time series must be co-aligned. To avoid unused areas in the ROI, the size of the region should be a multiple of the patch dimensions. Then the patches are created with a unique identifier within an image.

Step2*: *Calculate the NCD distance matrix: The patches corresponding to the same position of two co-aligned images are read in pixel wise and compared via the

previous equation. The result of the comparison of two satellite images is a grayscale change image.

Step3: *Apply a threshold:* A threshold for identifying changes has to be calculated based on the inflection point of the histogram from the distance matrix, as described in [22]. Here, the pixel value of the BCM is set with the following equation:

$$Pixel\ value = \begin{cases} NCD\,(x, y) \leq Threshold : 0 \\ NCD\,(x, y) > Threshold : 255 \end{cases}.$$

Step4: *Calculate an undirected tree*: To calculate a binary tree representation from the patches of a satellite image, the toolkit *Complearn* [23] was used. This tool applies different compression techniques to the given data. For demonstration, the compression library *bzlib* [23] was used to calculate the matrix and the toolkit *maketree* to compute the best-fitting binary tree.

4.3 Experimental Results

The images from the data set were tiled into patches of 64 × 64 pixels and applied to the previous procedure. The results were split according to tour two application scenarios.

4.3.1 Danube Delta Scenario

The following figure shows the area of interest and the selected ROI (see Fig. 4).

This area underwent changes between April and May 2016 when a number of agricultural land parcels were covered by water. Figure 5 shows the binary BCM in which the algorithm is able to detect the location of the flooded areas.

For the selected area before and after the flooding, the corresponding undirected graphs are represented in Fig. 6. The deformation of the graphs represents the impact of the flooding.

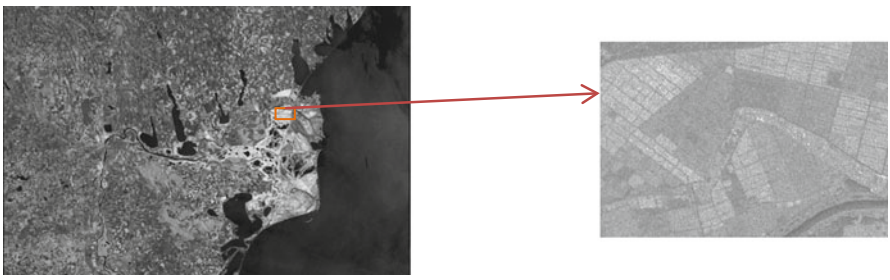


Fig. 4 Position of the selected ROI from an image of the Danube Delta

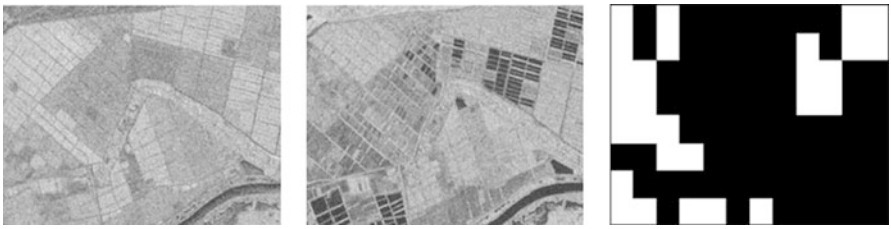


Fig. 5 (From left to right) The first image shows the test area prior to the flooding; the second image represents the region during the flooding (dark areas); and the third shows the changes (white areas) in form of a BCM

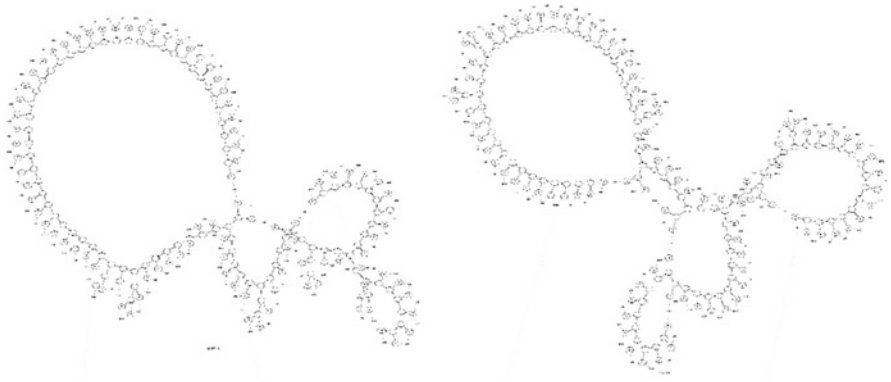


Fig. 6 The left graph illustrates the distribution of the patches in the selected ROI before the flooding, while the right graph depicts the area during the flooding following the approach of [22]

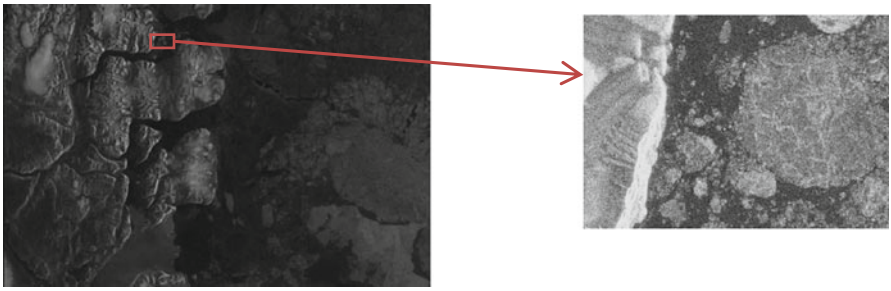


Fig. 7 Position of the selected ROI from an image of Belgica Bank

4.3.2 Belgica Bank Scenario

The same procedure is also applied to the second scenario. Figure 7 sows the area of interest and the selected area (e.g., the ROI).

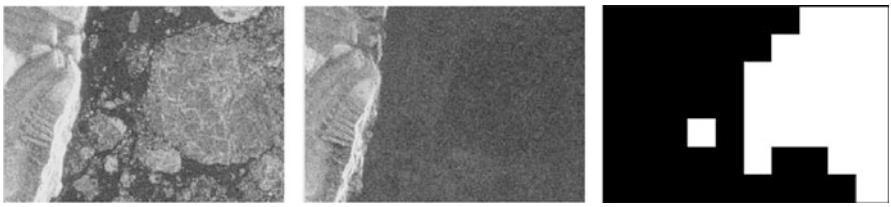


Fig. 8 (From left to right) The first image is an area of Belgica Bank acquired on April 17, 2018. The second image covers the same area acquired on June 16, 2018, where the ice has melted. The third one is the BCM of the ROI area

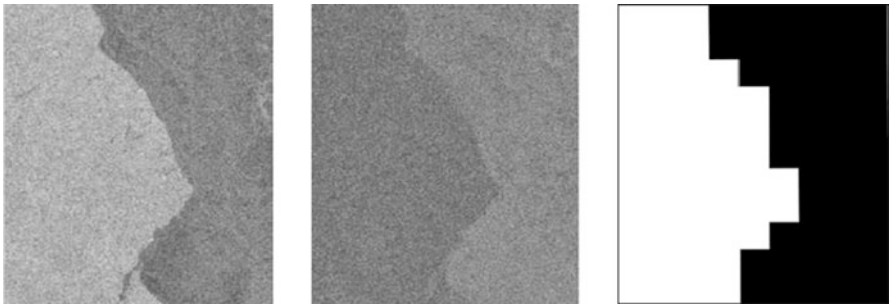


Fig. 9 (From left to right) The first image shows an ice cover acquired on April 17, 2018. The second image covers the same area acquired on June 16, 2018. The third one shows the BCM of the ROI area

Between April and June 2018, the ice, which had been floating in the sea, has melted; the BCM is able to detect that in this area, a change occurred (see Fig. 8).

The ice shelf depicted in Fig. 9 is reduced to a minimum between these two images. This change can also be seen in the BCM map.

4.4 Conclusions

Here, we described a robust and unsupervised and feature-free approach. For both scenarios, the results show, by visual evaluation, that for most subareas, the method works very well except for some exceptions. For the Danube Delta, this can be explained by currents on the Black Sea affecting the water surface backscatter or by different meteorological conditions. In the case of Belgica Bank, the seasonal change from winter to summer has affected the ice cover, and the ice has been moving between April and June 2018.

As future work, we plan to use the Burrows and Wheeler transformation [8] that is reordering the input values of each patch. The first test shows that the resulting images using this transform are similar to the NCD. That demonstrates the "robustness" of the NCD method.

5 Extracting Features with Variational Auto-encoders

5.1 Motivation

What if we could learn the features of satellite imagery with a neural network and without having to use extensive modelling and complicated approaches? What kind of network architecture is there to be used? How can we tailor this architecture to be specialized on some kind of data but flexible enough to be used on others?

In this section, we propose a new feature extractor using a variational auto-encoder. While the variational auto-encoder has been extensively used as a generative model, we now want to exploit its dimensionality reduction properties. By taking the input and mapping it to a more compact, lower-dimensional representation, we then take this mapping and construct a single feature vector which is then fed to a Support Vector Machine (SVM) and a k-NN classifier.

5.2 Methodology

5.2.1 Variational Inference and Auto-encoders

Suppose we have some data x (e.g., a 128×128 patch from a SAR product) which depends on some hidden variables x and we try to learn the parameters θ that best approximate the true distribution of the data $p^*(x)$. Using Bayes' rule, marginalization, and Jensen's inequality [24, 25], we can derive a lower bound for the log-likelihood of $p_\theta(x)$, i.e.,

$\ln(p_\theta(x)) \geq \mathbb{E}_{q_\varphi(z|x)}\left[\ln\left(p_\theta\left(x|z\right)\right)\right] - KL(q_\varphi\left(z|x\right) \mid |p_\theta(z)) = \mathcal{L}\left(x, \varphi, \theta\right)$, where $q_\varphi(z|x)$ is a recognition model used to approximate the true posterior $p_\theta(z|x)$. The main idea in variational Bayes theory is to "make the bound tight," which means to find the set of parameters $(\varphi, \theta)_{\text{opt}}$ which maximize $\mathcal{L}\left(x, \varphi, \theta\right)$.

In essence, $q_\varphi(z|x)$ can be interpreted as a function which maps the data to the hidden variables, an encoder. Then $p_\theta(x|z)$ can be interpreted as a decoder. If we use neural networks to compute $(\varphi, \theta)_{opt}$, then we talk about a variational auto-encoder.

In our case, we suppose the hidden variables $p_\theta(z)$ come from an isotropic Gaussian $\mathcal{N}(0, I)$ and the encoder is also a Gaussian with a mean vector $\boldsymbol{\mu}$ and a diagonal covariance matrix $\boldsymbol{\sigma^2} I$. Using the same re-parametrization method for variance reduction as in [6], we learn the best pair $\varphi = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ for different dimensionalities of the latent spaces, i.e., we set the dimensionality of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ to a certain number, e.g., 2, 4, 8, etc. We found out that 128 dimensions give the best classification results. For implementation purposes, we do not learn $\boldsymbol{\sigma}$ directly but $ln(\boldsymbol{\sigma})$ and then use the exponential function to retrieve $\boldsymbol{\sigma}$.

Figure 10 summarizes the computational flow for the training of the auto-encoder, in analogy to [26].

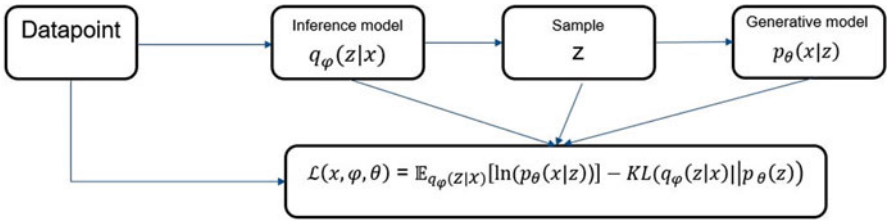


Fig. 10 Computational flow for the training of the auto-encoder. The data point, in our case the $128 \times 128 \times 2$ patch of a SAR image, is fed to the encoder, which is an inference model which produces a sample z. This sample is then used by the decoder to generate a new data point, which should be as close to the original one (in the MSE-sense). With this information, we construct the loss function $\mathcal{L}(x, \varphi, \theta)$, which is then optimized

5.2.2 Classifiers

5.2.2.1 Support Vector Machines (SVMs)

Support Vector Machines are a powerful type of classifiers and have been used extensively with great success in remote sensing applications [27]. They are representative of the family of large margin classifiers, meaning that the SVM tries to learn the hyperplane of greatest separation, i.e., to find the solution to the so-called primal problem:

$$\arg \min_{\mathbf{w},\mathbf{b}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i \quad s.t. \forall i: \quad y_i\left(\mathbf{w}^{\boldsymbol{T}}\boldsymbol{\varphi}_i \quad + b\right) \geq 1 - \xi_i,$$

where $\varphi_i = (\boldsymbol{\mu_i}, \boldsymbol{\sigma_i})$ are the features extracted from the ith patch $\mathbf{x_i}$ with a category label y_i.

Because the separating plane is in practice rarely described by a linear function, we use the kernel trick to map our nonlinear data to another domain, where the classification can become linear. In this chapter, we use the radial basis function kernel $k(\mathbf{x}_i, \mathbf{x}_j) = exp\,(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2)$, but also other kernels can work in principle.

For a more thorough description of this algorithm, we refer to [28]. The hyper-parameters C and γ are learned through a grid-search approach, and the optimal values for them turned out to be $C = 1000$ and $\gamma = 0.0019$.

5.2.2.2 k-Nearest Neighbors (k-NNs)

Another classifier used in this chapter is the k-Nearest Neighbors algorithm. Like the SVM, the k-NN takes patches $\mathbf{x_i}$ with category label y_i as its input. Unlike SVM where there is an optimization problem to be solved, in k-NN, each unassigned data point is being classified by assigning the label that is the most frequent among the

nearest k data points. Thus, the optimization problem becomes a search problem. For more on k-NNs, we refer to [29].

The choice of a metric as well as the constant k is up to the user. In our experiments, we found out that a weighted k-NN with a Euclidean metric, where we assigned a weight to each point being proportional to its distance, and $k = \#\,\text{categories} + 1$ worked best.

5.2.3 Overview of Our Methodology

The following section lists a short overview of the methodology being used as well as some implementation details.

***Step1*:** *Data preprocessing*: We gathered seven Sentinel-1 GRDH products covering nearly all kinds of terrain and water for the training of the auto-encoder. These products were tiled into 164,125 patches (131,300 patches for training, 32,825 patches for validation).

***Step2*:** *Network training*: In this phase, the auto-encoder presented in the section above was trained until the cost function $\mathcal{L}(.)$ attained its minimum. The optimization was made with the Adam method [30], and the function attained a value of 9.681 for the training set and 4.773 for the validation set after 50 cycles.

***Step3*:** *Feature extraction and classification*: After finishing the training, the encoder part of the network was used to extract features from the Danube Delta and the Belgica Bank data sets, respectively. The patches were fed to the encoder, and the output was a 256-dimensional vector containing pairs of 128-dimensional $(\boldsymbol{\mu}, \boldsymbol{\sigma})$ parameters. Using these parameters, SVM and k-NN classifiers were trained and tested on one of the products in each data set and then used as a prediction for the other products. The performance of the training was assessed using precision/recall and F1-score metrics.

5.3 Experimental Results

5.3.1 The Danube Delta Scenario

In the case of the Danube Delta data set, the highest values for precision, recall, and F1 in each category were 1, 1, and 1, while the lowest ones were 0.65, 0.81, and 0.79 (see Table 1 and Fig. 11a). The highest values were obtained for the categories with many examples such as sea/lakes and plowed agricultural land, while the lowest ones were seen for the categories with few examples such as flooded areas. In Fig. 11b, we show the change of the land cover in two different images by using the k-NN classifier for both data sets.

Following the methodology presented above, we got the following results for the weighted average of the metrics.

5.3.2 The Belgica Bank Scenario

In the case of the Belgica Bank data set, the maximum values for the metrics describing each category individually were 0.49, 0.80, and 0.62, while the minimum values were 0.40, 0.11, and 0.18 (see Table 2 and Fig. 12a). The corresponding change map is shown in Fig. 12b.

Following the methodology presented above, we got the following results for the weighted averages of the metrics.

In the classification scenario, *k*-NN, even though simpler, outperforms the SVM in all cases. An explanation can be given by visualizing the high-dimensional features in 2D by a projection with the help of *t*-SNE [31] using two different values for the perplexity parameter (see Figs. 13 and 14). The features for the examples of the same category do not always cluster together, but they create smaller clusters which are sometimes far apart from each other. Because the SVM does not directly take into account the distance between these, it cannot learn the very complicated decision boundary properly. Because we used a weighted Euclidean distance, the *k*-NN is thus robust to such distances between data points stemming from the same category.

5.4 Conclusion

We have presented a novel way of characterizing SAR imagery with the help of a variational auto-encoder that learns the optimal geometry of a Gaussian latent space from all kinds of polarizations. We have also shown that the parameters that govern this geometry make good feature descriptors by using them for classification and land cover change detection.

But are the classification results a robust measure of how good this feature descriptor is? From a practical viewpoint, the answer could be yes, because we know that having trained the feature extractor to be tailored on SAR data, we can expect it to also perform well on other datasets. On the other hand, the metrics used for assessing classification performance (precision, recall, F1-score) are all data dependent and do not give any guarantees that this procedure may work well on other data sets.

Table 1 The average performances obtained over all categories in the case of the Danube Delta scenario

Classifiers	Precision	Recall	F1
SVM Danube Delta	0.79	0.50	0.59
k-NN Danube Delta	0.97	0.96	0.97

Table 2 The average performances obtained over all categories in the case of Belgica Bank

Classifiers	Precision	Recall	F1
SVM Belgica Bank	0.75	0.55	0.59
k-NN Belgica Bank	0.97	0.97	0.96

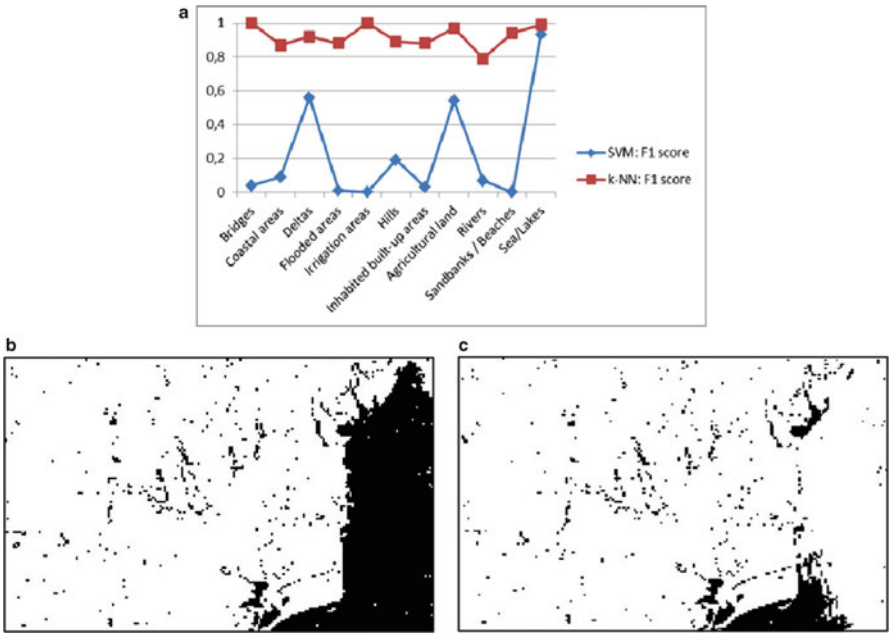


Fig. 11 (**a**) The performances obtain for all the categories in the case of the Danube Delta scenario. (**b**) Binary change map between two products showcasing the Danube Delta. The figure on the center shows the changes between May 18, 2016, and May 6, 2016, while the one on the right depicts the changes between May 18, 2016, and July 1, 2016

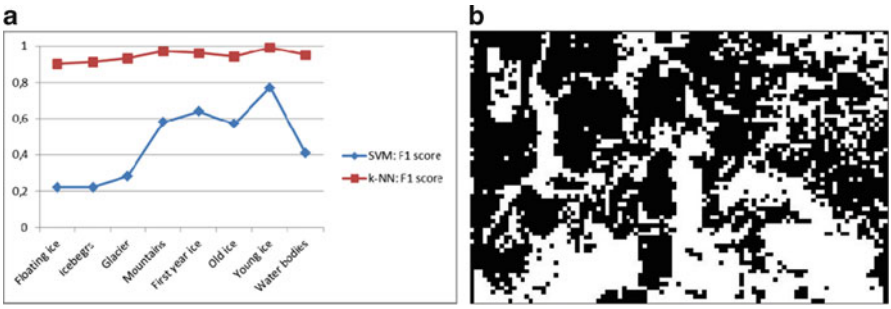


Fig. 12 (**a**) The performances obtained for all the categories in the case of the Danube Delta scenario. (**b**) Change map for Belgica Bank for two acquisitions made on April 17, 2018, and June 16, 2018

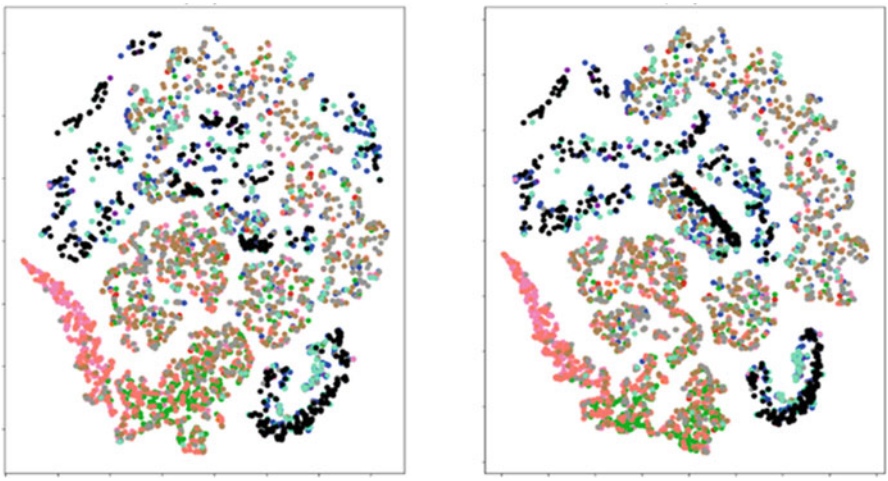


Fig. 13 2D projection of the feature space for the patches in the Danube Delta data set for different values of the perplexity parameter (left side $p = 30$ and right side $p = 50$) (see Ref. [31] for the role of this parameter in the projection)

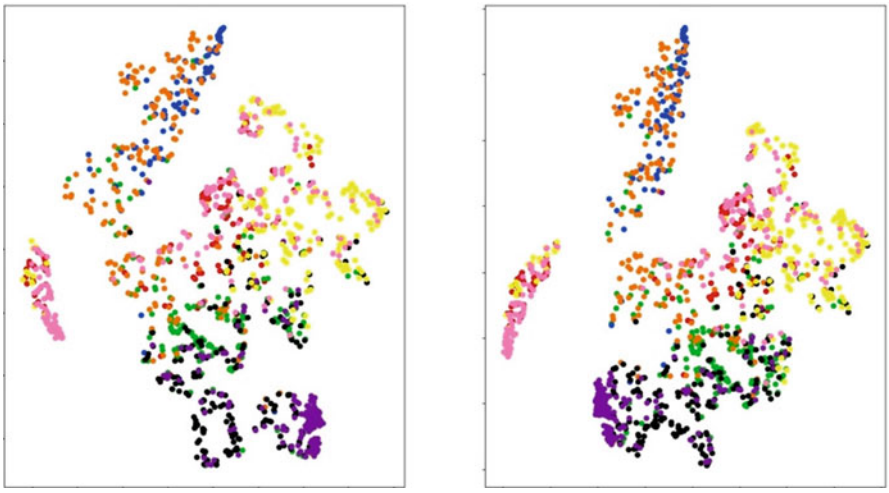


Fig. 14 2D projection of the feature space for the patches in the Belgica Bank data set for different values of the perplexity parameter (left side $p = 30$ and right side $p = 50$) (see [31] for the role of this parameter in the projection)

One may also ask the question, is it possible to get better than this, and if yes, how? We believe that this feature extractor should not only be assessed using empirical approaches but also through sound theoretical methods. We believe that information theory could give us an answer in terms of how good we can get

(the Cramer-Rao Lower Bound could be a characterization of this) and that novel methods of analyzing model predictions [32] could also help us understand our model better. This is a good starting point for future work.

6 Coastline and Ice Detection Using Polarimetric SAR Images

6.1 Motivation

Coastline detection is of great significance for the identification and exploitation of marine resources and is an important task in ocean remote sensing. Because Synthetic Aperture Radar (SAR) images can observe the actual situation of the coastline, the parameters of sea level and its surrounding environment are often used in marine management and disaster warning applications. Usually, coastline detection methods are based on traditional image processing methods, but these methods require complex arithmetic operations, which are computationally expensive, complex, and unstable in output. Therefore, the traditional methods are not suitable for long-term sequence SAR image processing. In this section, we describe a fast and effective method for detecting coastlines, which consists of three steps. First, the improved cross-correlation coefficients of two different but accurately overlapping polarimetric SAR images are calculated. With this improved polarization information feature, the distinction between ocean and land in SAR images becomes more apparent. Second, a histogram-based threshold selection criterion is set to distinguish ocean and land in the polarization feature image. Finally, by using a Canny edge detector to extract continuous coastlines, our detector maintains good detection performance even in the presence of noisy data. In the measured scene experiment, we summarize the time course of the coast and its surrounding areas from the coastline extracted from a time series of SAR images.

6.2 Methodology

For coastline/ice detection, the use of radar polarization information has become the mainstream method to distinguish land and sea as described by [33–36]. In particular, we use a new feature proposed in [34] to detect coastlines, based on radar polarization and an improved correlation coefficient. Then, for each pixel in the image, we can calculate its polarization characteristics, defined as

$$\gamma = \langle S_{xx} \cdot S_{xy} \rangle,$$

where S are the pixel amplitudes of the polarization channels, $\{x, y\} \in \{H, V\}$, and $\langle \cdot \rangle$ stands for local ensemble averages within windows of 21 $\times$ 21 pixels. The specific operations are shown in the next figure (Fig. 15).

Unlike the correlation coefficients determined in traditional interference calculations, our improved polarization correlation has no normalized operation. The authors of [34] discuss this in detail and use scattering theory to demonstrate that using such calculations will increase the separability of sea and land. After extracting theγfeature, we obtain the distribution of this parameter in the image. Since the observed area is around the coastline and the polarization-dependent gamma feature map is used, the resulting histogram generally has a bimodal distribution. Therefore, it is only necessary to determine a threshold to discriminate between ocean and land pixels. For a pixel distribution with a distinct bimodal structure, a simple and stable threshold determination method is to select the minimum between the two peaks as a threshold.

Once the segmentation threshold for the image is determined, the next step is to extract the edges. In this text, we use an improved method of coastline extraction, using the Canny edge detection operator instead of the Sobel edge detection operator being used in [7]. The Canny edge detection operator is robust and can easily generate continuous coastline edges. In addition, there may be ship targets around the shoreline, which have a higher radar cross section but a small size. These vessels may also be considered terrestrial when using polarization information. Therefore, depending on the size of the segmentation areas, we may lose some of the smaller vessels. For time series analysis, all images have to be co-aligned with a selected master image. Then our proposed methodology to extract coastlines from SAR image time series contains the following steps:

***Step 1**: Date preparation*: The first step is to select a SAR image product which includes coastlines and has two polarization channels (e.g., VV and VH, or HH and HV). All other images that are to be analyzed jointly are co-registered with this main image using the same reference system; then we use geocoding to perform the co-registration of all images.

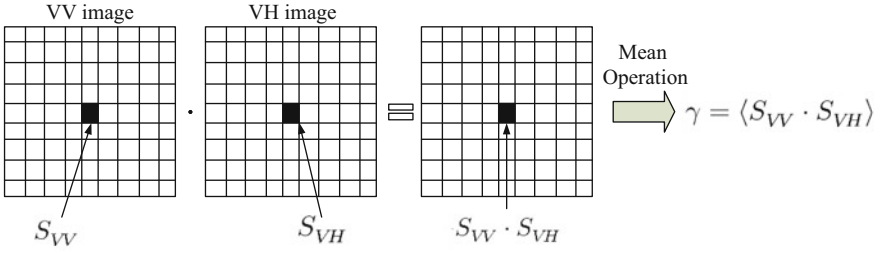


Fig. 15 Coastline detection steps by using two polarization images (VV and VH, or HH and HV)

Step 2: *Feature extraction*: From each SAR image, the equation given above is applied in order to extract the corresponding features. The output of this is a modified polarimetry-based correlation map.

Step 3: *Threshold determination*: The correlation map generated in *step 2* is used to distinguish between land and sea, or ice and sea. After a threshold selection, a binary image is output, where the ocean is in black, and the land/ice region is in white.

Step 4: *Edge extraction:* A Canny edge detector is applied to the binary image to extract all coastlines.

Step 5: *Time series analysis:* The previous steps are repeated for all the images from the selected data set. The time series data are then combined with weather information and other parameters (e.g., wind, etc.).

6.3 Experimental Results

6.3.1 Danube Delta Scenario

Following the steps presented above, for demonstration of the method, we selected the first scenario. Figure 16 shows the polarimetric correlation map and a quick-look of the original data provided by the Sentinel-1 product. The same figure also shows a histogram of the binary image used to separate land from sea [37].

By applying a Canny detector, we are able to determine the coastline very well. Figure 17 depicts two examples cut out from the full image where the algorithm is delimiting land from sea.

Using all the images acquired during the entire period over the Danube Delta and applying our algorithm, a number of interesting effects can be observed. For demonstration, we selected four out of 15 images from our data set and present them in Fig. 18.

Looking in detail to each region of the images, we noticed that the coastline is quite stable during that time but small changes are identified on the estuary area where the sand is accumulated when the water coming to the river goes to the sea. In hinterland, there are obvious changes during the period of monitoring, where a lot of vegetation is regarded as water body. The reason can be that during this period of the year, the area was covered by snow (snow melted) or by rain (flooding). Detailed results and comparisons are presented in [37].

6.3.2 Belgica Bank Scenario

Repeating the same procedure as described above, we show in Fig. 19 the polarimetric correlation map and a quick-look of the Sentinel-1 image data, together with a histogram of the binary image used to separate (in this case) ice from sea.

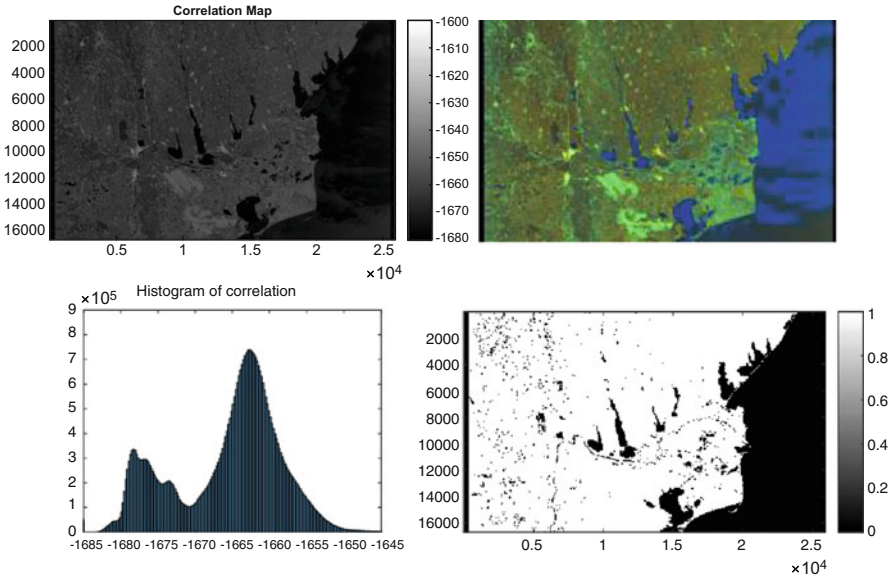


Fig. 16 The results in this figure are for the Danube Delta acquired on February 12, 2016. The top-left figure presents the correlation map, while the top-right figure depicts a quick-look of the original data on the same date. The bottom-left figure shows a histogram of the binary image, while the bottom-right figure is a binary separation between land and sea

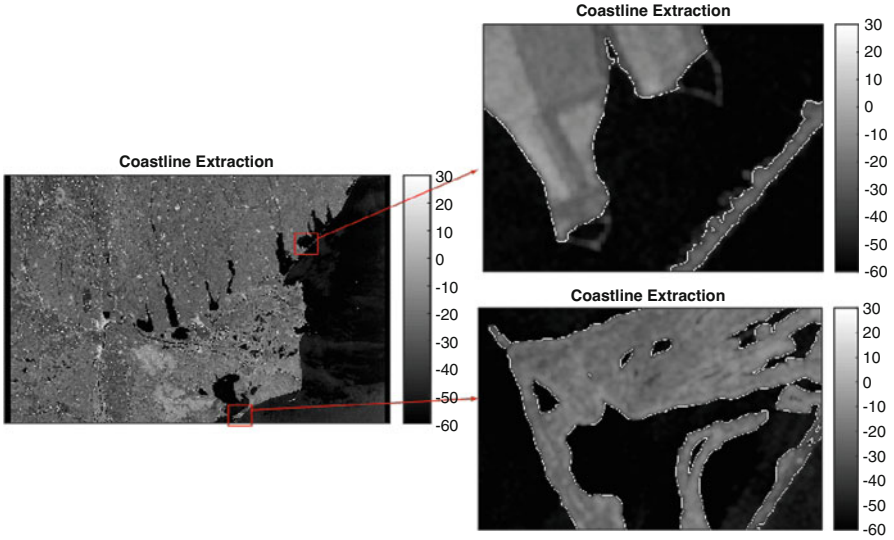


Fig. 17 Coastline detection by using a Canny detector (left) and two zoomed subareas from the full image (right)

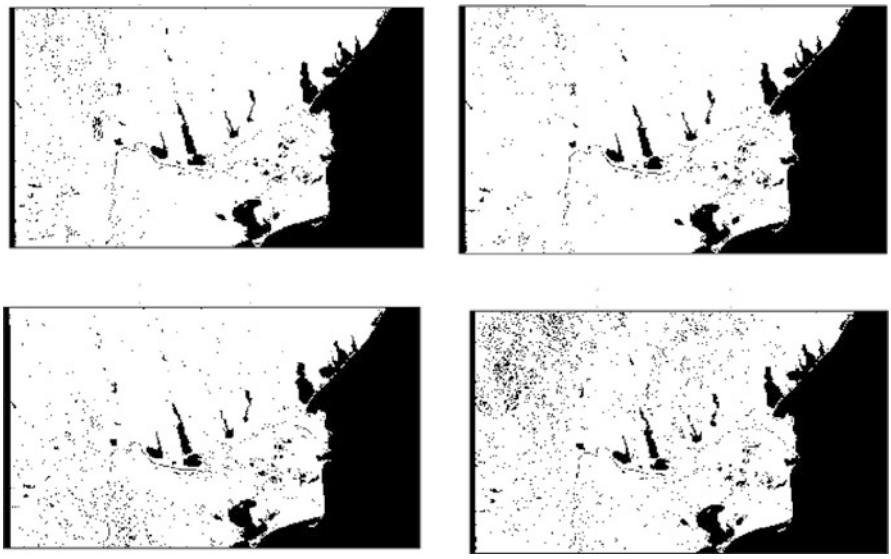


Fig. 18 Image time series that covers the Danube Delta in Romania (from left to right, top to bottom) acquired on February 12 and 24, 2016, on April 12, 2016, and on May 6, 2016

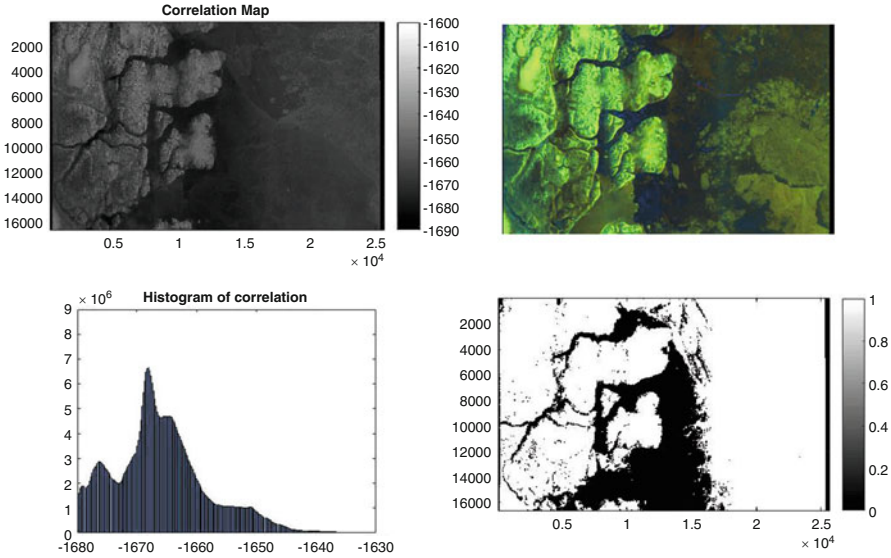


Fig. 19 Results for Belgica Bank acquired on June 16, 2018. The top-left figure presents the correlation map, while the top-right figure shows a quick-look image taken on the same date. The bottom-left figure depicts a histogram of the binary image, while the bottom-right figure is the binary image resulting from separation between ice and sea

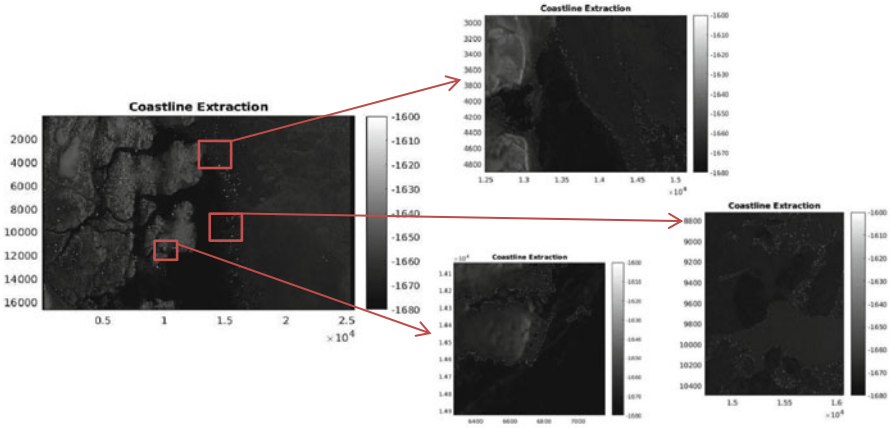


Fig. 20 Sea detection using a Canny detector (left) and three zoomed subareas from the full image (right)

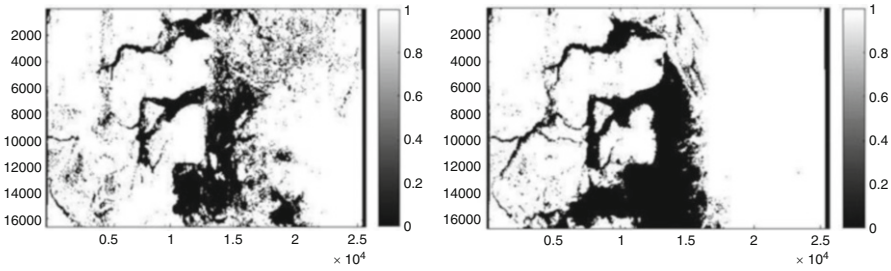


Fig. 21 An image time series covering Belgica Bank in Greenland (from left to right) acquired on April 17, 2018, and on June 16, 2018

Applying the Canny detector again, but this time for the second scenario, we are able to identify the coastline very well. Figure 20 shows three examples from the full image where the algorithm is delimiting very well sea from ice and land.

When analyzing the two images, one acquired in winter and one in summer over Belgica Bank and applying the algorithm, a number of interesting effects can be observed. For exemplification, Fig. 21 presents the results derived from these images. By comparing the two images, the different categories of ice from April 17, 2018, are transformed in June 16, 2018, into one of the other categories or into water. The separation between land/ice and water is very clear.

6.4 Conclusions

We proposed a method to easily and quickly extract coastlines from polarimetric SAR image time series based on a gamma computation between two SAR polarization channels (e.g., VV and VH, or HH and HV) and a histogram of the correlation map. Our general observations are the following:

- For the Danube Delta scenario [37]: for large water bodies, such as sea, lakes, and rivers, the coastline detection method is producing a stable result for image time series; for vegetated areas, like agricultural land and/or forest, the method is only detecting obvious changes within each area.
- For Belgica Bank scenario [3]: the proposed method detects and separates sea from ice and also from land (mountains in our case). We can see also that within the land, there are some areas where there are areas with water and floating ice; in this case, the method is able to delimit this if the ice is not melted.

7 Outlook

The image data processing concept described above represents established approaches and solutions based on today's technology for satellite instrumentation, and common on-ground processing facilities with their available hardware and software. In future, we expect to see a number of new developments when our ice classification tasks for satellite images receive additional support from photogrammetry (motion analysis of ice floes and icebergs with changing and disintegrating features), still better radiometric quality of the processed reflectance data (consistency of the calibration parameters among all Sentinel satellites, reliable classification of snow-covered surfaces), and inclusion of additional knowledge from modeling of snow/ice processes and climate change (visibility of physical processes and determination of their quantitative parameters such as freezing and melting of ice in radar images). Here, we expect that some innovative tools could simulate new sensors and their performance (transfer learning with quantitative physical background).

On the other hand, the launch of any new SAR satellites is a long-term process with considerable lead times and few players on the market. While optical multispectral imagers have become well-known data sources with a number of public or commercial data providers, radar data and their characteristics are still less commonly known. Currently, there are European plans for longer wavelength L-band instruments that are aiming primarily at vegetation monitoring over land surfaces (biomass and carbon balances of forested areas); the impact of such instruments on ice image understanding will be an interesting topic especially for the arctic waters as comparative ground truth measurements are easier to obtain than in Antarctica.

Finally, new developments in software tools (such as Python-like languages or easily configurable machine learning libraries) and new hardware standards (for instance, still more powerful graphics processing units to model three-dimensional objects with irregular structure) will probably become available in a few years.

Acknowledgments The first scenario, the protected area of the Danube Delta, was supported by the H2020 ECOPOTENTIAL project (under grant agreement No. 641762), while the selection of the second scenario, the area of Belgica Bank, was supported by the H2020 ExtremeEarth project (under grant agreement No. 825258). We would like to thank Nick Hughes from Norwegian Meteorological Institute, Norway, for his supporting discussions regarding the selection of the area for the second scenario.

References

1. Taini, G. et al. (2012). SENTINEL-1 satellite system architecture: Design, performances and operations. *IEEE International Geoscience and Remote Sensing Symposium*, Munich, pp. 1722–1725.
2. ExtremeEarth project. (2019). Available online: http://earthanalytics.eu/. Accessed in Apr 2019.
3. Singha, S., Johansson, M., Hughes, N., Hvidegaard, S. M., & Skourup, H. (2018). Arctic Sea ice characterization using spaceborne fully polarimetric L-, C-, and X-band SAR with validation by airborne measurements. *IEEE Transactions on Geoscience and Remote Sensing, 56*(7), 3715–3734.
4. ECOPOTENTIAL project. (2019). Available online: http://www.ecopotential-project.eu/. Accessed in Apr 2019.
5. Li, M., Chen, X., Li, X., Ma, B., & Vitányi, P. M. B. (2004). The similarity metric. *IEEE Transactions on Information Theory, 50*(12), 3250–3264.
6. Kingma, D. P., & Welling, M. (2014). *Auto-encoding variational Bayes*, arXiv:1312.6114v10 [stat.ML].
7. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 8*(6), 679–698.
8. Giancarlo, R., Restivo, A., & Sciortino, M. (2007). From first principles to the Burrows and Wheeler transform and beyond, via combinatorial optimization. *Elsevier Theoretical Computer Science, 387*, 236–248.
9. Sentinel-1 ESA hub. (2019). Available online: http://en.wikipedia.org/wiki/Sentinel-1. Accessed in Mar 2019.
10. ESA Thematic Exploitation Platforms. (2019). Available online: https://tep.eo.esa.int/. Accessed in Mar 2019.
11. Danube Delta. (2016). Available: http://romaniatourism.com/danube-delta.html. Accessed in Mar 2018.
12. Manual of Ice (MANICE). (2019). Available online: https://www.canada.ca/en/environment-climate-change/services/weather-manuals-documentation/manice-manual-of-ice.html. Accessed in Mar 2019.
13. Ulaby, F. T., Long, D., & Blackwell, W. J. (2014). *Microwave radar and radiometric remote sensing*. Ann Arbor: University of Michigan Press.
14. Dumitru, C. O., Schwarz, G., & Datcu, M. (2016). Land cover semantic annotation derived from high-resolution SAR images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9*(6), 2215–2232.
15. EOLib project. (2018). Available: http://wiki.services.eoportal.org/tiki-index.php?page=EOLib. Accessed in Feb 2018.

16. Dumitru, C. O., Schwarz, G., & Datcu, M. (2018). SAR image land cover datasets for classification benchmarking of temporal changes. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11*(5), 1571–1592.
17. Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 837–842.
18. Chen, J., et al. (2010). WLD: A robust local image descriptor. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 32*(9), 1705–1720.
19. Monetdb. (2019). Available: https://www.monetdb.org/. Accessed in Apr 2019.
20. Blanchart, P., Ferecatu, M., Cui, S., & Datcu, M. (2014). Pattern retrieval in large image databases using multiscale coarse-to-fine cascaded active learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 7*(4), 1127–1141.
21. Garcés, M. F. E.. (2019). *Climate and sustainable development for all | General Assembly of the United Nations*. Available online: https://www.un.org/pga/73/2019/03/28/climate-and-sustainable-development-for-all/. Accessed in Apr 2019.
22. Coca, M., Anghel, A., & Datcu, M. (2018). Normalized compression distance for SAR image change detection. In *Proceedings of the IGARSS 2018*, Valencia, Spain, pp. 1–3.
23. Cilibrasi, R. (2003). *CompLearn*. Available online: https://complearn.org. Accessed in Apr 2019.
24. Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. PhD dissertation, University College London.
25. Blei, D. M., et al. (2018). *Variational inference: A review for statisticians*, arXiv:1601.00670v9 [stat.CO].
26. Kingma, D. P. (2017). *Deep learning and variational inference: A new synthesis*. PhD thesis, Amsterdam, the Netherlands, 162 pages. Available online: https://hdl.handle.net/11245.1/8e55e07f-e4be-458f-a929-2f9bc2d169e8.
27. Mountrakis, G., Im, J., & Ogole, C. (2011). Support vector machines in remote sensing: A review. *ISPRS Journal of Photogrammetry and Remote Sensing, 66*(3), 247–259.
28. Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: A library for support vector machines*, In *ACM Transactions on Intelligent Systems and Technology*, 2(3). Available online: https://dl.acm.org/doi/10.1145/1961189.1961199.
29. Devroye, L., Gyorfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition* (Vol. 31). New York: Springer Applications of Mathematics.
30. Kingma, D. P., & Ba, J. (2015). *Adam: A method for stochastic optimization.* In *Proceedings of the 3rd International Conference on Learning Representations*, arXiv:1412.6980v9.
31. van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research, 9*, 2579–2605.
32. Lundberg, S., & Su-In, L. (2017). *A unified approach to interpreting model predictions.* In *Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, pp. 4768–4777.
33. Baghdadi, N., Pedreros, R., Lenotre, N., Dewez, T., & Paganini, M. (2007). Impact of polarization and incidence of the ASAR sensor on coastline mapping: Example of Gabon. *International Journal of Remote Sensing, 28*(17), 3841–3849.
34. Nunziata, F., Buono, A., Migliaccio, M., & Benassai, G. (2016). Dual-polarimetric C- and X-band SAR data for coastline extraction. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 9*(11), 4921–4928.
35. Baselice, F., & Ferraioli, G. (2013). Unsupervised coastal line extraction from SAR images. *IEEE Geoscience and Remote Sensing Letters, 10*(6), 1350–1354.
36. Nunziata, F., Migliaccio, M., Li, X., & Ding, X. (2014). Coastline extraction using Dual-Polarimetric COSMO-SkyMed PingPong mode SAR data. *IEEE Geoscience and Remote Sensing Letters, 11*(1), 104–108.
37. Ao, D., Dumitru, O., Schwarz, G., & Datcu, M. (2017). Coastline detection with time series of SAR images. In *Proceedings of SPIE Remote Sensing*, Warsaw, Poland, pp. 11–14.

Applications in Financial Industry: Use-Case for Fraud Management

Sorin Soviany and Cristina Soviany

1 Issues and Challenges for Application of Data Science Principles and Tools in Financial Industry

The actual major issues for application development in various domains, including the financial industry and the associated use-cases, concern the ways in which big data can be approached in order to meet the real-case constraints. Several *data analytics* tools can be applied by the financial organizations to perform specific tasks according to their partner and customer expectations and requirements various financial data sources and the way in which the provided data can be combined, consolidated, and integrated in order to take proper decisions in real cases [1].

The advances in artificial intelligence-based technologies together with the storage technologies and processing algorithms are the key factors for the fast adoption of data science principles and tools for an enlarging domain of applications with various constraints. The financial industry is just a relevant example of the sectors that potentially uses these approaches. According to the Financial Stability Report that was published in November 2017, the key driving factors for this evolution are grouped into the following main categories [2]:

- *Technological support factors*, a category that includes the most recent developments in *software*, *statistic*, and *analytic tools* and also various *process-*

S. Soviany
Communication Terminals and Telematics, National Communications Research Institute, Bucharest, Romania
e-mail: sorin.soviany@inscc.ro

C. Soviany (✉)
Features Analytics, Nivelles, Belgium
e-mail: cristina.soviany@features-analytics.com

© Springer Nature Switzerland AG 2020

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_11

ing algorithms for smart decision-making (including *machine learning*-based approaches). In this dynamic development framework, the challenge is given by the increasing availability of large data collections and datasets, as in the case of financial applications and especially for applications in which the data sources are provided from large financial markets. The financial application data provides information about the banking and e-commerce transactions, but also more complex information that involve operations on large financial markets and for applications in fields such as fraud management (detection and prevention). The innovative design approaches and the advances in storage technologies together with the cloud-based approaches still enlarge the potential of using data science tools and methodologies for the financial industry. In some cases, additional constraints are given by the real-time requirements for decision-making, especially for intensive applications with high computational expenses. On the one hand, an increased amount of data can be processed and analyzed in order to build and to run more accurate predictive learning models providing the required support for decisions; the same amount of data can be used also for unsupervised processes in financial applications, for instance, to detect abnormal cases on capital markets. On the other hand, the informative degree of the available data should be carefully considered in order to only exploit the most relevant data for the application purposes. The hardware advances as concerning the execution speed and storage capacity provide an increased computing power with reduced costs, also allowing the access to high-performance computational resources through cloud-based approaches.

- *Demanding and other market-related factors*, including the competition on the target markets and the needs for regulation in very sensitive cases such as fraud management. The artificial intelligence (AI) and particularly machine learning (ML) usages for various business processes already represent priorities for many financial institutions due to the opportunities for cost reduction, significant advantages of risk management, and productivity improvements. As stated in the report [2], the main goals of using data science principles (for data analytics) together with AI/ML design and development tools in financial industry are as follows:
  - Optimization of the tasks and activities in respect to the customer expectations and requirements
  - Increasing the interactivity degree between the designed/developed application systems (software) and the operational staff that is involved in decision-making processes
  - Enlarging and enriching of the products and services portfolio in order to properly respond to the customer's needs

  A key demanding factor is represented by the need for regulation and the way in which the compliance can be ensured. The new analytical tools that are based on data science principles can provide a reliable framework to ensure this compliance by enhancing the overall operational and business processes.

The application of *data science* principles is based on the following concepts [2]:

- *Data analytics*: An approach that looks to develop a data model in order to properly describe specific application scenarios. The typical goal is to provide useful information for prediction and decision-making based on the described application scenario. Data mining methods are used for the development of data analytics functional capabilities.
- *Business intelligence*: A concept that includes applications, infrastructures, development and evaluation tools, modeling techniques, and the specification of the best practices to be used for analytical processes performing in order to enhance the business processes and to meet the given requirements in real cases. Data visualization tools are also integrated into this procedural framework in order to provide a reliable way for the relevant information description and visual representation.
- *Big data*: A concept that refers to the management of a very large amount of data for storage, analysis, and processing in order to extract, generate, and provide meaningful information for application. The main attributes of big data are the following: *volume*, *velocity*, *variety*, *veracity*, and *value*. These attributes define new complexity dimensions to the design process in many application use-cases, and this is also true for the financial industry domain. For many application belonging to this field, the following assumptions are already proven to be true:
  - The amount of required data is huge, especially for use-cases that concern large financial markets (*volume*).
  - The data types are diversified (*variety*).
  - The real-time constraints in decision-making, meaning that the designed system should ensure a fast response capability (*velocity*).
  - The amount of uncertainty in the input data could be high (*veracity*).

The application of data science in financial industry is still an evolving (ongoing) process, while a lot of advanced analytical and processing tools are already available. The typical users of these technologies include banks but also other financial institutions and many actors that operate on capital markets. These entities are concerned to improve their analytical capabilities for specific tasks. Other purposes include [2] the following:

- Various financial-specific processes and activities, such as stock forecasting and portfolio management
- Investment risk analysis
- Predictions of bankruptcy and foreign exchange rate
- Fraud management: detection and prevention
- Customer behavior analysis

The *capital markets* have become a very important target for which these advanced data analytics and processing tools can be applied. This is a big challenge because of the actual tendency of the large financial markets to be interconnected in a complex

way. In these cases, a large amount of data, with many variables and real-time constraints, should be analyzed and processed in order to provide suitable decisions in applications [2]. The market surveillance requires very advanced tools in order to detect more sophisticated fraudulent activities.

The *risk analysis* is another key goal for the data science application in the financial industry. In this case, the following purposes are typical for data analytics and advanced data processing based on data science principles and tools [2]:

- The evaluation of the credit-associated risks
- The analysis of banking operation performances
- The evaluation of insurances
- Compliance and regulatory reporting

2 A Data Science Use-Case for Financial Industry: Fraud Detection with Data Analytics and Machine Learning

One of the most challenging use-cases of the data science principles and tools, within the context of artificial intelligence/machine learning-based approaches, is represented by the fraud management in the financial industry. The usage of the AI/ML methodologies for the fraud detection cases is encouraged by the efforts to overcome the drawbacks of the legacy anti-fraud solutions that are based on expert-specified rules.

The conventional solutions in which specific ***rules*** are defined and applied in order to recognize the fraudulent activities still have a higher rate of false positives, which is an obvious inconvenient for many real cases in which the genuine transactions or other nonmalicious activities are actually rejected from further processing, leading to additional penalties and financial losses for customers and financial service providers. On the other hand, the actual dynamic of the threats and the fraud types is another major issue that cannot be efficiently managed with conventional *rule-based fraud management solutions*. These legacy solutions are not very suitable for the cases that require dealing with the fast changing of the threats and fraudulent activity types.

These are the main reasons for the actual interest to design and develop fraud management solutions in which the principles of data sciences together with the advanced artificial intelligence-based tools and technologies can be applied. Machine learning, either supervised or unsupervised, sometimes applied with more advanced technologies including ensemble learning and innovative algorithms, represents the key approach to design high-performance solutions in respect to the real-case constraints. However, even this design option should be updated in respect to the real-case occurrences as much as the typical learning methodology with training and validation may be not sufficient to properly handle the new threats; these new cases are not seen during the training stage and therefore the supervised

learning process should be completed with unsupervised methods, statistical data processing, and outlier detection techniques.

The machine learning application for the fraud management use-cases in financial industry requires a careful consideration of data and algorithms, both for supervised and unsupervised methods. Therefore, the overall design process should take into account the data as well as the algorithms.

Working with data concerns several *quantitative (numerical)* and *qualitative* issues to be considered within the overall design and development process for high performance and optimized complexity.

The *quantitative* factors include *the amount of available data*, *the feature space dimensionality*, and *the representativeness degree of the classes within the training set*.

Given the specific properties of these applications and especially the need to properly train the supervised machine learning models, a huge amount of data should be processed in order to design high-performance models. The training periods should be fixed in order to ensure the most relevant data for the target recognition, in this case the fraudulent activities. This requirement is justified by the fact that in some periods the fraudulent activities could be sparse, while other times one can see an increasing number of malicious attempts.

On the other hand, the big data-based approaches cannot always ensure the best performances, and this is also the case of applying advanced machine learning techniques for the real-time fraud detection.

The *amount of data* remains a critical issue especially for some throughput issues like execution time and the number of transactions that are processed per second. The challenge is whether it is possible to achieve good results for *fraud detection* while reducing the *alert rate* even if the designed system is trained with lower *amounts of data*. The volume of historical data should be large enough in order to provide relevant statistical data allowing to build a reliable profile of the user actions and to accurately recognize the fraudulent transactions. However, the past events could not always provide enough information to deal with the continuously enlarging of the fraudster methods and techniques, especially for mobile payments that are more prone to frauds. This is why the supervised machine learning in fraud detection must be completed with the unsupervised machine learning methods for anomaly detection; also the training process should be tuned in order to cover new updates with relevant data concerning the new fraud types.

Another issue that should be considered is the *feature space dimensionality*; therefore the number of input variables (direct features) and generated (designed) new features is based on the provided raw data. In this case it is important to exploit the most informative features within the further classification stage.

For example, working with *less correlated features* could be a reliable way to improve the classification performance.

Also, an *analytical process to explore the overlaying structure of the available data* and to maximize the overall variance using advanced transformations like PCA (principal component analysis) is helpful to generate a reduced feature space with enriched discrimination potential. PCA typically maximizes the variance and also

reduces the correlation degree among the input data. This seems to be a very good approach in many cases, as it is strongly required to exploit the informativeness of the data as best as possible. The uncorrelated data are usually the most informative for the further discrimination step, allowing to avoid the redundancies and to provide an optimal trade-off performance vs. execution time.

The *class representativeness* within the training datasets is another important issue that should be considered in the modeling. This criterion is given by the number of training samples per each class. The representativeness degree of the two classes ("fraud" and "genuine," respectively) within the training dataset has a significant influence of the performance achievements. For this, use-case "fraud" is the target class. Typically most of the fraudulent transaction samples are taken for training, and this is because the fraudulent transactions represent only a very low percentage in the overall number of available data samples with information about transactions. The representativeness seems to be more challenging for "genuine" class, because the training dataset contains many samples belonging to this class. The challenge is how much from the total number of genuine samples can be retained for training in order to ensure the desired performances, especially for the false positive reduction.

The *qualitative* factors are focused on the informativeness property of the available data. The informative value of the training samples could be provided by a careful *selection of the features* that should be applied for the classification process (training and testing, respectively). The used features include the original input variables (direct features), but an important contribution for the performance achievement is provided by the designed features. In many real cases one can see that the new features that are derived from the original variables provide more relevant information with high discriminant power, allowing to discover hidden patterns for the cases. From this perspective, working with data means to design and compute new suitable features based on the available data, using different mathematical operators and various statistics that can be evaluated using the original variables. This is actually a problem of *feature engineering*.

Working with algorithms concerns the typical tasks of the whole modeling process for *data classification* and *performance estimation* while considering several optimization strategies in order to achieve the performance target. The categories of the algorithms to be applied depend on the specific application process:

- The *fraud detection/recognition* is typically approached with s*upervised machine learning algorithms*. In these cases, various optimization techniques can be applied in order to properly adjust the required parameters and hyperparameters that allow to improve the model performances as required in real application cases.
- The more challenging task of *anomaly detection* requires *unsupervised machine learning methods* like clustering together with customized and optimized techniques for the data space transformation that can be applied in order to find the most discriminant patterns. The results of the unsupervised learning processes

and data projections should allow to efficiently separate the normal cases about the market participant's behavior from the abnormal behavioral cases.

The major challenge remains on how to accurately identify the new abnormal cases that can occur in almost real time, given the new and often more sophisticated types of malicious online activities.

Working with algorithms also means how to find and to apply a proper parameterization of the selected models in order to ensure their best behavior and the stability over various customer datasets.

2.1 Fraud Management Solution Design Process with Supervised Learning

In the supervised approach for fraud detection, the full modeling process includes typical steps of *feature generation*, *feature selection*, *training*, and *validation* (for the performance estimation). Then the designed model is applied for customer data in order to perform the required tasks as specified within the operational environment of the application case.

In this use-case, the most common design setting is with two classes in which the target class is represented by the fraud cases, while the nontarget class contains all samples that represent genuine transactions. There are also particular application cases in which the multiclass approach should be used if the goal is to make the discrimination among several types of frauds.

The following two functional components are included in the overall design and development process:

- *Data analytics* that looks to enhance the raw data quality. This functional component is based on preprocessing operations (data cleaning and transformations) and *feature engineering* tasks that allow to enrich the original data space by the following:
  - Increasing the informative value of the original variables.
  - Finding potentially hidden patterns.
  - Extracting the most relevant information allowing to properly discriminate between fraud and genuine cases.
- *The learning process*, in which the classification model is selected, trained, and validated for its performance estimation on a small data subset that is drawn from the original design dataset. Typically the performance estimation is done with ROC (receiver operating characteristic) analysis, based on a set of operation points that are generated according to the fixed thresholding.

The full modeling process is performed in order to reach the desired KPIs (key performance indicators) as given according to the application requirements. A suitable set of KPIs should have the following target:

- A *detection rate* (true positive rate for the target class) around 90% or even more. The target class is represented by the fraud cases in this approach; therefore, this performance measure (TPR for the target class) represents actually fraud detection rate (FDR) for the use-case of fraud management in financial industry.
- An *alert rate* (false positive rate for the same target class) not exceeding 10% and even less than 1% for some particular cases. In real use-cases, it is very important to minimize the alert rate (FPR) in order to reduce the fraction of blocked genuine transactions.

These performance targets can be reached by working with data and working with learning algorithms.

2.1.1 Data Analytics in Fraud Management

The main goal of this functional component that should be integrated into any efficient fraud management solution is to ensure data with improved discriminant power to the advanced steps of the predictive modeling (data classification). This objective is reachable by working on the available data in order to remove the noisy and less meaningful elements. The data analytical process with data science principles is performed in the following steps that include *exploratory analytical activities* together with *feature engineering* operations in order to generate an enriched data space with added value in performance for the fraud detection use-case:

- The *initial examination of the customer data* in order to explore the full data space and to look for the variables that can be used in the overall modeling process. In this step one can see that the fraud cases only represent a very small fraction from the overall dataset. This is a very important issue for the predictive modeling, because the training is done with unbalanced classes. Typically for training most of the fraud cases should be considered, while only a small amount of genuine samples is considered for training. This is because in real operating environments (with banking transactions, e-commerce, or other activities), the genuine cases are prevalent. The genuine samples drawn for the model training is important in order to ensure the alert rate (FPR) minimization according to the fixed target.
- The *preprocessing* step in which the noisy and redundant (duplicated) samples must be discarded. This step includes *data cleaning* and *data transformation* in order to ensure a cleaned and consolidated dataset with a meaningful representation of the data space. The correlation analysis of the variables and the variable normalization can be applied in this step. An advanced exploration of the existing variables can provide additional information for the next processing steps.
- The *feature generation* step in which the original variables (after the cleaning and other preprocessing operations, if required) are further processed in order to enrich the data space with *new and more informative features*. This step includes operations in which the data science principles and tools are strongly

involved. The new features are designed using estimated statistics and various mathematical operators that are applied on the original variables. Additionally, the feature space transformation with data projections on reduced dimensionality subspaces can help discover *hidden patterns into the input data*, with high discriminant value between fraud and genuine cases. These transforms like PCA (principal component analysis) or LDA (linear discriminant analysis) are applied in order to maximize the overall variance of data (PCA), to eliminate the correlated variables (PCA), and to maximize the class separation (LDA) [3]. PCA can be applied in order to further explore the underlying structure of the available data and to make an additional exploration of the data complexity. However, sometimes it does not preserve the full class separation because of its unsupervised nature. An additional feature selection substep can ensure a more useful feature subset. All these operations, including feature design, feature extraction, and feature selection, represent *feature engineering* tasks.

In many real cases several sources of data must be integrated and *consolidated* into a single data structure that should contain all the relevant original variables together with the designed features. Some *computational preprocessing operations* can also be applied, for instance, *scaling* and *normalization* in order to ensure, if required, the compatibility among the variables and generated features, depending on their significance. The original variables provide information about transactions, time-related information, the involved devices (especially in case of mobile payment applications), their operating systems, and other technical attributes that can be associated with the operational environments of the involved transactions. Some of these attributes can be relevant for certain fraudulent activities and therefore the corresponding variables should be preserved in the overall modeling process for the learning. There could be also cases of attributes with lower relevancy for the application target (in this case, the recognition of fraud cases), and therefore the variables may be discarded from the original set. But this can be only done within a comprehensive analytical process in order to explore the given variable significance and utility for the application target. These analytical operations should be completed with *feature engineering* operations in order to enrich the original data space with new features providing a significant value for the discrimination fraud vs. genuine.

2.1.2 The Supervised Learning Process and Application in Fraud Management

The *supervised learning* process and the *predictive modeling* in fraud detection application use-cases are based on the following typical operations:

- *Training*, for the model selection and building, with its proper parameter adjustment.
- *Validation*, for the performance estimation using a small data subset. The most common training/validation splitting ratio is 70%/30%. The validation dataset

is an internal subset that is used for the ROC analysis in order to estimate the designed model performance. Sometimes a cross-validation is applied in order to enhance the generalization capacity of the designed model. The most common approach for cross-validation in these use-cases is K-fold.

- *Testing in real application operational environment*, in which the already trained and validated model is applied in the operational environment of the target applications in order to make decisions according to the customer expectation (in this case, to accurately detect the fraudulent cases in order to prevent their damaging impact for the clients and also for the financial service providers).

The available datasets contain usually a huge number of records or samples, out of which only a very small fraction of records are marked as fraud cases. There are a large number of genuine records that are present in the original dataset. Using all genuine cases for the full modeling process is not feasible because of the hardware constraints and the execution time for the typical modeling operations (training, validation). Therefore, in many real use-cases, the modeling is not applied on the original full dataset. A *subsampled version of the dataset* is used for modeling, but with care to ensure the representative information for the target, therefore the fraud cases.

This is why the subsampling should be only performed for the genuine cases while preserving as much as possible the fraud cases in the subsampled dataset. All fraud cases should be taken for the advanced modeling operations.

On the other hand, in this context of genuine cases sub-sampling, the new custom features should be designed and computed using the *overall consolidated dataset*, before the advanced learning steps, and also neglecting the genuine case subsampling. In this way the resulted features can describe in a reliable way the informative properties of the entire dataset in order to exploit the real-time behavior of the involved actions corresponding to the different transactions and other financial operations.

In many use-cases, the new features that are designed and computed, using data science-specific tools and methods, ensure the performance improvement for the supervised learning models. Figure 1 depicts the ROC (receiver operating characteristic) curves with performance estimation for an example of a fraud vs. genuine classification model. In this example the target KPIs (TPR, true positive rate, and FPR, false positive rate, respectively) are the following: a TPR (detection rate) of 90% should be achieved for a reduced FPR (alert rate) not exceeding 10%. These performance levels significantly exceed the performances of the legacy rule-based solutions using rules for the fraud detection.

One can see the estimated performance improvements when the new features are added to the original data space, using the *feature engineering* techniques.

In this example, the performance measures (TPR, FPR) are estimated on a small validation subset that is drawn from the initial training set. The ROC curves show the expected performance of the designed model on the validation subset.

True positive rate (*detection rate*) is estimated as the percentage of fraud cases that are detected out of the total number of frauds. False positive rate (alert rate) is

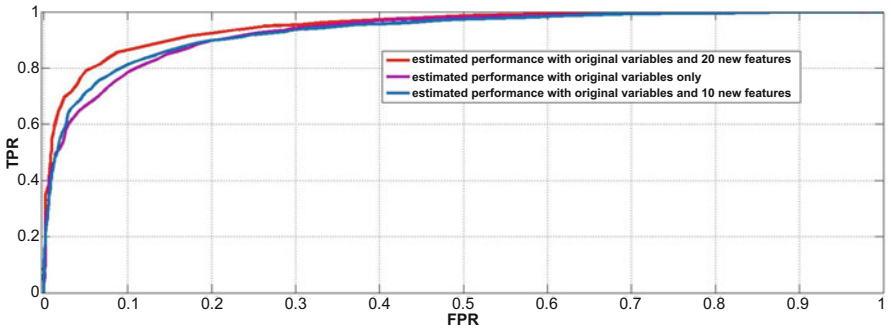


Fig. 1 Estimated performances (with ROC) on three cases: (**a**) without new features, (**b**) with 10 new features, and (**c**) with 20 new features

estimated as the fraction of genuine cases that are incorrectly marked as frauds over the total number of true genuine cases.

The overall predictive modeling process is applied on the following three cases in this example:

- An *initial model* which exclusively uses the *subsampled dataset without new features*, therefore only the original variables as provided for training and validation with the selected classification model. This model is applied in order to achieve a baseline performance estimation, the expectations in respect to which further improvements can be done. This baseline model without new features can be useful to explore the additional insights into the suitable modeling approaches that may be able to ensure the performance improvement. In Fig. 1 one can see the expected or estimated performance for the initial model without new features: the best operating point provides a detection rate (TPR) of 78% for an alert rate (FPR) of 10%, which may be not very convenient for the true cases of fraud recognition (in respect to the fixed target). However, the alert rate is significantly improved while comparing with the legacy rule-based solutions that still have a high false positives rate.
- An *improved model* with *the original variables and ten new features*, a case in which the supervised learning for predictive modeling is applied on a completed data space that includes the original variables together with additional ten features. These features are designed using *feature engineering* tasks and while taking into account the statistical properties of the original variables. One can see a performance improvement as concerning the detection rate (TPR) for the same alert rate (FPR): 82% vs. 10% (Fig. 1).
- An *improved model* with *the original variables and 20 new features*, a case that provides the best performance improvement for this modeling example, with a detection rate close to 90% for an alert rate around 10%.

Actually the full process to *design a suitable predictive model* (with *supervised learning*) for the fraud management (detection/prevention) use-case can be

approached in an iterative way, starting from the original variables and passing through several preprocessing and processing steps, with or without new feature design and computation. The main operations include

- *Data preprocessing* in order to ensure a cleaned dataset, sometimes applying computational preprocessing operations like scaling or normalization. In this stage a *data cleaning* operation is usually required in order to remove the noisy samples, duplicates, and also to efficiently handle the missing data cases.
- *Variables and feature selection* in order to exploit the most informative features (direct and designed features, respectively) for the next modeling stages.
- *Dimensionality adjustment*, usually to reduce the computational complexity and to efficiently handle problems like curse of dimensionality and performance peaking.
- *Data subsampling and training/validation split*, in order to take all the fraud cases and only a part of the genuine cases. The training/validation splitting is done using the typical ratio of 70% for training and 30% for validation, but depending on some particular requirements, other ratios can be considered, too.
- *Feature expansion for categorical cases*, actually an additional transformation of the feature space that is required in order to efficiently handle the categorical variable cases. A common approach for the categorical variable cases is to apply one-hot binary encoding for each of the possible categories or distinct values of those variables.
- *Classification model selection*, by exploring the behavior of several algorithms on the available datasets.
- *Classifier hyperparameter tuning* with a proper training process in order to ensure the model stability (to have a stable behavior in case of small changes in training dataset) and the desired performances.
- *ROC-based optimization* in order to find out the best operating point that ensures the expected performance in real case (the highest fraud detection rate for a fixed and reduced alert rate).

This iterative approach leads to a sequence of processing and learning tasks that finally generates an abstract mathematical representation of the input data which is the *data model*. The model should have some performances that are measured using the KPIs (key performance indicators): TPR and FPR. Their significance depends on the application objective. For the fraud management use-case, these amounts measure the designed system capacity to accurately detect the fraud cases (TPR) for a given alert rate (FPR) that should be minimized in order to prevent or to reduce the genuine case rejections.

On the other hand, one should remark that in many use-cases the performance that is *estimated* using *ROC* on the validation subset is slightly optimistic than the effective performance that is *evaluated* on real application operational environment (using, e.g., the *confusion matrix*). The *estimation* seems to be better than the *true performance* on the real operational cases. This is usually happening when the internal validation subset that is used for ROC is uniformly sampled from the overall training dataset; in such situations, the validation subset only represents the

training data patterns in a more efficient way than for the true testing cases. The generalization capability of the model can be improved using cross-validation in order to prevent or to reduce overfitting.

2.2 *Fraud Management Solution Design Process with Unsupervised Learning and Anomaly Detection*

The unsupervised learning approach for fraud management solutions should be considered in cases in which the absence of labeled samples has a significant impact on the performance of the designed and trained models.

The evolution of fraudulent activities types, the technological advances, the change in user's behavior, and the requirements for mobility in e-commerce and banking applications represent a few key factors that require considering other approaches such as *unsupervised learning* and *outlier detection*.

This is because the new types of frauds that occur may not be retrieved within the available training datasets for the *supervised learning*. On the other hand, the *unsupervised learning* can be applied for several goals that include the following:

- Exploring the dataset in order to find some hidden patterns.
- Clustering of data points based on specified similarity measures.
- Finding the outliers in a dataset. The outliers may represent abnormal cases for the application and they could be further investigated using statistics and more sophisticated data science tools in order to detect potential fraudulent activities or other malicious events/cases that otherwise could not be identified using the supervised learning.

The *cluster analysis* for the fraud management use-case allows for a more detailed exploration of the data space in order to determine the way in which the data points describing operational cases of transactions are grouped (Fig. 2).

Among the *clustering algorithms* that can be used in applications belonging to the financial field, including for fraud management, one can mention *DBSCAN* (density-based spatial clustering of applications with noise) [4, 5]. DBSCAN is a density-based algorithm that generates a partitional clustering in which the number of clusters is not predetermined but it automatically results from the algorithm running. The data points that belong to low-density regions are considered as representing noise. Usually DBSCAN does not produce a complete clustering. Another difference between DBSCAN and the most common clustering algorithm *K-means* is that DBSCAN can recognize the outliers (Fig. 3), and therefore it does not cluster the data points representing abnormal cases (*outliers* or anomalies for the application, in this case new types of frauds that cannot be seen during the training period of the supervised models).

Outlier or *anomaly detection* is a reliable design approach for applications in which the goal is to accurately detect new and very often more sophisticated

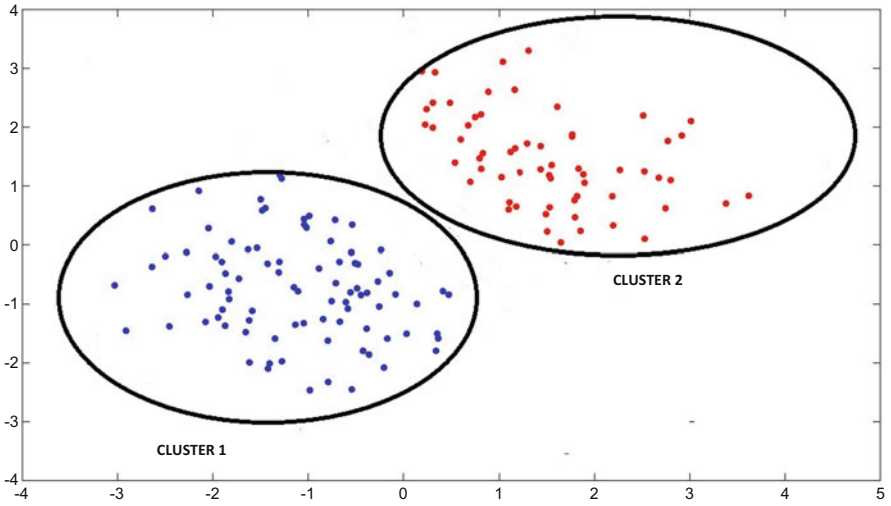


Fig. 2 Data clustering: example with two clusters in a two-dimensional data space

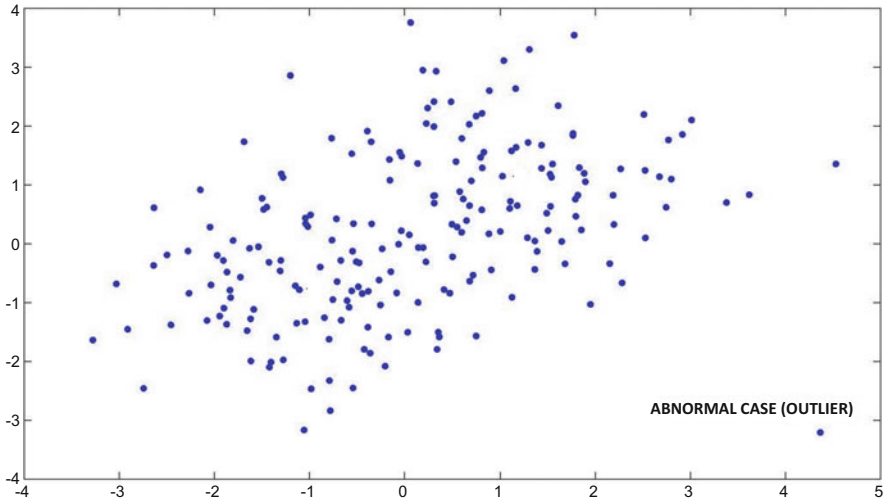


Fig. 3 An example with an abnormal case (outlier)

fraudulent actions, for example, on large financial markets. In this approach, the analytic and computational process looks to identify the observations that are inconsistent in respect to the other samples or data points within the available dataset. The advantage is that in this way the new fraudulent cases could be detected almost in real time, especially if the used detection technique is applied on an enriched data space with new features that were previously designed using statistics and various mathematical operators applied on the original variables. Therefore, the

key factors for an improved design in the field of fraud management for financial applications are to integrate innovative approaches including anomaly detection and the data space enriching with new features.

An *outlier* is an observed or measured value that significantly deviates from all the other observations that are given in a dataset, making it suspicious to be generated from a different and abnormal mechanism than the other cases [6]. An example is represented in Fig. 3, where there is a data point that is placed outside the region containing the normal cases. The distance is indeed significant in the depicted example. The outlier corresponding data point represents an abnormal case for the running application in its operational environment.

The outlier detection for fraud management in financial industry is justified by the fact that since most of the data samples represent genuine cases, the anomalies within the analyzed dataset should represent frauds/malicious behaviors of the market participants or, in rare cases, misuses or other exceptions that can occur within the transactions.

This is an application use-case of outlier detection in which the outliers must not be removed from the analytical and processing tasks, as the case in other fields. Actually the outliers represent the true target of the overall modeling process, because in most cases it can be assumed that the detected outliers represent

- Fraudulent activity cases in transactional processes (for banking and e-commerce applications)
- More sophisticated malicious activities, such as market manipulation on large capital markets and money laundering

This approach is applied using a multidimensional data space and the full process requires a multivariate analysis, taking into account the contributions of the original variables and the new designed features. In such enriched multidimensional feature space, the isolated data point (as represented in Fig. 3) is a multivariate outlier corresponding to an abnormal case. Such detected abnormal cases should be further investigated in order to recognize a potential new type of malicious activity or unsuitable behavior of the market participants.

The fraudulent activity detection in large financial markets is a typical case in which the outlier detection should be applied taking into account the multivariate nature of the problem. This is because several variables and designed features must be considered for analysis and processing.

The *multivariate outlier detection* deals with significant challenges. The most important challenge is that in these cases, the multidimensional observations cannot be always detected as representing *outliers* or *abnormal cases* if each of the involved variables and features is separately considered for analysis. An important requirement is to take into account the interactions among the variables and various relationships among the generated features. Two variables or features could describe normal cases if they are independently taken, but the testing could lead to opposite results if the variables are considered together within the analytical process [7].

The multivariate outlier detection can be approached with two main categories of methods [7]:

- *Statistical methods* that use estimated distribution parameters. These methods make use of some distance measures in order to determine the observations that are far from the representative or centroid of the data cluster (distribution). Mahalanobis distance is very often used in this approach, due to its properties (scaling invariance, correlation exploiting). The observations for which Mahalanobis distance is large can be considered as outliers.
- *Data mining-related methods* that are parameter-free in many cases. These methods do not assume a known generating model for the available data. The main application is for large databases and high-dimensional data spaces.

References

1. Perić, A., Polić, N, & Kozarević, E. (2016, January). *Application of data science in financial and other industries*. In *5th International Scientific Symposium "Economy of eastern Croatia – vision and growth"*
2. Report Financial Stability Board. (2017, November 1). *Artificial intelligence and machine learning in financial services. Market developments and financial stability implications*. http://www.fsb.org/2017/11/artificial-intelligence-and-machine-learning-in-financial-service/
3. Theodoridis, S., & Koutroumbas, K. (2009). *Pattern recognition* (4th ed.). Elsevier/Academic.
4. Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. (2019). *Introduction to data mining* (2nd ed.). New York: Pearson Education.
5. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *KDD, 96*, 226–231.
6. Hans-Peter Kriegel, Peer Kröger, Arthur Zimek: Outlier Detection Techniques, In: Tutorial Notes, SIAM SDM 2010, Columbus, Ohio, Proceedings of the 2010 SIAM International Conference on Data Mining.
7. Ben-Gal I., Outlier detection, In: Maimon O. and Rockach L. (Eds.) Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers," Kluwer Academic Publishers, 2005, ISBN 0-387-2443.

Stochastic Analysis for Short- and Long-Term Forecasting of Latin American Country Risk Indexes

Julián Pucheta, Gustavo Alasino, Carlos Salas, Martín Herrera, and Cristian Rodriguez Rivero

1 Introduction

In 1981, the economist of the International Finance Corporation, Antoine van Agtmael [1], coin the concept of emerging economy to refer to the transition between developed and developing economies. Since then and given the potential for growth and profitability, the financial markets of these economies have received important attention [2]. The methods of risk assessment and country economy analysis are mostly qualitative or quantitative deterministic, among which those that analyze the JP Morgan's Emerging Markets Bond Index (EMBI) have been arisen [3].

The EMBI is a daily measure of the additional risk with respect to a safe investment in a certain economy of a country or region. It is an indicator that daily measures the investor sentiment regarding the country risk. It reflects the perception of political events or economic news incorporated into the return of investment in these markets. EMBI has variants that comprise a selection of sovereign debt

J. Pucheta (✉)
Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de Córdoba, Córdoba, Argentina
e-mail: jpucheta@unc.edu.ar

G. Alasino
Universidad Torcuato Di Tella, Buenos Aires, Argentina

C. Salas · M. Herrera
FTyCA-Universidad Nacional de Catamarca, San Fernando del Valle de Catamarca, Catamarca, Argentina

C. R. Rivero
Faculteit der Natuurwetenschappen, Universiteit van Amsterdam, Amsterdam, Netherlands
e-mail: c.m.rodriguezrivero@uva.nl

© Springer Nature Switzerland AG 2020

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_12

securities in US dollars, by a selected group of emerging economies [4]. The indices include variables such as liquidity, maturity, and structural restrictions. EMBI was introduced in 1995 with data since December 31, 1993, to the present [4].

The spreads analysis of the sovereign bonds [5] has two aspects. One of them is the study of the causal factors that originate its spreads. The remaining ones are for projecting early warning of fiscal crisis [6, 7], so it serves as an indicator of fiscal vulnerability and currency crisis [8], among others. Low levels of the gap on the risk-free rate reveal low cost of financing in international markets, which highlights the fiscal resilience to face turbulence or sudden changes in external markets, hence the importance for studying the EMBI time series forecast. In this article, the EMBI Global index is used for forecasting the evolution of economies using historical time series obtained from public sources [9].

In this work, the countries under study have been selected by emerging market concept. This category groups countries with the ability to pay the external debt, whose credit ratings are located up to the BBB + / Baa12 category. Note that emerging markets imply those economies that are experiencing positive but unstable rates, opening their economies abroad, presenting risks, and having fiscal uncertainty given their political transformation. However, these countries are vulnerable in varying degrees to weakening in the face of an external shock or political changes in the domestic economy and, in such case, to suspend the payment of their financial obligations.

This article is structured as follows: after this Introduction, Sect. 2 describes an overview of the problem statement, with a glance on the amount of the experimental data and the objective of this work. Section 3 gives a brief description of the related work in this field. Section 4 shows the proposed approach of the tuning and the selection method based on fractional Brownian motion using a moving average nonlinear autoregressive model based on neural networks. Section 5 shows the sets of monthly EMBI series from Latin American countries, with emphasis on the statistical evaluation of the short-range case (800 training data) and the long-range case (2700 training data). Section 5 presents discussions, lessons learned, and concluding remarks.

2 Problem Formulation

It is well-known that nobody has the ability to capture and analyze data from the future [10], hence the search for a predictive model that allows to predict the future using data from the past. The EMBI prediction allows to have a trace of the index evolution and thus identify risks and opportunities to guide the decision maker to perform decisions in future transactions. In this case, those transactions were grouped in time periods of 30 days and 350 days, for short- and long-term decision, respectively. However, for first-world countries indexes [11] such as US, the class of long-term is assigned to periods larger than 1-year. Here, it focus on Latin American countries whose instability trends often turn up in one year.

The available data series for the selected countries under study have different lengths, i.e., Argentina, Brazil, Chile, and Mexico have registers since December 1998; the Regional EMBI is registered since June 1999, and the Colombia's EMBI since December 1999 [4]. However, the available historical data used in this study retrieved from free sources [9] are more limited (since year 2007) having 2781 registers for each time series by December 28, 2018 (3037 by Dec 18, 2019). Therefore, the time period forecast of 30 days for short term and 350 days for long term was stated for this study, with 2781 available historical data for each time series.

3 Related Work

The EMBI's community proposed references for background [12], where a description for a wide sort of schemes is detailed, and here, the multilayer perceptron for regression and function approximation is used. Some efforts have been made in forecasting EMBI time series from Latin American countries by using diffusion model [13] for forecasting or by analyzing the data [14] to state trends. The amount of data is relatively small with respect to deep learning approach [15, 16] which involves millions of parameter to tune which results in task to avoid overfitting. However, none of the popular machine learning algorithms have been created for time series forecasting, and time series data need to be preprocessed as indeed [17] that is useful when long dynamic ranges are present. In this work, the preprocessing task is performed prior the analysis for forecasting by using static scaling.

The related work focused on EMBI time series forecast proposes models for forecasting and analysis for fractional structures [13, 14] among others. In this work, a nonlinear model based on neural network is proposed with an analysis of its performance measuring the fractal structure into the forecasted time series by the Hurst parameter. The main contribution is a methodology to provide a time series forecast that presents the same roughness than that of the data time series.

4 Proposed Approach

In this chapter, a classical nonlinear autoregressive (AR) model based on neural networks (NN) whose parameters are batch tuned and its performance is evaluated by stochastic analysis is proposed. After training is completed, predictions are generated by using Monte Carlo with normal Gaussian noise (Gn) and fractional Gaussian noise (fGn). The noise is generated by fractional Brownian motion with the Hurst parameter $H \in (0,1)$, and normal Brownian motion is retrieved for $H = 0.5$. Also, if $H \in (0,0.5]$, then the time series is rough, whereas if $H \in [0.5,1)$, then the time series is smooth so H can be used as a measure for the time series. From a stochastic point of view, when H is greater or lower than 0.5, the series tends to present long- or

short-term dependence [18]. Here, the expectation from each prediction ensemble is computed for obtaining a deterministic time series both for Gn and fGn. The stochastic analysis of these time series is performed for determining the forecast with the appropriate stochastic roughness characteristic, and so it is chosen for the forecast. This section details the training and the model selection method based on fractional Brownian motion.

4.1 The Basic Problem

The classical prediction problem can be formulated as follows. Given past values uniformly time spaced of a process, namely, $x(n-1)$, $x(n-2)$, $\ldots$, $x(n-N)$, where N is the time series length, the present value $x(n)$ must be predicted. Here, a prediction device is designed by considering the given sequence $\{x_n\}$ at time n from which it can be obtained the best prediction $\{x_e\}$ for the following future 30/350 values sequence. Hence, it is proposed a predictor filter with an input array with l_x elements, which is obtained by applying the delay operator to the sequence $\{x_n\}$. Then, the filter output generates x_e as the next value, which will be equal to the present value x_n. So, the prediction error at time k can be evaluated as

$$e(k) = x_n(k) - x_e(k) \tag{1}$$

which is used by the learning rule to adjust the predictor filter parameters.

The predictor filter is implemented by an AR NN-based nonlinear adaptive filter. The NNs are used as a nonlinear model building; the better the model performs the underlying physical process dynamic behavior that generates the data, the better will be the forecast of the AR filter. So, a time-lagged feed-forward NN is used. The present value of the time series is used as the desired response for the adaptive filter, and the past values of the signal are supplied as adaptive filter input. Then, the adaptive filter output will be the one-step prediction signal as is stated by (1). In Fig. 1, the block diagram of the nonlinear prediction scheme based on a NN filter is shown. Therefore, our aim is to obtain the best prediction (in roughness sense) of the present values from a pseudo-random time series.

4.2 NN-Based AR Model

Several experiences had been obtained from previous works detailed on [19]. Here, a NN-based AR filter model following the scheme detailed in Fig. 1 is tuned. The NN used is a time-lagged feed-forward network class. The NN topology consists of l_x inputs, one hidden layer of H_o neurons, and one output neuron and is tuned as shown [19]. The learning rule used in the parameter tuning process is based on the Levenberg-Marquardt method [20].

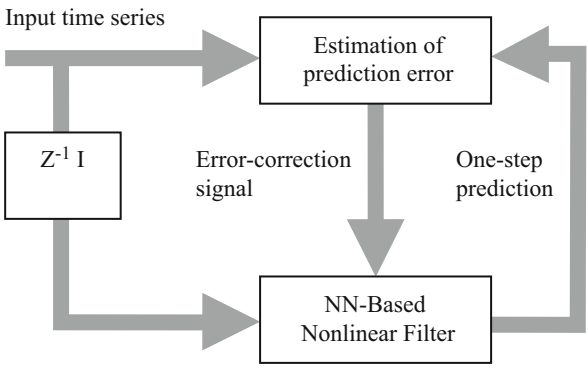


Fig. 1 Block diagram of the neural network-based nonlinear predictor

In order to predict the future of $\{x_n\}$ one step ahead, the first delay is taken from the tapped-line x_n and used as input. Therefore, the output prediction can be denoted by

$$f(n) = F_p\left(\left[x_{n-1} \;\; x_{n-2} \;\; \ldots \;\; x_{n-1-l_x}\right]^T, \mathbf{W}_1, \mathbf{W}_2\right) \tag{2}$$

where $F_p(\cdot, \mathbf{W}_1, \mathbf{W}_2)$ is the nonlinear predictor filter with l_x inputs from the data with an extra unitary input and $x_e(n)$ is the output prediction at time *n-1*. The activation function is hyperbolic tangent for the hidden layer with H_o processing nodes and linear for the output layer which has H_o inputs with an extra unitary one. Therefore, the matrixes that contains the parameters are defined as $\mathbf{W}_1 \in \Re^{H_o \times (l_x+1)}$ and $\mathbf{W}_2 \in \Re^{1\times(H_o+1)}$. Each one of the hidden processing nodes has *tanh(·)* as its activation function. Thus, the predictor filter contains tuning parameters $\{\mathbf{W}_1, \mathbf{W}_2\}$ that must be computed. The selection of *tanh(·)* assumes larger computing times than rectified linear (ReLU) and similar present in several big data approach [12], but given the amount of data, the computing time is still acceptable.

4.3 Monte Carlo Implementation Including Fractional Brownian Motion

In this work, the Hurst parameter is used as a statistical criterion for time series forecast by selecting the assessment in the algorithm. This H gives an idea of roughness and also determines its stochastic dependence. The definition of the Hurst parameter appears in the Brownian motion from generalizing its integral to a fractional one. The fBm is defined in the pioneering work by Mandelbrot [18] through its stochastic representation.

$$B_H(t,\omega) = \frac{1}{\Gamma\left(H+\frac{1}{2}\right)}\left(\int_{-\infty}^{0}\left((t-s)^{H-\frac{1}{2}} - (-s)^{H-\frac{1}{2}}\right)dB(s,\omega) + \int_0^t (t-s)^{H-\frac{1}{2}} dB(s,\omega)\right) \tag{3}$$

where $\Gamma(\cdot)$ represents the Gamma function

$$\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx \tag{4}$$

And $0 < H < 1$ is called the Hurst parameter. The integrator *B(t)* is a stochastic process, ordinary Brownian motion. Note that *B(t)* is recovered by taking $H = 0.5$ in (3). Here, it is assumed that *B(t)* is defined on some probability space (Ω, *F*, *P*), where $\omega \in \Omega$ is an element, and Ω, *F*, and *P* are the sample space, the sigma algebra (event space), and the probability measure, respectively. Thus, an fBm is a time continuous Gaussian process depending on the so-called Hurst parameter $0 < H < 1$. The ordinary Brownian motion is generalized to $H = 0.5$ and whose derivative is the white noise. The fBm is self-similar in distribution, and the variance of the increments is defined by

$$\mathrm{Var}\left(B_H(t,\omega) - B_H(s,\omega)\right) = \nu \cdot |t-s|^{2H} \tag{5}$$

where ν is a positive constant, 1 if $s = 0$. This special form of the variance of the increments suggests various ways to estimate the parameter *H*. In fact, there are different methods for computing the parameter *H* associated to Brownian motion [18]. In this work, the algorithm uses a wavelet-based method for estimating *H* from a trace path of the fBm with parameter *H* [21].

For generating data, e.g., three ensembles from fBm with different values of *H* are shown in Fig. 2, where the difference in the increment velocity and variances for each *H* can be noted. The figure shows synthetized traces using the method described in [22], where the black lines correspond to five traces for each *H*, the magenta lines show the theoretical variances using (5), and the blue lines show the variance estimated by

$$\mathrm{Var}\left(f(t,\omega)\right) = \frac{1}{N}\sum_{\omega}\left(f(t,\omega) - E\left[f(t,\omega)\right]\right)^2 \tag{6}$$

where *N* is the size of *f(t,ω)* for each time *t*. For implementing (5), in this chapter, the method detailed in [23] was used. This article proposes the use of fGn in the Monte Carlo; given the spread and scale of the fBm showed in Fig. 2, the generation of the ensemble has two aspects. One of them is *H*; the other one is the variance function that depends on time.

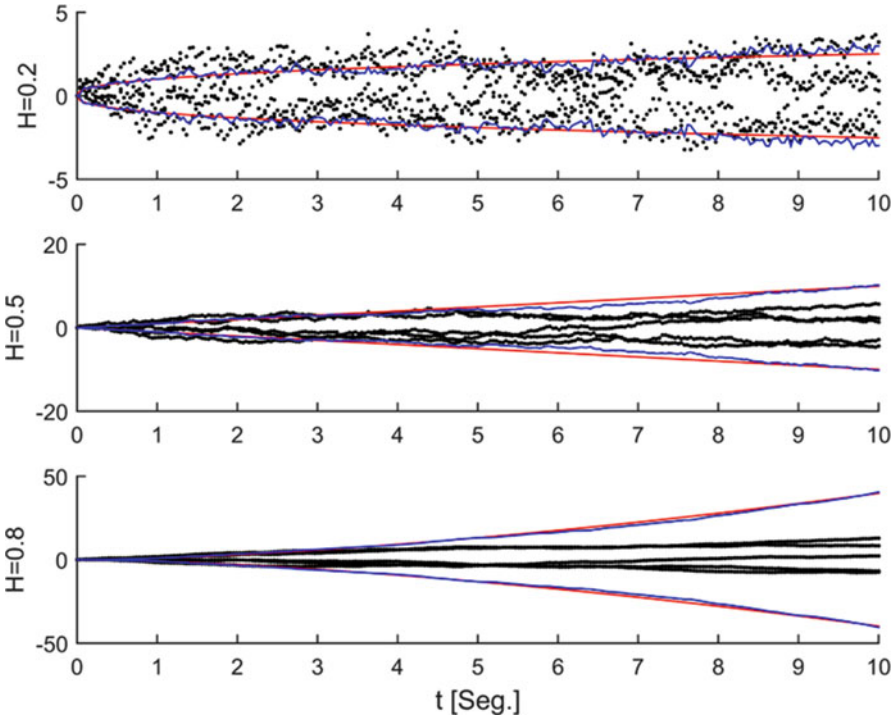


Fig. 2 Five sample paths from fractional Brownian motion for three values of H (estimated over 50-trace ensemble). Theoretical and numerical variances explicit in (5) and (6), respectively, are plotted

4.4 Algorithm Description

Our thesis asserts that if some process evolves along time with any H, it will do in the future with the same H, namely, keeping its smoothness or roughness. To do so, a classic nonlinear model based on NNs batch-tuned is proposed. In the tuning process, data are randomly split for generalizing and training, with a 15%/85% ratio, respectively. Such 15/85 was selected after a brief analysis to summit the bias variance trade-off [12, 20]. Furthermore, since the last data are the most important one given the series nature, the last or the last two are left for true data for test. Thus, the relevant data quantity to be considered from the series is then determined for performing the desired forecast. By this way, the parameters contained into the matrixes $\mathbf{W}_1$ and $\mathbf{W}_2$ are obtained.

After tuning the filter parameters defined in Eq. (2) $\{\mathbf{W}_1, \mathbf{W}_2\}$, the prediction traces are generated using normal Gaussian (Gn) and fractional Gaussian noise (fGn). In order to generate the ensemble with R traces for forecasting the time series, Eq. (2) is modified by

$$f\left(n+q,\omega\right)=F_p\left(\left[\,x_{n-1}\;x_{n-2}\;\ldots\;x_{n-1-l_x}\,\right]^T,\mathbf{W}_1,\mathbf{W}_2\right)+\theta\cdot\Delta B_H\left(q,\omega\right) \tag{7}$$

where $q=1,2,\ldots F_H$ sets the future time, F_H is the forecast horizon, $\omega = 1, 2, \ldots R$ denotes each trace, R is the ensemble size, and θ is a parameter for de-normalizing the increments of B_H.

For selecting a value for θ, one must take into account the length of the forecast horizon F_H, which in this work is either 30 or 350 days, and also the series dynamic range, by using

$$\theta=\left(\frac{1}{F_H}\sqrt{c\cdot\left(\frac{\max\left(\{x_n\}\right)-\min\left(\{x_n\}\right)}{\max\left(\{x_n\}\right)}\right)}\right)^{2H} \tag{8}$$

And specifically for the Gaussian case when $H = 0.5$, the exponent $2H$ is replaced by 0.5, the coefficient c was set to 0.001, and it was introduced for preserving suitable behavior of the ensemble for every H. The ensemble described by Eq. (7) is analyzed by classical statistical methods for obtaining the mean and the variance functions.

According to the stochastic behavior of the data time series, H can be greater or smaller than 0.5, which means that the data time series tends to present long- or short-term dependence, respectively [18]. This dependence will modify the input length of the filter and the number of processing units [19].

After the tuning process is completed, the ensemble average is computed for each future time of the forecast horizon by

$$\{x_e\}=E\left[f\left(e,\omega\right)\right],e=1,2,3\ldots F_H, \tag{9}$$

where $\{x_e\}$ is the mean forecast time series, which will be analyzed by the roughness criterion. It is defined a sequence pair, where one sequence comes from the data

$$\{x_n\},n=1,2,3\ldots N, \tag{10}$$

and the other one is composed by the former concatenated with the forecasted one which is the ensemble's expectation (9), hence

$$\{\{x_n\},\{x_e\}\},e=1,2,3\ldots F_H. \tag{11}$$

Both sequences (10) and (11) are analyzed by the method detailed in [21], and their estimated H, namely, $H\{x_n\}$ and $H\{x_n,x_e\}$, must match between them, given that if the underlying process is rough up to time n, it will continue with the same roughness. In order to move the estimate of $H\{x_n,x_e\}$, the ensemble (7) must be computed with a new value of H into its fBm. This value of H has different influence into the estimate of $H\{x_n,x_e\}$, so it must be varying from small quantities. Thus, the

Table 1 Algorithm that performs the short/long-term time series forecast

1.	Set the neural network (NN) topology by assigning l_x, H_o for dimensions of $\mathbf{W}_1\mathbf{W}_2$
2.	Chose the data for truth test that are just one or two of the last time series data that will not be used by training or validating processes
3.	Train the NN with 85% of the data and check the over-fitting with the remaining 15%
4.	Stop the training if the generalization error increases
5.	If the truth test data is not suitably forecasted, go to 1 for varying the topology as indicated in [19]
6.	Set θ: Via Eq. (8)
7.	Run the Monte Carlo including H described by Eq. (7)
8.	Compute the mean value by (9) for obtaining $\{x_e\}$
9.	Estimate the roughness either via $H(\{x_n\},\{x_e\})$ or $H(\{x_e\})$, for short- or long-term forecast
10.	Compare the obtained roughness with the data one $H(\{x_n\})$
11.	If the obtained roughness is close enough, then the obtained mean and variance of the time series are the algorithm results
12.	Else go to 6 and modify the noise by changing H, $0 < H < 1$
13.	Analyze trend, seasonality, and another typical characteristic. If any of those does not match, go to 1 to modify the topology by following [19]

ensemble mean of each prediction is taken, and the one with the suitable stochastic roughness H is chosen as the forecast.

Given that the method for estimation [21] has better performance with time series with about 200–500 registers, for the long-term analysis case, the study on the time series is defined by

$$\{x_e\}, e = 1, 2, 3 \ldots F_H. \tag{12}$$

Therefore, for long-term stochastic analysis, the time series under study are (10) and (12) giving the estimates $H\{x_n\}$ and $H\{x_e\}$, respectively. Table 1 details the method for tuning and selecting the most suitable model based on fBm and NNs, where the initial conditions were $l_x = 12$, $H_o = 24$ and $H = 0.5$.

5 Implementation with EMBI Time Series

The method has been implemented by considering that there are considerable data from free sources [9] for Latin American countries and Latin American region in itself. Here, it is proposed to build a model for forecasting the time evolution of the next 30 data following the current time, which is going to be considered as short range, and the next 350 data that corresponds to the long range. For both cases, the methodology uses a batch training with a validation set of 15% of the data taken at random. In the short-range case, 800 training data were used and 2700 for the long range. The last group of one to three data was left as a test data or truth test, mainly to

penalize the underestimation. The prediction was made by Monte Carlo simulation where the noise used was stationary, seasonal, and pseudorandom, although with a roughness characteristic determined by the Hurst parameter H, generated according to [22] by using [23]. Thus, the roughness is evaluated by H, which is computed using a wavelet-based method [21].

After the tuning process is completed, two sequence pairs are defined. One pair is

$$\{x_n\}, n = 1, 2, 3 \ldots N_S, \tag{13}$$

with

$$\{\{x_n\}, \{x_e\}\}, e = 1, 2, 3 \ldots F_S \tag{14}$$

for the short-term forecast given $N_S = 800$, $F_S = 30$. The other pair is

$$\{x_n\}, n = 2350, 2351, \ldots, N_L, \tag{15}$$

with

$$\{x_e\}, e = 1, 2, 3 \ldots F_L \tag{16}$$

for the long-term forecast given $N_L = 2700$, $F_L = 350$. Each pair of sequences must show the same H parameter estimated by [21] over the time series, namely, $H\{s\}$ for the time series s. The intention is that the time series shows a behavior with the same roughness along the data and the forecasted time series. Note that each data time series has more data than those used in the roughness comparison, given that the method [21] has lower sensitivity for time series extremely large.

The performance of the filter is evaluated using the well-known Symmetric Mean Absolute Percentage Error (SMAPE) proposed in the most of metric evaluation, defined by

$$\text{SMAPE}_S = \frac{1}{n_F} \sum_{e=1}^{n_F} \frac{|X_e - F_e|}{(X_e + F_e)/2} \cdot 100 \tag{17}$$

where e is the future time observation, n_F is the true set size either F_S or F_L both for short- or long-term forecast, s is each time series for every EMBI under study, and X_e and F_e are the true and the forecast time series values at time e, respectively. The values F_e for time series are obtained by averaging over ω Eq. (7) for every future time e.

5.1 Monthly EMBI Forecast

The obtained results by using the algorithm of Table 1 for the short-term forecast are summarized in Table 2. In order to compare the results, Table 2 has been divided into three parts, one of them with the original characteristic time series, the second for the fGn used for the Monte Carlo in (7), and the last one for reference when the normal Gn is used in the Monte Carlo (7). Additionally, each column contains the H parameter associated to the time series detailed in the first column. The first row contains values from the data series used in the computation, the second row shows data values from the expanded time series defined in (14), and the third row shows the figures with the stochastic ensemble recommended by the algorithm of Table 1. The fourth row has the values of the SMAPE index expressed by (17) for comparing the true values against the forecasted ones. Note that for each time series of the first row reveals the H parameter from the data, namely, original roughness. Therefore, the other values of each column must get as close as possible to this H value.

Every result is shown via a graphic representation for each Monte Carlo with its H parameter specified and the 50% and 95% confidence intervals indicated. So, in Fig. 3, the prediction for the LAT EMBI time series with a Gn is shown. For LAT forecasted time series, its original roughness is 0.64581, and the closest H belongs to the Gn whose forecasted time series (14) gives a roughness of 0.64505 as is detailed in Table 2. And then, the SMAPE is computed both for fGn and Gn time series, and its values are in the fourth and sixth rows of Table 2. In the case of Argentina's EMBI time series, the original roughness of the data series $H\{x_n\}$ is 0.66917, and the forecasted time series for Gn noise ensemble are shown in Fig. 4 which has an $H\{x_n,x_e\}$ of 0.66895 being the recommended forecast given that it is closest to the original $H\{x_n\}$ as shown in the second column in Table 2. Following the same procedure to analyze the Chile, Colombia, and Mexico time series, each ensemble generated by normal noise is shown in Figs. 5, 6, and 8. In Table 2, the values that indicate the recommended forecast by this method are in bold font. On the other hand, the Brazil time series forecast shows its best fit with respect to the roughness H for fGn, whose results are shown in Fig. 7, and its roughness values are in the sixth column of Table 2.

Table 2 Roughness results with the implementation of the 30-day algorithm (December 2019)

	Regional LAT	Argentina	Chile	Colombia	Brazil	Mexico
$H\{x_n\}$	0.64581	0.66917	0.503	0.503	0.63827	0.65115
$H\{x_n,x_e\}$ fGn	0.64683	0.66941	0.49902	0.49902	**0.63864**	0.65077
Monte Carlo	Figure 3	Figure 4	Figure 5	Figure 6	Figure 7	Figure 8
SMAPE	5.2997	21.0018	2.1712	4.8116	4.4028	1.5965
$H\{x_n,x_e\}$ Gn	**0.64505**	**0.66895**	**0.50227**	**0.50227**	0.6376	**0.65128**
SMAPE	5.6569	20.9321	2.1082	4.9689	4.5600	1.7397

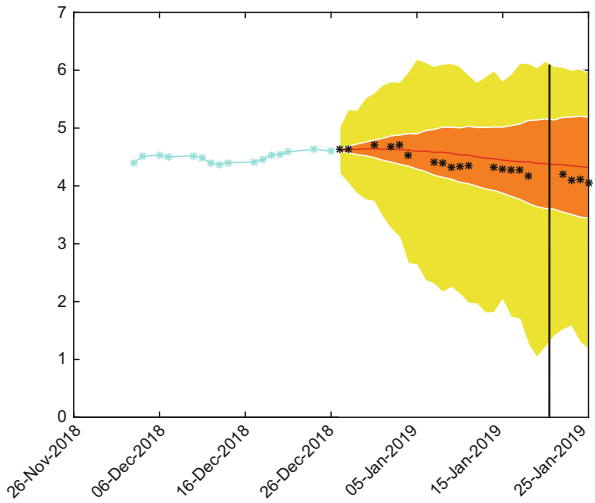


Fig. 3 Latin American EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

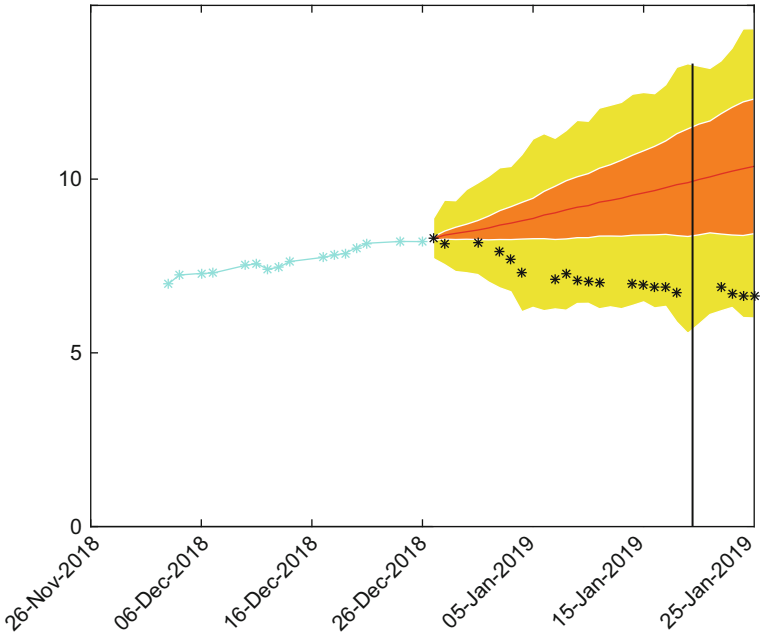


Fig. 4 Argentinean EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

5.2 Using the Monthly EMBI Forecast

Data series of the Latin American (LAT) EMBI index were used, available online [9] since 2007 up to December 18, 2019, so the data necessary to evaluate the results

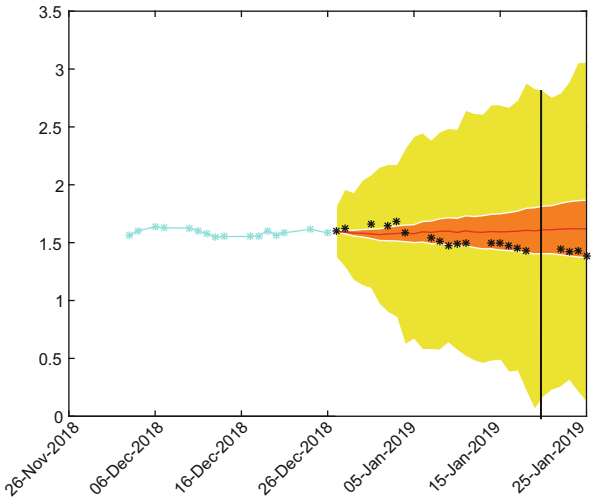


Fig. 5 Chilean EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

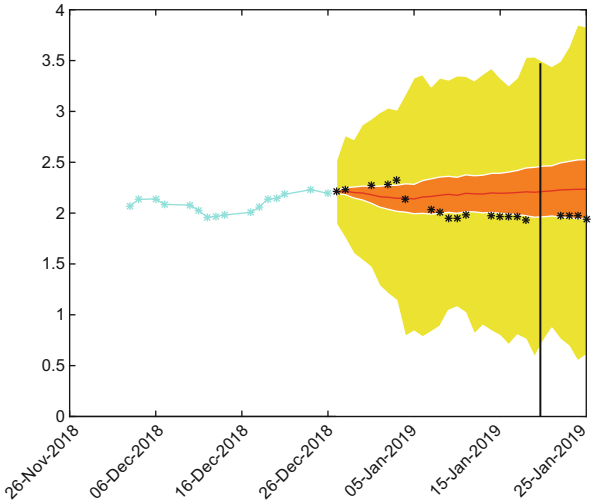


Fig. 6 Colombian EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

are included. The evaluation with true data consists of letting the practitioner make a decision in the LAT region. So, his focus is firstly on the region and then on a specific country. The LAT's EMBI has the characteristic acceptable to perform the forecast by this method, and its field test can be seen on the second column fourth row of Table 3. The first row has the trend information, whose interpretation will demand down or stable behaviors. The second row introduces information about the variation, although for this case, it is smooth for all cases. The third row has the range for 55% likelihood computed for 30 days. The fourth row has the true values, which may indicate the effectiveness of the forecast. Note that the only case out of computed range was the Argentinean one, even when the computed range turned out

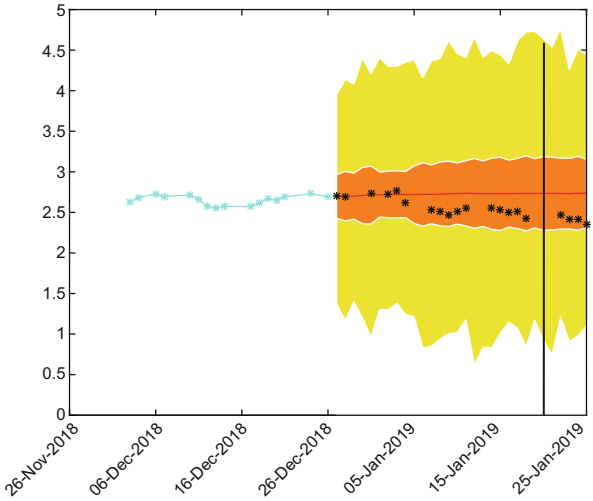


Fig. 7 Brazilian EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

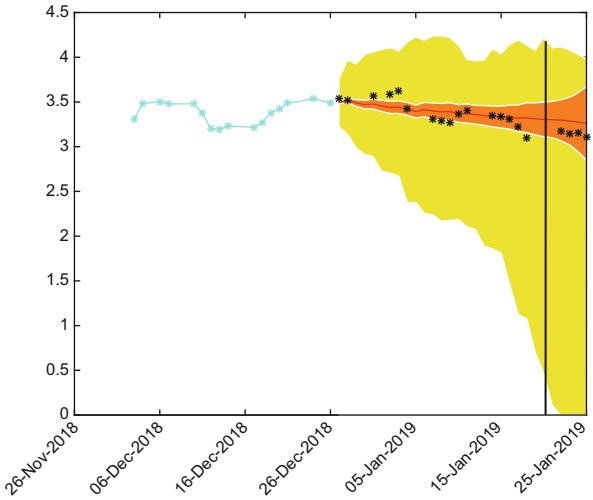


Fig. 8 Mexican EMBI 30-day forecast. NN topology: $l_x = 14$, $H_o = 27$

to be the broader one. Thus, the decision maker would tend to focus his investment analysis on the four last columns, given its trend, variation, and range for the risk indicated by the forecasted EMBI.

5.3 Annual EMBI Forecast

The annual case involves the criteria associated to the time series detailed in (16), where the time series is about ten times the monthly case. The forecast's results

Table 3 Short-term EMBI prediction and true values by January 19, 2019

	Region LAT	Argentina	Chile	Colombia	Brazil	Mexico
Trend	Down	Up	Stable	Stable	Stable	Stable
Variation	Smooth	Smooth	Smooth	Smooth	Smooth	Smooth
Range	350–500	750–1200	130–170	190–240	225–300	290–340
True values by January 19, 2019	407	678	145	191	235	330
Ensemble (scale 1:100)	Figure 3	Figure 4	Figure 5	Figure 6	Figure 7	Figure 8

Table 4 Roughness results with the implementation of the algorithm for 350-day horizon forecast (December 2019)

	Regional LAT	Argentina	Chile	Colombia	Brazil	Mexico
$H\{x_n\}$	0.64221	0.70392	0.40106	0.65106	0.647	0.54395
$H\{x_n,x_e\}$ fGn	0.76967	**0.78425**	0.6938	0.84175	**0.6444**	**0.29233**
Monte Carlo	Figure 9	Figure 10	Figure 11	Figure 12	Figure 13	Figure 14
SMAPE	1.9837	7.7683	4.4007	4.2908	3.8722	3.0972
$H\{x_n,x_e\}$ Gn	**0.65828**	0.58782	**0.47213**	**0.67277**	0.44172	0.83236
SMAPE	4.2451	8.6381	6.8195	9.6330	5.9993	3.0829

are summarized in Table 4. Following the scheme for monthly forecast results, here again, each column contain its H parameter associated to the time series detailed in the first column. The first row belongs to the data series used in the computation, namely, original; the second row shows the expanded time series defined in (15); and the last rows show the values of ensembles generated by the standard Gn used in the Monte Carlo. The fourth row indicates the square mean absolute predictive error between forecasted and real EMBI time series.

Every result has a graphic representation for each Monte Carlo with the H parameter associated to the noise used in its computation that gives the best fit. So, Fig. 9 shows the ensemble for the LAT EMBI time series forecast using Gn jointly with the generated by fGn, whose values are shown in the second column of Table 4. For LAT forecasted time series, its original roughness is 0.64221, and the closest H belongs to the Gn whose forecasted time series gives a roughness of 0.65828. In the case of EMBI time series belonging to Argentina, Brazil, and Mexico, the roughness measured by H of the forecasting data time series are 0.70392, 0.6444, and 0.29233, respectively. Each forecasted time series was generated by fGn as shown in Figs. 10, 13, and 14 which are the recommended forecast given the fit roughness. For reference, the roughness H measured in the generated forecasted time series by Gn is also shown in Table 4. Following similar reasoning to analyze the EMBI time series of Chile and Colombia, each ensemble generated by normal noise is shown in Figs. 9, 11, and 12 with their associated roughness values detailed in Table 4, respectively.

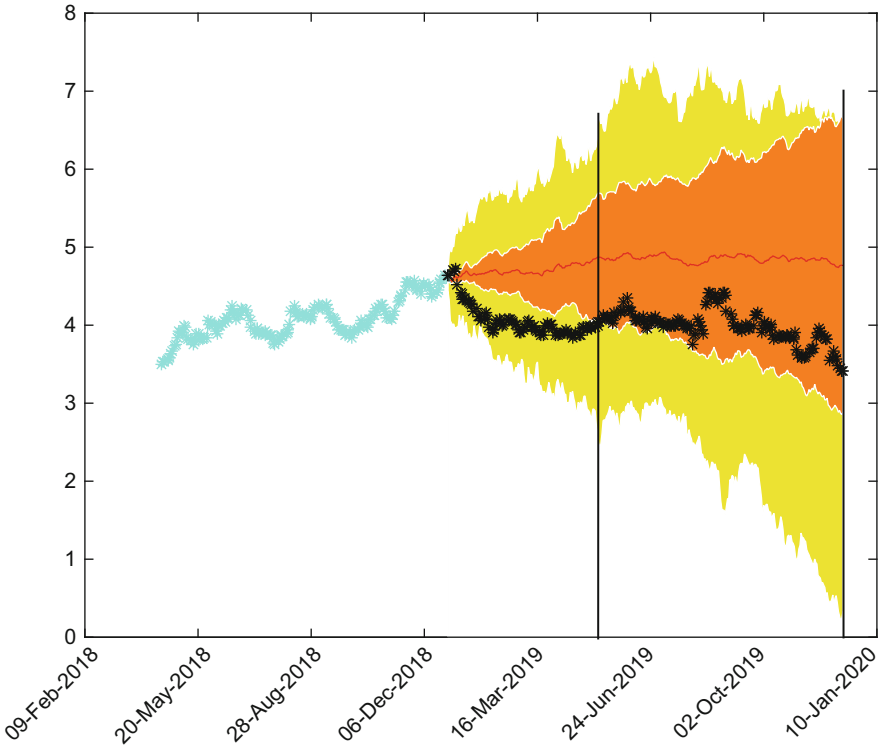


Fig. 9 Latin EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 18$

5.4 Using the Annual EMBI Forecast

Analogous to the monthly case, the evaluation with true data consists of the practitioner taking an investment decision into the Latin American region starting by December 2018 up to December 2019. Data series of the Latin American EMBI index for true evaluation were used, available online [9] up to December 18, 2019. Again, his focus is on the region first and then on the specific country. The LAT's EMBI has the characteristic acceptable to perform the forecast by this method, and its field test can be seen on the second column fourth row of Table 5. Here, two opportunities are analyzed, one of them by April 30, 2019, and another one on December 18, 2019, at the end of the computing period.

The first row has the trend information, whose interpretation will demand down or stable trend behaviors. The second row introduces information about the variation, although for this case, it is smooth for all cases except for Argentina. This label is derived from the forecasted plot with focus on the first quarter of 2019. The third row has the computed range for 55% likelihood by April 30, 2019. The fourth row has the true values, which may indicate the effectiveness of the forecast. Note

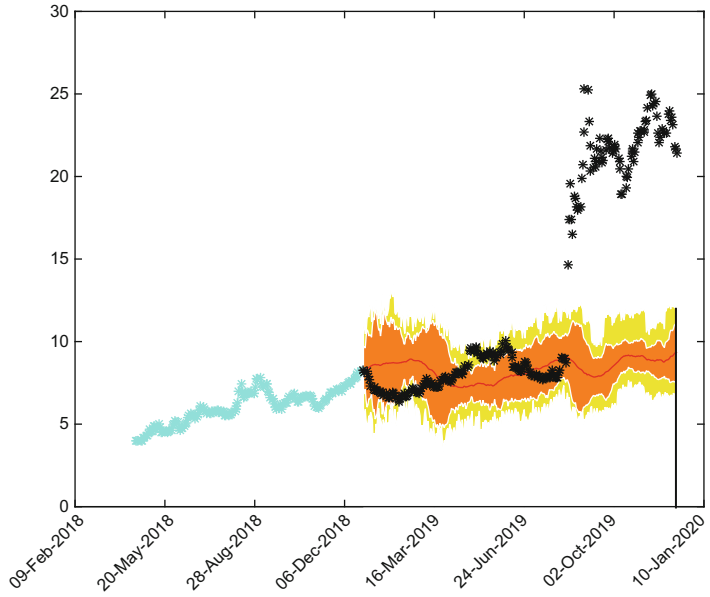


Fig. 10 Argentinean EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 27$

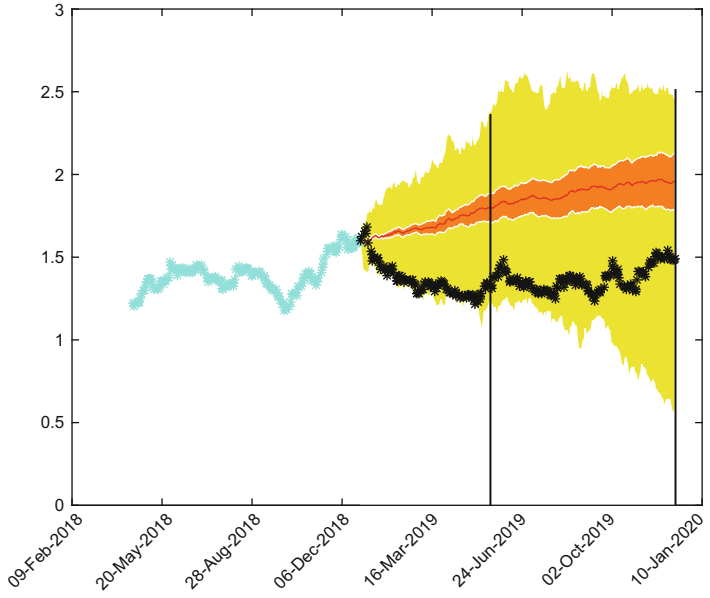


Fig. 11 Chilean EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 18$

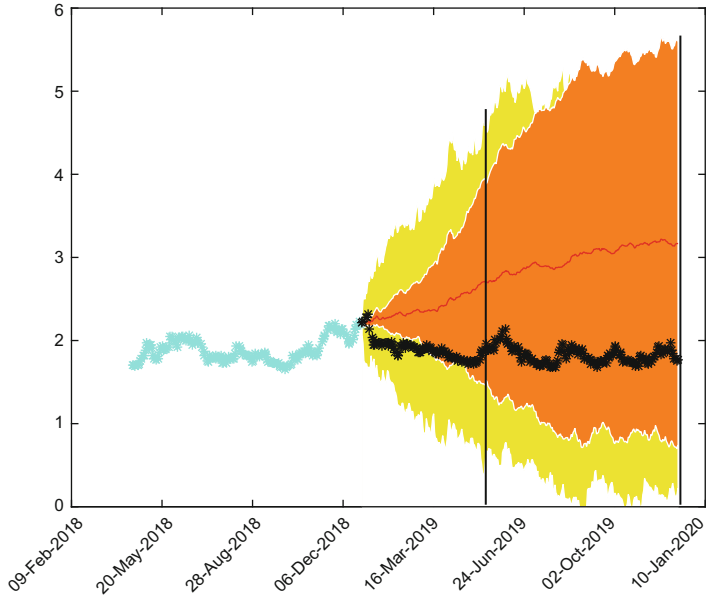


Fig. 12 Colombian EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 18$

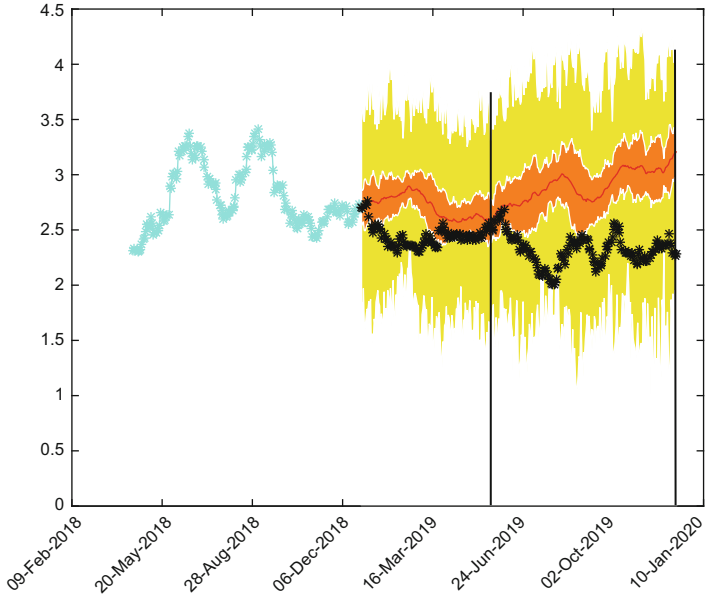


Fig. 13 Brazilian EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 27$

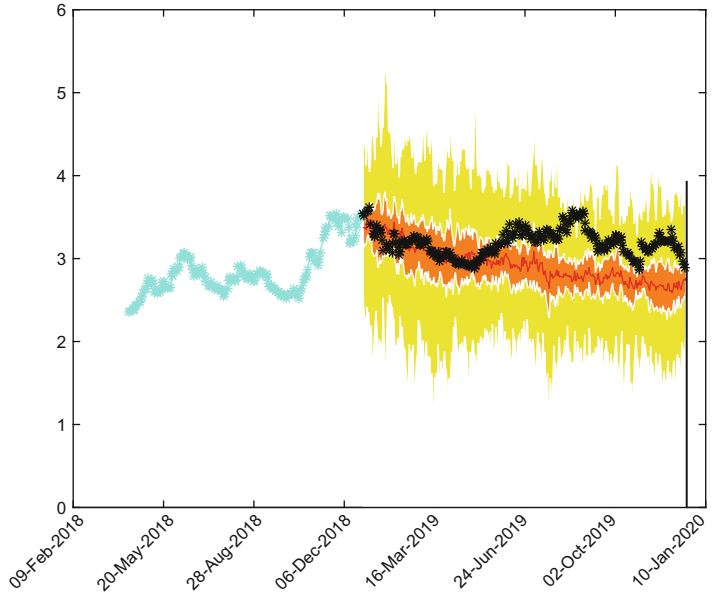


Fig. 14 Mexican EMBI 350-day forecast. NN topology: $l_x = 14$, $H_o = 18$

Table 5 2019 first quarter EMBI prediction and true values

	Region LAT	Argentina	Chile	Colombia	Brazil	Mexico
Trend	Soft rise	Soft rise	Soft rise	Soft rise	Soft down	Down
Variation	Smooth	Rough	Smooth	Smooth	Smooth	Smooth
Range	390–550	600–850	160–185	150–350	240–260	250–350
True values by April 30, 2019	398	950	127	176	245	292
Ensemble (scale 1:100)	Figure 9	Figure 10	Figure 11	Figure 12	Figure 13	Figure 14

that the forecast for Argentina's EMBI turned out of range with a defect error which is the undesired and worse case from the obtained results, given the possibility that the practitioner takes his action based on this information. However, this choice is unlikely given the relative high risk associated with this action because the ranges shown by Brazil or Mexico are less than a half of that of Argentina and those trends are less risky. Thus, the decision maker would tend to focus his investment analysis on the two last columns of Table 5, given its trend, variation, and range for the risk indicated by EMBI's forecast.

When the practitioner has to take a decision based on EMBI forecast with 350 days ahead, the date is December 18 for our study. The results are summarized in Table 6. Note that the first and second rows have differences with respect to those of Table 5, although the first column indicates that the Region LAT deserves the practitioner attention given its trend down. The EMBI variation suggests that

Table 6 2019 last day EMBI prediction and true values

	Region LAT	Argentina	Chile	Colombia	Brazil	Mexico
Trend	Down	Abrupt rise	Down	Down	Soft rise	Down
Variation	Smooth	Rough	Smooth	Smooth	Smooth	Smooth
Range	300–650	790–1100	175–200	90–500	280–310	250–290
True values by December 18, 2019	312	1888	136	159	207	289
Ensemble (scale 1:100)	Figure 9	Figure 10	Figure 11	Figure 12	Figure 13	Figure 14

Argentina must be a very high-risk option. Furthermore, the range with 55% of likelihood is the highest and the true value even more high. Brazil appears with a soft rise of its risk, so the practitioner's focus must go to Chile, Colombia, and Mexico. Here, the method detailed shows that the range computed contains the true value for the cases of Mexico and Colombia, while Chile lies 40 pb below the range which is acceptable.

5.5 Discussion and Lessons Learned

From a decision maker point of view, the result can be discussed as follows. For the Latin American region, its index begins around 460 bp with fluctuations that would position it in the 490–500 range for the middle of the quarter (March), the rise seems sustained and of very mild slope, but it falls again toward the end of the month close to 450, which is completely acceptable for analyzing a possible action. At first glance, the obtained EMBI time series behavior is stable, with trend increasing in Argentina's case, which is very volatile and decreases in the Mexico's case. The variation range estimated for Argentina is very broad. At the regional level, the variations of Chile and Brazil are moderate, and those of Colombia and Mexico are very small. Reading the Argentina obtained forecast, the country risk range in bp would be between 800 and 600 points in the period with an unstable downward trend after peaks greater than 840 in January 2019. The trend would indicate a fall in the quarter and then continues declining with respect to the historical range. Volatility affects the amplitude of the predicted values range with probable values for descending around 495 points in less than 90 days. The scenario of variations is high, with a decreasing trend with wide margins conditioned by the high fluctuation of its historical index behavior. The average monthly value would be around 700 points and with an annual downward unstable trend. This is an acceptable forecast given that 2019 is a year with scheduled presidential elections in Argentina. Finally, the true EMBI shows values that are two times higher than the forecasted values. This forecast error would not be determinant because Argentina was out of the

practitioner's focus in the Latin American region given the relatively high and unstable risk, so it was dismissed or under-considered.

For Chile's case study, its forecasted EMBI starts the year around 165 bp with an upward, soft trend that continues up to the end of March with 160 points, which indicates some internal or external factor that slightly presses the upside. Finally, its true EMBI was 28% lower than the forecasted one which is an acceptable error from the practitioner point of view.

Colombia's forecasted EMBI starts around 224 bp with a wide band of likelihood it would consolidate around 245 at the end of March, with some predictions of expansion toward the end of the period. The true value by the end of the period is the half of the forecasted EMBI, being an error in excess, and it is acceptable to avoid some crucial decisions by the practitioner.

For its part, according to this model, Brazil's EMBI would start the year around 272 bp, and it would have an ascending behavior no higher than 290 until March 20th. After that, it starts to return below 260 bp, with small variation and regular movements without brusque highs and lows. The true EMBI value for the final date was 207 bp, resulting in a 30% below the forecasted one which is conservative with respect to the practitioner-decision maker.

Finally, in the case of Mexico's EMBI, the prediction would begin in the range 350–360 bp, with a smooth descending trend toward the end of January, 300–330 bp, and the rest of the year would remain in 250–300. The true EMBI value for the final date was 289 bp which lies into the forecasted bands of 55% likelihood which is clearly acceptable from the practitioner-decision maker point of view.

By observing the obtained results from the forecaster point of view, the method is suitable for the short-term case and shows some limitations in the long-term case. As can be noted from the results shown in Tables 2 and 4, several SMAPE values were better for the reference forecaster model than for the recommended by the method proposed here.

For short-term forecast, the recommended results were 6.31, 3.17, and 8.23% higher than the reference one in the EMBI time series from LAT, Colombia, and Mexico, respectively, and were 0.33, 2.99, and 3.57 lower than those of reference for Argentina, Chile, and Brazil, respectively. Additionally, for the long-term forecast case, the recommended results were 53.27, 35.47, 55.46, and 0.46% higher than the reference one in the EMBI time series from LAT, Chile, Colombia, and Mexico, respectively, and were 11.2 and 54.93 lower than those of reference for Argentina and Brazil, respectively. However, both for short- and long-term forecast, the results was above the true values, a fact that is valuable to the practitioner given that the method provides a trend from which the time series would evolve below such trend, with the exception of some short periods.

6 Conclusions

In this work, a methodology based on neural network filter to model the underlying process behavior that causes the EMBI measurement evolution of the short and long term was detailed. The methodology consists of generating an artificial intelligence-based predictor filter with a stochastic analysis of its future behavior using fractional Gaussian noise. Results of this analysis were shown to determine which series is the most coincident according to the Hurst parameter H. This H is an additional measure used in accordance with that of seasonality, trend, linearity, and stationarity that must be consistent between the data series and the forecasted time series.

The information generated is not the exact value but gives an idea about which will be the EMBI trend and behavior based on its historical values in such a way the decision maker can use. As was discussed, the method presents good performance for the practitioner activity, and in this sense, the method could be improved by including further measurement into the algorithm. However, the errors generated by the algorithm do not affect the decision maker given that the difficult appear when the time series present erratic behavior such as the case of Argentina's EMBI.

The incorporation of this index forecast into decision-making by estimating the predicted values impact it may have on the business plan or the investment portfolio was shown. In the latter case, the EMBI provides a daily measure of the investor's sentiment regarding country risk. Thus, given the alarm sense that brings the EMBI rise, inability to pay or monetary insolvency, this reason makes relevant to generate the projection of this series for the region and some countries that comprise it. Based on data predictions, it is possible to optimize the exposure in instruments, both bonds as shares, at country level and help to reduce the exposure in moments that anticipate an EMBI increase.

By this way, the proposed forecast algorithm identifies relationships between different factors that allow assessing associated risks or likelihood based on a set of conditions. This guides the decision maker during the operations of the organization, identifying the occurrence likelihood of events and the consequent value of the index in the short and medium term. It generates clarity on the scenario of the required return rate in dollars of the countries under analysis and allows the portfolio manager to reaffirm investment and exposure objectives considering the individuality of the risk of each country and the whole or reformulate the investment.

Acknowledgments The authors wish to thank Universidad Nacional de Córdoba (UNC) and Universidad Nacional de Catamarca (UNCa) for their financial support of this work. The authors also would like to thank the reviewers for their thoughtful comments and efforts toward improving the manuscript.

References

1. Van Agtmael, A. (2007 January 9) The emerging markets century: How a new breed of world-class companies is overtaking the world. Free Press; Annotated edition.
2. IFC. (2010). *The first six decades leading the way in private sector development a history* (2nd ed.). Washington D.C. Available on line at https://www.ifc.org/wps/wcm/connect/6285ad53-0f92-48f1-ac6e-0e939952e1f3/IFC-History-Book-Second-Edition.pdf?MOD=AJPERES
3. Cavanagh, J., & Long, R. (1999 August 3). *Introducing the J.P. Morgan Emerging Markets Bond Index Global (EMBI Global)*. New York: J.P. Morgan Securities Inc. Emerging Markets Research.
4. https://www.jpmorgan.com/country/US/EN/jpmorgan/investbk/solutions/research/indices/product
5. Comelli, F. (2012). *Emerging market sovereign bond spreads: Estimation and back-testing* (International Monetary Fund Working Paper WP/12/212). Washington, DC: International Monetary Fund.
6. Baldacci, E., Petrova, I., Belhocine, N., Dobrescu, G., & Mazraani, S. (2011). *Assessing fiscal stress* (IMF Working Paper No. WP/11/100). Washington, DC: International Monetary Fund.
7. Schaechter, A., Emre Alper, C., Arbatli, E., Caceres, C., Callegari, G., Gerard, M., Jonas, J., Kinda, T., Shabunina, A., & Weber, A. (2012). *A toolkit to assessing fiscal vulnerabilities and risks in advanced economies* (IMF Working Paper No. WP/12/11). Washington, DC: International Monetary Fund). https://www.imf.org/external/pubs/ft/wp/2012/wp1211.pdf.
8. Candelon, B., Dumitrescu E. and C. Hurlin. How to evaluate and early-warning system: Toward a unified statistical framework for assessing financial crises forecasting methods, IMF economic review, 60, 1 (Washington, DC, International Monetary Fund). 2012.
9. https://www.invenomica.com.ar/riesgo-pais-embi-america-latina-serie-historica/
10. Davenport, T. (2014, September). A predictive analytics primer. *Havard Business Review*. https://hbr.org/2014/09/a-predictive-analytics-primer
11. Lihn, S. H. T. (2018). *Jubilee tectonic model: Forecasting long-term growth and mean reversion in the U.S. stock market* (April 4, 2018). SSRN: 3156574. http://dx.doi.org/10.2139/ssrn.3156574.
12. Kolanovic, M., & Krishnamachari, R.. (2017). *Big data and AI strategies. Machine learning and alternative data approach to investing. Quantitative and derivatives strategy*. Available at https://www.cfasociety.org/cleveland/Lists/Events%20Calendar/Attachments/1045/BIG-Data_AI-JPMmay2017.pdf
13. Cervelló-Royo, R., Cortés, J.-C., Sánchez-Sánchez, A., Santonja, F.-J., & Villanueva, R. (2013). Forecasting Latin America's country risk scores by means of a dynamic diffusion model. *Abstract and Applied Analysis, 2013*., Article ID 264657, 1–11. https://doi.org/10.1155/2013/264657.
14. Caporale, G., Carcel, H., & Gil-Alana, L. (2018 March). The EMBI in Latin America: Fractional integration, non-linearities and breaks. *Finance Research Letters, 24*, 34–41.
15. Siddiqui, S. A., Mercier, D., Munir, M., Dengel, A., & Ahmed, S. (2018). *TSViz: Demystification of deep learning models for time-series analysis*. Volume 4, 2016 1 arXiv:1802.02952v2 [cs.LG] 13 December 2018. https://arxiv.org/pdf/1802.02952.pdf
16. Tipirisetty, A. (2018). *Stock price prediction using deep learning*. San Jose State University. Master's theses and Graduate Research. https://www.math.cuhk.edu.hk/~jwong/MATH4900F_19/Schedule/Paper_SP1.pdf
17. Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting, 36*, 75–85.
18. Dieker, T. (2004). *Simulation of fractional Brownian motion*. MSc theses, University of Twente, Amsterdam, The Netherlands.

19. Rodríguez Rivero, C., Pucheta, J. Patiño, D., Puglisi, J., Otaño, P., Franco, L., Juarez, G., Gorrostieta E., & Orjuela-Cañón, A. (2019 December). Bayesian inference for training of long short term memory models in chaotic time series forecasting. In *Applications of computational intelligence* (Pages 197–208. Pp 279). Cham: Springer. eBook ISBN 978-3-030-36211-9. Series ISSN 1865-0929. https://doi.org/10.1007/978-3-030-36211-9_16
20. Bishop, C. (2006). *Pattern recognition and machine learning*. Boston: Springer.
21. Abry, P., Flandrin, P., Taqqu, M. S., & Veitch, D. (2003). Self-similarity and long-range dependence through the wavelet lens. In *Theory and applications of long-range dependence* (pp. 527–556). Basel: Birkhäuser.
22. Hosking, J. R. M. (1984). Modeling persistence in hydrological time series using fractional Brownian differencing. *Water Resources Research, 20*, 1898–1908.
23. http://www.columbia.edu/~ad3217/fbm/hosking.c

Correction to: Principles of Data Science

Hamid R. Arabnia, Kevin Daimi, Robert Stahlbock, Cristina Soviany, Leonard Heilig, and Kai Brüssau

Correction to:
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

The original version of this book was inadvertently published without adding second affiliation for the editor "Robert Stahlbock". The affiliation "FOM University of Applied Sciences, Hamburg/Essen, Germany" has now been added in the Copyright page and Acknowledgments on page vii.

The updated online version of this book can be found at
https://doi.org/10.1007/978-3-030-43981-1

© Springer Nature Switzerland AG 2020

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_13

Index

© Springer Nature Switzerland AG 2020

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